I Know There Is No Answer: Modeling Answer Validation for Machine Reading Comprehension

Chuanqi Tan[1(✉)], Furu Wei[2], Qingyu Zhou[3], Nan Yang[2], Weifeng Lv[1],
and Ming Zhou[2]

[1] Beihang University, Beijing, China
tanchuanqi@nlsde.buaa.edu.cn, lwf@buaa.edu.cn
[2] Microsoft Research Asia, Beijing, China
{fuwei,nanya,mingzhou}@microsoft.com
[3] Harbin Institute of Technology, Harbin, China
qyzhgm@gmail.com

Abstract. Existing works on machine reading comprehension mostly focus on extracting text spans from passages with the assumption that the passage must contain the answer to the question. This assumption usually cannot be satisfied in real-life applications. In this paper, we study the reading comprehension task in which whether the given passage contains the answer is not specified in advance. The system needs to correctly refuse to give an answer when a passage does not contain the answer. We develop several baselines including the answer extraction based method and the passage triggering based method to address this task. Furthermore, we propose an answer validation model that first extracts the answer and then validates whether it is correct. To evaluate these methods, we build a dataset SQuAD-T based on the SQuAD dataset, which consists of questions in the SQuAD dataset and includes relevant passages that may not contain the answer. We report results on this dataset and provides comparisons and analysis of the different models.

Keywords: Machine reading comprehension · Answer validation

1 Introduction

Machine reading comprehension, which attempts to enable machines to answer questions after reading a passage, has attracted much attention from both research and industry communities in recent years. The release of large-scale manually created datasets such as SQuAD [12] and TriviaQA [5] has brought great improvement for model training and testing of machine learning algorithms on the related research area. However, most existing reading comprehension datasets assume that there exists at least one correct answer in the passage set. Current models therefore only focus on extracting text spans from passages to

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11108, pp. 85–97, 2018.
https://doi.org/10.1007/978-3-319-99495-6_8

answer the question, but do not determine whether an answer even exists in the passage for the question. Although the assumption simplifies the problem, it is unrealistic for real-life applications. Modern systems usually rely on an independent component to pre-select relevant passages, which cannot guarantee that the candidate passage contains the answer.

In this paper, we study the reading comprehension task in which whether the given passage contains the answer is not specified in advance[1]. For the question whose passage contains the answer, the system needs to extract the correct text span to answer the question. For the question whose passage does not contain the answer, the system needs to correctly refuse to give the answer. We develop several baseline methods following previous work on answer extraction [12] and answer triggering [21]. We implement the answer extraction model [19] to predict the answer. We then use the probability of the answer to judge whether it is correct. In addition, we propose two methods to improve the answer extraction model by considering that there may be no answer. The first is to add a no-answer option with a padding position for the passage that does not contain the answer and supervise the model to predict this padding position when there is no answer. The second is to control the probability of the answer by modifying the objective function for the passage that does not contain the answer. Second, we develop the passage triggering based method, which first determines whether the passage contains the answer then extracts the answer only in the triggered passage. Finally, we propose the answer validation method, which first extracts the answer in the passage then validates whether it is correct.

To test the above methods, we build a new dataset SQuAD-T based on the SQuAD dataset. For each question in the SQuAD dataset, we use Lucene[2], an off-the-shelf tool, to retrieve the top relevant passage from the whole SQuAD passage set. If the top passage is the original corresponding passage in the SQuAD dataset that contains the answer, we treat the question and passage pair as a positive example. Otherwise, we treat the question and the top-ranked passage that does not contain the answer as a negative example. Table 1 shows two examples in the SQuAD-T dataset. In the first example, the passage contains the correct answer "Denver Broncos" (underlined). In the second example, the passage does not contain the answer. We use precision, recall and F_1 scores for the positive examples and overall accuracy for all data to evaluate this task.

Experiments show that both the answer extraction model with the no-answer option and the modified objective function improve the results of the answer extraction model. Our answer validation model achieves the best F_1 score and overall accuracy on the SQuAD-T test set. Further analysis indicates that our proposed answer validation model performs better in refusing to give the answers when passages do not contain the answers without performance degradation when passages contain the answers.

