S-NET: FROM ANSWER EXTRACTION TO ANSWER GENERATION FOR MACHINE READING COMPREHENSION

Chuanqi Tan[†*], **Furu Wei**[‡], **Nan Yang**[‡], **Bowen Du**[†], **Weifeng Lv**[†], **Ming Zhou**[‡]
[†] State Key Laboratory of Software Development Environment, Beihang University, Beijing, China
[‡] Microsoft Research, Beijing, China
tanchuanqi@nlsde.buaa.edu.cn {dubowen,lwf}@buaa.edu.cn
{fuwei,nanya,mingzhou}@microsoft.com

ABSTRACT

In this paper, we present a novel approach to machine reading comprehension for the MS-MARCO dataset. Unlike the SQuAD dataset that aims to answer a question with exact text spans in a passage, the MS-MARCO dataset defines the task as answering a question from multiple passages and the words in the answer are not necessary in the passages. We therefore develop an extraction-then-synthesis framework to synthesize answers from extraction results. Specifically, the answer extraction model is first employed to predict the most important sub-spans from the passage as evidence, and the answer synthesis model takes the evidence as additional features along with the question and passage to further elaborate the final answers. We build the answer extraction model with state-of-the-art neural networks for single passage reading comprehension, and propose an additional task of passage ranking to help answer extraction in multiple passages. The answer synthesis model is based on the sequence-to-sequence neural networks with extracted evidences as features. Experiments show that our extraction-then-synthesis method outperforms state-of-the-art methods.

1 INTRODUCTION

Machine reading comprehension (Rajpurkar et al., 2016; Nguyen et al., 2016), which attempts to enable machines to answer questions after reading a passage or a set of passages, attracts great attentions from both research and industry communities in recent years. The release of the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) and the Microsoft MAchine Reading COmprehension Dataset (MS-MARCO) (Nguyen et al., 2016) provides the large-scale manually created datasets for model training and testing of machine learning (especially deep learning) algorithms for this task. There are two main differences in existing machine reading comprehension datasets. First, the SQuAD dataset constrains the answer to be an exact sub-span in the passage, while words in the answer are not necessary in the passages in the MS-MARCO dataset. Second, the SQuAD dataset only has one passage for a question, while the MS-MARCO dataset contains multiple passages.

Existing methods for the MS-MARCO dataset usually follow the extraction based approach for single passage in the SQuAD dataset. It formulates the task as predicting the start and end positions of the answer in the passage. However, as defined in the MS-MARCO dataset, the answer may come from multiple spans, and the system needs to elaborate the answer using words in the passages and words from the questions as well as words that cannot be found in the passages or questions.

Table 1 shows several examples from the MS-MARCO dataset. Except in the first example the answer is an exact text span in the passage, in other examples the answers need to be synthesized or generated from the question and passage. In the second example the answer consists of multiple text spans (hereafter evidence snippets) from the passage. In the third example, the answer contains

[*] Contribution during internship at Microsoft Research.

words from the question. In the fourth example, the answer has words that cannot be found in the passages or question. In the last example, all words are not in the passages or questions.

In this paper, we present an extraction-then-synthesis framework for machine reading comprehension shown in Figure 1, in which the answer is synthesized from the extraction results. We build an evidence extraction model to predict the most important sub-spans from the passages as evidence, and then develop an answer synthesis model which takes the evidence as additional features along with the question and passage to further elaborate the final answers.

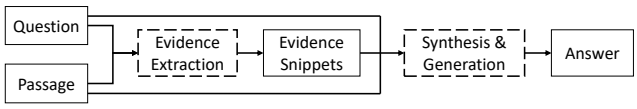

Figure 1: Overview of S-Net. It first extracts evidence snippets by matching the question and passage, and then generates the answer by synthesizing the question, passage, and evidence snippets.

