SelQA: A New Benchmark for Selection-based Question Answering

Tomasz Jurczyk **Michael Zhai** **Jinho D. Choi**
Mathematics and Computer Science
Emory University
Atlanta, GA 30322, USA
{tomasz.jurczyk,michael.zhai,jinho.choi}@emory.edu

Abstract

This paper presents a new dataset to benchmark selection-based question answering. Our dataset contains contexts drawn from the ten most prevalent topics in the English Wikipedia. For the generation of a large, diverse, and challenging dataset, a new annotation scheme is proposed. Our annotation scheme involves a series of crowdsourcing tasks that can be easily followed by any researcher. Several systems are compared on the tasks of answer sentence selection and answer triggering, providing strong baseline results for future work to improve upon. We hope that providing a large corpus will enable researchers to work towards more effective open-domain question answering.

1 Introduction

Selection-based question answering is a question answering task that finds a context containing an answer to an open-domain question, where the context comprises of one or more sentences. Let us define a context as being a document section, multi-sentence selection, or sentence. In answer sentence selection, the answer context is guaranteed to exist within the provided contexts. The task is defined as answering triggering when there is no such guarantee. Several corpora have been created for these tasks (Wang et al., 2007; Yang et al., 2015; Feng et al., 2015), allowing researchers to build effective question answering systems (Yu et al., 2014; Wang and Nyberg, 2015; Severyn and Moschitti, 2015). However, most of these datasets contain a limited number of questions and answers. Moreover, the questions in these datasets are restricted in their scope of topics. We attempt to mitigate these limitations, allowing for a more through evaluation of open domain question answering systems.

This paper presents a new corpus of annotated question answer pairs drawn from contexts of various topics. An effective annotation scheme is proposed to create a large corpus that is both challenging and realistic. Questions are additionally annotated with its topic, type, and paraphrase that enable to make comprehensive analyses of the answer sentence selection and answering triggering tasks. Two recent state-of-the-art systems based on convolutional and recurrent neural networks are implemented to analyze this corpus and to provide strong baseline measures for future work. In addition, our systems are evaluated on another dataset, WikiQA (Yang et al., 2015), for a fair comparison to previous work. Our analysis suggests extensive ways of evaluating selection-based question answering, providing meaningful benchmarks to question answering systems. The contributions of this work include:[1]

- Creating a new corpus for answer sentence selection and answer triggering (Section 3).
- Developing QA systems using the latest advances in neural networks (Section 4).
- Analyzing various aspects of selection-based question answering (Section 5).

2 Related Work

The TREC QA datasets have been a popular choice for evaluating answer sentence selection.[2] Wang et al. (2007) combined the TREC-[8-12] datasets for training and divided the TREC-13 dataset for development and evaluation. This dataset, known as QASent, has been used as the standard benchmark for answer sentence selection although it is rather small (277 questions with manually picked answer contexts). Yang et al. (2015) introduced a lager

[1] All our work is publicly available at: anonymous_url

[2] http://trec.nist.gov/data/qa.html

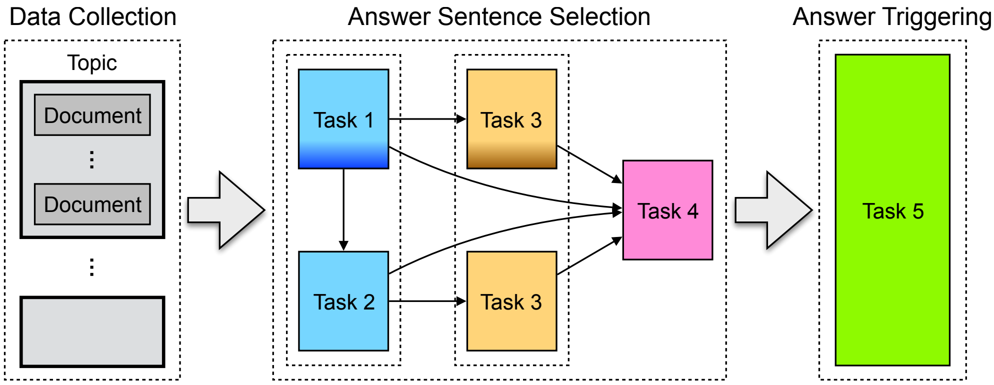

Figure 1: The overview of our data collection (Section 3.1) and annotation scheme (Section 3.2).

dataset, WikiQA, consisting of questions collected from the user logs of the Bing search engine. Our corpus is similar to WikiQA but has more diverse question topics, a larger number of questions (about 6 times larger for answer sentence selection and 2.5 times larger for answer triggering), and makes use of more contexts by using the entire article instead of only the abstract. Feng et al. (2015) distributed another dataset, InsuranceQA, including questions in the insurance domain. WikiQA introduced the task of answer triggering and and was the only answer triggering dataset. Our corpus provides a new automatically generated answer triggering dataset.

