

Multi-Task Learning with Generative Adversarial Training for Multi-Passage Machine Reading Comprehension

Qiyu Ren, Xiang Cheng,* Sen Su

State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing, China
{qyren, chengxiang, susen}@bupt.edu.cn

Abstract

Multi-passage machine reading comprehension (MRC) aims to answer a question by multiple passages. Existing multi-passage MRC approaches have shown that employing passages with and without golden answers (i.e. labeled and unlabeled passages) for model training can improve prediction accuracy. In this paper, we present MG-MRC, a novel approach for multi-passage MRC via multi-task learning with generative adversarial training. MG-MRC adopts the extract-then-select framework, where an extractor is first used to predict answer candidates, then a selector is used to choose the final answer. In MG-MRC, we adopt multi-task learning to train the extractor by using both labeled and unlabeled passages. In particular, we use labeled passages to train the extractor by supervised learning, while using unlabeled passages to train the extractor by generative adversarial training, where the extractor is regarded as the generator and a discriminator is introduced to evaluate the generated answer candidates. Moreover, to train the extractor by backpropagation in the generative adversarial training process, we propose a hybrid method which combines boundary-based and content-based extracting methods to produce the answer candidate set and its representation. The experimental results on three open-domain QA datasets confirm the effectiveness of our approach.

Introduction

Machine reading comprehension (MRC), which aims to answer questions about one or more passages, is a long-standing objective in natural language processing. A significant milestone is that several MRC models have exceeded the performance of human annotators on the SQuAD dataset (Rajpurkar et al. 2016)¹. Considering the ultimate goal of reading the whole web, however, this success is still not adequate while the SQuAD dataset only requires MRC models to extract answer on single passage. Therefore, several latest datasets (Dunn et al. 2017; Dhingra, Mazaitis, and Cohen 2017; He et al. 2018) attempt to design MRC tasks in more realistic settings. For each question, they use the search engines to retrieve multiple passages and the multi-passage

MRC models are required to read these passages in order to give the final answer.

Several recent multi-passage MRC approaches adopt an extract-then-select framework (Wang et al. 2018b; 2018c; 2018d; Hu et al. 2019), where an extractor is first used to predict answer candidates from each retrieved passage, then a selector is used to choose one of the answer candidates as the final answer. Some of them have shown that using both labeled and unlabeled passages to train the extractor can improve prediction accuracy. However, the usage of unlabeled passages for model training in these approaches has the following drawbacks. The rule-based method, proposed by Wang et al. (2018c), labels the answer candidates in unlabeled passages for training the extractor, which inevitably incurs the wrong labeling problem. For the reinforcement learning-based method proposed by Wang et al. (2018d), it jointly trains the extractor and selector via policy gradient using unlabeled passages and uses the correctness of the final answer as the reward for each answer candidate. However, since there exist both right and wrong answer candidates in the answer candidate set, using the same reward for them will introduce noises in the training process and degrade the performance of the extractor.

In this paper, we propose a novel multi-passage MRC approach, called MG-MRC, which adopts the extract-then-select framework and leverages labeled and unlabeled passages to train the extractor by multi-task learning (Ruder 2017) with generative adversarial training (Goodfellow et al. 2014). In MG-MRC, we adopt multi-task learning to train the extractor by different tasks where labeled and unlabeled passages are used respectively. In particular, we train the extractor to predict the golden answer from labeled passages. Meanwhile, we regard the extractor as a generator and introduce a discriminator to train them by generative adversarial training using unlabeled passages, where the extractor is trained to obtain a higher score from the discriminator. Thanks to the generative adversarial training, we do not have to label the unlabeled passages and can evaluate each answer candidate independently. As a result, we avoid the wrong labeling problem and the same reward problem. Existing study (Goodfellow et al. 2014) has shown the advantage of adopting backpropagation in the generative adversarial training.

*The corresponding author of this paper is Xiang Cheng.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹<https://rajpurkar.github.io/SQuAD-explorer/>

Therefore, we propose a hybrid method to train the extractor by backpropagation in the generative adversarial training process. In this hybrid method, we combine the boundary-based and content-based extracting method to produce the answer candidate set and its representation which is a continuous latent variable. In doing so, we can use the continuous latent variable as the input of the discriminator, which enables the gradient to backpropagate from the discriminator to the extractor. After the training process, we use the extractor and selector to predict the final answer of the given question from multiple passages.

In summary, our work makes the following contributions:

- We propose a multi-passage MRC approach based on the extract-then-select framework, which adopts multi-task learning with generative adversarial training to train the extractor by using both labeled and unlabeled passages.
- We propose a hybrid method for the extractor, which represents the answer candidates by a continuous latent variable to enable the extractor to be trained by backpropagation in the generative adversarial training process.
- We conduct experiments on three public datasets: Quasart (Dhingra, Mazaitis, and Cohen 2017), SearchQA (Dunn et al. 2017) and TriviaQA (Joshi et al. 2017). The results confirm the effectiveness of applying multi-task learning with generative adversarial training in multi-passage MRC task.

