Reinforced History Backtracking for Conversational Question Answering

Minghui Qiu[1,*], Xinjing Huang[2,*], Cen Chen[1,†], Feng Ji[1], Chen Qu[1], Wei Wei[3], Jun Huang[1], Yin Zhang[2,†]

[1] Alibaba Group, China
[2] Zhejiang University
[3] Huazhong University of Science and Technology
minghui.qmh@alibaba-inc.com,chencen.cc@antfin.com,{huangxinjing,zhangyin98}@zju.edu.cn,weiw@hust.edu.cn

Abstract

To model the context history in multi-turn conversations has become a critical step towards a better understanding of the user query in question answering systems. To utilize the context history, most existing studies treat the whole context as input, which will inevitably face the following two challenges. First, modeling a long history can be costly as it requires more computation resources. Second, the long context history consists of a lot of irrelevant information that makes it difficult to model appropriate information relevant to the user query. To alleviate these problems, we propose a reinforcement learning based method to capture and backtrack the related conversation history to boost model performance in this paper. Our method seeks to automatically backtrack the history information with the implicit feedback from the model performance. We further consider both immediate and delayed rewards to guide the reinforced backtracking policy. Extensive experiments on a large conversational question answering dataset show that the proposed method can help to alleviate the problems arising from longer context history. Meanwhile, experiments show that the method yields better performance than other strong baselines, and the actions made by the method are insightful.

Introduction

Conversational **Q**uestion **A**nswering (ConvQA) have recently interested a lot of researchers in both research community and industry (Choi et al. 2018b; Reddy, Chen, and Manning 2018a). Under the ConvQA setting, the current query in each dialogue turn may not be self-contained and depends greatly on the dialogue history, as the phenomenon of coreference and pragmatic omission frequently occurs in the dialogs. Thus modeling the dialogue history becomes a critical step towards better understanding the query. Most of the existing works tend to model the semantic changes using the whole history as the input and perform coreference resolution in a single model, such as FlowQA (Huang, Choi, and Yih 2018), FlowDelta (Yeh and Chen 2019) and MC^2(Zhang 2019). Recent state-of-the-art studies append all dialogue history by using history answer embedding (Qu et al. 2019a) or question attention (Qu et al. 2019b), which can be viewed as a form of soft selection for related history.

However, considering the whole history in the single model will inevitably face some challenges. First, it requires more computation resources to incorporate the representation of all the history, including both the relevant and irrelevant ones, which may be unnecessary for understanding the query. Moreover, this issue gets even worse when we adopt a heavy model such as BERT large, as a longer input sequence with the whole history need to be maintained. Second, existing works that model the whole history usually employ attention or gating based mechanisms to selectively attend to different history turns (Yeh and Chen 2019; Huang, Choi, and Yih 2018). However, those methods still achieve less ideal results due to the irrelevant parts that appeared in the dialogue history. In other words, the existing methods can benefit from an additional step of relevant history extraction.

To alleviate the above-mentioned problems, in this paper we work from a different perspective and seek to make meaningful selections of conversation history. The advantage of our method is that it can avoid the negative impact of unimportant history turns from the source by not considering them. We model the ConvQA task as two subtasks: a conversational QA task using a neural MRC model and a conversation history selection task with a reinforced backtracker. The reinforced backtracker is an agent that interacts with the environment constructed by the ConvQA.

More specifically, for each query, we view the process of finding the related history as a sequential decision-making process. The agent acts on the available conversation history and backtracks the history question-answer pairs turn by turn to decide whether it is relevant/useful based on the observations. The MRC model then uses the selected history turns to help itself answer the current question and generates a reward to evaluate the utility of the history selection. However, the rewards generated by the MRC model are sparse, as they can only be obtained at the end of the decision process. To address this sparse reward problem, we further propose a novel training scheme, in which the agent first learns from the examples with only one history turn, followed by learning from examples with two history turns, and so on so forth.

As irrelevant histories are filtered, the MRC model can be better trained with a more sophisticated mechanism and concentrate on fitting the history turns with higher confi-

[*] Equal contributions.
[†] Corresponding authors.

dence. Moreover, as the reinforced backtracker is a separate module, it can be flexibly adapted and further improved with techniques such as transfer learning in the future.