The answer is an exact text span in the passage. Q: how tall is jack griffo P: Jack Griffo Height : **5'6 (167.64 cm)**. Standing at a height of 5 feet, 6 inches tall Jack Griffo is taller than 11.9% of all men, as reflected by the figure's fill height %.Conversely, at this height Jack Griffo is not as tall as 88.1% of all men. A: 5'6 (167.64 cm)
All words in the answer are in the passage but from multiple text spans. Q: who did odysseus see in the underworld P_1: The souls that Odysseus saw in the Underworld On seeing **Achilles**' soul, said Odysseus: Achilles, the most fortunate man that ever was or will be honored as though you were a god and now you are a mighty prince among the dead. P_2: Odysseus talked to his mother Anticlea, who died of grief when he did not return home after the Trojan War. Odysseus was also surprised to see **Elphenor**, the youngest member of his crew, in the Underworld. A: Elphenor and Achilles.
All words in the answer are in the passage and question. Q: what do **producers need to make food** P: Plants are producers. Producers are living things that can make their own food using **air, light, soil, and water**. Plants use a process called photosynthesis to make food. A: Producers need air, light, soil, and water to make food.
Part of words in the answer are not found in the passage or question. Q: why conversion observed in body P: Conversion disorder **symptoms** may appear suddenly after a stressful event or trauma, whether physical or psychological. Signs and symptoms that affect movement function may include: 1 Weakness or paralysis. 2 Abnormal movement, such as tremors or difficulty walking. 3 Loss of balance. A: Due to symptoms in the body
All Words in the answer are not found in the passages or question. Q: is there an age limit for learning speech P: Age is not a detriment to language learning, and by all accounts, learning a second (or third etc) language actually keeps the older language learners mind active. People of all ages can benefit from learning languages. A: No

Table 1: Representative examples for different kinds of answer according to the necessary of synthesis in the MS-MARCO dataset. Q, P, and A represent question, passage, and answer, respectively. The text in bold is the evidence in the passage or the word in the question that matches the answer.

Specifically, we develop the answer extraction model with state-of-the-art attention based neural networks which predict the start and end positions of evidence snippets. As multiple passages are provided for each question in the MS-MARCO dataset, we propose incorporating passage ranking as an additional task to improve the results of evidence extraction under a multi-task learning frame-

work. We use the bidirectional recurrent neural networks (RNN) for the word-level representation, and then apply the attention mechanism (Rocktäschel et al., 2015) to incorporate matching information from question to passage at the word level. Next, we predict start and end positions of the evidence snippet by pointer networks (Vinyals et al., 2015a). Moreover, we aggregate the word-level matching information of each passage using the attention pooling, and use the passage-level representation to rank all candidate passages as an additional task. For the answer synthesis, we apply the sequence-to-sequence model to synthesize the final answer based on the extracted evidence. The question and passage are encoded by a bi-directional RNN in which the start and end positions of extracted snippet are labeled as features. We combine the question and passage information in the encoding part to initialize the attention-equipped decoder to generate the answer.

We conduct experiments on the MS-MARCO dataset. The results show our extraction-then-synthesis framework outperforms our baselines and all other existing methods in terms of ROUGE-L and BLEU-1.

Our contributions can be summarized as follows:

- We propose an extraction-then-synthesis framework for machine reading comprehension in which words in answer are not necessary in the passages.
- We incorporate passage ranking to pure answer span prediction, which improves the extraction result in the multiple passages reading comprehension.
- We develop an answer synthesis model that applies the sequence-to-sequence model to generate the answer with extracted evidences as features, which outperforms pure answer extraction methods and all other existing methods on the MS-MARCO dataset.

2 Related Work

Benchmark datasets play an important role in recent progress in reading comprehension and question answering research. Richardson et al. (2013) release MCTest whose goal is to select the best answer from four options given the question and the passage. CNN/Daily-Mail (Hermann et al., 2015) and CBT (Hill et al., 2016) are the cloze-style datasets in which the goal is to predict the missing word (often a named entity) in a passage. Different from above datasets, the SQuAD dataset (Rajpurkar et al., 2016) whose answer can be much longer phrase is more challenging. The answer in SQuAD is a segment of text, or span, from the corresponding reading passage. Similar to the SQuAD, MS-MARCO (Nguyen et al., 2016) is the reading comprehension dataset which aims to answer the question given a set of passages. The answer in MS-MARCO is generated by human after reading all related passages and not necessarily sub-spans of the passages.

To the best of our knowledge, the existing works on the MS-MARCO dataset follow their methods on the SQuAD. Wang & Jiang (2016b) combine match-LSTM and pointer networks to produce the boundary of the answer. Xiong et al. (2016) and Seo et al. (2016) employ variant co-attention mechanism to match the question and passage mutually. Xiong et al. (2016) propose a dynamic pointer network to iteratively infer the answer. Wang et al. (2017) apply an additional gate to the attention-based recurrent networks and propose a self-matching mechanism for aggregating evidence from the whole passage, which achieves the state-of-the-art result on SQuAD dataset. Other works which only focus on the SQuAD dataset may also be applied on the MS-MARCO dataset (Yu et al., 2016; Lee et al., 2016; Yang et al., 2016).

The sequence-to-sequence model is widely-used in many tasks such as machine translation (Luong et al., 2015), parsing (Vinyals et al., 2015b), response generation (Gu et al., 2016), and summarization generation (Zhou et al., 2017). We use it to generate the synthetic answer with the start and end positions of the evidence snippet as features.

