Denoising Distantly Supervised Open-Domain Question Answering

Yankai Lin, Haozhe Ji, Zhiyuan Liu[*], Maosong Sun
State Key Lab on Intelligent Technology and Systems,
Department of Computer Science and Technology,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing, China
{linyk14,jhz16}@mails.tsinghua.edu.cn, {liuzy,sms}@tsinghua.edu.cn

Abstract

Distantly supervised open-domain question answering (DS-QA) aims to find answers in collections of unlabeled text. Existing DS-QA models usually retrieve related paragraphs from a large-scale corpus and apply reading comprehension technique to extract answers from the most relevant paragraph. They ignore the rich information contained in other paragraphs. Moreover, distant supervision data inevitably accompanies with the wrong labeling problem, and these noisy data will substantially degrade the performance of DS-QA. To address these issues, we propose a novel DS-QA model which employs a paragraph selector to filter out those noisy paragraphs and a paragraph reader to extract the correct answer from those denoised paragraphs. Experimental results on real-world datasets show that our model can capture useful information from noisy data and achieve significant improvements on DS-QA as compared to all baselines. The source code and data of this paper can be obtained from `https://github.com/thunlp/OpenQA`

1 Introduction

Reading comprehension, which aims to answer questions about a document, has recently become a major focus of NLP research. Many reading comprehension systems (Chen et al., 2016; Dhingra et al., 2017a; Cui et al., 2017; Shen et al., 2017; Wang et al., 2017) have been proposed and achieved promising results since their multi-layer architectures and attention mechanisms allow them to reason for the question. To some extent, reading comprehension has shown the ability of recent neural models for reading, processing, and comprehending natural language text.

Despite their success, existing reading comprehension systems rely on pre-identified relevant texts, which do not always exist in real-world question answering (QA) scenarios. Hence, reading comprehension technique cannot be directly applied to the task of open domain QA. In recent years, researchers attempt to answer open-domain questions with a large-scale unlabeled corpus. Chen et al. (2017) propose a distantly supervised open-domain question answering (DS-QA) system which uses information retrieval technique to obtain relevant text from Wikipedia, and then applies reading comprehension technique to extract the answer.

Although DS-QA proposes an effective strategy to collect relevant texts automatically, it always suffers from the noise issue. For example, for the question “Which country’s capital is Dublin?”, we may encounter that: (1) The retrieved paragraph “*Dublin is the largest city of Ireland ...*” does not actually answer the question; (2) The second “*Dublin*” in the retrieved paragraph ‘*Dublin is the capital of Ireland. Besides, Dublin is one of the famous tourist cities in Ireland and ...*” is not the correct token of the answer. These noisy paragraphs and tokens are regarded as valid instances in DS-QA. To address this issue, Choi et al. (2017) separate the answer generation in DS-QA into two modules including selecting a target paragraph in document and extracting the correct answer from the target paragraph by reading comprehension. Further, Wang et al. (2018a) use reinforcement learning to train target paragraph selection and answer extraction jointly.

These methods only extract the answer according to the most related paragraph, which will lose a large amount of rich information contained in

[*] Corresponding author: Zhiyuan Liu

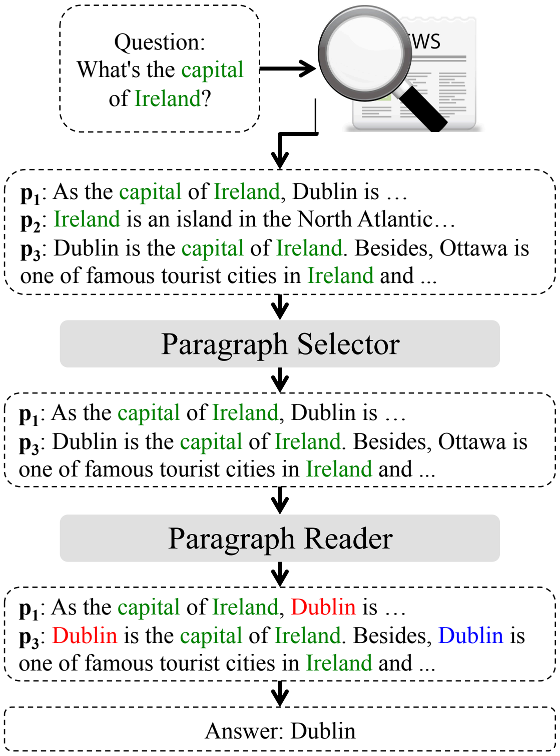
Question:
What's the capital of Ireland?
WS
p1: As the capital of Ireland, Dublin is …
p2: Ireland is an island in the North Atlantic…
p3: Dublin is the capital of Ireland. Besides, Ottawa is one of famous tourist cities in Ireland and ...
Paragraph Selector
p1: As the capital of Ireland, Dublin is …
p3: Dublin is the capital of Ireland. Besides, Ottawa is one of famous tourist cities in Ireland and ...
Paragraph Reader
p1: As the capital of Ireland, Dublin is …
p3: Dublin is the capital of Ireland. Besides, Dublin is one of famous tourist cities in Ireland and ...
Answer: Dublin