In all, our contributions can be summarized as follows:

1. We propose a novel solution for modeling the conversation history in the ConvQA setting. We incorporate a reinforced backtracker in the traditional MRC model to filter the irrelevant history turns instead of evaluating them as a whole. As a consequence, the MRC model can concentrate on the more relevant history and obtain better performance.
2. We model the conversation history selection problem as a sequential decision-making process, which can be solved by reinforcement learning (RL). By interacting with a pretrained MRC model, the reinforced backtracker is able to generate good selection policies. We further propose a novel training scheme to address the sparse reward issue.
3. We conduct extensive experiments on a large conversational question answering dataset QuAC (Choi et al. 2018a), and the results show that the learned conversation history selection policy by RL could help boost answer prediction performance.

The rest of our paper is organized as follows. We first formulate the conversation history selection problem in the ConvQA setting and thoroughly elaborate our proposed approach that uses reinforcement learning to train backtracking policy for useful history-turn selection. We then conduct detailed experiments on the QuAC dataset. Finally, we present related works and conclude the work.

Models

In this section, we first define our task and then present our proposed reinforced backtracker.

Task Definition

We define the conversation history selection task on top of the ConvQA task. We formulate our task into two subtasks: a conversational QA task and a conversation history selection task. Given the current question Q_k and dialogue history $H = \{(Q_i, A_i)|_{i=0}^{i=k-1}\}$, our reinforced backtracker aims to find a subset $H' \in H$ of most relevant history turns, to maximize the performance of the ConvQA task.

Model Overview

As illustrated in Figure 1, we model the history selection problem as a sequential decision-making process. Given the current query, the agent backtracks the history and obtains the state representation in each dialogue turn through the state network. The policy network takes the state representation and the last action to decide whether this history turn is related to the query. Subsequently, the MRC model uses the selected history turns and the passage as the inputs to predict the answer span. The history selection quality has a direct impact on the answer prediction performance. Thus, the MRC model is able to generate a reward to evaluate the utility of the history selection. Finally, the reward is used to update the policy network. We now introduce the state, agent, environment, and reward in detail in the following sections.

State

The state of a given history turn (Q_i, A_i) is denoted as a continuous real valued vector $\mathbf{S_i} \in R^l$, where l is the dimension of the state vector. The state vector S in i-th selection is the concatenation of the following features:

$$\mathbf{S_i} = [h_i \oplus V(a_{i-1}) \oplus V(i) \oplus \omega_i] \tag{1}$$

- **Sentence Vector** h_i**.** We adopt the average of the word's hidden representation generated by the vanilla BERT model as the sentence vector, where the input to the BERT is a sentence pair as follows: `[CLS]`$Q_i A_i$`[SEP]`.
- **Last Action's Vector** $V(a_{i-1})$**.** We embed the last action into an action vector with a length of 20.
- **Position Vector** $V(i)$. We embed the current relative step into this vector, which is designed to inject the position information.
- **Segment Embedding** ω. This vector is defined as the average of past sentence embeddings whose corresponding action is 1 (denotes being selected), formally:

$$\omega_i = \sum_{m=1}^{i} h_m, \ where\ a_m = 1. \tag{2}$$

RL Agent

Policy Network. Given the state, our policy network is a fully connected neural network, defined as follows:

$$P = softmax(W \times S). \tag{3}$$

At the training stage, we calculate the action distribution and sample actions from the distribution. At the evaluation stage, we select the action according to the max probability.

The policy gradient is calculated as followings:

$$\nabla_\theta J(\theta) = E\Big(\sum_{t=1}^{L}(R - b(\tau))\nabla_\theta \log \pi_\theta(a_t|s_t)\Big), \tag{4}$$

where b is the baseline, designed to reduce the variance. We adopt the average return in the batch as our baseline. R is the accumulated reward, which will be discussed in the section .

Action. Since our goal is to select the related history turns, the agent has two possible options for each turn, i.e., 0 (ignored) or 1 (selected).