3 Our Approach

Following the overview in Figure 1, our approach consists of two parts as evidence extraction[1] and answer synthesis. The two parts are trained in two stages. The evidence extraction part aims to

[1] In our model, we use "evidence extraction" to represent the pure "answer extraction" in previous work.

extract evidence snippets related to the question and passage. The answer synthesis part aims to generate the answer based on the extracted evidence snippets. We propose a multi-task learning framework for the evidence extraction shown in Figure 2, and use the sequence-to-sequence model with additional features of the start and end positions of the evidence snippet for the answer synthesis shown in Figure 3.

3.1 Gated Recurrent Unit

We use Gated Recurrent Unit (GRU) (Cho et al., 2014) instead of basic RNN. Equation 1 describes the mathematical model of the GRU. r_t and z_t are the gates and h_t is the hidden state.

$$
\begin{aligned}
z_t &= \sigma(W_{hz}h_{t-1} + W_{xz}x_t + b_z) \\
r_t &= \sigma(W_{hr}h_{t-1} + W_{xr}x_t + b_r) \\
\hat{h_t} &= \Phi(W_h(r_t \odot h_{t-1}) + W_x x_t + b) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h_t}
\end{aligned} \tag{1}
$$

3.2 Evidence Extraction

We propose a multi-task learning framework for evidence extraction. Unlike the SQuAD dataset, which only has one passage given a question, there are several related passages for each question in the MS-MARCO dataset. In addition to annotating the answer, MS-MARCO also annotates which passage is correct. To this end, we propose improving text span prediction with passage ranking. Specifically, as shown in Figure 2, in addition to predicting a text span, we apply another task to rank candidate passages with the passage-level representation.

3.2.1 Evidence Snippet Prediction

Consider a question Q = $\{w_t^Q\}_{t=1}^m$ and a passage P = $\{w_t^P\}_{t=1}^n$, we first convert the words to their respective word-level embeddings and character-level embeddings. The character-level embeddings are generated by taking the final hidden states of a bi-directional GRU applied to embeddings of characters in the token. We then use a bi-directional GRU to produce new representation $u_1^Q, \ldots, u_m^Q$ and $u_1^P, \ldots, u_n^P$ of all words in the question and passage respectively:

$$
\begin{aligned}
u_t^Q &= \text{BiGRU}_Q(u_{t-1}^Q, [e_t^Q, char_t^Q]) \\
u_t^P &= \text{BiGRU}_P(u_{t-1}^P, [e_t^P, char_t^P])
\end{aligned} \tag{2}
$$

Given question and passage representation $\{u_t^Q\}_{t=1}^m$ and $\{u_t^P\}_{t=1}^n$, Rocktäschel et al. (2015) propose generating sentence-pair representation $\{v_t^P\}_{t=1}^n$ via soft-alignment of words in the question and passage as follows:

$$
v_t^P = \text{GRU}(v_{t-1}^P, c_t^Q) \tag{3}
$$

where $c_t^Q = att(u^Q, [u_t^P, v_{t-1}^P])$ is an attention-pooling vector of the whole question (u^Q):

$$
\begin{aligned}
s_j^t &= \text{v}^{\text{T}}\tanh(W_u^Q u_j^Q + W_u^P u_t^P) \\
a_i^t &= \exp(s_i^t)/\Sigma_{j=1}^m \exp(s_j^t) \\
c_t^Q &= \Sigma_{i=1}^m a_i^t u_i^Q
\end{aligned} \tag{4}
$$

Wang & Jiang (2016a) introduce match-LSTM, which takes u_j^P as an additional input into the recurrent network. Wang et al. (2017) propose adding gate to the input ($[u_t^P, c_t^Q]$) of RNN to determine the importance of passage parts.

$$
\begin{aligned}
g_t &= \text{sigmoid}(W_g[u_t^P, c_t^Q]) \\
[u_t^P, c_t^Q]^* &= g_t \odot [u_t^P, c_t^Q] \\
v_t^P &= \text{GRU}(v_{t-1}^P, [u_t^P, c_t^Q]^*)
\end{aligned} \tag{5}
$$