Figure 1: An overview of our model. For the question 'What's the capital of Dublin?", our paragraph selector selects two paragraphs p_1 and p_3 which actually correspond to the question from all retrieved paragraphs. And then our paragraph reader extracts the correct answer "Dublin" (in red color) from all selected paragraphs. Finally, our system aggregates the extracted results and obtains the final answer.

those neglected paragraphs. In fact, the correct answer is often mentioned in multiple paragraphs, and different aspects of the question may be answered in several paragraphs. Therefore, Wang et al. (2018b) propose to further explicitly aggregate evidence from across different paragraphs to re-rank extracted answers. However, the reranking approach still relies on the answers obtained by existing DS-QA systems, and fails to solve the noise problem of DS-QA substantially.

To address these issues, we propose a coarse-to-fine denoising model for DS-QA. As illustrated in Fig. 1, our system first retrieves paragraphs according to the question from a large-scale corpus via information retrieval. After that, to utilize all informative paragraphs, we adopt a fast paragraph selector to skim all retrieved paragraphs and filter out those noisy ones. And then we apply a precise paragraph reader to perform careful reading in each selected paragraph for extracting the answer. Finally, we aggregate the derived results of all chosen paragraphs to obtain the final answer. The fast skimming of our paragraph selector and intensive reading of our paragraph reader in our method enables DS-QA to denoise noisy paragraphs as well as maintaining efficiency.

The experimental results on real-world datasets including Quasar-T, SearchQA and TriviaQA show that our system achieves significant and consistent improvement as compared to all baseline methods by aggregating extracted answers of all informative paragraphs. In particular, we show that our model can achieve comparable performance by selecting a few informative paragraphs, which greatly speeds up the whole DS-QA system. We will publish all source codes and datasets of this work on Github for further research explorations.

2 Related Work

Question answering is one of the most important tasks in NLP. Many efforts have been invested in QA, especially in open-domain QA. Open-domain QA has been first proposed by (Green Jr et al., 1961). The task aims to answer open-domain questions using external resources such as collections of documents (Voorhees et al., 1999), webpages (Kwok et al., 2001; Chen and Van Durme, 2017), structured knowledge graphs (Berant et al., 2013a; Bordes et al., 2015) or automatically extracted relational triples (Fader et al., 2014).

Recently, with the development of machine reading comprehension technique (Chen et al., 2016; Dhingra et al., 2017a; Cui et al., 2017; Shen et al., 2017; Wang et al., 2017), researchers attempt to answer open-domain questions via performing reading comprehension on plain texts. Chen et al. (2017) propose a DS-QA system, which retrieves relevant texts of the question from a large-scale corpus and then extracts answers from these texts using reading comprehension models. However, the retrieved texts in DS-QA are always noisy which may hurt the performance of DS-QA. Hence, Choi et al. (2017) and Wang et al. (2018a) attempt to solve the noise problem in DS-QA via separating the question answering into paragraph selection and answer extraction and they both only select the most relevant paragraph among all retrieved paragraphs to extract answers. They lose a large amount of rich information contained in those neglected paragraphs. Hence, Wang et al. (2018b) propose strength-base and coverage-based re-ranking approaches, which can aggregate the results extracted from each paragraph by existing DS-QA system to better determine the answer. However, the method relies on the pre-extracted answers of existing DS-QA models and still suffers from the noise issue in distant supervision data because it considers all retrieved paragraphs indiscriminately. Different from these methods, our model employs a paragraph selector to filter out those noisy paragraphs and keep those informative paragraphs, which can make full use of the noisy DS-QA data.

Our work is also inspired by the idea of coarse-to-fine models in NLP. Cheng and Lapata (2016) and Choi et al. (2017) propose a coarse-to-fine model, which first selects essential sentences and then performs text summarization or reading comprehension on the chosen sentences respectively. Lin et al. (2016) utilize selective attention to aggregate the information of all sentences to extract relational facts. Yang et al. (2016) propose a hierarchical attention network which has two levels of attentions applied at the word and sentence level for document classification. Our model also employs a coarse-to-fine model to handle the noise issue in DS-QA, which first selects informative retrieved paragraphs and then extracts answers from those selected paragraphs.

3 Methodology

In this section, we will introduce our model in details. Our model aims to extract the answer to a given question in the large-scale unlabeled corpus. We first retrieve paragraphs corresponding to the question from the open-domain corpus using information retrieval technique, and then extract the answer from these retrieved paragraphs.