Our convolutional neural network model is inspired by the previous work utilizing the tree-edit distance and the tree kernel (Heilman and Smith, 2010; Wang and Manning, 2010; Severyn and Moschitti, 2013), although we introduce a different way of performing subtree matching facilitating word embeddings. Our recurrent neural network model with attention mechanisms is based on established state-of-the-art systems for answer sentence selection (Tan et al., 2015; Santos et al., 2016).

3 Corpus

Our annotation scheme provides a framework for any researcher to create a large, diverse, pragmatic, and challenging dataset for answer sentence selection and answer triggering, while maintaining a low cost using crowdsourcing.

3.1 Data Collection

A total of 486 articles are uniformly sampled across the following 10 topics from the English Wikipedia, dumped on August, 2014:

Arts, *Country*, *Food*, *Historical Events*, *Movies*, *Music*, *Science*, *Sports*, *Travel*, *TV*.

These are the most prevalent topics categorized by DBPedia.[3] The original data is preprocessed into smaller chunks. First, each article is divided into sections using the section boundaries provided in the original dump.[4] Each section is then segmented into sentences by the open-source toolkit, NLP4J.[5] In our corpus, documents refer to individual sections in the Wikipedia articles.

Type	Count
Total # of articles	486
Total # of sections	8,481
Total # of sentences	113,709
Total # of tokens	2,810,228

Table 1: Lexical statistics of our corpus.

3.2 Annotation Scheme

Four annotation tasks are conducted in sequence on Amazon Mechanical Turk for answer sentence selection (Tasks 1-4), and a single task is conducted for answer triggering using Elasticsearch (Task 5; see Figure 1 for the overview).

Task 1

Approximately two thousand sections are randomly selected from the 486 articles in Section 3.1. All the selected sections consist of 3 to 25 sentences; we found that annotators experienced difficulties accurately and timely annotating longer sections. For each section, the annotator is instructed to generate a question and select one or more sentences containing the answer from the provided section. The annotator is provided with the instruction, the topic, the article title, the section title, and the list of numbered sentences in the section (Table 2).

[3] http://dbpedia.org
[4] https://dumps.wikimedia.org/enwiki
[5] https://github.com/emorynlp/nlp4j

Topic: TV, **Article**: Criminal Minds, **Section**: Critical reception

1. The premiere episode was met with mixed reviews, receiving a score of 42 out of 100 on aggregate review site Metacritic, indicating "mixed or average" reviews.
2. Dorothy Rabinowitz said, in her review for the Wall Street Journal, that "From the evidence of the first few episodes, *Criminal Minds* may be a hit, and deservedly"...
3. The New York Times was less than positive, saying "The problem with *Criminal Minds* is its many confusing maladies, applied to too many characters" and felt that "as a result, the cast seems like a spilled trunk of broken toys, with which the audience - and perhaps the creators - may quickly become bored."
4. The Chicago Tribune reviewer, Sid Smith, felt that the show "May well be worth a look" though he too criticized "the confusing plots and characters".

Task 1	How was the premiere reviewed?
Task 2	Who felt that Criminal Minds had confusing characters?
Task 3.1	How were the initial reviews?
Task 3.2	Who was confused by characters on Criminal Minds?
Task 4.3.1	How were the initial reviews in Criminal Minds?

Table 2: Given a section, Task 1 asks to generate a question regarding to the section. Task 2 crosses out the sentence(s) related to the first question (line 1), and asks to generate another question. Task 3 asks to paraphrase the first two questions. Finally, Task 4 asks to rephrase ambiguous questions.

Task 2

Annotators are asked to create another set of ≈2K questions from the same selected sections, but this time, sentences regarding to the questions in Task 1 are discarded. The goal of Task 2 is to generate questions from contexts different from those used in the Task 1. The annotators are provided with the same information, except that the sentences used as the answer contexts in Task 1 are crossed out (line 1 in Table 2). Annotators are instructed not to use these sentences to generate new questions.