Related Work

Machine reading comprehension (MRC) has made rapid progress in recent years. The mainstream approaches for MRC (Dhingra et al. 2017; Seo et al. 2016; Wang and Jiang 2016; Wang et al. 2017; Devlin et al. 2019) treat the MRC task as the problem of answer boundary prediction from a given passage. These approaches have proven to be effective for single-passage MRC datasets (Rajpurkar et al. 2016).

More recently, multi-passage MRC has attracted increasing attention (Nguyen et al. 2017; Dunn et al. 2017; Dhingra, Mazaitis, and Cohen 2017; He et al. 2018). In this task, a question is paired with multiple passages, which are often collected by exploiting unstructured passages or web data. We categorize existing multi-passage MRC approaches into two groups. Specifically, the first group of approaches (Tan et al. 2018; Wang et al. 2018a; Clark and Gardner 2018; Pang et al. 2019; Das et al. 2019) adopts a rank-then-extract framework, which first employs a ranker to choose the most relevant passage and then employs an extractor to extract the final answer from this passage. However, these approaches depend only on one passage when producing the answer, hence put great demands on the precision of both components.

The second group of approaches (Wang et al. 2018b; 2018c; 2018d; Hu et al. 2019) adopts an extract-then-select framework, which first employs an extractor to predict the answer candidates from multiple passages and then employs a selector to choose the final answer from the answer candidates. Therefore, these approaches can combine the information from different passages to predict the final answer. In particular, Wang et al. (2018b) propose a re-ranking method

to rank answer candidates extracted from multiple passages. Hu et al. (2019) further propose a unified question answering model to re-rank the answer candidates based the encoding of BERT (Devlin et al. 2018). However, they ignore the information between different answer candidates and evaluate each answer candidate isolatedly. To solve this problem, Wang et al. (2018c) propose to leverage the answer candidates to verify each other and rule out the noisy answers. In particular, they choose a segment of text from each unlabeled passage by the ROUGE-L score as the training label and train the extractor using all the passages. However, this simple rule-based labeling might introduce much noise, as the segment of text from unlabeled passage which achieves the highest ROUGE-L score may not be an answer candidate. In addition, Wang et al. (2018d) propose to employ reinforcement learning to fine-tune the extractor and selector together using all the passages. They use the word overlap between the final answer and the golden answer as the reward to train the extractor and selector. Since they assign the same reward to all of the answer candidates whatever they are right or wrong, it might produce noise when training the extractor. For instance, the rewards of right answer candidates will be negative when the chosen final answer is wrong, which will suppress the convergence significantly.

Preliminaries

Problem Definition

Multi-passage MRC task involves a question Q and multiple retrieved passages $\mathcal{P} = \{P_1, P_2, \dots, P_N\}$, and aims to find out the answer A . Both the question Q and each passage P_i can be represented by a sequence of tokens, i.e., $Q = \{w_Q^j\}_{j=1}^{|Q|}$, $P_i = \{w_{P_i}^j\}_{j=1}^{|P_i|}$. Similarly, the answer A can be also represented by a sub-sequence of tokens in the t^{th} passage P_t , i.e., $A = \{w_{P_t}^j\}_{j=s}^e$ ($1 \leq s \leq e \leq |P_t|$), where s and e denote the start and end positions of A in P_t , respectively. The learning algorithm for multi-passage MRC is to learn a function $f(Q, \mathcal{P}) \rightarrow A$. Each item in the training dataset contains a query Q , a set of retrieved passages \mathcal{P} and the golden answer A , which can be denoted by a triple $\langle Q, \mathcal{P}, A \rangle$.

Extract-then-Select Framework

Different from extracting answer directly in single-passage MRC task, Wang et al. (2018d) propose an extract-then-select framework for multi-passage task. First, the extractor predicts K answer candidates $\mathcal{C}_i = \{C_{i1}, C_{i2}, \dots, C_{iK}\}$ from each retrieved passage P_i independently, i.e., $\mathcal{C}_i = f_E(Q, P_i)$, similar to a single-passage MRC task. Then, the selector combines the information from the answer candidate set $\mathcal{C} = \bigcup_1^N \mathcal{C}_i = \{C_{11}, C_{12}, \dots, C_{NK}\}$ using attention mechanism to choose the final answer A from \mathcal{C} , i.e., $A = f_S(Q, \mathcal{P}, \mathcal{C})$. In this paper, we take the selector proposed by Wang et al. (2018d) as our selector, and focus on improving the performance of the extractor.

