SelQA: A New Benchmark for Selection-based Question Answering

Tomasz Jurczyk
Mathematics and Computer Science
Emory University
Atlanta, GA 30322, USA
tomasz.jurczyk@emory.edu

Michael Zhai
Mathematics and Computer Science
Emory University
Atlanta, GA 30322, USA
michael.zhai@emory.edu

Jinho D. Choi
Mathematics and Computer Science
Emory University
Atlanta, GA 30322, USA
jinho.choi@emory.edu

Abstract—This paper presents a new selection-based question answering dataset, SelQA. The dataset consists of questions generated through crowdsourcing and sentence length answers that are drawn from the ten most prevalent topics in the English Wikipedia. We introduce a corpus annotation scheme that enhances the generation of large, diverse, and challenging datasets by explicitly aiming to reduce word co-occurrences between the question and answers. Our annotation scheme is composed of a series of crowdsourcing tasks with a view to more effectively utilize crowdsourcing in the creation of question answering datasets in various domains. Several systems are compared on the tasks of answer sentence selection and answer triggering, providing strong baseline results for future work to improve upon.

I. INTRODUCTION

Selection-based question answering is the task of selecting a segment of text or interchangeably a context from a provided set of contexts that best answers a posed question. Let us define a context as a single document section, a group of contiguous sentences or a single sentence. Selection-based question answering is subdivided into answer sentence selection and answer triggering. Answer sentence selection is defined as ranking sentences that answer a question higher than the irrelevant sentences. Answer triggering is defined as selecting any number ($n >= 0$) of sentences from a set of candidate sentences, where the set of candidate sentences may or may not contain answer sentences. Several corpora have been created for these tasks [1, 2, 3], allowing researchers to build effective question answering systems [4, 5, 6] with the aim of improving reading comprehension through understanding and reasoning of natural language. However, most of these datasets are constrained in the number of examples and scope of topics. We attempt to mitigate these limitations to allow for a more thorough reading comprehension evaluation of open-domain question answering systems.

This paper presents a new corpus with annotated question answering examples of various topics drawn from Wikipedia. An effective annotation scheme is proposed to create a large corpus that is both challenging and realistic. Questions are additionally annotated with their corresponding topics, types, and paraphrases that enable comprehensive analyses of system performance on the answer sentence selection and answer 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 [2], for a fair comparison to previous work. The contributions of this work include:[1]

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

II. RELATED WORK

The TREC QA competition datasets have been a popular choice for evaluating answer sentence selection.[2] Wang et al. [1] 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. [2] introduced a larger dataset, WikiQA, consisting of questions collected from the user logs of the Bing search engine. Our corpus is similar to WikiQA but covers more diverse topics, consists of 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 extracting contexts from the entire article instead of from only the abstract. Feng et al. [3] distributed another dataset, InsuranceQA, including questions in the insurance domain. WikiQA introduced the task of answer triggering and was the only answer triggering dataset. Our corpus provides a new automatically generated answer triggering dataset.

Due to increasing complexity in question answering, deep learning has become a popular trend in solving difficult problems. Yu et al. [7] proposed a convolutional neural network with a single convolution layer, average pooling and logistic regression at the end for factoid question answering. Further, more convolutional neural network based frameworks have been proposed as solutions for question answering [8, 9, 10, 11, 12].
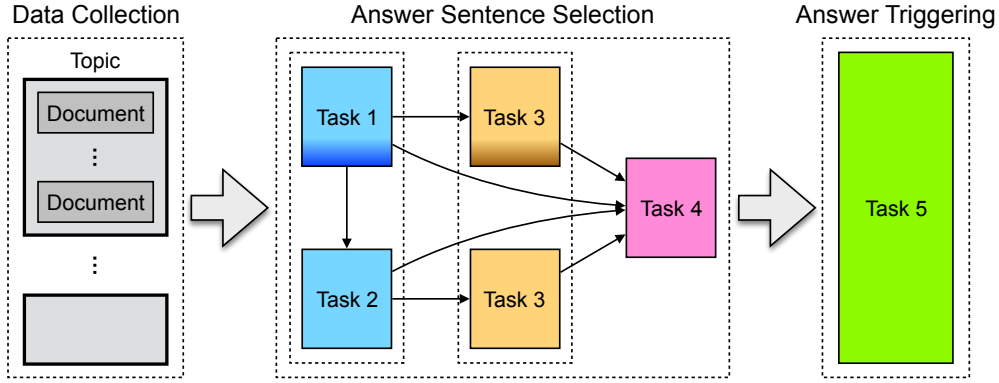
---