Formally, given a question $q = (q^1, q^2, \cdots, q^{|q|})$, we retrieve m paragraphs which are defined as $P = \{p_1, p_2, \cdots, p_m\}$ where $p_i = (p_i^1, p_i^2, \cdots, p_i^{|p_i|})$ is the i-th retrieved paragraph. Our model measures the probability of extracting answer a given question q and corresponding paragraph set P. As illustrated in Fig. 1, our model contains two parts:

1. Paragraph Selector. Given the question q and the retrieved paragraph P, the paragraph selector measures the probability distribution $\Pr(p_i|q, P)$ over all retrieved paragraphs, which is used to select the paragraph that really contains the answer of question q.

2. Paragraph Reader. Given the question q and a paragraph p_i, the paragraph reader calculates the probability $\Pr(a|q, p_i)$ of extracting answer a through a multi-layer long short-term memory network.

Overall, the probability $\Pr(a|q, P)$ of extracting answer a given question q can be calculated as:

$$\Pr(a|q, P) = \sum_{p_i \in P} \Pr(a|q, p_i) \Pr(p_i|q, P). \quad (1)$$

3.1 Paragraph Selector

Since the wrong labeling problem inevitably occurs in DS-QA data, we need to filter out those noisy paragraphs when exploiting the information of all retrieved paragraphs. It is straightforward that we need to estimate the confidence of each paragraph. Hence, we employ a paragraph selector to measure the probability of each paragraph containing the answer among all paragraphs.

Paragraph Encoding. We first represent each word p_i^j in the paragraph p_i as a word vector $\mathbf{p}_i^j$, and then feed each word vector into a neural network to obtain the hidden representation $\hat{\mathbf{p}}_i^j$. Here, we adopt two types of neural networks including:
1. Multi-Layer Perceptron (MLP)

$$\hat{\mathbf{p}}_i^j = \text{MLP}(\mathbf{p}_i^j), \quad (2)$$

2. Recurrent Neural Network (RNN)

$$\{\hat{\mathbf{p}}_i^1, \hat{\mathbf{p}}_i^2, \cdots, \hat{\mathbf{p}}_i^{|p_i|}\} = \text{RNN}(\{\mathbf{p}_i^1, \mathbf{p}_i^2, \cdots, \mathbf{p}_i^{|p_i|}\}), \quad (3)$$

where $\hat{\mathbf{p}}_i^j$ is expected to encode semantic information of word p_i^j and its surrounding words. For RNN, we select a single-layer bidirectional long short-term memory network (LSTM) as our RNN unit, and concatenate the hidden states of all layers to obtain $\hat{\mathbf{p}}_i^j$.

Question Encoding. Similar to paragraph encoding, we also represent each word q^i in the question as its word vector $\mathbf{q}^i$, and then fed them into a MLP:

$$\hat{\mathbf{q}}_i^j = \text{MLP}(\mathbf{q}_i^j), \quad (4)$$

or a RNN:

$$\{\hat{\mathbf{q}}^1, \hat{\mathbf{q}}^2, \cdots, \hat{\mathbf{q}}^{|q|}\} = \text{RNN}(\{\mathbf{q}^1, \mathbf{q}^2, \cdots, \mathbf{q}^{|q|}\}). \quad (5)$$

where $\hat{\mathbf{q}}^j$ is the hidden representation of the word q^j and is expected to encode the context information of it. After that, we apply a self attention operation on the hidden representations to obtain the final representation $\mathbf{q}$ of the question q:

$$\hat{\mathbf{q}} = \sum_j \alpha^j \hat{\mathbf{q}}^j, \tag{6}$$

where α_j encodes the importance of each question word and is calculated as:

$$\alpha_i = \frac{\exp(\mathbf{w}\mathbf{q}_i)}{\sum_j \exp(\mathbf{w}\mathbf{q}_j)}, \tag{7}$$

where $\mathbf{w}_b$ is a learned weight vector.

Next, we calculate the probability of each paragraph via a max-pooling layer and a softmax layer:

$$\Pr(p_i|q,P) = \text{softmax}\left(\max_j(\hat{\mathbf{p}}_i^j \mathbf{W}\mathbf{q})\right), \tag{8}$$

where $\mathbf{W}$ is a weight matrix to be learned.

3.2 Paragraph Reader

The paragraph reader aims to extract answers from a paragraph p_i. Similar to paragraph reader, we first encode each paragraph p_i as $\{\bar{\mathbf{p}}_i^1, \bar{\mathbf{p}}_i^2, \cdots, \bar{\mathbf{p}}_i^{|p_i|}\}$ through a multi-layers bidirectional LSTM . And we also obtain the question embedding $\bar{\mathbf{q}}$ via a self-attention multi-layers bidirectional LSTM.