Generative Adversarial Training

Generative Adversarial Training is a novel way to train a generator G , which captures the real data distribution. It in-

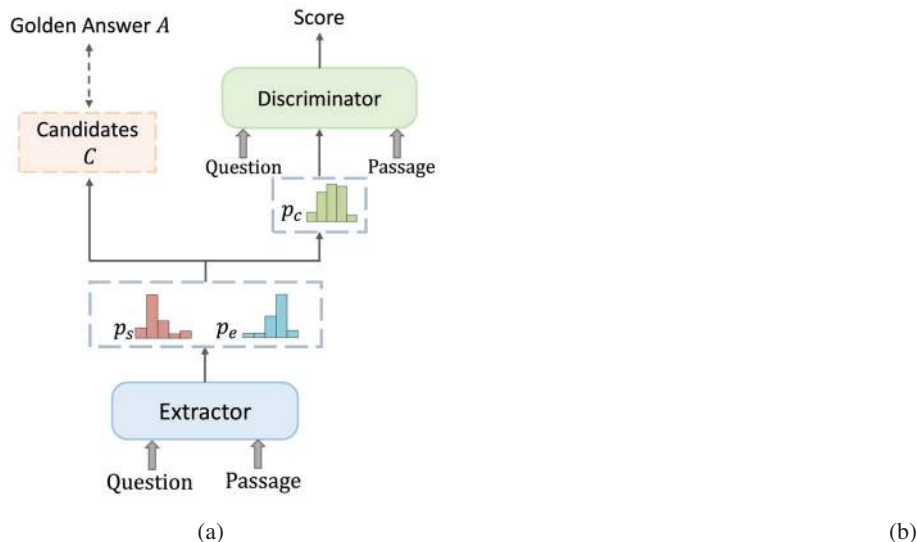


Figure 1: An overview of the proposed approach MG-MRC, which consists of an extractor (blue), a discriminator (green) and a selector (red). Figure 1a shows the training process of the extractor, where labeled and unlabeled passages are used, respectively. And Figure 1b shows the prediction process, where the extractor and the selector are used to answer a given question from multiple passages.

roduces a discriminator D to estimate the probability that a sample came from the real data rather than G .

To learn a distribution p_g over data x , the generator builds a mapping function from a noise distribution $p_z(z)$ to data space as $G(z)$. Meanwhile, the discriminator $D(x)$ outputs a single scalar representing the probability that x came from real data rather than synthetic data generated by G .

G and D are both trained simultaneously: the parameters for G are optimized to minimize $\log(1 - D(G(z)))$ and the parameters for D are optimized to maximize $\log D(x)$, as if they are following the two-player min-max game with value function $V(G, D)$:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim \mathbb{P}_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim \mathbb{P}_z(z)} [\log(1 - D(G(z)))]. \quad (1)$$

The advantage of such formulation is that it does not require an explicit loss function and instead uses the discriminator to optimize the generator.

Methodology

Overview

As shown in Figure 1, MG-MRC consists of three components: an extractor, a discriminator and a selector. In particular, the extractor adopts a hybrid method to produce answer candidate set \mathcal{C}_i and its representation p_{c_i} based on the question Q and passage P_i , i.e., $(\mathcal{C}_i, p_{c_i}) = f_E(Q, P_i)$. Additionally, the discriminator is fed with p_{c_i} to evaluate the correctness of \mathcal{C}_i , i.e., $score = f_D(Q, P_i, p_{c_i})$. The selector combines the information from answer candidate set \mathcal{C} and chooses the final answer A from it, i.e., $A = f_S(Q, \mathcal{P}, \mathcal{C})$.

In the training phase, we first adopt multi-task learning to train the extractor by using the labeled and unlabeled passages (as shown in Figure 1a). Specifically, the first task is to train the extractor to predict golden answers using the labeled passages, and the second task is to train the extractor together with discriminator using the unlabeled passages by generative adversarial training, where the extractor is regarded as the generator. Then, we train the selector independently.

After training the extractor and selector separately, we use them to perform the multi-passage MRC task (as shown in Figure 1b).

Extractor

Given a question $Q = \{w_Q^j\}_{j=1}^{|Q|}$ and a single passage $P_i = \{w_{P_i}^j\}_{j=1}^{|P_i|}$, the extractor predicts answer candidate set $\mathcal{C}_i = \{C_{i1}, C_{i2}, \dots, C_{iK}\}$ from P_i , and computes a continuous latent variable p_{c_i} to represent \mathcal{C}_i .

Passage Representation We first represent all tokens $\{w_{P_i}^j\}_{j=1}^{|P_i|}$ in passage P_i as a sequence of feature vectors. Each feature vector is the concatenation of two vectors x_w^j and x_l^j . In particular, x_w^j is the word embedding of the j^{th} word in P_i , and x_l^j is the concatenation of three linguistic embeddings of the j^{th} word: 1) a 56-dimensional POS tagging embedding, 2) an 18-dimensional NER tagging embedding, and 3) a 3-dimensional binary exact match feature, indicating whether $w_{P_i}^j$ can be matched to one question word in Q , either in its original, lowercase or lemma form.

Then, we feed the feature vectors $\{x_{P_i}^j\}_{j=1}^{|P_i|}$ into a bi-directional long short-term memory network (Bi-LSTM)

(Hochreiter and Schmidhuber 1997) and obtain:

$$\{u_{P_i}^j\}_{j=1}^{|P_i|} = \text{Bi-LSTM}(\{x_{P_i}^j\}_{j=1}^{|P_i|}), \quad (2)$$

where $u_{P_i}^j$ is expected to encode useful context information around token $w_{P_i}^j$.

