CMU OAQA at TREC 2017 LiveQA: A Neural Dual Entailment Approach for Question Paraphrase Identification

Di Wang and **Eric Nyberg**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{diwang,ehn}@cs.cmu.edu

Abstract

In this paper, we present CMU's question answering system that was evaluated in the TREC 2017 LiveQA Challenge. Our overall approach this year is similar to the one used in 2015 and 2016. This system answers real-user submitted questions from Yahoo! Answers website and medical questions, which involves retrieving relevant web pages, extracting answer candidate texts, ranking and selecting final answer text. The main improvement this year is the introduction of our new question paraphrase identification module based on a neural dual entailment model. The model uses bidirectional recurrent neural network to encode the premise question into *phrase* vectors, and then *align* corresponding phrase vectors from the candidate question with the attention mechanism. The final similarity score is produced based on aggregated phrase-wise comparisons of both entailment directions. In the TREC 2017 LiveQA evaluations, human assessors gave our system an average score of 1.139 on a three-point scale and the median score was 0.777 for all the systems evaluated. Overall, our approach received the highest average scores among automatic systems in main tasks of 2015, 2016 and 2017, and also the highest average score in the new medical subtask of 2017.

1 Introduction

Similar to the LiveQA tracks in 2015 [1] and 2016 [2], the main objective of LiveQA 2017 was still to provide automatic answers to real-user questions in real time. In previous years, the test collection came from a stream of questions that freshly submitted to the Yahoo! Answers website and have not been previously answered by humans. This year, due to technical difficulties, the participant systems were fed with cached questions instead. Participant systems need to return answer texts with no more than 1000 characters in length and within 1 minute. There are no additional restrictions on the resources that can be used, with the exception of the human answers to the same question in Yahoo! Answers. System responses were then judged by TREC assessors on a 4-level Likert scale.

CMU's Open Advancement of Question Answering (OAQA) group continued the work from last two years' LiveQA submissions. We improved the real-time web-based Question Answering (QA) system, and submitted one run to 2017 evaluation. Our QA pipeline begins with candidate passages retrieval and extraction, then answer candidate ranking and tiling. In this paper, we focus on discussing our recent development on the question paraphrase identification module. Being considered as one of the key challenges of developing effective QA system, the question paraphrase identification is to verify whether a retrieved candidate question is a paraphrase of input question. Two questions are paraphrases of each other if they can be adequately answered by the exact same answer. During the official run, our QA server received one question per minute for 24 hours and

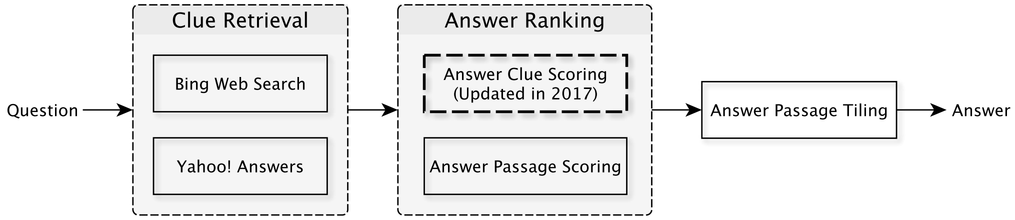

Figure 1: Architecture of the CMU-OAQA LiveQA system

provided answers within one minute for 98% of the input questions. On a normalized three-point average score metric, CMU-OAQA received a score of 1.139 on the main task, which was significantly higher than the median score of 0.777 from all systems. A new subtask focused specifically on medical questions is also offered this year. Our system answered medical questions the same way as the main task, and also achieved the highest average score of 0.637 comparing to the median score of 0.431. In the rest of this paper, we will briefly sketch the OAQA LiveQA system structure, and then introduce the new question paraphrase identification model in more details.