[1] We notice Rajpurkar et al. also address this problem [11] when this paper is under review.

[2] http://lucene.apache.org.

Table 1. Examples in the SQuAD-T dataset. The first example contains the answer "Denver Broncos" (underlined). The second example does not contain the answer to the question.

Question: Which NFL team represented the AFC at Super Bowl 50? **Passage:** The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24-10 to earn their third Super Bowl title
Question: Where did Super Bowl 50 take place? **Passage:** In addition to the Vince Lombardi Trophy that all Super Bowl champions receive, the winner of Super Bowl 50 will also receive a large, 18-karat gold-plated "50"

2 Related Work

Previous methods achieve promising results on the SQuAD dataset for reading comprehension. Since the passage must contain the answer to the question in the SQuAD dataset, state-of-the-art methods usually answer the question by predicting the start and end positions of the answer in the passage [4,13,18,20]. Unlike the SQuAD dataset that only has one passage for a question, the TriviaQA dataset [5] and the MS-MARCO dataset [9] contain multiple paragraphs or passages for a question. However, since the datasets still guarantee that it must contain the answer, state-of-the-art methods do not discriminate which passage contains the answer, but concatenate all passages to predict one answer [3,15,18].

Yang et al. [21] propose an answer triggering task with the WikiQA dataset. It aims to detect whether there is at least one correct answer in the set of candidate sentences for the question, and selects one of the correct answer sentences from the candidate sentence set if yes. Several feature-based methods [21] and deep learning methods [6,22] are proposed for this task. Chen et al. [1] tackle the problem of open-domain question answering, which combines the document retrieval (finding the relevant articles) with machine comprehension of text (identifying the answer spans from those articles). It only evaluates the coverage of the retrieval result and the accuracy of the final answer, but does not address the problem of the retrieved document not containing the answer.

3 Approach

Previous reading comprehension tasks usually aim to extract text spans from the passage to answer the question. In this work, the task is advanced that whether the given passage contains the answer is not specified. For the question whose passage contains the answer, the system needs to correctly extract the answer. Otherwise, the system needs to refuse to answer the question that there is no answer in the passage.

To solve this problem, we develop three categories of methods. First, we implement an answer extraction model and propose two methods to improve it for the passage that may not contain the answer. Second, we develop the passage triggering based method, which first judges whether the passage contains the answer then extracts the answer only in the triggered passage. Finally, we propose an answer validation model, which first extracts the answer then validates whether it is correct.

3.1 Answer Extraction Based Method

In this work, we implement the answer extraction model following match-LSTM [17] and R-Net [19], which have shown the effectiveness in many reading comprehension tasks.

Consider a question $\mathrm{Q} = \{w_t^Q\}_{t=1}^m$ and a passage $\mathrm{P} = \{w_t^P\}_{t=1}^n$, 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 [2] 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:

$$u_t^Q = \mathrm{BiGRU}_Q(u_{t-1}^Q, [e_t^Q, char_t^Q]), u_t^P = \mathrm{BiGRU}_P(u_{t-1}^P, [e_t^P, char_t^P]) \tag{1a}$$

Given question and passage representations $\{u_t^Q\}_{t=1}^m$ and $\{u_t^P\}_{t=1}^n$, [17] introduce match-LSTM, which combines the passage representation u_j^P with the passage-aware question representation c_t^Q to aggregate the question information to words in the passage, 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. [19] propose adding a gate to the input ($[u_t^P, c_t^Q]$) of GRU to determine the of passage parts.

$$s_j^t = \mathrm{v}^{\mathrm{T}}\mathrm{tanh}(W_u^Q u_j^Q + W_u^P u_t^P + W_v^P v_{t-1}^P) \tag{2a}$$

$$a_i^t = \exp(s_i^t)/\Sigma_{j=1}^m \exp(s_j^t) \tag{2b}$$

$$c_t^Q = \Sigma_{i=1}^m a_i^t u_i^Q \tag{2c}$$

$$g_t = \mathrm{sigmoid}(W_g[u_t^P, c_t^Q]) \tag{2d}$$

$$[u_t^P, c_t^Q]^* = g_t \odot [u_t^P, c_t^Q] \tag{2e}$$

$$v_t^P = \mathrm{GRU}(v_{t-1}^P, [u_t^P, c_t^Q]^*) \tag{2f}$$

We then obtain the question-aware passage representation v_t^P for all positions in the passage.