[1] https://github.com/emorynlp/question-answering
[2] http://trec.nist.gov/data/qa.html

Fig. 1. The overview of our data collection (Section III-A) and annotation scheme (Section III-B).

TABLE I
LEXICAL STATISTICS OF OUR CORPUS.

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

Our convolutional neural network model is inspired by the previous work utilizing the tree-edit distance and the tree kernel [13, 14, 15], although we introduce a different way of performing subtree matching facilitating word embeddings.

III. 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.

A. Data Collection

A total of 486 articles are uniformly sampled from the following 10 topics of 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 segmented into sentences by the open-source toolkit, NLP4J.[5] In our corpus, documents refer to individual sections in the Wikipedia articles.

B. 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 only Elasticsearch (Task 5; see Figure 1 for the overview).

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

Task 1

Approximately two thousand sections are randomly selected from the 486 articles in Section III-A. 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, annotators are instructed to generate a question that can be answered in one or more sentences in the provided section, and select the corresponding sentence or sentences that answer the question. The annotators are provided with the instructions, the topic, the article title, the section title, and the list of numbered sentences in the section (Table II).

Task 2

Annotators are asked to create another set of ≈2K questions from the same selected sections excluding the sentences selected as answers in Task 1. The goal of Task 2 is to generate questions that can be answered from sentences different from those used to answer questions generated in the Task 1. The annotators are provided with the same information as in Task 1, except that the sentences used as the answer contexts in Task 1 are crossed out (line 1 in Table II). 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 tend to generate questions with some lexical overlap with the corresponding contexts. The intention of this task is to mitigate the effects of annotators' tendency to generating questions with similar vocabulary and phrasing to answer contexts. This is a necessary step in creating a corpus that evaluates reading comprehension rather than ability to model word co-occurrences. The annotators are provided with the previously generated questions and answer contexts and are instructed to paraphrase these questions using different terms.

Task 4

Most questions generated by Tasks 1-3 are of high quality, that is they can be answered by a human when given the corresponding contexts; however, there are some questions that are ambiguous in meaning and difficult for humans to answer

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?

	$\mathbf{Q_s}$	$\mathbf{Q_m}$	$\mathbf{Q_{s+m}}$	$\mathbf{\Omega_q}$	$\mathbf{\Omega_a}$	$\mathbf{\Omega_f}$	**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	-	-

correctly. These difficult questions often assuming incorrectly that the related sections are provided with the questions. For instance, it is impossible to answer the question from Task 3.1 in Table II unless the related section is provided with the question. These questions are sent to the annotators for revision.

Elasticsearch, a Lucene-based open-source search engine is used to find ambiguous questions[6]. 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.

C. Corpus Analysis

The entire annotation took about 130 hours, costing $770 in total; each mechanical turk job took on average approximately 1 minute and costed about 10¢. A total 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 III). 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 [16], 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 [2]. 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% higher), 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 inaccessible to most researchers. The new proposed annotation scheme can prove useful for researchers needing to create a corpus for selection-based QA.

---

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

Our answer triggering dataset contains 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. In WikiQA, 40.76% of the questions have corresponding answer contexts for answer triggering, as compared to 39.25% in ours.