2 System Architecture

Our overall system architecture remains the same to the one used in 2015 [9] and 2016 [10]. The pipeline is briefly described here for completeness, please refer to our reports for the last two years for a complete description. As illustrated in Figure 1, the architecture of our system decomposes the solution into three major processing phases:

1. **Clue Retrieval**. Given a question title and its full-text description, we formulate search engine queries and issue them to different search engines (Bing Web Search, Yahoo! Answers) in order to retrieve web pages related to the question.
2. **Answer Ranking**. Answer candidates (title/body/answer tuples that represent either conceptional questions or answer texts) are extracted from web pages, and ranked based on a relevance estimator. The most effective relevance estimator we found was a heuristically-weighted combination of: a) optimized BM25 similarity scoring over the title and body texts, b) an attentional encoder-decoder recurrent neural networks model [10, 11] that estimates the relevance of a candidate answer text given a question text, and c) a novel neural dual entailment based question paraphrase identification model that predicts the relevance between input question and titles of answer candidates.
3. **Answer Passage Tiling**. Finally, a simple greedy algorithm is used to select a subset of highest-ranked answer candidates; these are simply concatenated without further processing in order to produce the final answer.

3 Question Paraphrase Identification by Neural Dual Entailment

Able to determine if two questions have the same meaning semantically can greatly improve the candidate answer ranking performance. However, it is a challenging task even for human experts. Questions from online forums can range from highly technical subjects to ambiguous thoughts. The question texts are also not always grammatically correct and often contain misspelled words.

In this section, we discuss the problem of question paraphrase identification which predicting a binary label for whether two input questions convey the same meaning. We introduce a new neural dual entailment model for the question paraphrase identification task. This method uses bidirectional recurrent neural networks to encode the premise question into phrase vectors, and then fetch the softly aligned corresponding phrase embedding from the candidate question with the attention mechanism. The final similarity score is produced based on aggregated phrase-wise comparisons of both entailment directions. Our model is fully symmetric and then parameters are shared on both sides. This allows the model parameters can be trained effectively even on the medium sized dataset.

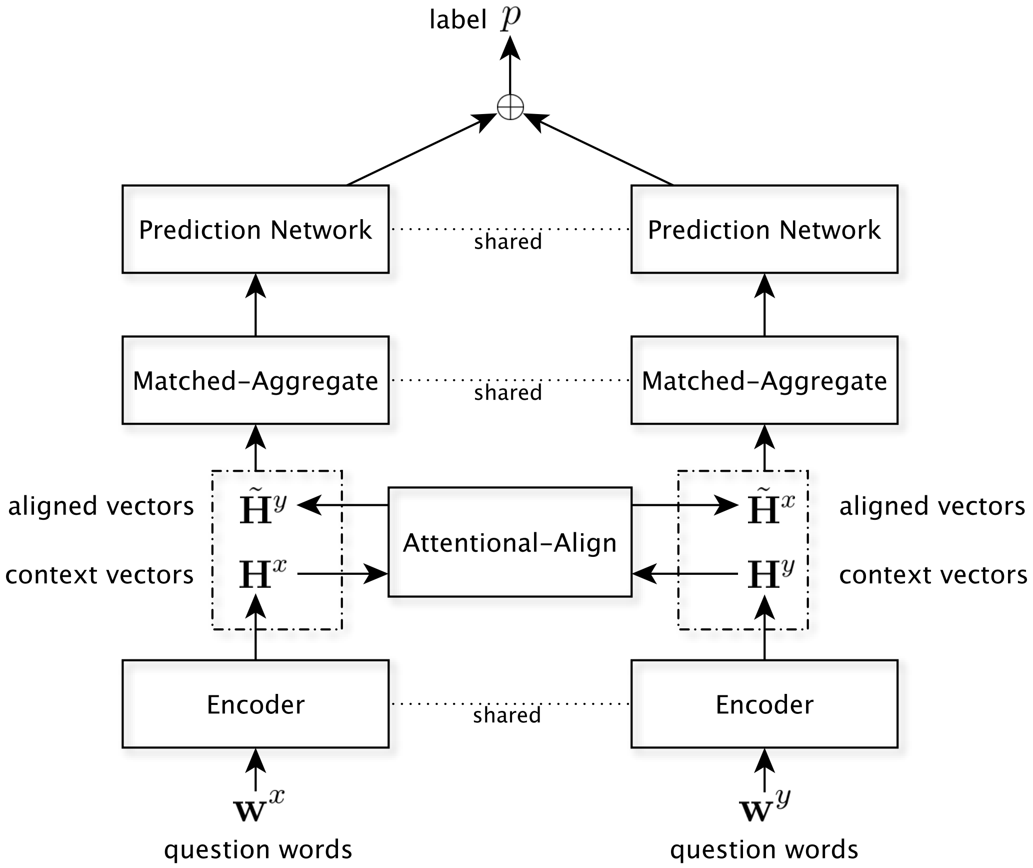

Figure 2: Model Structure of our Neural Dual Entailment Model

We evaluated our model on the Quora Question Pairs dataset.[1] Experimental results show that our model achieves the state-of-the-art performance on this benchmark dataset.