Following previous methods used on the SQuAD, we use pointer networks [16] to predict the start and end positions of the answer. Given the passage representation $\{v_t^P\}_{t=1}^n$, the attention mechanism is utilized as a pointer to select the start position (p^1) and end position (p^2) from the passage, which can be formulated as follows:

$$s_j^t = \mathrm{v}^{\mathrm{T}}\tanh(W_h^P v_j^P + W_h^a h_{t-1}^a) \tag{3a}$$

$$a_i^t = \exp(s_i^t)/\Sigma_{j=1}^n \exp(s_j^t) \tag{3b}$$

$$p^t = \mathrm{argmax}(a_1^t, \ldots, a_n^t) \tag{3c}$$

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:

$$c_t = \Sigma_{i=1}^n a_i^t v_i^P, h_t^a = \mathrm{GRU}(h_{t-1}^a, c_t) \tag{4a}$$

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:

$$s_j = \mathrm{v}^{\mathrm{T}}\tanh(W_u^Q u_j^Q + W_v^Q v_r^Q) \tag{5a}$$

$$a_i = \exp(s_i)/\Sigma_{j=1}^m \exp(s_j) \tag{5b}$$

$$r^Q = \Sigma_{i=1}^m a_i u_i^Q \tag{5c}$$

The objective function is to minimize the following cross entropy:

$$\mathcal{L} = -\Sigma_{t=1}^2 \Sigma_{i=1}^n [y_i^t \log a_i^t + (1 - y_i^t)\log(1 - a_i^t)] \tag{6a}$$

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

This model is trained on the positive examples in the SQuAD-T dataset. When predicting the answer, the answer extraction model outputs two probabilities at the start and end positions, respectively. We multiply them for the probability of each text span to select the answer. If the probability of the answer is higher than a threshold pre-selected on the development set, we output it as the final answer, otherwise we refuse to answer this question.

Answer Extraction with No-Answer Option

The answer extraction model has two issues. First, we can only train it with positive examples in which the passage contains the answer. Second, the score is relative since the probability of the answer is normalized in each passage. To handle these issues, we propose improving the answer extraction model with a no-answer option. Levy et al. [8] propose adding a trainable bias to the confidence score p^t to allow the model to signal that there is no answer in the relation extraction task. Similarly, we add a padding position for the passage and supervise the model to predict this position when the passage does not contain the answer. In addition to the prediction strategy in the answer extraction model, we refuse to give an answer when the model predicts the padding position.

Answer Extraction with Modified Objective Function

We develop another strategy to improve the answer extraction model by modifying the objective function. For the positive example, we use the original objective function in the answer extraction, for which the probability is set to 1 for correct start and end positions, otherwise it is 0. For the negative example, we modify the objective function as follows:

$$\mathcal{L} = -\Sigma_{t=1}^{2} \Sigma_{i=1}^{n} [y_i^t \log a_i^t + (1 - y_i^t) \log(1 - a_i^t)] \tag{7a}$$

where $y_i^t = \frac{1}{n}$ for all positions.

3.2 Passage Triggering Based Method

Unlike the answer extraction based methods that extract and judge the answer in one model, the passage triggering based method divides this task into two steps. We first apply a passage triggering model to determine whether the passage contains the answer. We then apply the answer extraction model only on the triggered passage for the answer.