Task 3

Although our instruction encourages the annotators to create questions in their own words, annotators will still often use vocabulary similar to the target context to create the question. The intention of this task is to mitigate the bias of using similar vocabulary for the question and answer contexts because the goal of this corpus is to evaluate a model's ability to understand the question and answer contexts, not word co-occurances. Annotators are asked to generate another set of questions by paraphrasing the questions from the previous tasks. The annotators are provided with the same information as the first two tasks and the previously generated questions and answer contexts; they are instructed to paraphrase these questions using different terms.

Task 4

Most questions generated by Tasks 1-3 are of high quality; however, few of them are ambiguous and difficult for even humans to answer correctly. These ambiguous questions often assume that the contexts are provided with the questions. For instance, it is impossible to answer the question from Task 3.1 in Table 2 unless the related section is provided with the question. These ambiguous questions are sent back to the annotators for revision.

For finding ambiguous questions, Elasticsearch is used,[6] a Lucene-based open-source search engine. First, an inverted index of 8,481 sections is built, where each section is considered a document. Each question is queried to this search engine. If the answer context is not included within the top 5 sections in the search result, the question is considered 'suspicious' although it may not be ambiguous. Among 7,904 questions generated by Tasks 1-3, 1,338 of them are found to be suspicious. These questions are sent to the annotators, and rephrased by the annotators if deemed necessary.

Task 5

By using the previously generated answer sentence selection data, the answer triggering corpus can be automatically generated again using Elasticsearch. To generate answer contexts for answer triggering, all 14M sections from the entire English Wikipedia are indexed, and each question from Tasks 1-4 is queried. Every sentence in the top 5 highest scoring sections from Elasticsearch are collected as candidates, which may or may not include the answer context that resolves the question.

3.3 Corpus Analysis

The entire annotation took about 130 hours, costing \$770 in total; each mturk job took on average approximately 1 minute and costed about ¢10. A total

[6] www.elastic.co/products/elasticsearch

	Q_s	Q_m	Q_{s+m}	Ω_q	Ω_a	Ω_f	Time	Credit
Task 1	1,824	154	1,978	44.99	23.65	28.88	71 sec.	$ 0.10
Task 2	1,828	148	1,976	44.64	23.20	28.62	64 sec.	$ 0.10
Task 3	3,637	313	3,950	38.03	19.99	24.41	41 sec.	$ 0.08
Task 4	682	55	737	31.09	19.41	21.88	54 sec.	$ 0.08
Our corpus	7,289	615	**7,904**	40.54	21.51	**26.18**	-	-
WikiQA	1,068	174	1,242	39.31	9.82	15.03	-	-

Table 3: $Q_{s|m}$: number of questions whose answer contexts consist of single|multiple sentences, $\Omega_{q|a}$: macro avg. of overlapping words between q and a, normalized by the length of $q|a$, $\Omega_f = {}^{(2 \cdot \Omega_q \cdot \Omega_a)}/_{(\Omega_q + \Omega_a)}$, Time|Credit: avg. time|credit per mturk job. WikiQA statistics here discard questions w/o answer contexts.

of 7,904 questions were generated from Tasks 1-4, where 92.2% of them found their answers in single sentences. It is clear that Task 3 was effective in reducing the percentage of overlapping words between question and answer pairs (about 4%; Ω_f in Table 3). The questions from Task 3 can be used to develop paraphrasing models as well. Multiple pilot studies on different tasks were conducted to analyze quality and cost; Tasks 1-4 were proved to be the most effective in the pilot studies. Following Ho et al. (2015), we paid incentives to those who submitted outstanding work, which improved the overall quality of our annotation.

Our corpus could be compared to WikiQA that was created with the intent of providing a challenging dataset for selection-based question answering (Yang et al., 2015). Questions in this dataset were collected from the user logs of the Bing search engine, and associated with the specific sections in Wikipedia, namely the first sections known as the abstracts. We aim to provide a similar yet more exhaustive dataset by broadening the scope to all sections. A notable difference was found between these two corpora for overlapping words (about 11% difference), which was expected due to the artificial question generation in our scheme. Although questions taken from the search queries are more natural, real search queries are inacessible to most researchers. The new annotation scheme proposed here can prove useful for researchers needing to create a corpus for selection-based QA.

Our answer triggering dataset has 5 times more answer candidates per question than WikiQA because WikiQA includes only sections clicked on by users. Manual selection is eliminated from our framework, making our corpus more practical since finding the relevant section no longer rely on the user clicks. In WikiQA, 40.76% of the questions have corresponding answer contexts for answer triggering, as compared to 39.25% in our corpus.