IV. SYSTEMS

Two models using convolutional neural networks are developed, one is our replication of the best model in Yang et al. [2], and the other is an improved model using subtree matching (Section IV-A). Two more models using recurrent neural networks are developed, one is our replication of the attentive pooling model in Santos et al. [17], and the other is a simpler model using one-way attention (Section IV-B). These are inspired by the latest state-of-the-art approaches, providing sensible evaluations.

A. Convolutional Neural Networks

Our CNN model is motivated by Yang et al. [2]. First, a convolutional layer is applied on the image of text (embedding) 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 [18]. For our experiments, we concatenate 40 question word embeddings and 40 answer word embeddings. Next, the max pooling is applied,[7] and the sentence vectors for q and a are generated. Unlike Yang et al. [2] 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.

Next, we use a logistic regression model, where the CNN score from the output layer is used as one of the features. Other features in the logistic regression are the raw and idf normalized numbers of overlapping words between q and a and the question length. While the logistic regression model could be merged directly with our CNN model, it has been empirically shown that it is more effective to construct this last phase as a separate model.

For the answer sentence selection task, the predictions for each question are treated as a ranking and mean average precision (MAP) and mean reciprocal rank (MRR) are calculated (Section V-B). On the other hand, in the answer triggering task (Section V-C) a threshold is applied to the logistic regression scores; the candidate with the highest score is considered the answer if it is above the threshold found during development; otherwise, the model predicts the answer context does not exist.

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 [19]. First, a set of co-occurring words between q and a, say T, is created. For each $w_i^o \in T$, w_i^o's parents (p_i^q, p_i^a), siblings (S_i^q, S_i^a), and children (C_i^q, C_i^a) are

---

[7]We also experimented with the average pooling as [2], which led to a marginally lower accuracy.

Input: T: a set of co-occurring words between a question and answer.
D^q, D^a: sets of slices for a question and answer.
f_m: a metrics function.
f_c: a comparator function.
Output: S_{dep}: A triplet of dependency similarity.
$S_{dep} \leftarrow [0, 0, 0]$;
foreach *word w_i^o in T* **do**
  $p_i^q \leftarrow$ getParent(D_i^q)
  $p_i^a \leftarrow$ getParent(D_i^a)
  $S_{dep}[0] \leftarrow S_{dep}[0] + f_c(p_i^q, p_i^a)$
  $S_i^q \leftarrow$ getSiblings(D_i^q)
  $S_i^a \leftarrow$ getSiblings(D_i^a)
  $vals \leftarrow []$
  foreach *sibling s_j^q in S_i^q* **do**
    foreach *sibling s_k^a in S_i^a* **do**
      $vals$.append($f_c(s_j^q, s_k^a)$)
    end
  end
  $S_{dep}[1] \leftarrow S_{dep}[1] + f_m(vals)$
  $C_i^q \leftarrow$ getChildren(D_i^q)
  $C_i^a \leftarrow$ getChildren(D_i^a)
  $vals \leftarrow []$
  foreach *child c_j^q in C_i^q* **do**
    foreach *child c_k^a in C_i^a* **do**
      $vals$.append($f_c(c_j^q, c_k^a)$)
    end
  end
  $S_{dep}[2] \leftarrow S_{dep}[2] + f_m(vals)$
end
Algorithm 1: Algorithm of our subtree matching mechanism

---

extracted from the dependency slices of q and 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, the triplet S_{dep} is used as the additional features to the logistic regression model. Algorithm 1 presents the entire process in detail. Although our subtree matching mechanism adds only 3 more features, our experiments show significant performance gains for both the answer sentence selection and answer triggering strengthening our hypothesis that to solve question answering problems more effectively, deeper contextual similarity is required.

B. Recurrent Neural Network

Our RNN model is based on the bidirectional Long Short-Term Memory (LSTM) using attentive pooling introduced by Santos et al. [17], except that our network uses a gated recurrent unit (GRU; Cho et al. [20]) 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:

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

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. [21]. We did not use the one-way attention from Tan et al. [22] 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 $r^a = A\sigma^a$.