Environment

Given the current question Q_k, a subset of the dialogue history H' and passage P, the environment prepends the conversation history to the current question to convert the multi-turn conversational QA task to the single-turn QA task. Then the model predicts the answer span and generates a reward to evaluate the utility of the history selection for predicting the answer.

In this paper, we adopt BERT (Devlin et al. 2018) as our MRC model. The input for BERT is defined as

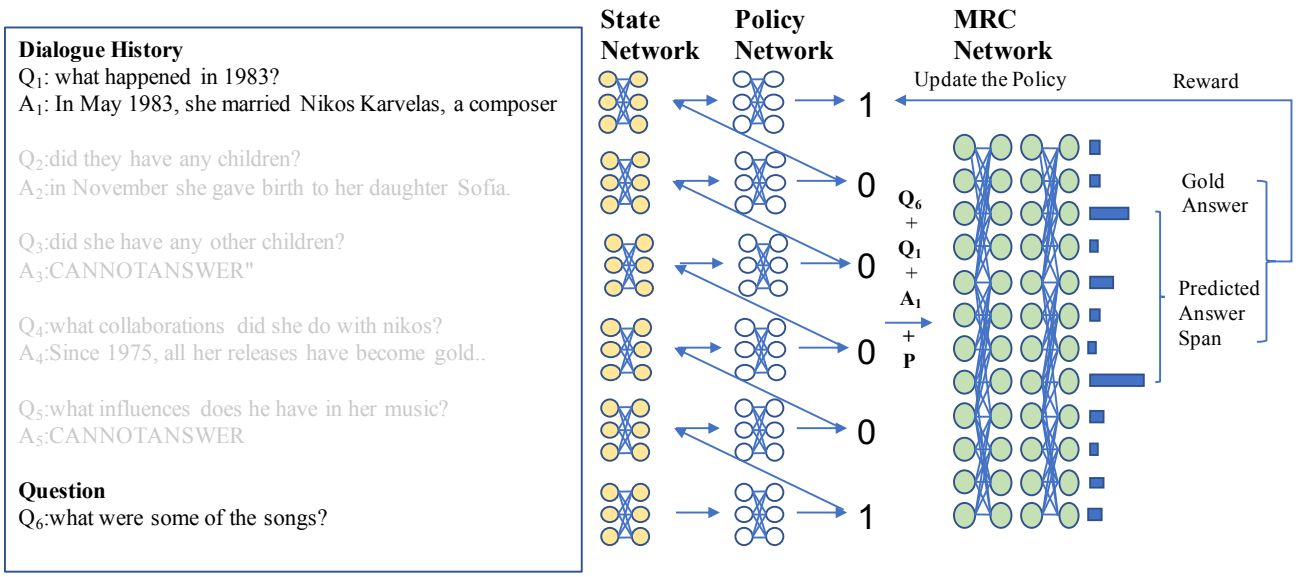

Figure 1: Overview of the our proposed MRC model with reinforced backtracker for the ConvQA task. The left part is an example of a dialogue selected from the QuAC dataset. Given the current question Q_6, the agent (middle part) starts to backtrack the history. It obtains the state representation in each dialogue turn through the state network. The policy network takes state representation and the last action to decide whether this turn is related to Q_6. Subsequently, the MRC network takes the selected history turns (e.g. Q_6, Q_1, A_1) and the passage P as the inputs, to predict the answer span. Finally, the reward is used to update the policy network.

$\texttt{[CLS]}\, Q_k\, \texttt{[SEP]}\, H'\, \texttt{[SEP]}\, P\, \texttt{[SEP]}$, where Q_k and P refer to the current k-th question and the passage, H' is the set of the selected history turns. We denote the output of BERT as $H_{rc} \in R^{L \times D_m}$, where L is the passage length and D_m is the dimension of the vector. Formally, we predict the start and end positions of the answer as the following:

$$P_s = Softmax(W_s H_{rc}^T + b_s), \quad (5)$$

$$P_e = Softmax(W_e H_{rc}^T + b_e), \quad (6)$$

where W_s, $W_e \in R^{1 \times D_m}$, b_s, $b_e \in R^{1 \times L}$, and s stands for the start while e stands for the end.