4 Systems

Two models using convolutional neural networks are developed, one is our replication of the best model in Yang et al. (2015), and the other is an improved model using subtree matching (Section 4.1). Two more models using recurrent neural networks are developed, one is our replication of the attentive pooling model in Santos et al. (2016), and the other is a simpler model using one-way attention (Section 4.2). These are inspired by the latest state-of-the-art approaches, providing sensible evaluations.

4.1 Convolutional Neural Networks

Our CNN model is motivated by Yang et al. (2015). First, a convolutional layer is applied on the image of text using the hyperbolic tangent activation function. The image consists of rows standing for consecutive words in two sentences, the question (q) and the answer candidate (a), where the words are represented by their embeddings (Mikolov et al., 2013). Next, the max pooling is applied,[7] and the sentence vectors for q and a are generated. Unlike Yang et al. (2015) who performed the dot product between these two vectors, we added another hidden layer to learn their weights. Finally, the sigmoid activation function is applied and the entire network is trained using the binary cross-entropy.

The CNN score from the output layer is used as one of the features for logistic regression, in addition to the number of overlapping words between q and a, say Ω, Ω normalized by the IDF, and the question length. A threshold is applied on each logistic regression prediction; the candidate with the highest score is considered the answer context if it is above the threshold found during development; otherwise, the model assumes no existence of the answer context in this document for that question.

[7] We also experimented with the average pooling as Yang et al. (2015), which led to a marginally lower accuracy.

Subtree Matching

We propose a subtree matching mechanism for measuring the contextual similarity between two sentences. All sentences are automatically parsed by the NLP4J dependency parser (Choi and McCallum, 2013). First, a set of co-occurring words between q and a, say T, is created. For each $w_i \in T$, w_i's parents (p_i^q, p_i^a), siblings (S_i^q, S_i^a), and children (C_i^q, C_i^a) are extracted from q and a. Then, three matching scores are measured as follows:

$$\mu_p = \sum_{w_i \in T} f_c(p_i^q, p_i^a)$$
$$\mu_s = \sum_{w_i \in T} f_m(\{f_c(x, y) : \forall(x, y) \in S_i^q \times S_i^a\})$$
$$\mu_c = \sum_{w_i \in T} f_m(\{f_c(x, y) : \forall(x, y) \in C_i^q \times C_i^a\})$$

When the word-forms are used as the comparator, $f_c(x, y)$ returns 1 if x and y have the same form; otherwise, 0. When the word embeddings are used as the comparator, $f_c(x, y)$ returns the cosine similarity between x and y. The function f_m takes a list of scores and returns either the `sum`, `avg`, or `max` of the scores. Finally, μ_p, μ_s, and μ_c are used as the additional features to the logistic regression model. Although it adds only 3 more features, our experiments show significant performance gains for both answer sentence selection and answer triggering.

4.2 Recurrent Neural Network

Our RNN model is based on the bidirectional Long Short-Term Memory (LSTM) using attentive pooling introduced by Santos et al. (2016), except that our network uses a gated recurrent unit (GRU; Cho et al. (2014)) instead of LSTM. From our preliminary experiments, we found that GRU converged faster than LSTM while achieving similar performance for these tasks. Let $w_i^q \in q$, $w_j^a \in a$, where q is the question and a is the answer candidate, and $e(w)$ returns the embedding of a word w. Embeddings are encoded by a single bidirectional GRU g that consists of the forward ($\overrightarrow{g}$) and the backward ($\overleftarrow{g}$) GRUs, each with h hidden units. Given w, g outputs the vector concatenation of the hidden states of $\overrightarrow{g}$ and $\overleftarrow{g}$:

$$g(e(w)) = \overrightarrow{g}(e(w)) || \overleftarrow{g}(e(w))$$

Let $c = 2 \cdot h$ represent the dimensionality of the output of g. Then, sentence embedding matrices $Q \in \mathbb{R}^{|q| \times c}$ and $A \in \mathbb{R}^{|a| \times c}$ are generated by g as $Q_i = g(e(w_i^q))$ and $A_j = g(e(w_j^a))$.

Both the attentive pooling and one-way attention models below are trained by minimizing the pairwise hinge ranking loss. In addition, RMSProp is used for the optimization and the ℓ_2 weight penalty is applied on all parameters except for embeddings. All network parameters except the embeddings are initialized using orthogonal initialization.