3.1 Problem Formulation

Let $\mathbf{w}^x = (w_1^x, \dots, w_{\ell^x}^x)$ and $\mathbf{w}^y = (w_1^y, \dots, w_{\ell^y}^y)$ be two input questions consisting of ℓ^x and ℓ^y words, respectively. Each word in both sentences, w_i^x or w_j^y, belongs to the vocabulary V_w. The training data is in the form of labeled text pairs $\{(\mathbf{w}^x, \mathbf{w}^y)^{(n)}, p^{(n)}\}_{n=1}^N$, where $p^{(n)} \in \{0, 1\}$ is a binary label indicating whether $\mathbf{w}^x$ is a paraphrase of $\mathbf{w}^y$. The goal is to predict the correct label p given a previously unseen text pair $(\mathbf{w}^x, \mathbf{w}^y)$.

3.2 Model Structure

As outlined in Figure 2, our model decomposes the paraphrase identification problem into four steps: *Encode*, *Attentional-Align*, *Matched-Aggregate*, and *Dual-Predict*.

Encode. The goal of this step is to obtain a dense representation of each word w_i in $\mathbf{w}^x$ and $\mathbf{w}^y$ that captures word meaning along with contextual information.

We first individually map each word w_i into a d-dimensional vector by a mixed embedding matrix $\mathbf{E} \in \mathbb{R}^{|V_w| \times d}$, where $\mathbf{E} = \mathbf{E}_{pre} + \mathbf{E}_{tune}$. Here $\mathbf{E}_{pre}$ is a word embedding matrix that unsupervisedly pre-trained on a large corpus and will be fixed during the model training. On the other hand, $\mathbf{E}_{tune}$ is a trainable embedding matrix that are randomly initialized. This mixture setup aims to fine-tune

[1] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

the pre-trained embeddings to recognize domain specific semantics of word usage, while avoiding deviating from the pre-trained representations too early.

Then the encoder converts the word embedding sequence into a list of contextual vectors $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{\ell^w}\}$, whose size varies with regard to the length of the passage. This context representation is generated using a two-layer stacked bidirectional recurrent neural networks (BRNN), which utilize context of both sides. Specifically, the encoder BRNN processes the data from both directions with two separate hidden sub-layers, where

$$\begin{aligned} \overrightarrow{\mathbf{h}_i} &= \overrightarrow{\Psi}\left(\overrightarrow{\mathbf{h}}_{i-1}, \mathbf{E}\left[w_i\right]\right) \\ \overleftarrow{\mathbf{h}_i} &= \overleftarrow{\Psi}\left(\overleftarrow{\mathbf{h}}_{i+1}, \mathbf{E}\left[w_i\right]\right). \end{aligned} \tag{1}$$

Here Ψ is a recurrent activation unit that we employ in the Long Short-Term Memory (LSTM) [5]. The output of current time step is then generated by concatenating both direction's hidden vector $\overrightarrow{\mathbf{h}_i}$ and $\overleftarrow{\mathbf{h}_i}$. The stacked upper layer BRNN treats the output from the lower layer $\mathbf{h}_i^1$ as the input, and further outputs $\mathbf{h}_i^2$. Finally, the encoded contextual vector is aggregated as the sum of embedding and outputs from both layers:

$$\mathbf{h}_i = \mathbf{E}\left[w_i\right] + \mathbf{h}_i^1 + \mathbf{h}_i^2. \tag{2}$$

Attentional-Align. Regarding each $\mathbf{h}_i$, this step attempts to gather a softly aligned context vector from the other sentence's encoded sequence $\bar{\mathbf{H}}$. It starts by computing the content-based score of each pair of context vectors as:

$$e_{i,j} = \mathbf{h}_i^\top \bar{\mathbf{h}}_j. \tag{3}$$

This score measures similarity between $\mathbf{h}_i$ and the j-th context vector of $\bar{\mathbf{H}}$. These relevance scores are further normalized by the softmax function:

$$\alpha_{i,j} = \mathrm{softmax}(e_{i,j}) = \frac{\exp(e_{i,j})}{\sum_{k=1}^{\ell} \exp(e_{i,k})}, \tag{4}$$

and we call $\alpha_{i,j}$ the attention weight. The softly aligned phrase vector $\tilde{\mathbf{h}}_i$ is then the weighted sum of the context vectors with their attention weights from above:

$$\tilde{\mathbf{h}}_i = \sum_{j=i}^{\ell} \alpha_{i,j} \bar{\mathbf{h}}_j. \tag{5}$$