Question Representation We first feed the word embeddings $\{x_Q^j\}_{j=1}^{|Q|}$ of question Q into a Bi-LSTM and obtain:

$$\{u_Q^j\}_{j=1}^{|Q|} = \text{Bi-LSTM}(\{x_Q^j\}_{j=1}^{|Q|}). \quad (3)$$

Then, we apply a self-attention operation (Vaswani et al. 2017) on the hidden units $\{u_Q^j\}_{j=1}^{|Q|}$ to obtain one single vector v_Q as the representation of Q :

$$v_Q = \sum_{j=1}^{|Q|} \alpha^j u_Q^j, \quad (4)$$

where α^j encodes the importance of the j^{th} word of Q and is calculated as:

$$\alpha^j = \frac{\exp(w_b u_Q^j)}{\sum_k \exp(w_b u_Q^k)}, \quad (5)$$

where $w_b \in \mathbb{R}^d$ is a learned weight vector.

Candidate Prediction There are two methods for predicting the answer candidates. The boundary-based method employs the pointer network (Vinyals, Fortunato, and Jaitly 2015) to predict the start and end positions of the answer candidates. The content-based method, proposed by Wang et al. (2018c), predicts a probability vector, which is the probability of each word appearing in an answer candidate.

However, although these two methods can utilize labeled passages for training the extractor, neither of them is suitable for our extractor. Specifically, the boundary-based method directly treats the predicted answer candidate set as a latent variable for evaluation. Since the latent variable is discrete, adopting this method will prevent the extractor from being trained together with the discriminator by backpropagation. For the content-based method, since it uses the probability vector as a continuous latent variable to represent the answer candidates, it enables the gradient to backpropagate from the discriminator to the extractor. However, applying this method cannot determine the boundaries of the answer candidates.

To this end, we propose a hybrid method for the extractor, which first predicts the probability of each word to be the start and end positions of an answer candidate, and then combines the boundary-based and the content-based methods to produce answer candidate set \mathcal{C}_i and its representation p_{C_i} , which is a continuous latent variable. As a result, unlabeled passages can be used to train the extractor by backpropagation in the generative adversarial training process.

Specifically, we first employ the pointer network to respectively compute the probabilities of each word to be the

start and end positions of answer candidate C :

$$p_s(j|Q, P_i) = \frac{\exp(v_Q W_s u_{P_i}^j)}{\sum_k \exp(v_Q W_s u_{P_i}^k)}, \quad (6)$$

$$p_e(j|Q, P_i) = \frac{\exp(v_Q W_e u_{P_i}^j)}{\sum_k \exp(v_Q W_e u_{P_i}^k)},$$

where $W_s \in \mathbb{R}^{d \times d}$ and $W_e \in \mathbb{R}^{d \times d}$ are the learnable parameters, $p_s(j|Q, P_i)$ and $p_e(j|Q, P_i)$ are the probabilities of the j^{th} word to be the start and end positions of C .

Then, we build the answer candidate set \mathcal{C}_i by choosing the top K boundary scores, where the boundary score is computed as the product of p_s and p_e :

$$p(C_{ij}|Q, P_i) = p_s(s_{ij}|Q, P_i)p_e(e_{ij}|Q, P_i), \quad (7)$$

where C_{ij} denotes the j^{th} answer candidate in P_i , and s_{ij} and e_{ij} denote the start and end positions of C_{ij} subject to the constraint $s_{ij} \leq e_{ij}$.

Meanwhile, we calculate $p_{C_i}(j|Q, P_i)$ as the probability of the j^{th} word in P_i appearing in C_i . Since the probability of the j^{th} word appearing in the answer can be considered as the probability that the j^{th} word is within the answer's boundary, $p_{C_i}(j|Q, P_i)$ can be calculated by p_s and p_e :

$$p_{C_i}(j|Q, P_i) = \sum_{s=1}^j \sum_{e=j}^{|P_i|} p_s(s|Q, P_i)p_e(e|Q, P_i). \quad (8)$$

In doing so, we can use the vector p_{C_i} to represent \mathcal{C}_i , and feed it into the discriminator. As a result, the gradients can backpropagate from the discriminator to the extractor in the generative adversarial training process.

Discriminator

After we obtain the continuous latent variable p_{C_i} of the answer candidate set \mathcal{C}_i based on the given question Q and passage P_i , we introduce a discriminator to evaluate \mathcal{C}_i .

Candidate Representation To enable the discriminator to better evaluate \mathcal{C}_i , we need to encode the information of both Q and P_i into p_{C_i} . To this end, we first build a question-aware passage representation $\{r_{P_i}^j\}_{j=1}^{|P_i|}$, and then build the new representation of \mathcal{C}_i based on $\{r_{P_i}^j\}_{j=1}^{|P_i|}$ and p_{C_i} , which is denoted by r_{C_i} . The details are shown as follows.