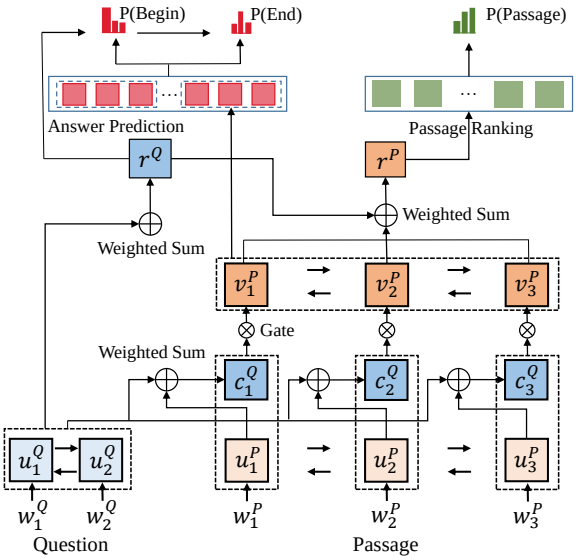

Figure 2: Evidence Extraction Model

We use pointer networks (Vinyals et al., 2015a) to predict the position of evidence snippets. Following the previous work (Wang & Jiang, 2016b), we concatenate all passages to predict one span for the evidence snippet prediction. Given the representation $\{v_t^P\}_{t=1}^N$ where N is the sum of the length of all passages, the attention mechanism is utilized as a pointer to select the start position (p^1) and end position (p^2), which can be formulated as follows:

$$\begin{aligned} s_j^t &= \mathrm{v}^{\mathrm{T}} \tanh(W_h^P v_j^P + W_h^a h_{t-1}^a) \\ a_i^t &= \exp(s_i^t) / \Sigma_{j=1}^N \exp(s_j^t) \\ p^t &= \mathrm{argmax}(a_1^t, \ldots, a_N^t) \end{aligned} \tag{6}$$

Here h_{t-1}^a represents the last hidden state of the answer recurrent network (pointer network). The input of the answer recurrent network is the attention-pooling vector based on current predicted probability a^t:

$$\begin{aligned} c_t &= \Sigma_{i=1}^N a_i^t v_i^P \\ h_t^a &= \mathrm{GRU}(h_{t-1}^a, c_t) \end{aligned} \tag{7}$$

When predicting the start position, h_{t-1}^a represents the initial hidden state of the answer recurrent network. We utilize the question vector r^Q as the initial state of the answer recurrent network. $r^Q = att(u^Q, v_r^Q)$ is an attention-pooling vector of the question based on the parameter v_r^Q:

$$\begin{aligned} s_j &= \mathrm{v}^{\mathrm{T}} \tanh(W_u^Q u_j^Q + W_v^Q v_r^Q) \\ a_i &= \exp(s_i) / \Sigma_{j=1}^m \exp(s_j) \\ r^Q &= \Sigma_{i=1}^m a_i u_i^Q \end{aligned} \tag{8}$$

For this part, the objective function is to minimize the following cross entropy:

$$\mathcal{L}_{AP} = -\Sigma_{t=1}^2 \Sigma_{i=1}^N [y_i^t \log a_i^t + (1 - y_i^t) \log(1 - a_i^t)] \tag{9}$$

where $y_i^t \in \{0, 1\}$ denotes a label. $y_i^t = 1$ means i is a correct position, otherwise $y_i^t = 0$.

3.2.2 Passage Ranking

In this part, we match the question and each passage from word level to passage level. Firstly, we use the question representation r^Q to attend words in each passage to obtain the passage representation

r^P where $r^P = att(v^P, r^Q)$.

$$
\begin{aligned}
s_j &= \mathrm{v}^{\mathrm{T}} \tanh(W_v^P v_j^P + W_v^Q r^Q) \\
a_i &= \exp(s_i) / \Sigma_{j=1}^{n} \exp(s_j) \\
r^P &= \Sigma_{i=1}^{n} a_i v_i^P
\end{aligned} \tag{10}
$$

Next, the question representation r^Q and the passage representation r^P are combined to pass two fully connected layers for a matching score,

$$
g = v_g^{\mathrm{T}}(\tanh(W_g[r^Q, r^P])) \tag{11}
$$

For one question, each candidate passage P_i has a matching score g_i. We normalize their scores and optimize following objective function:

$$
\begin{aligned}
\hat{g}_i &= \exp(g_i) / \Sigma_{j=1}^{k} \exp(g_j) \\
\mathcal{L}_{PR} &= -\sum_{i=1}^{k} [y_i \log \hat{g}_i + (1 - y_i) \log(1 - \hat{g}_i)]
\end{aligned} \tag{12}
$$

where k is the number of passages. $y_i \in \{0, 1\}$ denotes a label. $y_i = 1$ means P_i is the correct passage, otherwise $y_i = 0$.

3.2.3 Joint Learning

The evident extraction part is trained by minimizing joint objective functions:

$$
\mathcal{L}_E = r\mathcal{L}_{AP} + (1 - r)\mathcal{L}_{PR} \tag{13}
$$

where r is the hyper-parameter for weights of two loss functions.