Reward

Our goal is to maximize the accuracy of the MRC model's prediction through the inputs selected by the agent. So an intuitive way is to adopt the word-level F1 score between the predicted answer and the gold answer as our reward R. If the input information is not sufficient for the model to predict the answer correctly, the F1 score can be low. Formally, the F1 score is defined as follows:

$$F1 = \frac{2 * P * R}{P + R}, \quad (7)$$

where P is the overlap percentage of the gold answer that counts in the predicted answer, R is the overlap percentage that counts in the gold answer. We then define the delayed reward as the difference of F1 scores after model updates, i.e., $R^{delayed} = \Delta F1$.

Note that the above F1 score serves as the delayed reward as it is obtained after all the selection actions at all the history turns. Such sparse reward poses challenges for the RL policy. Hence, we further consider incorporating immediate rewards for all actions to help RL training. Let's denote the representation for the history sentences $H = \{(Q_i, A_i)|_{i=0}^{i=k-1}\}$ as $S_H = \{S_i|_{i=0}^{i=k-1}\}$ and the action for the i-step as a_i. Then the immediate reward is defined based on the similarity between the current sentence vector and history segment embeddings as defined in Eq. 2. The idea is we shall encourage adding a sentence that is close to the history segment embeddings, as it may provide coherent information to help history modeling. Formally, we have:

$$R_i^{immediate} = sign(a_i == 1) cos(h_i, \omega_{i-1}), \quad (8)$$

where $sign(\cdot)$ is 1 if the action a_i is 1, and -1 otherwise. Clearly, if action is 1, the reward is higher if the current sentence is closer to the segment embeddings.

The final reward for state i is obtained as follows:

$$R_i = \delta^{T-i} R^{delayed} + R_i^{immediate}, \quad (9)$$

where δ is the discount factor and T is the final step index.

Algorithm

As illustrated in Figure 1, our method consists of three modules: a state network using a vanilla BERT, a policy network (i.e., the agent), and a pretrained MRC model. The state network provides state representations for the policy network to make history backtracking actions. The resulting history turns and the user question are feed to the MRC model to obtain the prediction results. The policy network is updated according to the defined rewards computed based

Algorithm 1 RL Training Procedure

Require: Environment M: A pre-trained MRC model with latest 8 history turns;
Sentence Representation Model: (Vanilla) $BERT_s$
Policy Network: P_n

for j in range(MAX History Turn) **do**
  for $Q_k, H = \{(Q_i, A_i)|_{i=0}^{i=k-1}\}$ in training data **do**
    $V_q = BERT_s(Q_k)$
    if $len(H) > j$ **then**
      continue
    end if
    Actions=[]
    for (Q_i, A_i) in H **do**
      $h_i = BERT_s(Q_i \oplus A_i)$
      $State = h_i \oplus \omega \oplus V(a_{i-1}) \oplus V_i$
      $a_i = P_n(State)$
      Actions.append(a_i)
    end for
    Obtain history subset H' according to $Actions$
  end for
  Obtain reward R according to Eq. 9
  Update the policy network according to Eq. 4
end for

on the prediction results and the ground truth answer. As for the state network, we adopt the pre-trained model from github[1] and freeze its weights. And we use REINFORCE algorithm (Williams 1992) to update the policy of the agent. The detailed algorithm is presented in Algorithm 1.

Training Scheme Inspired by curriculum learning (Bengio et al. 2009), we consider gradually increasing the difficulty of learning to help the training of the RL agent. The agent first learns policy from the episodes with only one history, which can be viewed as a simplified selection procedure. Then we increase the length of the episodes to help the agent developing its context modeling strategies.

Experiments

Datasets

We conduct experiments on the QuAC dataset[2]. QuAC is a machine reading comprehension task with multi-turn interactions, where the current question often refers to the dialogue history. Some dialogue behaviors often occur such as topic shift, drill down, and topic return. There are mainly 100k questions and 10k dialogue in the dataset. The maximum round in the dialogue is 12.

We also evaluate the methods on an additional Canard [3] dataset. This dataset contains manually-generated questions based on the context history, which can serve as ground-truth for context modeling. This dataset can help to examine whether the competing methods have the ability to choose proper context history to generate high quality rewritten questions close to the manually-generated questions.