To obtain the question-aware passage representation $\{r_{P_i}^j\}_{j=1}^{|P_i|}$, we first apply a variants of Match-LSTM model as mentioned by (Wang et al. 2018a):

$$\{r_{P_i}^j\}_{j=1}^{|P_i|} = \text{Match-LSTM}(\{u_{P_i}^j\}_{j=1}^{|P_i|}, \{u_Q^j\}_{j=1}^{|Q|}). \quad (9)$$

We then adopt a weighted-sum method to obtain the new representation r_{C_i} by using p_{C_i} and $\{r_{P_i}^j\}_{j=1}^{|P_i|}$:

$$\gamma_j = \frac{p_{C_i}(j|Q, P_i)}{\sum_{k=1}^{|P_i|} p_{C_i}(k|Q, P_i)}, \quad (10)$$

$$r_{C_i} = \sum_{k=1}^{|P_i|} \gamma_k r_{P_i}^k.$$

Candidate Scoring We apply a linear transformation and the sigmoid function to compute the score of the answer candidate set:

$$f_D(Q, P_i, p_{c_i}) = \text{sigmoid}(w_c r_{C_i}), \quad (11)$$

where $w_c \in \mathbb{R}^d$ is the learnable parameter and $f_D \in [0, 1]$ is the score function to evaluate the answer candidates extracted from P_i .

Training and Prediction

In the training phase, we first train extractor by multi-task learning using both labeled and unlabeled passages, and then train the selector independently in the same way as Wang et al. (2018d). We detail the training process of the extractor as follows.

The first task is to train the extractor by using labeled passages. In particular, we train the extractor by minimizing the negative log probabilities of the true start and end positions of the golden answer:

$$\mathcal{L}_E = -\log p_s(s|Q, P_t) - \log p_e(e|Q, P_t), \quad (12)$$

where s and e denote the start and end positions of the golden answer A in the passage P_t .

The second task is to train the extractor by using unlabeled passages, where we adopt generative adversarial training to train the extractor and discriminator together. In particular, we train the extractor to obtain a higher score from the discriminator:

$$\mathcal{L}'_E = \log(1 - f_D(Q, P_i, p_{c_i})). \quad (13)$$

Meanwhile, we train the discriminator by using labeled passages:

$$\mathcal{L}_D = y \log f_D(Q, P_t, p_{c_t}) + (1 - y) \log(1 - f_D(Q, P_t, p_{c_t})), \quad (14)$$

where $y \in \{1, 0\}$ denotes whether the golden answer A appears in the candidate answer set \mathcal{C}_t or not.

We define the joint objective function \mathcal{L}''_E of extractor as follows:

$$\mathcal{L}''_E = \alpha \mathcal{L}_E + (1 - \alpha) \mathcal{L}'_E, \quad (15)$$

where α is a hyper-parameter that controls the weights of the two tasks.

In the prediction phase, given a question Q with a set of passages \mathcal{P} , we use the extractor and selector to predict the final answer. Specifically, the extractor first predicts K answer candidates from each passage and calculates the boundary score for each answer candidate. Then, the selector calculates the selection score for each answer candidate. Finally, the final answer is selected from these answer candidates according to the product of these two scores. This simple strategy that multiplies these two scores turns out to be helpful, as we will show in experiment.

Experimental Settings

Datasets

To verify the effectiveness of our approach on multi-passage MRC, we conduct experiments on the Quasar-T (Dhingra,

Dataset	#q(train)	#q(dev)	#q(test)	#p	#g
Quasar-T	37,012	3,000	3,000	100	14.8
SearchQA	99,811	13,893	27,247	50	16.5
TriviaQA-unfiltered	66,828	11,313	10,832	100	16.0

Table 1: The statistics of the datasets. #q represents the number of questions for each split of the dataset. #p is the number of passages for each question. #g means the number of passages that contain the golden answer in average.

Mazaitis, and Cohen 2017), SearchQA (Dunn et al. 2017) and TriviaQA (Joshi et al. 2017) datasets whose statistics are shown in Table 1.

Quasar-T² consists of 43,000 open-domain trivia questions and corresponding answers obtained from various internet sources. Each question is paired with 100 sentence-level passages retrieved from ClueWeb09 (Callan et al. 2009) based on Lucene.

SearchQA³ starts from existing question-answer pairs, which are crawled from J!Archive, and is augmented with text snippets retrieved by Google, resulting in more than 140,000 question-answer pairs with each pair having 49.6 snippets on average.