3.3 Answer Synthesis

As shown in Figure 3, we use the sequence-to-sequence model to synthesize the answer with the extracted evidences as features. We first produce the representation h_t^P and h_t^Q of all words in the passage and question respectively. When producing the answer representation, we combine the basic word embedding e_t^p with additional features f_t^s and f_t^e to indicate the start and end positions of the evidence snippet respectively predicted by evidence extraction model. $f_t^s = 1$ and $f_t^e = 1$ mean the position t is the start and end of the evidence span, respectively.

$$
\begin{aligned}
h_t^P &= \mathrm{BiGRU}(h_{t-1}^P, [e_t^p, f_t^s, f_t^e]) \\
h_t^Q &= \mathrm{BiGRU}(h_{t-1}^Q, e_t^Q)
\end{aligned} \tag{14}
$$

On top of the encoder, we use GRU with attention as the decoder to produce the answer. At each decoding time step t, the GRU reads the previous word embedding w_{t-1} and previous context vector c_{t-1} as inputs to compute the new hidden state d_t. To initialize the GRU hidden state, we use a linear layer with the last backward encoder hidden state $\overleftarrow{h}_1^P$ and $\overleftarrow{h}_1^Q$ as input:

$$
\begin{aligned}
d_t &= \mathrm{GRU}(w_{t-1}, c_{t-1}, d_{t-1}) \\
d_0 &= \tanh(W_d[\overleftarrow{h}_1^P, \overleftarrow{h}_1^Q] + b)
\end{aligned} \tag{15}
$$

where W_d is the weight matrix and b is the bias vector.

The context vector c_t for current time step t is computed through the concatenate attention mechanism (Luong et al., 2015), which matches the current decoder state d_t with each encoder hidden state h_t to get the weighted sum representation. Here h_i consists of the passage representation h_t^P and the question representation h_t^Q.

$$
\begin{aligned}
s_j^t &= v_a^{\mathrm{T}} \tanh(W_a d_{t-1} + U_a h_j) \\
a_i^t &= \exp(s_i^t) / \Sigma_{j=1}^{n} \exp(s_j^t) \\
c_t &= \Sigma_{i=1}^{n} a_i^t h_i
\end{aligned} \tag{16}
$$

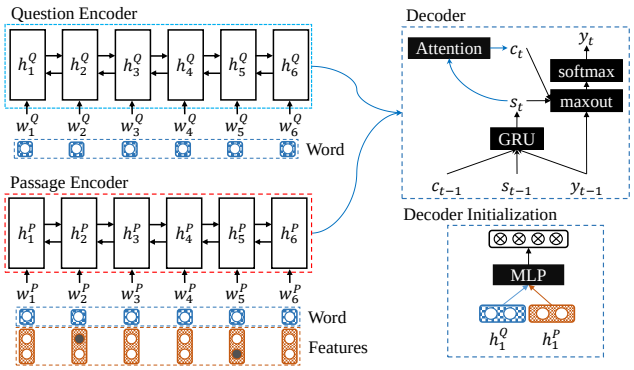

Figure 3: Answer Synthesis Model

We then combine the previous word embedding w_{t-1}, the current context vector c_t, and the decoder state d_t to construct the readout state r_t. The readout state is then passed through a maxout hidden layer (Goodfellow et al., 2013) to predict the next word with a softmax layer over the decoder vocabulary.

$$\begin{aligned} r_t &= W_r w_{t-1} + U_r c_t + V_r d_t \\ m_t &= [\max\{r_{t,2j-1}, r_{t,2j}\}]^{\mathrm{T}} \\ p(y_t|y_1, \ldots, y_{t-1}) &= \mathrm{softmax}(W_o m_t) \end{aligned} \tag{17}$$

where W_a, U_a, W_r, U_r, V_r and W_o are parameters to be learned. Readout state r_t is a $2d$-dimensional vector, and the maxout layer (Equation 17) picks the max value for every two numbers in r_t and produces a d-dimensional vector m_t.

Our goal is to maximize the output probability given the input sentence. Therefore, we optimize the negative log-likelihood loss function:

$$\mathcal{L}_S = -\frac{1}{|\mathcal{D}|}\Sigma_{(X,Y)\in\mathcal{D}} \log p(Y|X) \tag{18}$$

where $\mathcal{D}$ is the set of data. X represents the question and passage including evidence snippets, and Y represents the answer.

4 Experiment

We conduct our experiments on the MS-MARCO dataset (Nguyen et al., 2016). We compare our extraction-then-synthesis framework with pure extraction model and other baseline methods on the leaderboard of MS-MARCO. Experimental results show that our model achieves better results in official evaluation metrics. We also conduct ablation tests to verify our method, and compare our framework with the end-to-end generation framework.