For passage triggering, we follow the above-mentioned matching strategy to obtain the question-aware passage representation $\{v_j^P\}_{j=1}^n$ in Eq. 2 and the question representation r^Q in Eq. 5. We apply an attention pooling to aggregate the matching information to a fix length vector.

$$s_j = \mathrm{v}^{\mathrm{T}} \tanh(W_v^P v_j^P + W_v^Q r^Q) \tag{8a}$$

$$a_i = \exp(s_i) / \Sigma_{j=1}^{n} \exp(s_j) \tag{8b}$$

$$r^P = \Sigma_{i=1}^{n} a_i v_i^P \tag{8c}$$

We then feed r^P to a multi-layers perceptron for the decision. The objective function is to minimize the following cross entropy:

$$\mathcal{L} = -\sum_{i=1}^{N} [y_i \log p_i + (1 - y_i) \log(1 - p_i)] \tag{9a}$$

where p_i is the probability that the passage contains the answer. y_i denotes a label, $y_i = 1$ means the passage contains the answer, otherwise it is 0.

When predicting the answer, we first judge whether the passage contains the answer by comparing the probability with a pre-selected threshold on the development set. For the triggered passage, we then apply the extraction model for the answer.

3.3 Answer Validation Based Method

There is an issue posed by answer information not being considered in the passage triggering based method. To this end, we propose the answer validation model, which first extracts an answer then validates whether it is correct.

We first apply the answer extraction model to obtain the answer span. Next, we incorporate the answer information into the encoding part by adding additional features f_t^s and f_t^e, to indicate the start and end positions of the extracted answer span. $f_t^s = 1$ and $f_t^e = 1$ mean the position t is the start and end of the answer span, respectively.

$$u_t^P = \text{BiGRU}_P(u_{t-1}^P, [e_t^P, char_t^P, f_t^s, f_t^e]) \tag{10a}$$

Unlike the answer extraction that predicts the answer on the passage side, answer validation needs to judge whether the question is well answered. Therefore, we reverse the direction of all above-mentioned equations to aggregate the passage information with the question. Specifically, we reverse Eq. 2 to obtain the passage-aware question representations,

$$s_j^t = \text{v}^{\text{T}}\tanh(W_u^P u_j^P + W_u^Q u_t^Q + W_v^Q v_{t-1}^Q) \tag{11a}$$

$$a_i^t = \exp(s_i^t)/\varSigma_{j=1}^n \exp(s_j^t) \tag{11b}$$

$$c_t^P = \varSigma_{i=1}^n a_i^t u_i^P \tag{11c}$$

$$g_t = \text{sigmoid}(W_g[u_t^Q, c_t^P]) \tag{11d}$$

$$[u_t^Q, c_t^P]^* = g_t \odot [u_t^Q, c_t^P] \tag{11e}$$

$$v_t^Q = \text{GRU}(v_{t-1}^Q, [u_t^Q, c_t^P]^*) \tag{11f}$$

Based on the Eq. 11, we obtain the v_t^Q for each position of questions. We then make the decision by judging whether each question word is well answered by the passage and answer with three steps. First, we measure the passage-independent importance of question words. We hold that the importance of each word in the question should not vary no matter what the passage and answer are. For example, the interrogative and name entity are usually more important than the conjunction and stopwords. Therefore, we apply the gate mechanism to select the important information, which is produced by the original representation of each question word.

$$g_t = \text{sigmoid}(W_g u_t^Q), v_t^Q* = g_t \odot v_t^Q \tag{12a}$$

Next, we obtain the matching score of each question word by a multi-layers perceptron,

$$s_t^Q \propto \exp(W_2(\tanh(W_1 v_t^Q*))) \tag{13a}$$

Finally, we combine the matching score of question words adaptively. We apply the attention mechanism on the matching vector v_t^Q* based on the learned parameter v_s to obtain the weight of each question, and then apply it to weighted-sum the score s_t^Q for the final score s.

$$s_j^t = \text{v}^{\text{T}}\tanh(W_v v_s + W_u^Q v_t^Q*) \tag{14a}$$

$$a_i^t = \exp(s_i^t)/\varSigma_{j=1}^m \exp(s_j^t) \tag{14b}$$

$$s = \varSigma_{i=1}^m a_i^t s_t^Q \tag{14c}$$

As both the score and the weight of each question word are normalized, we treat the final score s as the probability that the answer is correct.

$$\mathcal{L} = -\sum_{i=1}^{N}[y_i \log s + (1 - y_i)\log(1 - s)] \tag{15a}$$

where y_i denotes a label, $y_i = 1$ means the answer is correct, otherwise 0.