V. EXPERIMENTS

Our systems are evaluated for the answer sentence selection (Section V-B) and answer triggering (Section V-C) tasks on both WikiQA and our corpus.

A. SelQA: Selection-based QA Corpus

Table IV 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 III-B).

TABLE IV
DISTRIBUTIONS OF OUR CORPUS. Q/SEC/SEN: NUMBER OF
QUESTIONS/SECTIONS/SENTENCES.

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

B. Answer Sentence Selection

Table V shows results from ours and the previous approaches on WikiQA. Two metrics are used, MAP and MRR for the evaluation of this task. CNN_0 is our replication of the best model in Yang et al. [2]. CNN_1 and CNN_2 are the CNN models using the subtree matching in Section IV-A, where the comparator of f_c is either the word form or the word embedding respectively, and $f_m = \texttt{avg}$. The subtree matching models consistently outperforms the baseline model. Note that among the three metrics of f_m, $\texttt{avg}$, $\texttt{sum}$, and $\texttt{max}$, $\texttt{avg}$ outperformed the others in our experiments for the answer sentence selection task although no significant differences were found. RNN_0 and RNN_1 are the RNN models using the one-way attention and the attentive pooling in Section IV-B. 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 assists 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.

TABLE V
ANSWER SENTENCE SELECTION RESULTS ON THE DEVELOPMENT AND
EVALUATION SETS OF WIKIQA.

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. [2]	-	-	65.20	66.52
Santos et al. [17]	-	-	68.86	69.57
Miao et al. [23]	-	-	68.86	70.69
Yin et al. [24]	-	-	69.21	71.08
Wang et al. [25]	-	-	70.58	72.26
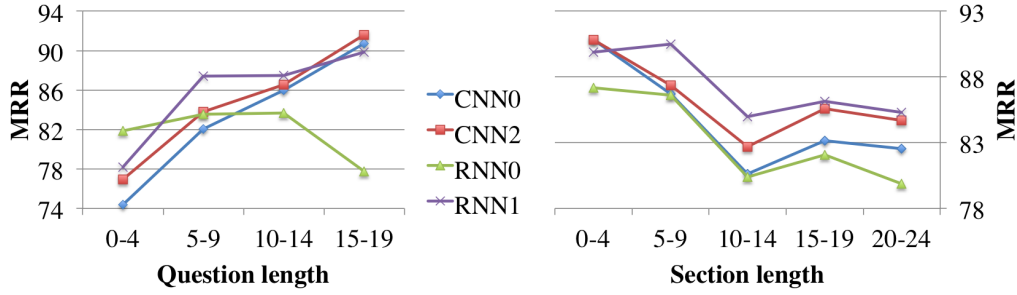
Fig. 2. Answer sentence selection on the SelQA evaluation set w.r.t. question and section lengths.

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 VI shows the results achieved by our models on SelQA. CNN_2 outperforms the other CNN models, indicating the effectiveness 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 V, 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.

Table VII 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 can be explained by the small variety of questions in these topics. 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 are found between question and answer pairs in these documents, which also consist of many segments caused by bullet points.

TABLE VII
MRR SCORES ON THE SELQA EVALUATION SET FOR ANSWER SENTENCE
SELECTION WITH RESPECT TO TOPICS.

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 VIII
MRR SCORES ON THE SELQA EVALUATION SET FOR ANSWER SENTENCE
SELECTION W.R.T. PARAPHRASING.

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 VIII shows comparisons between questions from Tasks 1 and 2 (original) and Task 3 (paraphrase) in Section III-B. 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.

TABLE IX
MRR SCORES ON THE SELQA EVALUATION SET FOR ANSWER SENTENCE
SELECTION W.R.T. QUESTION TYPES.

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 IX 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. 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, 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 set of entities.