4.1 Dataset and Evaluation Metrics

For the MS-MARCO dataset, the questions are user queries issued to the Bing search engine and the context passages are from real web documents. The data has been split into a training set (82,326 pairs), a development set (10,047 pairs) and a test set (9,650 pairs).

The answers are human-generated and not necessarily sub-spans of the passages so that the metrics in the official tool of MS-MARCO evaluation are BLEU (Papineni et al., 2002) and ROUGE-L (Lin, 2004). In the official evaluation tool, the ROUGE-L is calculated by averaging the score per question, however, the BLEU is normalized with all questions. We hold that the answer should be evaluated case-by-case in the reading comprehension task. Therefore, we mainly focus on the result in the ROUGE-L.

4.2 Implementation Details

4.2.1 Training

The evidence extraction and the answer synthesis are trained in two stages.

For evidence extraction, since the answers are not necessarily sub-spans of the passages, we choose the span with the highest ROUGE-L score with the reference answer as the gold span in the training. Moreover, we only use the data whose ROUGE-L score of chosen text span is higher than 0.7, therefore we only use 71,417 training pairs in our experiments.

For answer synthesis, the training data consists of two parts. First, for all passages in the training data, we choose the best span with highest ROUGE-L score as the evidence, and use the corresponding reference answer as the output. We only use the data whose ROUGE-L score of chosen evidence snippet is higher than 0.5. Second, we apply our evidence extraction model to all training data to obtain the extracted span. Then we treat the passage to which this span belongs as the input.

4.2.2 Parameter

For answer extraction, we use 300-dimensional uncased pre-trained *GloVe* embeddings (Pennington et al., 2014)[2] for both question and passage without update during training. We use zero vectors to represent all out-of-vocabulary words. Hidden vector length is set to 150 for all layers. We also apply dropout (Srivastava et al., 2014) between layers, with dropout rate 0.1. The weight r is set to 0.8.

For answer synthesis, we use an identical vocabulary set for the input and output collected from the training data. We set the vocabulary size to 30,000 according to the frequency and the other words are set to <unk>. All word embeddings are updated during the training. We set the word embedding size to 300, set the feature embedding size of start and end positions of the extracted snippet to 50, and set all GRU hidden state sizes to 150.

The model is optimized using AdaDelta (Zeiler, 2012) with initial learning rate of 1.0. All hyper-parameters are selected on the MS-MARCO development set.

4.2.3 Decoding

When decoding, we first run our extraction model to obtain the extracted span, and run our synthesis model with the extracted result and the passage that contains this span. We use the beam search with beam size of 12 to generate the sequence. After the sequence-to-sequence model, we post-process the sequence with following rules:

- We only keep once if the sequence-to-sequence model generates duplicated words or phrases.
- For all "<unk>" and the word as well as phrase which are not existed in the extracted answer, we try to refine it by finding a word or phrase with the same adjacent words in the extracted span and passage.
- If the generated answer only contains a single word "<unk>", we use the extracted span as the final answer.

4.3 Baseline Methods

We conduct experiments with following settings:

S-Net (Extraction): the model that only has the evidence extraction part.

S-Net: the model that consists of the evidence extraction part and the answer synthesis part.

We implement two state-of-the-art baselines on reading comprehension, namely BiDAF (Seo et al., 2016) and Prediction (Wang & Jiang, 2016b), to extract text spans as evidence snippets. Moreover, we implement a baseline that only has the evidence extraction part without the passage ranking.

[2] `http://nlp.stanford.edu/data/glove.6B.zip`.

Method	ROUGE-L	BLEU-1
FastQAExt	33.67	33.93
Prediction	37.33	40.72
ReasoNet	38.81	39.86
R-Net	42.89	42.22
S-Net (Extraction)	41.45	44.08
S-Net (Extraction, Ensemble)	42.92	44.97
S-Net	45.23	43.78
S-Net*	**46.65**	**44.78**
Human Performance	47	46

Table 2: The performance on the MS-MARCO test set. *Using the ensemble result of extraction models as the input of the synthesis model.

Method	Extraction	Extraction +Synthesis
FastQAExt	33.7	-
BiDAF	34.89	38.73
Prediction	37.54^{+}	41.55
S-Net (w/o Passage Ranking)	39.62	43.26
S-Net	42.23	45.95
S-Net*	**44.11**	**47.76**

Table 3: The performance on the MS-MARCO development set in terms of ROUGE-L. *Using the ensemble result of extraction models as the input of the synthesis model. $^{+}$Wang & Jiang (2016b) report their Prediction with 37.3.