The paragraph reader aims to extract the span of tokens which is most likely the correct answer. And we divide it into predicting the start and end position of the answer span. Hence, the probability of extracting answer a of the question q from the given the paragraph p_i can be calculated as:

$$\Pr(a|q,p_i) = P_s(a_s)P_e(a_e), \tag{9}$$

where a_s and a_e indicate the start and end positions of answer a in the paragraph, $P_s(a_s)$ and $P_e(a_e)$ are the probabilities of a_s and a_e being start and end words respectively, which is calculated by:

$$P_s(j) = \text{softmax}(\bar{\mathbf{p}}_i^j \mathbf{W}_s \bar{\mathbf{q}}), \tag{10}$$

$$P_e(j) = \text{softmax}(\bar{\mathbf{p}}_i^j \mathbf{W}_e \bar{\mathbf{q}}), \tag{11}$$

where $\mathbf{W}_s$ and $\mathbf{W}_e$ are two weight matrices to be learned. In DS-QA, since we didn't label the position of the answer manually, we may have several tokens matched to the correct answer in a paragraph. Let $\{(a_s^1, a_e^1), (a_s^2, a_e^2), \cdots, (a_s^{|a|}, a_e^{|a|})\}$ be the set of the start and end positions of the tokens matched to answer a in the paragraph p_i. The equation (9) is further defined using two ways:

(1) **Max**. That is, we assume that only one token in the paragraph indicates the correct answer. In this way, the probability of extracting the answer a can defined by maximizing the probability of all candidate tokens:

$$\Pr(a|q,p_i) = \max_j \mathop{\Pr}_s(a_s^j) \mathop{\Pr}_e(a_e^j) \tag{12}$$

(2) **Sum**. In this way, we regard all tokens matched to the correct answer equally. And we define the answer extraction probability as:

$$\Pr(a|q,p_i) = \sum_j \mathop{\Pr}_s(a_s^j) \mathop{\Pr}_e(a_e^j). \tag{13}$$

Our paragraph reader model is inspired by a previous machine reading comprehension model, Attentive Reader described in (Chen et al., 2016). In fact, other reading comprehension models can also be easily adopted as our paragraph reader. Due to the space limit, in this paper, we only explore the effectiveness of Attentive Reader.

3.3 Learning and Prediction

For the learning objective, we define a loss function L using maximum likelihood estimation:

$$L(\theta) = -\sum_{(\bar{a},q,P)\in T} \log \Pr(a|q,P) - \alpha R(P), \tag{14}$$

where θ indicates the parameters of our model, a indicates the correct answer, T is the whole training set and $R(P)$ is a regularization term over the paragraph selector to avoid its overfitting. Here, $R(P)$ is defined as the KL divergence between $\Pr(p_i|q,P)$ and a probability distribution $\mathcal{X}$ where $\mathcal{X}_i = \frac{1}{c_P}$ (c_P is the number of paragraphs containing correct answer in P) if the paragraph contains correct answer, otherwise 0. Specifically, $R(P)$ is defined as:

$$R(P) = \sum_{p_i \in P} \mathcal{X}_i \log \frac{\mathcal{X}_i}{\Pr(p_i|q,P)}. \tag{15}$$

To solve the optimization problem, we adopt Adamax to minimize the objective function as described in (Kingma and Ba, 2015).

During testing, we extract the answer $\hat{a}$ with the highest probability as below:

$$\begin{aligned} \hat{a} &= \arg\max_a \Pr(a|q,P) \\ &= \arg\max_a \sum_{p_i \in P} \Pr(a|q,p_i) \Pr(p_i|q,P) \end{aligned} \tag{16}$$

Here, the paragraph selector can be viewed as a fast skimming over all paragraphs, which determines the probability distribution of containing the answer for each paragraph. Hence, we can simply aggregate the predicting results from those paragraphs with higher probabilities for acceleration.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate our model on five public open-domain question answering datasets.

Quasar-T[1] (Dhingra et al., 2017b) consists of $43,000$ open-domain trivia question, and their answers are extracted from ClueWeb09 data source, and the paragraphs are obtained by retrieving 50 sentences for each question from the ClueWeb09 data source using LUCENE.

SearchQA[2] (Dunn et al., 2017) is a large-scale open domain question answering dataset, which consists of question-answer pairs crawled from J! Archive, and the paragraphs are obtained by retrieving 50 webpages for each question from Google Search API.

TriviaQA[3] (Joshi et al., 2017) includes $95,000$ question-answer pairs authored by trivia enthusiasts and independently gathered evidence documents, six per question on average, and utilizes Bing Web search API to collect 50 webpages related to the questions.