C. Answer Triggering

Due to the nature 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
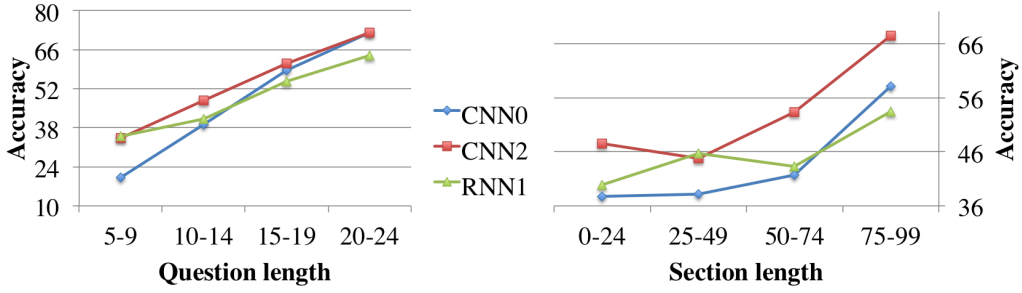
Fig. 3. Answer triggering on the SelQA evaluation set w.r.t. question and section lengths.

TABLE X
ANSWER TRIGGERING RESULTS ON WIKIQA.

Model	Development			Evaluation		
	P@1	R	F1	P@1	R	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. [2]	-	-	-	27.96	37.86	32.17

TABLE XI
ANSWER TRIGGERING RESULTS ON SELQA.

Model	Development			Evaluation		
	P@1	R	F1	P@1	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

answer sentence selection task is a ranking 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. [2] as the evaluation for this task, which we follow.

Table X 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 = 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, retraining word embeddings (CNN_3 + emb+), performs 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. [2], 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 for future work.

Table XI shows the answer triggering results on SelQA. Unlike the results on WikiQA (Table X), CNN_2 outperforms CNN_3 on our corpus. On the other hand, RNN_1 shows a similar score to Yang et al. [2] as it does on WikiQA. CNN_2 using subtree matching gives over a 5% improvement to the baseline model, which is significant.

Table XII shows the precision@1 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 checked in the set of correct answers. Similar to answer sentence selection, CNN_2 stills

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

shows strength on topics such as 'Country' and 'Historical Events', but the trend is not as clear for the other models. The worst performing topics are 'TV', 'Music' and 'Art'. Such a noticeable difference might be caused by the unusual semantic sentence constructions of the text. Sections in these categories often contain listings, bullet-pointed texts etc., which is problematic for the models to properly take care of. How to correctly understand and solve question from such context will be a challenge to the future systems. Also, 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.

TABLE XII
PRECISION@1 ON THE SELQA EVALUATION SET FOR ANSWER TRIGGERING WITH RESPECT TO TOPICS.

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 XIII shows the precision@1 on SelQA with respect to paraphrasing, which is similar to the trend found in Table VIII for answer sentence selection.

Table XIV shows the precision@1 on SelQA with respect to question types. Interestingly, each model shows different strength on different types, which may suggest further improvements using an ensemble. Finally, Figure 3 shows the performance difference with respect to question and section lengths for the answer triggering task. All the models tend to

TABLE XIII
PRECISION@1 ON THE SELQA EVALUATION SET FOR ANSWER
TRIGGERING W.R.T. PARAPHRASING.

Type	CNN$_0$	CNN$_2$	RNN$_1$	Q
Original	46.15	55.13	44.36	810
Paraphrase	31.52	38.52	42.21	789

perform better as questions become longer. When the section becomes longer, the precision increases. We hypothesize that such a behavior might be caused by the fact that it is easier for the models to decide whether the context of the section is the same as the context of the question when there is more information (sentences) in the section. Thus, this phenomenon is related to the task of answer triggering, where the model not only chooses the sentence with the answer, but must decide if the context matches first.

TABLE XIV
PRECISION@1 ON THE SELQA EVALUATION SET FOR ANSWER
TRIGGERING W.R.T. QUESTION TYPES.