Attentive Pooling

Attentive Pooling (AP) is a framework-independent two-way attention mechanism that jointly learns a similarity measure between q and a. AP learns the similarity measure over the hidden states of q and a. The AP matrix $H \in \mathbb{R}^{|q| \times |a|}$ has a bilinear form and is followed by a hyperbolic tangent non-linearity, where $U \in \mathbb{R}^{c \times c}$:

$$H = \tanh(Q^T U A)$$

The importance vectors $h^q \in \mathbb{R}^{|q|}$ and $h^a \in \mathbb{R}^{|a|}$ are generated from the column-wise and row-wise max pooling over H, respectively:

$$[h^q]_j = \max_{i \in [1, |q|]} [H_{j,i}]$$

The normalized attention vectors σ^q and σ^a are created by applying the softmax activation function on h^q and h^a:

$$\sigma^q = \frac{\exp([h^q]_j)}{\sum_{i \in [1, |q|]} \exp([h^q]_i)}$$

The final representations $r^q = Q\sigma^q$ and $r^a = A\sigma^a$ for q and a are created using the dot products of the sentence representations and their corresponding attention vectors. The score is computed for each pair using cosine similarity:

$$s(q, a) = \frac{{r^q}^T r^a}{||r^q|| ||r^a||}$$

One-Way Attention

Our one-way attention model is a simplified version of the attentive pooling model above, which is most similar to the global attention model introduced by Luong et al. (2015). We did not use the one-way attention from Tan et al. (2015) to avoid deviating the attention mechanism significantly. Replacing Q with $Q_{|q|}$, the last hidden state of g, H becomes the importance vector h. Again, we create the normalized attention vector σ^a by applying the softmax activation function. The final representations are $r^q = Q_{|q|}$ and $r^a = A\sigma^a$.

5 Experiments

Our systems are evaluated for the answer sentence selection (Section 5.2) and answer triggering (Section 5.3) tasks on both WikiQA and our corpus.

5.1 SelQA: Selection-based QA Corpus

Table 4 shows the distributions of our corpus, called SelQA. Our corpus is split into training (70%), development (10%), and evaluation (20%) sets. The answer triggering data (AT) is significantly larger than the answer sentence selection data (ASS), due to the extra sections added by Task 5 (Section 3.2).

Set	Q	ASS		AT	
		Sec	Sen	Sec	Sen
`TRN`	5,529	5,529	66,438	27,645	205,075
`DEV`	785	785	9,377	3,925	28,798
`TST`	1,590	1,590	19,435	7,950	59,845

Table 4: Distributions of our corpus. Q/Sec/Sen: number of questions/sections/sentences.

5.2 Answer Sentence Selection

Table 5 shows results from ours and the previous approaches on WikiQA. Two metrics are used, mean average precision (MAP) and mean reciprocal rank (MRR), for the evaluation of this task.

Model	Development		Evaluation	
	MAP	MRR	MAP	MRR
CNN_0: baseline	69.93	70.66	65.62	66.46
CNN_1: `avg` + word	**70.75**	**71.46**	67.40	69.30
CNN_2: `avg` + emb	69.22	70.18	**68.78**	**70.82**
RNN_0: one-way	**71.19**	**71.80**	66.64	68.70
RNN_1: attn-pool	67.56	68.31	**67.47**	**68.92**
Yang et al. (2015)	-	-	65.20	66.52
Santos et al. (2016)	-	-	68.86	69.57
Miao et al. (2015)	-	-	68.86	70.69
Yin et al. (2015)	-	-	69.21	71.08
Wang et al. (2016)	-	-	70.58	72.26

Table 5: Answer sentence selection results on the development and evaluation sets of WikiQA.

CNN_0 is our replication of the best model in Yang et al. (2015). CNN_1 and CNN_2 are the CNN models using the subtree matching in Section 4.1, where the comparator of f_c is either the word form or the word embedding respectively, and f_m = `avg`. The subtree matching models consistently outperform the baseline model. Note that among the three metrics of f_m, `avg`, `sum`, and `max`, `avg` outperformed the others in our experiments for the answer sentence selection task although no significant difference was found.

RNN_0 and RNN_1 are the RNN models using the one-way attention and the attentive pooling in Section 4.2. Note that RNN_1 converged much faster than RNN_0 at the same learning rate and fixed number of parameters in our experiments, implying that two-way attention helps with optimization. It is interesting to see how CNN_1 and RNN_0 outperform CNN_2 and RNN_1 respectively on the development set, but not on the evaluation set. This result may be explained by the larger percentage of overlapping words in the development set, enabling the simpler models perform more effectively.