CuratedTREC[4] (Voorhees et al., 1999) is based on the benchmark from the TREC QA tasks, which contains $2,180$ questions extracted from the datasets from TREC1999, 2000, 2001 and 2002.

WebQuestions[5] (Berant et al., 2013b) is designed for answering questions from the Freebase knowledge base, which is built by crawling questions through the Google Suggest API and the paragraphs are retrieved from the English Wikipedia using .

For Quasar-T, SearchQA and TriviaQA datasets, we use the retrieved paragraphs provided by (Wang et al., 2018a). For CuratedTREC and WebQuestions datasets, We use the 2016-12-21 dump of English Wikipedia as our knowledge source used to answer the question and then build a Lucene index system on it. After that, we take each input question as a query to retrieve top-50 paragraphs.

The statistics of these datasets are shown in Table 1.

Dataset	#Train	#Dev	#Test
Quasar-T	37,012	3,000	3,000
SearchQA	99,811	13,893	27,247
TriviaQA	87,291	11,274	10,790
CuratedTREC	1,486	-	694
WebQuestions	3,778	-	2,032

Table 1: Statistics of the dataset.

Following (Chen et al., 2017), we adopt two metrics including ExactMatch (EM) and F1 scores to evaluate our model. EM measures the percentage of predictions that match one of the ground truth answers exactly and F1 score is a metric that loosely measures the average overlap between the prediction and ground truth answer.

4.2 Baselines

For comparison, we select several public models as baselines including: (1) **GA** (Dhingra et al., 2017a), a reading comprehension model which performs multiple hops over the paragraph with gated attention mechanism; (2) **BiDAF** (Seo et al., 2017), a reading comprehension model with a bi-directional attention flow network. (3) **AQA** (Buck et al., 2017), a reinforced system learning to re-write questions and aggregate the answers generated by the re-written questions; (4) **R^3** (Wang et al., 2018a), a reinforced model making use of a ranker for selecting most confident paragraph to train the reading comprehension model.

And we also compare our model with its naive version, which regards each paragraph equally and sets a uniform distribution to the paragraph selection. We name our model as "Our+FULL" and its naive version "Our+AVG".

4.3 Experimental Settings

In this paper, we tune our model on the development set and use a grid search to determine the optimal parameters. We select the hidden size of LSTM $n \in \{32, 64, \mathbf{128}, \cdots, 512\}$, the number of LSTM layers for document and question encoder among $\{1, 2, \mathbf{3}, 4\}$, regularization weight α among $\{0.1, \mathbf{0.5}, 1.0, 2.0\}$ and the batch size among $\{4, 8, 16, \mathbf{32}, 64, 128\}$. The optimal parameters are highlighted with bold faces. For other

[1] https://github.com/bdhingra/quasar

[2] https://github.com/nyu-dl/SearchQA

[3] http://nlp.cs.washington.edu/triviaqa/

[4] https://github.com/brmson/dataset-factoid-curated/tree/master/trec

[5] https://github.com/brmson/dataset-factoid-webquestions

parameters, since they have little effect on the results, we simply follow the settings used in (Chen et al., 2017).

For training, our Our+FULL model is first initialized by pre-training using Our+AVG model, and we set the iteration number over all the training data as 10. For pre-trained word embeddings, we use the 300-dimensional GloVe[6] (Pennington et al., 2014) word embeddings learned from 840B Web crawl data.

4.4 Effect of Different Paragraph Selectors

As our model incorporates different types of neural networks including MLP and RNN as our paragraph selector, we investigate the effect of different paragraph selector on the Quasar-T and SearchQA development set.

As shown in Table 3, our RNN paragraph selector leads to statistically significant improvements on both Quasar-T and SearchQA. Note that Our+FULL which uses MLP paragraph selector even performs worse on Quasar-T dataset as compared to Our+AVG. It indicates that MLP paragraph selector is insufficient to distinguish whether a paragraph answers the question. As RNN paragraph selector consistently improves all evaluation metrics, we use it as the default paragraph selector in the following experiments.

4.5 Effect of Different Paragraph Readers

Here, we compare the performance of different types of paragraph readers and the results are shown in Table 4.

From the table, we can see that all models with Sum or Max paragraph readers have comparable performance in most cases, but Our+AVG with Max reader has about 3% increment as compared to the one with Sum reader on the SearchQA dataset. It indicates that the Sum reader is more susceptible to noisy data since it regards all tokens matching to the answer as ground truth. In the following experiments, we select the Max reader as our paragraph reader since it is more stable.

4.6 Overall Results

In this part, we will show the performance of different models on five DS-QA datasets and offer some further analysis. The performance of our models are shown in Table 2. From the results, we can observe that:

(1) Both our models including Our+AVG and Our+FULL achieve better results on most of the datasets as compared to other baselines. The reason is that our models can make full use of the information of all retrieved paragraphs to answer the question, while other baseline models only consider the most relevant paragraph. It verifies our claim that incorporating the rich information of all retrieved paragraphs could help us better extract the answer to the question.