Matched-Aggregate. Given the $\mathbf{H}$ and its softly aligned $\tilde{\mathbf{H}}$, the problem has been reduced to comparing a sequence of aligned phrase vectors. Each pair of aligned vectors are first merged into a matched vector m_i with a Multi-Layer Perceptron M:

$$m_i = M\left(\left[\mathbf{h}_i; \tilde{\mathbf{h}}_j; (\mathbf{h}_i - \tilde{\mathbf{h}}_j)^2\right]\right). \tag{6}$$

The list of matched vectors $\{m_1, \ldots, m_\ell\}$ are later scanned by an RNN layer, and its output of the last timestamp is served as the final aggregated vector.

Dual-Predict. The prediction network consists of a two-layer batch-normalized multi-layer perceptron (MLP), MaxOut neurons [4] , and a linear layer to produce an entailment score based on the matched aggregate vector above. As yet, the *Attentional-Align*, *Matched-Aggregate*, and the prediction network can only to check whether the information of one sentence is phrase-wise covered by another sentence. To identify paraphrase, we use the shared network to calculate entailment scores from both directions of question pairs. Finally, two scores are summed and followed by a logistic layer, to predict the label p.

3.3 Experiments

3.3.1 Dataset

We evaluated our models on the Quora question paraphrase dataset which contains over 404,000 question pairs with binary labels. The dataset has approximately 37% positive and 63% negative pairs. We use the same data split and tokenization provided by Wang et al. [2] [12]. Each of the devel-

[2] This partition of Quora dataset can be downloaded at `https://zhiguowang.github.io`.

Method	Dev Acc.	Test Acc.
Siamese-CNN	-	79.60
Multi-Perspective CNN	-	81.38
Siamese-LSTM	-	82.58
Multi-Perspective-LSTM	-	83.21
L.D.C	-	85.55
BiMPM	88.69	88.17
pt-DecAtt$_{\mathrm{word}}$	88.44	87.54
pt-DecAtt$_{\mathrm{char}}$	88.89	88.40
Ours model	88.61	**88.92**

Table 1: Evaluation results on the Quora paraphrase identification dataset in terms of accuracy. The first eight rows are taken from [8, 12].

opment and test datasets has 5,000 positive and 5,000 negative instances. The remaining instances are then used as the training set. We further augment the training set with 20,000 positive pairs with identical question pairs to make sure the model can generalize well on easy situation as well.

3.3.2 Network Setup and Parameter Optimization

The embeddings $\mathbf{E}_{pre}$ was initialized with the 300-dimensional GloVe [7] word vectors pre-trained from the 840B Common Crawl corpus. Out-of-vocabulary words were initialized as zeros. Our encoder RNN contains two-layer stacked LSTMs. Each layer of LSTM has a memory size of 300. The MLP networks used ReLU as activation functions and dropout rate of 0.2. The network weights are randomly initialized using a uniform distribution $(-0.08, 0.08)$, and are trained with the ADAM optimizer [6], with an initial learning rate of 0.002. Gradients were clipped so their norm does not exceed 5. Each mini-batch contains 1000 question pairs.

3.3.3 Results

In this section, we compare the performance of our approach with the baseline models, bilateral multi-perspective matching (BiMPM) [12] model, and Paralex pre-trained decomposable attention (pt-DecAtt) [8] models, which have benchmarked on this dataset before. The BiMPM model employs both character-level and word-level LSTMs to represent input, four different types of multi-perspective matching functions, an bi-LSTM aggregation layer, and a MLP with softmax output for final prediction. The pt-DecAtt model uses sums of character n-gram embeddings to represent words, both bi-attention and self-attention to softly align phrase vectors, a MLP followed by sum as aggregation layer, and another MLP with linear output for prediction. pt-DecAtt also utilized Paralex [3] question paraphrase corpus to pre-train both its character n-gram embeddings and rest of the model.