Model	Development		Evaluation	
	MAP	MRR	MAP	MRR
CNN_0: baseline	84.62	85.65	83.20	84.20
CNN_1: `avg` + word	85.04	86.17	84.00	84.94
CNN_2: `avg` + emb	**85.70**	**86.67**	**84.66**	**85.68**
RNN_0: one-way	82.26	83.68	82.06	83.18
RNN_1: attn-pool	**87.06**	**88.25**	**86.43**	**87.59**

Table 6: Answer sent. selection results on SelQA.

Table 6 shows the results achieved by our models on SelQA. CNN_2 outperforms the other CNN models, indicating the power of subtree matching coupled with word embeddings. RNN_1 outperforms RNN_0, indicating the importance of attention over the questions. Unlike the results on WikiQA in Table 5, CNN_2 and RNN_0 show the best performance on both the development and evaluation sets, implying the robustness of these models on our corpus.

Topic	CNN_0	CNN_2	RNN_0	RNN_1	Q
Arts	80.45	82.83	84.22	83.51	135
Country	**87.12**	**89.03**	**87.43**	**93.87**	178
Food	85.30	86.11	84.72	86.74	147
H. Events	**91.72**	**92.61**	**85.95**	**91.52**	164
Movies	84.43	86.50	82.42	88.41	164
Music	81.38	80.39	84.57	84.38	155
Science	86.37	86.50	83.59	84.63	179
Sports	81.83	83.69	79.05	86.86	168
Travel	83.78	86.03	84.29	87.79	165
TV	77.34	81.23	76.18	86.82	135

Table 7: MRR scores on the SelQA evaluation set for answer sentence selection with respect to topics.

Table 7 shows the MRR scores from our models on SelQA with respect to different topics. All models show strength on topics such as 'Country' and 'Historical Events', which is comprehensible since questions in these topics tend to be deterministic. On the other hand, most models show weakness on topics such as 'TV', 'Arts', or 'Music'. This may be due to the fact that not many overlapping words

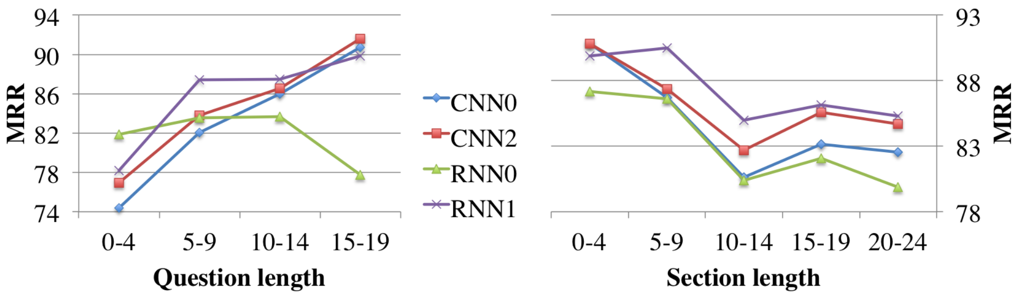

Figure 2: Answer sentence selection on the SelQA evaluation set w.r.t. question and section lengths.

are found between question and answer pairs in these documents, which also consist of many segments caused by bullet points.

Type	CNN_0	CNN_2	RNN_0	RNN_1	Q
What	84.54	85.36	83.50	87.66	678
How	81.92	84.01	82.04	87.32	233
Who	**85.46**	**88.17**	80.36	85.99	195
When	84.21	85.56	**86.16**	**90.35**	180
Where	83.78	87.44	84.59	82.54	85
Why	78.55	82.64	80.61	84.07	41
Misc.	84.17	84.80	85.20	89.66	215

Table 8: MRR scores on the SelQA evaluation set for answer sentence selection w.r.t. question types.

Table 8 shows the MRR scores with respect to question types. The CNN models show strength on the 'who' type, whereas the RNN models show strength on the 'when' type. Each model varies on showing their weakness, which we will explore in the future.

Type	CNN_0	CNN_2	RNN_0	RNN_1	Q
Original	86.70	88.31	85.57	89.90	810
Paraphrase	81.67	83.00	81.12	85.24	789

Table 9: MRR scores on the SelQA evaluation set for answer sentence selection w.r.t. paraphrasing.

Table 9 shows comparisons between questions from Tasks 1 and 2 (original) and Task 3 (paraphrase) in Section 3.2. As expected, noticeable performance drop is found for the paraphrased questions, which have much fewer overlapping words to the answer contexts than the original questions.