[1]https://github.com/google-research/bert

[2]https://quac.ai/

[3]https://sites.google.com/view/qanta/projects/canard

Evaluation Metrics The QuAC challenge provides two evaluation metrics, i.e., the word-level F1 and the human equivalence score (HEQ). The word-level F1 evaluates the overlap of the prediction and the ground truth answer span. It is a classic metric used in (conversational) machine comprehension tasks (Rajpurkar et al. 2016; Reddy, Chen, and Manning 2018b). HEQ measures the percentage of examples for which system F1 exceeds or matches human F1. Intuitively, this metric judges whether a system can provide answers as good as an average human. This metric is computed on the question level (HEQQ) and the dialog level (HEQD).

Environment

We test our method on the following varied environments. We adopt the same model architecture but with different inputs and training corpus.

- **Env-ST (Single Turn)** We denote the method of training a single turn MRC model on the first turn of dialogues in QuAC dataset as Env-ST. This is to avoid influence brought by introducing dialogue history for the MRC model. Note the number of examples is the same as the number of dialogues in QuAC. The training dataset has 11,567 examples.
- **Env-Canard (Canard Dataset))** As the Canard dataset has re-written questions based on the history turns, it can serve as a perfect environment to examine different history modeling policies. We denote the method of training the MRC model on the re-written questions from Canard as Env-Canard. It has about 31k training examples.
- **Env-ConvQA (Multi-turn)** We denote the method of appending the latest k history question-answer pairs to its current question as Env-ConvQA. Formally, the current question Q_k and its latest 8 history turns $H_8 = \{(Q_k - i, A_k - i)|_{i=1}^{i=min(8,k)}\}$ are concatenated to be a new long question and then accepted as the input of the model. This method is a strong baseline as shown in (Zhu, Zeng, and Huang 2018; Ju et al. 2019).

Baselines

We compare our method with the recent studies on ConvQA (Choi et al. 2018a; Qu et al. 2019b,a). Note that these studies can be treated as variants of our method where we define a rule-based policy to select the latest k history question-answer pairs. Hence these variants are denoted as **Rule-K**, where K is the number of selected history pairs.

The Necessity of the Selection

As shown in the previous study (Yatskar 2019), topic shift and topic-return are common in conversations, thus it is necessary to prepend history turns in the model. But there still remains the question: *is it true that, the more history turns appended, the better the performance will be?*

To examine this, we conduct experiments on training ConvQA model with various history turns. As shown in Table 1, the performance will increase when we append 8 history turns instead of 4, but decrease when we append the latest 12 turns. In general, it is beneficial to incorporate appropriate

Models	F1	HEQQ	HEQD	Total
ConvQA w/o history	55.93	49.43	3.3	108.66
ConvQA w/ 4 avg	63.84	59.29	5.8	128.93
ConvQA w/ 8 avg	**64.02**	**59.59**	**6.3**	**129.91**
ConvQA w/ 12 avg	63.12	58.37	5.5	126.99

Table 1: The model performance on ConvQA. Here we append the past k history question-answer pairs to the current question. We report the averaged results on the development dataset for k in {0,4,8,12} over 5 runs.

Env-Canard	F1	HEQQ	HEQD	Total
Rule-0	48.61	41.43	2.2	92.24
Rule-4	44.26	36.92	0.9	82.08
Rule-8	44.25	36.91	0.9	82.06
Rule-12	44.25	36.91	0.9	82.06
Agent	**49.90**	**42.89**	**2.3**	**95.09**

Table 2: The model performance on Env-Canard.

Env-ST	F1	HEQQ	HEQD	Total
Rule-0	33.60	**27.58**	1.0	62.18
Rule-4	31.00	21.89	0.5	53.39
Rule-8	31.05	21.92	0.5	53.47
Rule-12	31.05	21.92	0.5	53.47
Agent	**33.62**	27.49	**1.1**	**62.21**

Table 3: The model performance on single-turn environment (Env-ST).