TriviaQA-unfiltered⁴ includes 95,000 open-domain question-answer pairs authored by trivia enthusiasts, and utilizes Bing Web search API to collect 50 webpages related to the questions. We focus on the open domain version of this dataset, which contains the unfiltered documents.⁵

Baselines

The public baselines include⁶: GA (Dhingra et al. 2017; Dhingra, Mazaitis, and Cohen 2017), a reading comprehension model with gated-attention; BiDAF (Seo et al. 2016), a RC model with bi-directional attention flow; AQA (Buck et al. 2018), a reinforced learning system to aggregate the answers generated by the re-written questions; R³ (Wang et al. 2018a), a reinforced model making use of a ranker for selecting passages to train the RC model; DDS-QA (Lin et al. 2018), a distantly supervised QA model employing a selector to filter out the noisy passages; Re-Ranker (Wang et al. 2018b), a RC model making use of two re-rank models for selecting answer; JT-QA (Wang et al. 2018d), a joint RC model, which contains an extractor and a selector, and trains them by reinforcement learning; Multi-step-reasoner (Das et al. 2019), an open domain QA framework iteratively retrieving and reading passages; DocumentQA (Clark and Gardner 2018), a RC model employing a confidence-based method to globally choose the final answer; HAS-QA (Pang et al. 2019), an open domain QA model based on a three-level hierarchical structure to choose the final answer; RE³QA (Hu et al. 2019), a unified question answering model based on

²<https://github.com/bdhingra/quasar>

³<https://github.com/nyu-dl/SearchQA>

⁴<http://nlp.cs.washington.edu/triviaqa/>

⁵Since TriviaQA’s test set is unknown, we split a development set from the train set and evaluate on the official development set. In particular, Pang et al. (2019) also tested their model in such setting.

⁶We only compare to the results from the public papers.

	Quasar-T		SearchQA		TriviaQA-unfiltered	
	EM	F1	EM	F1	EM	F1
GA (Dhingra et al. 2017)	26.4	26.4	-	-	-	-
BiDAF (Seo et al. 2016)	25.9	28.5	28.6	34.6	-	-
AQA (Buck et al. 2018)	-	-	38.7	45.6	-	-
R ³ (Wang et al. 2018a)	35.3	41.7	49.0	55.3	47.3	53.7
Multi-step-reasoner (Das et al. 2019)	40.6	46.9	56.3	61.4	55.9	61.7
DDS-QA (Lin et al. 2018)	42.2	49.3	58.8	64.5	48.7	56.3
Re-Ranker (Wang et al. 2018b)	42.3	49.6	57.0	63.2	50.6	57.3
DocumentQA (Clark and Gardner 2018)	-	-	-	-	61.6	68.0
HAS-QA (Pang et al. 2019)	43.2	48.9	62.7	68.7	63.6	68.9
JT-QA (Wang et al. 2018d)	45.9	53.9	58.3	64.2	-	-
RE ³ QA (Hu et al. 2019)	-	-	-	-	65.5	71.2
MG-MRC	47.1	54.2	63.0	68.8	63.4	68.4
JT-QA (without selector)	35.4	41.6	44.7	51.2	-	-
MG-MRC (without selector)	39.0	46.1	56.5	61.8	58.9	63.6
Human Performance	54.7	60.4	43.9	-	-	-

Table 2: Experimental results on the test sets of Quasar-T and SearchQA and TriviaQA-unfiltered.

BERT.

We also compare MG-MRC with two internal baselines:

MG-MRC* To evaluate the effect of introducing the discriminator to train the extractor by using unlabeled passages, we introduce MG-MRC*, which contains only the extractor and the selector. In MG-MRC*, the extractor is trained only by Eq.(12) using the labeled data $\langle Q, P_t, A \rangle$.

MG-MRC[#] To evaluate the effectiveness of the hybrid method, we introduce MG-MRC[#], which adopts the policy gradient to train the extractor in the generative adversarial training process. In MG-MRC[#], the extractor only predicts the boundaries of answer candidates. The representations of answer candidates are based on the boundaries of answer candidates, and thus Eq.(10) is rewritten as follows:

$$r_{C_{ij}} = r_{P_i}^{s_{ij}} + r_{P_i}^{e_{ij}}, \quad (16)$$

where s_{ij} and e_{ij} are the start and end positions of the j^{th} answer candidate C_{ij} extracted from passage P_i . During the generative adversarial training process, the extractor is trained by policy gradient. Accordingly, the objective function of the extractor in the generative adversarial training process (i.e. Eq.(13)) is rewritten as follows:

$$\nabla \mathcal{L}'_E \approx - \sum_{j=1}^K [f_D(Q, P_i, C_{ij}) \nabla (\log p_s(s_{ij}|Q, P_i) + \log p_e(e_{ij}|Q, P_i))]. \quad (17)$$

Implementation Details

In the implementation of MG-MRC, we use the spaCy⁷ tool to tokenize both the passages and questions, and generate lemma, part-of-speech and named entity tags. we initialize word embeddings of the passage and question with the 300-dimensional Glove vectors⁸ (Pennington, Socher, and Manning 2014) for the extractor and discriminator. The number

⁷<https://spacy.io>

⁸<http://nlp.stanford.edu/data/wordvecs/glove.840B.300d.zip>

of Bi-LSTMs used to encode the embeddings is set to 3, the hidden size d is set to 128, the multi-task weight α is set to 0.7 and the number K of answer candidates extracted from single passage is set to 3. For the hyper-parameters of the selector, we follow the setting described in Wang et al. (2018d).