Then we apply the answer synthesis part on top of their results. We also compare with other methods on the MS-MARCO leaderboard, including FastQAExt (Weissenborn et al., 2017), ReasoNet (Shen et al., 2016), and R-Net (Wang et al., 2017).

4.4 Result

Table 2 shows the results on the MS-MARCO test data[3]. Our extraction model achieves 41.45 and 44.08 in terms of ROUGE-L and BLEU-1, respectively. Next we train the model 30 times with the same setting, and select models using a greedy search[4]. We sum the probability at each position of each single model to decide the ensemble result. Finally we select 13 models for ensemble, which achieves 42.92 and 44.97 in terms of ROUGE-L and BLEU-1, respectively, which achieves the state-of-the-art results of the extraction model. Then we test our synthesis model based on the extracted evidence. Our synthesis model achieves 3.78% and 3.73% improvement on the single model and ensemble model in terms of ROUGE-L, respectively. Our best result achieves 46.65 in terms of ROUGE-L and 44.78 in terms of BLEU-1, which outperforms all existing methods with a large margin and are very close to human performance. Moreover, we observe that our method only achieves significant improvement in terms of ROUGE-L compared with our baseline. The reason is that our synthesis model works better when the answer is short, which almost has no effect on BLEU as it is normalized with all questions.

Since answers on the test set are not published, we analyze our model on the development set. Table 3 shows results on the development set in terms of ROUGE-L. As we can see, our method outperforms the baseline and several strong state-of-the-art systems. For the evidence extraction part, our proposed multi-task learning framework achieves 42.23 and 44.11 for the single and ensemble model in terms of ROUGE-L. For the answer synthesis, the single and ensemble models improve 3.72% and 3.65% respectively in terms of ROUGE-L. We observe the consistent improvement when

[3]Baseline results are extracted from MS-MARCO leaderboard `http://www.msmarco.org/leaders.aspx` on Sept. 10, 2017.

[4]We search models from high to low by their performances on the development set. We keep the model if adding it improves the result, otherwise discard.

Method	P@1	ROUGE-L
Extraction w/o Passage Ranking	34.6	56.7
Passage Ranking then Extraction	28.3	52.9
S-Net (Extraction)	**38.9**	**59.4**

Table 4: Results of passage ranking. -w/o Passage Ranking: the model that only has evidence extraction part, without passage ranking part. -Passage Ranking then Extraction: the model that selects the passage firstly and then apply the extraction model only on the selected passage.

Category	Extraction	Extraction +Synthesis
max = 1.0 (63.95%)	50.74	49.59
0.8≤max<1.0 (20.06%)	40.95	41.16
0.6≤max<0.8 (5.78%)	31.21	33.21
0.4≤max<0.6 (1.54%)	21.97	22.44
0.2≤max<0.4 (0.29%)	13.47	13.49
max<0.2 (8.38%)	0.01	49.18

Table 5: The performance of questions in different levels of necessary of synthesis in terms of ROUGE-L on MS-MARCO development set.

applying our answer synthesis model to other answer span prediction models, such as BiDAF and Prediction.

4.5 Discussion

4.5.1 Ablation Test on Passage Ranking

We analyze the result of incorporating passage ranking as an additional task. We compare our multi-task framework with two baselines as shown in Table 4. For passage selection, our multi-task model achieves the accuracy of 38.9, which outperforms the pure answer prediction model with 4.3. Moreover, jointly learning the answer prediction part and the passage ranking part is better than solving this task by two separated steps because the answer span can provide more information with stronger supervision, which benefits the passage ranking part. The ROUGE-L is calculated by the best answer span in the selected passage, which shows our multi-task learning framework has more potential for better answer.

4.5.2 Extraction vs. Synthesis

We compare the result of answer extraction and answer synthesis in different categories grouped by the upper bound of extraction method in Table 5. For the question whose answer can be exactly matched in the passage, our answer synthesis model performs slightly worse because the sequence-to-sequence model makes some deviation when copying extracted evidences. In other categories, our synthesis model achieves more or less improvement. For the question whose answer can be almost found in the passage (ROUGE-L≥0.8), our model achieves 0.2 improvement even though the space that can be raised is limited. For the question whose upper performance via answer extraction is between 0.6 and 0.8, our model achieves a large improvement of 2.0. Part of questions in the last category (ROUGE-L<0.2) are the polar questions whose answers are "yes" or "no". Although the answer is not in the passage or question, our synthesis model can easily solve this problem and determine the correct answer through the extracted evidences, which leads to such improvement in this category. However, in these questions, answers are too short to influence the final score in terms of BLEU because it is normalized in all questions. Moreover, the score decreases due to the penalty of length. Due to the limitation of BLEU, we only report the result in terms of ROUGE-L in our analysis.