Table 1 shows the accuracy results of baseline models implemented in [12], BiMPM model, pt-DecAtt model, and our method. From the results, we can see that our model can outperform the previous best performance on the test set. Notes that our model is also simpler since it does not employ character encoding, self-attention, multi-perspective matching, and Paralex pre-training as required in previous methods.

4 Official Evaluation Results

In this year's LiveQA evaluation, there is a total of 1180 questions for the main task and an additional set of 102 medical subtask questions. Our system returned answers for 1162 main task questions and 99 medical subtask questions. Submitted answers were judged and scored by TREC assessors using the same 4-level Likert scale same as last two years:

- **4**: Excellent: "a significant amount of useful information, fully answers the question"
- **3**: Good: "partially answers the question"

Task	System ID	Avg score (0-3)	Success@ 2+	 3+	 4+	Precision@ 2+	 3+	 4+
Medical	Median of all runs	0.417	0.245	0.142	0.059	0.331	0.178	0.078
	CMU-OAQA	**0.637**	**0.392**	**0.265**	**0.098**	**0.404**	**0.273**	**0.101**
Main	Median of all runs	0.777	0.421	0.250	0.126	0.482	0.286	0.144
	CMU-OAQA	**1.139**	**0.567**	**0.387**	**0.198**	**0.577**	**0.393**	**0.201**

Table 2: Official TREC 2017 LiveQA track evaluation results.

- **2**: Fair: “marginally useful information”
- **1**: Bad: “contains no useful information for the question”
- **-2**: “the answer is unreadable (only 15 answers from all runs in 2015)”

The evaluation metrics are defined as follows:

- **avg-score (0-3)**: “average score over all queries (transferring 1-4 level scores to 0-3, hence comparing 1-level score with no-answer score, also considering -2-level score as 0)”
- **succ@i+**: “number of questions with i+ score (i=1..4) divided by number of all questions”
- **prec@i+**: “number of questions with i+ score (i=2..4) divided by number of answered only questions”

Table 2 summarizes the results of our system run and median scores from all submitted runs. We believe the overall performance of our system to be promising, as it suggests that our system can provide a useful answer (fair, good, or excellent) for more than 56% of the questions. It is also encouraging that our system can generalize to medical domain and receive the highest score without any domain-specific processing or modification.

5 Conclusion

This paper presented the improved question matching module and evaluation results for our LiveQA 2017 system. We showed that efficient parameter sharing and mixed word embeddings result in state-of-the-art accuracy on question paraphrase identification task even without complex neural architectures, character-level embeddings, or pre-training the full model. The new matching module also shown its robustness when integrated into our LiveQA system, and helped our system achieved highest average score among automatic systems in both main task and medical subtask.

References

[1] Eugene Agichtein, David Carmel, Dan Pelleg, Yuval Pinter, and Donna Harman. Overview of the TREC 2015 liveqa track. In *Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015*, 2015.

[2] Eugene Agichtein, David Carmel, Dan Pelleg, Yuval Pinter, and Donna Harman. Overview of the TREC 2016 LiveQA track. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, 2016.

[3] Anthony Fader, Luke S. Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1608–1618, 2013.

[4] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML’13, pages III–1319–III–1327. JMLR.org, 2013.

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Comput.*, 9(8): 1735–1780, 1997.

[6] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6, 2014.

[7] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[8] Gaurav Singh Tomar, Thyago Duque, Oscar Täckström, Jakob Uszkoreit, and Dipanjan Das. Neural paraphrase identification of questions with noisy pretraining. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 142–147. Association for Computational Linguistics, 2017.

[9] Di Wang and Eric Nyberg. CMU OAQA at TREC 2015 LiveQA: Discovering the Right Answer with Clues. In *Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015*, 2015.

[10] Di Wang and Eric Nyberg. CMU OAQA at TREC 2016 LiveQA: An attentional neural encoder-decoder approach for answer ranking. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, 2016.

[11] Di Wang, Nebojsa Jojic, Chris Brockett, and Eric Nyberg. Steering output style and topic in neural response generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2140–2150. Association for Computational Linguistics, 2017.

[12] Radu Florian Zhiguo Wang, Wael Hamza. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4144–4150, 2017. doi: 10.24963/ijcai.2017/579.