(2) On all datasets, Our+FULL model outperforms Our+AVG model significantly and consistently. It indicates that our paragraph selector could effectively filter out those meaningless retrieved paragraphs and alleviate the wrong labeling problem in DS-QA.

(3) On TriviaQA dataset, our+AVG model has worse performance as compared to R^3 model. After observing the TriviaQA dataset, we find that in this dataset only one or two retrieved paragraphs actually contain the correct answer. Therefore, simply using all retrieved paragraphs equally to extract answer may bring in much noise. On the contrary, Our+FULL model still has a slight improvement by considering the confidence of each retrieved paragraph.

(4) On CuratedTREC and WebQuestions datasets, our model only has a slight improvement as compared to R^3 model. The reason is that the size of these two datasets is tiny and the performance of these DS-QA systems is heavily influenced by the gap with the dataset used to pre-trained.

4.7 Paragraph Selector Performance Analysis

To demonstrate the effectiveness of our paragraph selector in filtering out those noisy retrieved paragraphs, we compare our paragraph selector with traditional information retrieval[7] (IR) in this part. We also compare our model with a new baseline named Our+INDEP which trains the paragraph reader and the paragraph selector independently. To train the paragraph selector, we regard all the paragraph containing the correct answer as ground truth and learns it with Eq. 14.

First, we show the performance in selecting informative paragraphs. Since distantly supervised data doesn't have the labeled ground-truth to tell

[6] http://nlp.stanford.edu/data/glove.840B.300d.zip

[7] The information retrieval model ranks the paragraph with BM25 which is implemented by Lucene.

Datasets	Quasar-T		SearchQA		TriviaQA		CuratedTREC	WebQuestions	
Models	EM	F1	EM	F1	EM	F1	REM	EM	F1
GA (Dhingra et al., 2017a)	26.4	26.4	-	-	-	-	-	-	-
BiDAF (Seo et al., 2017)	25.9	28.5	28.6	34.6	-	-	-	-	-
AQA (Buck et al., 2017)	-	-	40.5	47.4	-	-	-	-	-
R^3 (Wang et al., 2018a)	35.3	41.7	49.0	55.3	47.3	53.7	28.4	17.1	24.6
Our + AVG	38.5	45.7	55.6	61.0	42.6	48.2	28.6	17.8	24.5
+ FULL	**42.2**	**49.3**	**58.8**	**64.5**	**48.7**	**56.3**	**29.1**	**18.5**	**25.6**

Table 2: Experimental results on four open-domain QA test datasets: Quasar-T, SearchQA, TriviaQA, CuratedTREC and WebQuestions. TriviaQA, CuratedTREC and WebQuestions do not provide the leader board under the open-domain setting. Therefore, there is no public baselines in this setting and we only report the result of the R^3 baseline. The result of TriviaQA dataset is on its development set. CuratedTREC dataset is evaluated by regular expression matching (REM).

Datasets		Quasar-T		SearchQA	
Models	Selector	EM	F1	EM	F1
Our + AVG		38.6	45.8	57.3	62.7
+ FULL	MLP	37.1	43.5	59.9	65.1
+ FULL	RNN	41.7	49.1	62.3	67.9

Table 3: Effect of Different Paragraph Selector on the Quasar-T and SearchQA development set.

Datasets		Quasar-T		SearchQA	
Models	Reader	EM	F1	EM	F1
Our + AVG	Max	38.6	45.8	57.3	62.7
+ FULL		41.7	49.1	62.3	67.9
Our + AVG	Sum	39.1	46.3	54.0	59.4
+ FULL		42.3	49.4	61.9	67.4

Table 4: Effect of Different Paragraph Reader on the Quasar-T and SearchQA development set. The paragraph selector used in Our+FULL is RNN.

which paragraphs actually answer the question, we adopt a held-out evaluation instead. It evaluates our model by comparing the selected paragraph with pseudo labels: we regard a paragraph as ground-truth if it contains a token matched to the correct answer. We use Hit@N which indicates the proportion of proper paragraphs being ranked in top-N as evaluation metrics. The result is shown in Table 5. From the table, we can observe that:

(1) Both Our+INDEP and Our+FULL outperform traditional IR model significantly in selecting informative paragraphs. It indicates that our proposed paragraph selector is capable of catching the semantic correlation between question and paragraphs.

(2) Our+FULL has similar performance as compare with Our+SINGLE from Hits@1 to Hits@5 to select valid paragraphs. The reason is that the way of our evaluation of paragraph selection is consistent with the training objective of the ranker in Our+SINGLE.