When predicting the answer, we compare s with a threshold pre-selected on the development set to determine whether to answer the question with the extracted answer.

3.4 Implementation Details

For all above-mentioned models, we use 300-dimensional uncased pre-trained *GloVe* embeddings [10][3] 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 apply dropout [14] between layers, with a dropout rate of 0.2. The model is optimized using Adam [7] with default learning rate of 0.002.

4 Experiments

To evaluate methods in this task, we build a new dataset SQuAD-T based on the SQuAD dataset and propose using F-measure on the positive examples and overall accuracy for all data for evaluation. We report results of all above-mentioned models. Experimental results show that our answer validation model achieves the best F_1 score and accuracy on the SQuAD-T dataset. In addition, we provide detailed comparisons and analysis of all methods.

4.1 Dataset Construction

In real-life application (i.e. search engine), given a question (or query), it usually first retrieves the relevant passage then discriminates whether there is an answer. In this work, we simulate this process to build the SQuAD-T dataset based on the SQuAD dataset. Specifically, we use Lucene to index all passages in the SQuAD dataset. Then for each question in the SQuAD dataset, we obtain one relevant passage by searching the question with Lucene using its default ranker based on the vector space model and TF-IDF[4]. We observe that only 65.67% of questions whose most related passages are still original corresponding passages in the SQuAD dataset. We then treat these question and passage pairs as the positive examples in which the passages contain the answer. For other questions, we select the top-ranked passage that does not contain the answer as

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

[4] Details can be found in https://lucene.apache.org/core/2_9_4/api/core/org/apache/lucene/search/Similarity.html.

the negative example. As the author of SQuAD only publishes the training set and the development set, we split the 10,570 instances in the development set to 5,285 for development and 5,285 for test. The statistics of the SQuAD-T dataset are shown in Table 2.[5]

Table 2. Statistics of the SQuAD-T dataset.

	Train	Dev	Test
Question	86,830	5,285	5,285
Positive	57,024	3,468	3,468
Negative	29,806	1,817	1,817

4.2 Evaluation Metrics

Previous work adopts Exact Match and F_1[6] to evaluate the performance of the reading comprehension model [12]. These metrics are to evaluate the extracted answer in the case that the passage must contain the answer, and hence are not suitable for data in which there is no answer. In this work, we propose using precision, recall and F_1 scores at the question level to evaluate this task. A question is treated as a positive case only if it contains the correct answer in the corresponding passage. Given the question set Q, Q^+ is the set where there is an answer in the passage to answer the question, otherwise Q^-. T^+ is the set where the model gives an answer, otherwise T^-. A^+ is the set where the given answer is correct, otherwise A^-. We define the F-measure as follows:

$$Precision = \frac{|A^+|}{|T^+|}, Recall = \frac{|A^+|}{|Q^+|}, F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{16a}$$

In addition, we define the overall accuracy as follows:

$$Acc = \frac{|A^+| + |T^- \cap Q^-|}{|Q|} \tag{17a}$$

4.3 Main Result

We show the result in terms of precision, recall, and F_1 in Table 3, and illustrate the Precision-Recall curves on the development and test set in Fig. 1, respectively. The answer extraction model only achieves 68.59 and 67.50 in terms of F_1 on the development set and test set, respectively. Since the probability of the answer is normalized on each passage, the score is relative and therefore

[5] We release the dataset in https://github.com/chuanqi1992/SQuAD-T.

[6] Here the F_1 score is calculated at the token level between the true answer and the predicted answer.

Table 3. Results in terms of precision, recall, and F_1 on the SQuAD-T development and test set. *Significant improvement over the baseline method of the answer extraction (underlined) (t-test, $p < 0.05$).