Finally, Figure 2 shows the performance difference with respect to question and section lengths. All models except for RNN_0 tend to perform better as questions become longer. This makes sense since longer questions are usually more informative. On the other hand, models generally perform worse as sections become longer, which also makes sense because the models have to select the answer contexts from larger pools.

5.3 Answer Triggering

Due to the characteristic of answer triggering, metrics used for evaluating answer sentence selection are not used here, because those metrics assume that models are always provided with contexts including the answers. Broadly speaking, the answer sentence selection task is a raking problem, while answer triggering is a binary classification task with additional constraints. Thus, the F1-score on the question level was proposed by Yang et al. (2015) as the evaluation for this task, which we follow.

Model	Development P	Development R	Development F1	Evaluation P	Evaluation R	Evaluation F1
CNN_0: baseline	41.86	42.86	42.35	29.70	37.45	32.73
CNN_1: max + word	44.53	45.24	**44.88**	29.77	42.39	34.97
CNN_2: max + emb	43.07	46.83	44.87	29.77	42.39	34.97
CNN_3: max + emb+	44.44	44.44	44.44	29.43	48.56	**36.65**
RNN_1: attn-pool	25.95	38.10	30.87	24.32	47.74	32.22
Yang et al. (2015)	-	-	-	27.96	37.86	32.17

Table 10: Answer triggering results on WikiQA.

Table 10 shows the answer triggering results on WikiQA. Note that RNN_0 using one-way attention was dropped for these experiments because it did not show comparable performance against the others for this task. Interestingly, the CNN model with $f_m = \texttt{max}$ outperformed the other metrics for answer triggering, although avg was found to be the most effective for answer sentence selection. The CNN subtree matching models consistently gave over 2% improvements to the baseline model.

In addition, CNN_3 was experimented by retraining word embeddings (emb+), which performed slightly worse on the development set, but gave another 1.68% improvement on the evaluation set.[8] RNN_1 showed a very similar result to Yang et al. (2015), which was surprising since it performed so much better for answer sentence selection. This can be due to a lack of hyper-parameter optimization, which we leave as a future work.

[8]Retraining word embeddings was not found to be useful for answer sentence selection.

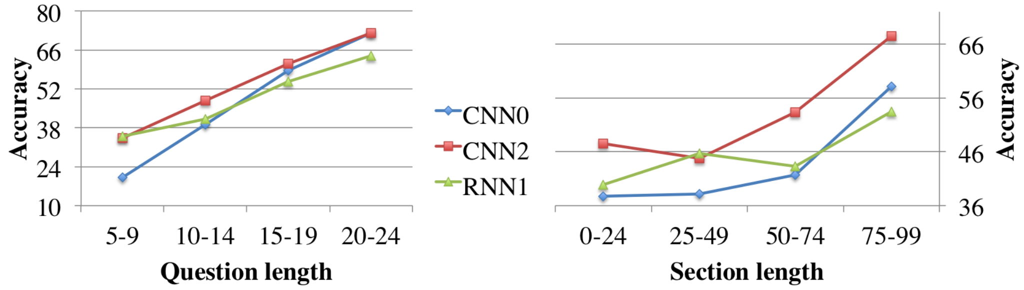

Figure 3: Answer triggering on the SelQA evaluation set w.r.t. question and section lengths.

Table 11 shows the answer triggering results on SelQA. Unlike the results on WikiQA (Table 10), CNN_2 outperforms CNN_3 on our corpus. On the other hand, RNN_1 shows a similar score to Yang et al. (2015) as it does on WikiQA. CNN_2 using subtree matching gives over a 5% improvement to the baseline model, which is significant.

Model	Development			Evaluation		
	P	R	F1	P	R	F1
CNN_0: baseline	50.63	40.60	45.07	52.10	40.34	45.47
CNN_1: `max` + word	48.15	47.99	48.07	52.22	47.30	49.64
CNN_2: `max` + emb	49.32	48.99	**49.16**	53.69	48.38	**50.89**
CNN_3: `max` + emb+	47.16	47.32	47.24	52.14	47.14	49.51
RNN_1: attn-pool	45.52	42.62	44.02	47.96	43.59	45.67

Table 11: Answer triggering results on SelQA.

Table 12 shows the accuracies on SelQA with respect to different topics. The accuracy is measured on the subset of questions that contain at least one answer among candidates; the top ranked sentence is taken and checked for the correct answer. Similar to answer sentence selection, CNN_2 stills shows strength on topics such as 'Country' and 'Historical Events', but the trend is not as clear for the other models. Interestingly, the standard deviation is much smaller for RNN_1 (3.9%) compared to the CNN models (10-12%) although RNN_1's overall performance is lower.