During training process, we employ the Adam algorithm (Kingma and Ba 2015) with the initial learning rate as 0.002 and the minibatch size as 64. In addition, we use a dropout rate of 0.35 in each training process.

Metrics

Similar to existing work (Chen et al. 2017), we consider Exact Match (EM) and F1 score as the performance metrics. In particular, EM measures the percentage of predictions that match the ground truth answers exactly, and F1 score measures the average overlap between the prediction and ground truth answer.

Results and Analysis

Overall Results

As shown in Table 2, we can clearly observe that the full model of MG-MRC reaches the state-of-the-art result on Quasar-T and SearchQA, and MG-MRC without the selector also surpasses the model of Wang et al. (2018d) without the selector. These results demonstrate the effectiveness of adopting multi-task learning with generative adversarial training to train the extractor using all the passages. MG-MRC also performs better than most baselines that do not use BERT on the open-domain version of TriviaQA, but slightly worse than RE³QA. In TriviaQA, the golden answers are mostly Wikipedia entities and labeled with Wikipedia aliases. Since RE³QA uses those Wikipedia aliases as the label of answer candidates to train the extractor, it performs slightly better than our model on TriviaQA. However, in most cases, the golden answer is often not the Wikipedia entity, and hard to label the alias of the golden

Quasar-T	EM	F1
MG-MRC*	46.2	53.2
MG-MRC [#]	46.5	53.5
MG-MRC	47.1	54.2

Table 3: Comparison with internal baselines on the test set of Quasar-T.

MG-MRC	EM	F1
Product of two scores	46.9	54.5
- boundary score	46.6	54.3
- selection score	38.5	45.8

Table 4: Ablation results on the Quasar-T development set. Both the boundary score and the selection score impact the results.

answer. This is the reason why RE³QA performs worse than our model on Quasar-T and SearchQA where the golden answers are not Wikipedia entities. What’s more, in the future, we plan to build MG-MRC on BERT to boost the performance of it, such as the method in Wang et al. (2019).

Further Analysis

Compared with Internal Baselines As shown in Table 3, both MG-MRC and the internal baseline MG-MRC[#] achieve better performance on the test set of Quasar-T than MG-MRC*, which trains the extractor only using labeled passages. This means it’s useful to introduce the discriminator to train the extractor using the unlabeled passages.

From Table 3, we can also find that MG-MRC achieves better performance on the test set of Quasar-T than MG-MRC[#]. This is because the policy gradient adopted in MG-MRC[#] produces noise when sampling the answer candidates, whereas MG-MRC can avoid such noise by training the extractor using backpropagation. It indicates that our hybrid method can improve the performance of extractor by enabling it to be trained by backpropagation in the generative adversarial training process.

Ablation Study We conduct ablation experiments to illustrate the effect of multiplying the boundary score and the selection score, and report the results in Table 4. From the results, we can see that the selection score plays an important role in choosing the final answer, as it helps to model the relationship between different answer candidates. Meanwhile, by incorporating the boundary score, the performance can be further improved slightly.

Impact of K The number K of answer candidates extracted from each single passage takes an important role in the selection of the final answer. Therefore, we test the influence of different K on the development set of Quasar-T. The results are shown in Table 5. For K from 1 to 3, the performance of MG-MRC improves as K increases. However, taking $K=4$ cannot improve the performance further. This is because, although a larger K leads to a larger number of correct answers in the answer candidate set, it will intro-

	K	EM	F1
MG-MRC	1	46.3	54.0
	2	46.8	54.4
	3	46.9	54.5
	4	46.7	54.3

Table 5: The performance of MG-MRC with different number of extracted answer candidates on the development set of Quasar-T.

Q	In the body, what is the CNS?
A	central nervous system
P ₁	In general, reflexes control much of what the body must do every day.
P ₂	The CNS acts as the command center of the body.
P ₃	Autonomic nervous system: Much of what occurs in the body every day occurs without an individual being consciously aware.
P ₄	These peripheral nerves serve as the communications link from the body to the CNS.
P ₅	The CNS consists of the brain and the spinal cord which are located in the dorsal body cavity.
P ₆	The central nervous system (CNS) controls most functions of the body and mind.
P ₇	In the human body, the trunk is called the central nervous system , or CNS, and the branches are known as the peripheral nervous system.
P ₈	They help inform the CNS about the state of activity of the muscles and therefore the position and balance of the body.

Table 6: An example from Quasar-T. Answer candidates extracted from passages are in **bold** font. The answer candidates predicted from passage P₂ and P₄ can help to verify the correct answer “central nervous system”.

duce more noise when the selector chooses the final answer. Therefore, we choose $K=3$ as the default candidate number in MG-MRC.