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

VI. 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 of current QA approaches that benefit further research.

We plan to continue our work on providing large scale corpora for open-domain question answering. Also, we intend to continue working towards providing context-aware frameworks.

REFERENCES

[1] M. Wang, N. A. Smith, and T. Mitamura, "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*, ser. EMNLP-CoNLL'07, 2007, pp. 22–32.

[2] Y. Yang, W.-t. Yih, and C. Meek, "WIKIQA: A Challenge Dataset for Open-Domain Question Answering," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP'15, 2015, pp. 2013–2018.

[3] M. Feng, B. Xiang, M. R. Glass, L. Wang, and B. Zhou, "Applying Deep Learning to Answer Selection: A Study and An Open Task," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2015, pp. 813–820.

[4] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman, "Deep Learning for Answer Sentence Selection," in *Proceedings of the NIPS Deep Learning Workshop*, 2014.

[5] D. Wang and E. Nyberg, "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*, ser. ACL'15, 2015, pp. 707–712.

[6] A. Severyn and A. Moschitti, "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*, ser. SIGIR '15, 2015, pp. 373–382.

[7] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman, "Deep learning for answer sentence selection," *arXiv preprint arXiv:1412.1632*, 2014.

[8] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher, and H. Daumé III, "A neural network for factoid question answering over paragraphs," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 633–644.

[9] L. Dong, F. Wei, M. Zhou, and K. Xu, "Question answering over freebase with multi-column convolutional neural networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 1, 2015, pp. 260–269.

[10] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "Abcnn: Attention-based convolutional neural network for modeling sentence pairs," *arXiv preprint arXiv:1512.05193*, 2015.

[11] W.-t. Yih, X. He, and C. Meek, "Semantic parsing for single-relation question answering," in *Proceedings of ACL*, 2014.

[12] P. Blunsom, E. Grefenstette, N. Kalchbrenner *et al.*, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014.

[13] M. Heilman and N. A. Smith, "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*, ser. HLT'10, 2010, pp. 1011–1019.

[14] M. Wang and C. Manning, "Probabilistic Tree-Edit Models with Structured Latent Variables for Textual Entailment and Question Answering," in *Proceedings of the 23rd International Conference on Computational Linguistics*, ser. COLING'10, 2010, pp. 1164–1172.

[15] A. Severyn and A. Moschitti, "Automatic Feature Engineering for Answer Selection and Extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP'13, 2013, pp. 458–467.

[16] C.-J. Ho, A. Slivkins, S. Suri, and J. W. Vaughan, "Incentivize High Quality Crowdwork," in *Proceedings of the 24th World Wide Web Conference*, ser. WWW'15, 2015.

[17] C. N. d. Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," *CoRR*, vol. abs/1602.03609, 2016. [Online]. Available: http://arxiv.org/abs/1602.03609

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Proceedings of Advances in Neural Information Processing Systems 26*, ser. NIPS'13, 2013, pp. 3111–3119.

[19] J. D. Choi and A. McCallum, "Transition-based Dependency Parsing with Selectional Branching," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ser. ACL'13, 2013, pp. 1052–1062.

[20] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP'14, 2014, pp. 1724–1734.

[21] T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP'15, 2015, pp. 1412–1421.

[22] M. Tan, B. Xiang, and B. Zhou, "LSTM-based Deep Learning Models for Non-factoid Answer Selection," *arXiv*, vol. arXiv:1511.04108, 2015.

[23] Y. Miao, L. Yu, and P. Blunsom, "Neural Variational Inference for Text Processing," *arXiv*, vol. arXiv:1511.06038, 2015.

[24] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs," *arXiv*, vol. arXiv:1512.05193, 2015.

[25] Z. Wang, H. Mi, and A. Ittycheriah, "Sentence Similarity Learning by Lexical Decomposition and Composition," *arXiv*, vol. arXiv:1602.07019, 2016.