Topic	CNN_0	CNN_2	RNN_1	Q
Arts	27.45	31.37	43.14	135
Country	43.59	**61.54**	38.46	178
Food	31.40	44.19	46.51	147
H. Events	**60.32**	**63.49**	38.10	164
Movies	37.74	45.28	39.62	164
Music	29.31	36.21	44.83	155
Science	45.00	57.50	43.75	179
Sports	50.00	58.11	47.30	168
Travel	42.68	50.00	**48.78**	165
TV	32.79	32.79	39.34	135

Table 12: Accuracies on the SelQA evaluation set for answer triggering with respect to topics.

Table 13 shows the accuracies on SelQA with respect to question types. Interestingly, each model shows different strength on different types, which may suggest a possibility of an ensemble model.

Type	CNN_0	CNN_2	RNN_1	Q
What	40.68	50.19	44.11	678
How	36.63	43.56	44.55	233
Who	**44.94**	50.56	38.20	195
When	33.33	43.06	38.89	180
Where	33.33	**51.85**	40.74	85
Why	42.11	47.37	**57.89**	41
Misc.	44.90	51.02	46.94	215

Table 13: Accuracies on the SelQA evaluation set for answer triggering w.r.t. question types.

Table 14 shows the accuracies on SelQA with respect to paraphrasing, which is similar to the trend found in Table 9 for answer sentence selection.

Type	CNN_0	CNN_2	RNN_1	Q
Original	46.15	55.13	44.36	810
Paraphrase	31.52	38.52	42.21	789

Table 14: Accuracies on the SelQA evaluation set for answer triggering w.r.t. paraphrasing.

6 Conclusion

In this paper we present a new benchmark for two major question answering tasks: answer sentence selection and answer triggering. Several systems using neural networks are developed for the analysis of our corpus. Our analysis shows different aspects about the current QA approaches, beneficial for further enhancement.

Researchers devoted to relatively small datasets reveal useful characteristics of the question answering tasks. Techniques that result in improvements on smaller datasets are often significantly diminished with larger datasets. Current hardware trends and the availability of larger datasets make large scale question answering more accessible. We plan to continue our work on providing large scale corpora for open-domain question answering. We also intend to work towards providing context-aware frameworks for question answering.

References

[Cho et al.2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP'14, pages 1724–1734.

[Choi and McCallum2013] Jinho D. Choi and Andrew McCallum. 2013. Transition-based Dependency Parsing with Selectional Branching. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL'13, pages 1052–1062.

[Feng et al.2015] 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 *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 813–820.

[Heilman and Smith2010] 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*, HLT'10, pages 1011–1019.

[Ho et al.2015] Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. 2015. Incentivize High Quality Crowdwork. In *Proceedings of the 24th World Wide Web Conference*, WWW'15.

[Luong et al.2015] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP'15, pages 1412–1421.

[Miao et al.2015] Yishu Miao, Lei Yu, and Phil Blunsom. 2015. Neural Variational Inference for Text Processing. *arXiv*, arXiv:1511.06038.

[Mikolov et al.2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of Advances in Neural Information Processing Systems 26*, NIPS'13, pages 3111–3119.

[Santos et al.2016] Cícero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR*, abs/1602.03609.

[Severyn and Moschitti2013] Aliaksei Severyn and Alessandro Moschitti. 2013. Automatic Feature Engineering for Answer Selection and Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP'13, pages 458–467.

[Severyn and Moschitti2015] 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.

[Tan et al.2015] Ming Tan, Bing Xiang, and Bowen Zhou. 2015. LSTM-based Deep Learning Models for Non-factoid Answer Selection. *arXiv*, arXiv:1511.04108.

[Wang and Manning2010] Mengqiu Wang and Christopher 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*, COLING'10, pages 1164–1172.

[Wang and Nyberg2015] Di Wang and Eric Nyberg. 2015. A Long Short-Term Memory Model for Answer Sentence Selection in Question Answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, ACL'15, pages 707–712.

[Wang et al.2007] Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL'07, pages 22–32.

[Wang et al.2016] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence Similarity Learning by Lexical Decomposition and Composition. *arXiv*, arXiv:1602.07019.

[Yang et al.2015] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WIKIQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP'15, pages 2013–2018.

[Yin et al.2015] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. *arXiv*, arXiv:1512.05193.

[Yu et al.2014] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep Learning for Answer Sentence Selection. In *Proceedings of the NIPS Deep Learning Workshop*.