Case Study The example in Table 6 shows the answer candidates predicted by the extractor. Although passages P₂ and P₄ do not contain the correct answer, our extractor still predicts two good answer candidates “command center” and “nerves” from these two passages. These two answer candidates provide valuable distinguishable information to verify the correct answer (i.e. “central nervous system”).

Conclusion

In this paper, we propose a multi-passage MRC approach based on the extract-then-select framework, which adopts multi-task learning to train the extractor using both labeled and unlabeled passages. In particular, we adopt generative adversarial training to train the extractor by using unlabeled passages. Moreover, we propose a hybrid method to enable the extractor to be trained by backpropagation in the generative adversarial training process. The experimental results confirm the effectiveness of our approach. In the future, we

plan to use some extra information, such as commonsense knowledge, to improve the performance of the selector.

Acknowledgments

We acknowledge anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China (Grant No. 61872045), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 61921003).

References

- Buck, C.; Bulian, J.; Ciaramita, M.; Gajewski, W.; Gesmundo, A.; Houlisby, N.; and Wang, W. 2018. Ask the right questions: Active question reformulation with reinforcement learning. *In Proceedings of ICLR*.
- Callan, J.; Hoy, M.; Yoo, C.; and Zhao, L. 2009. Clueweb09 data set.
- Chen, D.; Fisch, A.; Weston, J.; and Bordes, A. 2017. Reading wikipedia to answer open-domain questions. *In Proceedings of ACL*.
- Clark, C., and Gardner, M. 2018. Simple and effective multi-paragraph reading comprehension. *In Proceedings of ACL*.
- Das, R.; Dhuliawala, S.; Zaheer, M.; and McCallum, A. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. *In Proceedings of ICLR*.
- Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv: Computation and Language*.
- Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *In Proceedings of NAACL*.
- Dhingra, B.; Liu, H.; Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2017. Gated-attention readers for text comprehension. *In Proceedings of ACL*.
- Dhingra, B.; Mazaitis, K.; and Cohen, W. W. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*.
- Dunn, M.; Sagun, L.; Higgins, M.; Guney, V. U.; Cirik, V.; and Cho, K. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *In Proceedings of NIPS*.
- He, W.; Liu, K.; Liu, J.; Lyu, Y.; Zhao, S.; Xiao, X.; Liu, Y.; Wang, Y.; Wu, H.; She, Q.; et al. 2018. Dureader: a chinese machine reading comprehension dataset from real-world applications. *In Proceedings of ACL*.
- Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.
- Hu, M.; Peng, Y.; Huang, Z.; and Li, D. 2019. Retrieve, read, rerank: Towards end-to-end multi-document reading comprehension. *In Proceedings of ACL*.
- Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. S. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *In Proceedings of ACL*.
- Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Lin, Y.; Ji, H.; Liu, Z.; and Sun, M. 2018. Denoising distantly supervised open-domain question answering. *In Proceedings of ACL*.
- Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2017. Ms marco: A human-generated machine reading comprehension dataset. *In Proceedings of NIPS*.
- Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Su, L.; and Cheng, X. 2019. Has-qa: Hierarchical answer spans model for open-domain question answering. *arXiv preprint arXiv:1901.03866*.
- Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. *In Proceedings of EMNLP*.
- Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. *In Proceedings of EMNLP*.
- Ruder, S. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.
- Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.
- Tan, C.; Wei, F.; Yang, N.; Du, B.; Lv, W.; and Zhou, M. 2018. S-net: From answer extraction to answer synthesis for machine reading comprehension. *In Proceedings of AAAI*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *In Proceedings of NIPS*.
- Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. *In Proceedings of NIPS*.
- Wang, S., and Jiang, J. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.
- Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated self-matching networks for reading comprehension and question answering. *In Proceedings of ACL*.
- Wang, S.; Yu, M.; Guo, X.; Wang, Z.; Klinger, T.; Zhang, W.; Chang, S.; Tesauro, G.; Zhou, B.; and Jiang, J. 2018a. R³: Reinforced reader-ranker for open-domain question answering. *In Proceedings of AAAI*.
- Wang, S.; Yu, M.; Jiang, J.; Zhang, W.; Guo, X.; Chang, S.; Wang, Z.; Klinger, T.; Tesauro, G.; and Campbell, M. 2018b. Evidence aggregation for answer re-ranking in open-domain question answering. *In Proceedings of ICLR*.
- Wang, Y.; Liu, K.; Liu, J.; He, W.; Lyu, Y.; Wu, H.; Li, S.; and Wang, H. 2018c. Multi-passage machine reading comprehension with cross-passage answer verification. *In Proceedings of ACL*.
- Wang, Z.; Liu, J.; Xiao, X.; Lyu, Y.; and Wu, T. 2018d. Joint training of candidate extraction and answer selection for reading comprehension. *In Proceedings of ACL*.
- Wang, Z.; Ng, P.; Ma, X.; Nallapati, R.; and Xiang, B. 2019. Multi-passage bert: A globally normalized bert model for open-domain question answering. *In Proceedings of EMNLP*.