In fact, this way of evaluation may be not enough to distinguish the performance of different paragraph selector. Therefore, we further report the overall answer extraction performance of Our+FULL and Our+INDEP. From the table, we can see that Our+FULL performs better in answer extraction as compared to Our+SINGLE although they have similar performance in paragraph selection. It demonstrates that our paragraph selector can better determine which tokens matched to the answer are actually answering the question by joint training with paragraph reader.

4.8 Performance with different numbers of paragraphs

Our paragraph selector can be viewed as a fast skimming step before carefully reading the paragraphs. To show how much our paragraph selector can accelerate the DS-QA system, we compare the performance of our model with top paragraphs selected by our paragraph selector (Our+FULL) or traditional IR model.

The results are shown in Fig. 2. There is no doubt that with the number of paragraphs increasing, the performance of our+IR and our+FULL model will increase significantly. From the figure, we can find that on both Quasar-T and SearchQA datasets, our+FULL can use only half of the retrieved paragraphs for answer extraction without performance deterioration, while our+IR suffers from the significant performance degradation when decreasing the number of paragraphs. It demonstrates that our model can extract answer with a few informative paragraphs selected by paragraph selector, which will speed up our whole DS-QA system.

4.9 Potential improvement

To show the potential improvement in aggregating extracted answers with answer re-ranking models of our DS-QA system, we provide statistical anal-

Datasets	Quasar-T					SearchQA				
Task	Paragraph Selection			Overall		Paragraph Selection			Overall	
Models	Hits@1	Hits@3	Hits@5	EM	F1	Hits@1	Hits@3	Hits@5	EM	F1
IR	6.3	10.9	15.2	-	-	13.7	24.1	32.7	-	-
Our + INDEP	26.8	36.3	41.9	40.6	46.9	59.2	70.0	75.7	57.0	62.3
Our + FULL	27.7	36.8	42.6	41.1	48.0	58.9	69.8	75.5	58.8	64.5

Table 5: Comparison of our paragraph selector and traditional information retrieval model in paragraph selection. The Our+AVG and Our+FULL model used in WebQuestions dataset is pre-trained with Quasart-T dataset.

Question:	Who directed the 1946 'It's A Wonderful Life'?
Ground Truth:	Frank Capra
Paragraph1	It's a Wonderful Life (1946): directed by **Frank Capra**, starred by James Stewart, Donna Reed ...
Paragraph2	It's a Wonderful Life, the 1946 film produced and directed by **Frank Capra** and starring ...
Paragraph3	It's a Wonderful Life Guajara in other languages: Spanish, Deutsch, French, Italian ...
Question:	What famous artist could write with both his left and right hand at the same time
Ground Truth:	Leonardo Da Vinci
Paragraph1	**Leonardo Da Vinci** was and is best known as an artist,...
Paragraph2	... the reason **Leonardo da Vinci** used his left hand exclusively was that his right hand was paralyzed.
Paragraph3	... forced me to use my right-hand,... beat my left-hand fingers with ... so that i use the right hand.

Table 6: The examples of the answers to the given questions extracted by our model. The token in bold are the extracted answers in each paragraph. The paragraphs are sorted according to the probabilities output by our paragraph selector.

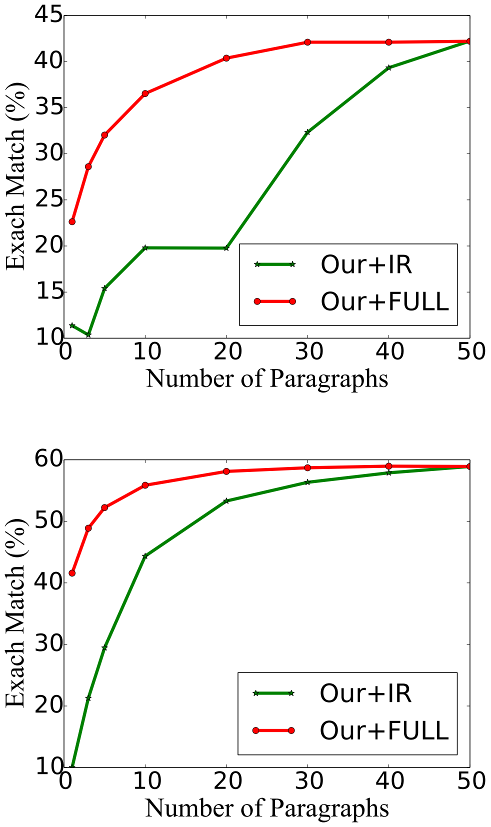

Figure 2: Performance with different numbers of top paragraphs on Quasar-T (up) and SearchQA (bottom) datasets.

ysis to the upper bound of our system performance on the development set. Here, we compare our model with R^3 model by evaluating the F1/EM scores among the top-k extracted answers. This top-k performance of our system can be viewed as the upper bound of our system to re-rank the top-k extracted answers.