Method	Development set			Test set		
	Prec	Rec	F_1	Prec	Rec	F_1
Answer extraction	66.70	70.59	68.59	66.35	68.69	67.50
Answer extraction with no-answer option	74.48	67.50	70.82	74.63	67.44	70.86
Answer extraction with modified objective function	74.63	67.68	70.98	73.09	66.35	69.55
Passage triggering then extraction	59.65	74.65	66.32	58.27	73.33	64.94
Answer validation	69.73	73.41	**71.53***\	68.74	73.93	**71.24***

performs worse for judging whether it is a real answer. In addition, this model achieves 78.374 and 77.105 in terms of Exact Match on the positive examples in the development and test set, respectively, which is used to provide the extraction result for the passage triggering based model and answer validation model. It determines the max recall of these related models shown in Fig. 1.

Improving the answer extraction model by adding the no-answer option and modifying the objective function greatly improves the result. The passage triggering based method only achieves 66.32 and 64.94 on the development set and test set, respectively. We observe that the answer validation model obviously outperforms the passage triggering based method since it incorporates the answer information. Our answer validation model outperforms all other baselines and achieves best the F_1 score with 71.53 and 71.24 in the development set and test set, respectively.

We show the overall accuracy on the SQuAD-T development and test set in Table 4. The answer extraction model with the no-answer option and modified objective function consistently improve the result of the answer extraction. Our answer validation model achieves the best overall accuracy of 74.60 on the SQuAD-T test set.

Table 4. Results in terms of overall accuracy on the SQuAD-T development and test set.

Method	Dev	Test
Answer extraction	65.26	64.34
Answer extraction with no-answer option	**74.65**	74.48
Answer extraction with modified objective function	74.14	73.08
Passage triggering then extraction	63.28	62.44
Answer validation	73.98	**74.60**

4.4 Model Analysis

Figure 1 shows the precision-recall curves on the development set and test set, respectively. We observe that the answer extraction based method achieves better precision when the recall is relatively low. With the increase of the recall, the precision of the answer extraction model obviously decreases, which indicates that the score of the answer extraction is not suitable for judging whether it is a real answer. Improving the answer extraction model by the no-answer option or modified objective function can partly relieve this issue. However, we observe that the max recall of these two improved methods is much lower than the answer extraction model in Fig. 1, which indicates that training these two models with negative examples leads to worse extraction precision on the positive examples. Therefore, we argue that the answer extraction model should only be trained on the positive example for better extraction precision. The answer validation model achieves better performance than the passage triggering based method. Our answer validation model almost maintains stable precision with the increase of the recall, which leads to the best F_1 score on the SQuAD-T dataset.

Table 5 shows the detailed result distribution of all methods. We observe that the answer extraction model with no-answer option and modified objective function achieve the lower ratio in Q^-T^+ and higher ratio in Q^-T^-, which shows the effectiveness of refusing to give an answer when there is no answer. However, these two methods sacrifice the precision as they have much lower ratios in $Q^+T^+A^+$. In addition, they also have higher ratios in Q^+T^-. Therefore, if we calculate the F-measure of negative examples, the result of answer extraction with no-answer option and modified function are 80.49 and 78.96, respectively,

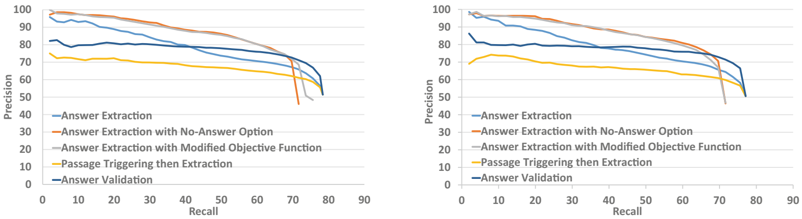

Fig. 1. Precision-Recall curves on the development and test set.

Table 5. The result distribution on the SQuAD-T test set. The values are percentages in corresponding categories.