Method	ROUGE-L
S2S (Question)	8.9
S2S (Question + All Passages)	28.75
S2S (Question + Selected Passage)	37.70
Matching + S2S	6.28

Table 6: The performance on MS-MARCO development set of end-to-end methods.

4.5.3 Comparison with the End-to-End Generation Framework

We compare our extraction-then-synthesis model with several end-to-end generation models in Table 6. S2S represents the sequence-to-sequence framework shown in Figure 3. The difference among our synthesis model and all entries in the Table 6 is the information we use in the encoding part. The authors of MS-MACRO publish a baseline of training a sequence-to-sequence model with the question and answer, which only achieves 8.9 in terms of ROUGE-L. Adding all passages to the sequence-to-sequence model can obviously improve the result to 28.75. Then we only use the question and the selected passage to generate the answer. The only difference with our synthesis model is that we add the position features to the basic sequence-to-sequence model. The result is still worse than our synthesis model with a large margin, which shows the matching between question and passage is very important for generating answer. Next, we build an end-to-end framework combining matching and generation. We apply the sequence-to-sequence model on top of the matching information by taking question sensitive passage representation v_t^P in the Equation 5 as the input of sequence-to-sequence model, which only achieves 6.28 in terms of ROUGE-L. Above results show the effectiveness of our model that solves this task with two steps. In the future, we hope the reinforcement learning can help the connection between evidence extraction and answer synthesis.

5 Conclusion and Future Work

In this paper, we propose S-Net, an extraction-then-synthesis framework, for machine reading comprehension. The extraction model aims to match the question and passage and predict most important sub-spans in the passage related to the question as evidence. Then, the synthesis model synthesizes the question information and the evidence snippet to generate the final answer. We propose a multi-task learning framework to improve the evidence extraction model by passage ranking to extract the evidence snippet, and use the sequence-to-sequence model for answer synthesis. We conduct experiments on the MS-MARCO dataset. Results demonstrate that our approach outperforms pure answer extraction model and other existing methods.

We only annotate one evidence snippet in the sequence-to-sequence model for synthesizing answer, which cannot solve the question whose answer comes from multiple evidences, such as the second example in Table 1. Our extraction model is based on the pointer network which selects the evidence by predicting the start and end positions of the text span. Therefore the top candidates are similar as they usually share the same start or end positions. By ranking separated candidates for predicting evidence snippets, we can annotate multiple evidence snippets as features in the sequence-to-sequence model for questions in this category in the future.

Acknowledgement

We thank the MS-MARCO organizers for help in submissions.

References

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1724–1734, 2014.

Ian J. Goodfellow, David Warde-farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *ICML*, 2013.

Jiatao Gu, Zhengdong Lu, Hang Li, and O.K. Victor Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1631–1640. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1154. URL http://aclweb.org/anthology/P16-1154.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pp. 1693–1701, 2015.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. In *Proceedings of the International Conference on Learning Representations*, 2016.

Kenton Lee, Tom Kwiatkowski, Ankur Parikh, and Dipanjan Das. Learning recurrent span representations for extractive question answering. *arXiv preprint arXiv:1611.01436*, 2016.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.

Thang Luong, Hieu Pham, and D. Christopher Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421. Association for Computational Linguistics, 2015. doi: 10.18653/v1/D15-1166. URL http://aclweb.org/anthology/D15-1166.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268, 2016.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1532–1543, 2014.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392. Association for Computational Linguistics, 2016.

Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 193–203, 2013.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. Reasoning about entailment with neural attention. *CoRR*, 2015.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016.*, 2016.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 2692–2700. Curran Associates, Inc., 2015a.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pp. 2773–2781, 2015b.

Shuohang Wang and Jing Jiang. Learning natural language inference with LSTM. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, 2016a.

Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*, 2016b.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 189–198. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1018. URL `http://aclanthology.coli.uni-saarland.de/pdf/P/P17/P17-1018.pdf`.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. Fastqa: A simple and efficient neural architecture for question answering. *arXiv preprint arXiv:1703.04816*, 2017.

Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*, 2016.

Zhilin Yang, Bhuwan Dhingra, Ye Yuan, Junjie Hu, William W. Cohen, and Ruslan Salakhutdinov. Words or characters? fine-grained gating for reading comprehension. *CoRR*, abs/1611.01724, 2016.

Yang Yu, Wei Zhang, Kazi Hasan, Mo Yu, Bing Xiang, and Bowen Zhou. End-to-end reading comprehension with dynamic answer chunk ranking. *arXiv preprint arXiv:1610.09996*, 2016.

Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.