Datasets		Quasar-T		SearchQA	
Model	TOP-k	EM	F1	EM	F1
R^3	1	35.3	41.6	51.2	57.3
	3	46.2	53.5	63.9	68.9
	5	51.0	58.9	69.1	73.9
	10	56.1	64.8	75.5	79.6
Our + FULL	1	42.2	49.3	58.8	67.4
	3	53.1	62.0	72.9	77.4
	5	56.4	66.4	76.9	81.0
	10	60.7	71.3	81.2	85.1

Table 7: Potential improvement on DS-QA performance by answer re-ranking. The performance is based on the Quasar-T and SearchQA development dataset.

From Table 7, we can see that:

(1) There is a clear gap between top-3/5 and top-1 DS-QA performance (10-20%). It indicates that our DS-QA model is far from the upper performance and still has a high probability to be improved by answer re-ranking.

(2) The Our+FULL model outperforms R^3 model in top-1, top-3 and top-5 on both Quasar-T and SearchQA datasets by 5% to 7%. It indicates that aggregating the information from all informative paragraphs can effectively enhance our model in DS-QA, which is more potential using answer re-ranking.

4.10 Case Study

Table 6 shows two examples of our models, which illustrates that our model can make full use of informative paragraphs. From the table we find that:

(1) For the question “Who directed the 1946 ‘It’s A Wonderful Life’?”, our model extracts the answer “Frank Capra” from both top-2 paragraphs ranked by our paragraph selector.

(2) For the question “What famous artist could write with both his left and right hand at the same time?”, our model identifies that “Leonardo Da Vinci” is an artist from the first paragraph and could write with both his left and right hand at the same time from the second paragraph.

5 Conclusion and Future Work

In this paper, we propose a denoising distantly supervised open-domain question answering system which contains a paragraph selector to skim over paragraphs and a paragraph reader to perform an intensive reading on the selected paragraphs. Our model can make full use of all informative paragraphs and alleviate the wrong labeling problem in DS-QA. In the experiments, we show that our models significantly and consistently outperforms state-of-the-art DS-QA models. In particular, we demonstrate that the performance of our model is hardly compromised when only using a few top-selected paragraphs.

In the future, we will explore the following directions:

(1) An additional answer re-ranking step can further improve our model. We will explore how to effectively re-rank our extracted answers to further enhance the performance.

(2) Background knowledge such as factual knowledge, common sense knowledge can effectively help us in paragraph selection and answer extraction. We will incorporate external knowledge bases into our DS-QA model to improve its performance.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC No. 61572273, 61661146007 and 61572273). This paper is also partially funded by Microsoft Research Asia FY17-RES-THEME-017.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013a. Semantic parsing on freebase from question-answer pairs. In *Proceedings of EMNLP*. pages 1533–1544.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013b. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of EMNLP*. pages 1533–1544.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075* .

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Andrea Gesmundo, Neil Houlsby, Wojciech Gajewski, and Wei Wang. 2017. Ask the right questions: Active question reformulation with reinforcement learning. *arXiv preprint arXiv:1705.07830* .

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of ACL*. pages 2358–2367.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the ACL*. pages 1870–1879.

Tongfei Chen and Benjamin Van Durme. 2017. Discriminative information retrieval for question answering sentence selection. In *Proceedings of EACL*. pages 719–725.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of ACL*. pages 484–494.

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-fine question answering for long documents. In *Proceedings of ACL*. pages 209–220.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of ACL*. pages 593–602.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017a. Gated-attention readers for text comprehension. In *Proceedings of ACL*. pages 1832–1846.

Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017b. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904* .

Matthew Dunn, Levent Sagun, Mike Higgins, Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179* .

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of SIGKDD*. pages 1156–1165.

Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: an automatic question-answerer. In *Proceedings of IRE-AIEE-ACM*. pages 219–224.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of ACL*. pages 1601–1611.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Cody Kwok, Oren Etzioni, and Daniel S Weld. 2001. Scaling question answering to the web. *TOIS* pages 242–262.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*. pages 2124–2133.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*. pages 1532–1543.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *Proceedings of ICLR*.

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of SIGKDD*. ACM, pages 1047–1055.

Ellen M Voorhees et al. 1999. The trec-8 question answering track report. In *Proceedings of TREC*. pages 77–82.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. 2018a. R3: Reinforced ranker-reader for open-domain question answering. In *Proceedings of AAAI*.

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018b. Evidence aggregation for answer re-ranking in open-domain question answering. In *Proceedings of ICLR*.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of ACL*. pages 189–198.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL*. pages 1480–1489.