Method	$Q^+T^+A^+$	$Q^+T^+A^-$	Q^+T^-	Q^-T^+	Q^-T^-
Answer extraction	45.07	7.72	12.83	15.14	19.24
+ No-answer option	44.26	10.88	10.49	**4.16**	**30.22**
+ Modified objective function	43.54	11.18	10.90	**4.84**	**29.54**
Passage triggering then extraction	48.12	14.40	**3.10**	20.06	14.32
Answer validation	**48.51**	13.77	**3.33**	8.29	26.09

which is still lower than 81.79 for the answer validation model. The passage triggering based method achieves a worse result with the highest ratio in $Q^{-}T^{+}$ because it does not consider the answer information when making the decision. The answer validation model achieves the better results in $Q^{+}T^{+}A^{+}$. Meanwhile, it also achieves relative lower ratio in $Q^{-}T^{+}$ and relative higher ratio in $Q^{-}T^{-}$ compared with other methods using the answer extraction model to extract the answer, which indicates that it performs better in refusing to give the answers when passages do not contain the answers without performance degradation when passages contain the answers. Our answer validation model therefore achieves the best result in terms of F_1 and overall accuracy in the SQuAD-T test set.

5 Conclusion and Future Work

In this paper, we study the machine reading comprehension task in which whether the passage contains the answer is not specified. Therefore the system needs to correctly refuse to give an answer when a passage does not contain the answer. We develop several baseline methods including the answer extraction based method, the passage triggering based method, and propose the answer validation method for this task. Experiments show that our proposed answer validation model outperforms all other baseline methods on the SQuAD-T test set. We notice that Rajpurkar et al. build a dataset SQuAD2.0 in which questions are written by humans. We will test our methods on this benchmark dataset in the future.

Acknowledgments. We greatly thank Hangbo Bao for helpful discussions. Chuanqi Tan and Weifeng Lv are supported by the National Key R&D Program of China (No. 2017YFB1400200) and National Natural Science Foundation of China (No. 61421003 and 71501003).

References

1. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading Wikipedia to answer open-domain questions. In: ACL (2017)
2. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP, pp. 1724–1734. Association for Computational Linguistics (2014)
3. Clark, C., Gardner, M.: Simple and effective multi-paragraph reading comprehension. arXiv preprint arXiv:1710.10723 (2017)
4. Huang, H.Y., Zhu, C., Shen, Y., Chen, W.: FusioNnet: Fusing via fully-aware attention with application to machine comprehension. In: ICLR (2018)
5. Joshi, M., Choi, E., Weld, D., Zettlemoyer, L.: Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In: ACL, pp. 1601–1611. Association for Computational Linguistics (2017)

6. Jurczyk, T., Zhai, M., Choi, J.D.: SelQA: a new benchmark for selection-based question answering. In: 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 820–827. IEEE (2016)
7. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
8. Levy, O., Seo, M., Choi, E., Zettlemoyer, L.: Zero-shot relation extraction via reading comprehension. In: CoNLL, pp. 333–342. Association for Computational Linguistics (2017)
9. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: MS MARCO: a human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268 (2016)
10. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
11. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: unanswerable questions for squad. In: ACL (2018)
12. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. In: EMNLP (2016)
13. Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H.: Bidirectional attention flow for machine comprehension. In: International Conference on Learning Representations (2017)
14. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014)
15. Tan, C., Wei, F., Yang, N., Du, B., Lv, W., Zhou, M.: S-Net: from answer extraction to answer generation for machine reading comprehension. AAAI (2018)
16. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 28. Curran Associates, Inc. (2015)
17. Wang, S., Jiang, J.: Learning natural language inference with LSTM. In: The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 (2016)
18. Wang, S., Jiang, J.: Machine comprehension using match-LSTM and answer pointer. In: International Conference on Learning Representations (2017)
19. Wang, W., Yang, N., Wei, F., Chang, B., Zhou, M.: Gated self-matching networks for reading comprehension and question answering. In: Proceedings of the 55th ACL, pp. 189–198. Association for Computational Linguistics (2017)
20. Xiong, C., Zhong, V., Socher, R.: Dynamic coattention networks for question answering. arXiv preprint arXiv:1611.01604 (2016)
21. Yang, Y., Yih, W.t., Meek, C.: WikiQA: a challenge dataset for open-domain question answering. In: Proceedings of EMNLP, pp. 2013–2018. Citeseer (2015)
22. Zhao, J., Su, Y., Guan, Z., Sun, H.: An end-to-end deep framework for answer triggering with a novel group-level objective. In: EMNLP, pp. 1276–1282. Association for Computational Linguistics (2017)