Env-ConvQA	F1	HEQQ	HEQD	Total
Rule-0	46.98	38.14	1.8	86.92
Rule-4	66.05	61.89	**7.3**	135.24
Rule-8	66.09	61.97	**7.3**	135.36
Rule-12	66.09	61.97	**7.3**	135.36
Agent	**66.11**	**62.06**	**7.3**	**135.47**

Table 4: The model performance on a tailored Env-ConvQA with the same training samples from Env-Canard but without re-written questions.

history turns. The incorporation of history poses challenges for modeling as well, thus it is not always better to incorporate more history turns. The potential reasons behind this are: 1) Information conflict. It is hard for the model to automatically capture the dependencies between related parts. 2) Length limitation. The pretrained BERT model is limited to a length of 512. The more turns we appended, the more passage or history terms need to be truncated to fit the model, which makes the model difficult to extract the key information from the input. This motivates us to consider history selection to further improve the model performance.

Reinforcement Learning vs. Rule Policy

In this section, we aim to examine the benefits of our reinforcement learning method. We conduct reinforcement learning on three environments Env-ConvQA, Env-ST, and Env-Canard as discussed above.

As shown in Table 2, we compare the performance of different settings of our agent learning and rule policy. For Env-Canard, the training samples are backed with manually rewritten questions, which can be viewed as a good environment without the need for history modeling. We use such an environment to examine how the appended history pairs affect model performance. We can see that Rule-0, the policy with no history, can achieve the best performance. And with more histories appended, the performances of Rule-4, Rule-8, and Rule-12 start to drop dramatically. This is intuitive as the re-written question with no history has already contained all the useful information, more history turns may bring more noise information which makes the modeling more challenging. When applied our method, the performance increases greatly. This shows our agent can help to backtrack helpful history turns to achieve better performance, i.e., dig out useful information from history turns. Surprisingly, our method even outperforms Rule-0, which shows our method has the potential to track more useful information to further enhance the manually crafted questions.

We further test our method on Env-ST to examine the effectiveness if no rewritten dataset is provided. As shown in Table 3, all model performances drop drastically. The reason is that without a perfect environment, the RL and rule-based agent have less satisfactory performance. But, our method can still boost the performance over all the rule-based agents, which shows the effectiveness of our policy.

We also report the results of our agent on Env-ConvQA to examine the model performance on datasets without re-written questions. Note that for fair comparison with Env-Canard, Env-ConvQA is tailored to only incorporate the same training samples from Env-Canard but without re-written questions. As shown in Table 4, the more history turns are appended, the better performance the rule policy can obtain. But it stops increasing when we append 8 history turns. This further echoes our findings that it is not always better to incorporate longer history turns. Despite of this, when applied our reinforcement learning scheme, the model performance can be further boosted, thanks to its history selection policy.

In a nutshell, we find that it is not always better to incorporate more history turns in ConvQA, as longer history poses new challenges for history modeling. We then propose a history selection policy to alleviate this problem and examine the effectiveness of the proposed method on three environments. The above results show our method consistently outperforms the competing baselines, which demonstrates the advantage of our proposed method.

The Comparison of Our Learning Scheme and Episode Learning

Recall that our training scheme is inspired by curriculum learning (Bengio et al. 2009), where we first learn the policy from examples with only one history, followed by learning from examples with two history turns, and so on so forth.

Learning Scheme	F1	HEQQ	HEQD	Total
Our Scheme	47.69	40.48	2.5	90.67
Episode Learning	43.26	35.98	1.1	80.34

Table 5: Different learning procedure of reinforcement learning on Env-Canard on the full corpus.

We examine the effectiveness of the training scheme in this section.

As shown in Table 5, we conduct experiments with different learning methods. The agent with Episode learning interacts with the environment with examples in the natural order that appeared in datasets. We can see that the training scheme performs much better than episode learning. A potential reason is that the training scheme can provide a warm start stage, as it first learns the policy from examples with less history turns, followed by learning from examples with more history turns. It can be viewed as a student learning from easy courses to hard courses.

The Comparison of Our Method and other ConvQA Methods

We compare our method on Env-ConvQA with the following state-of-the-art ConvQA methods:

- BiDAF++: a classic ConvQA model without history (Choi et al. 2018a).
- BiDAF++ w/ 2-C: BiDAF++ with two history turns.
- FlowQA: a ConvQA model with history flow modeling (Huang, Choi, and tau Yih 2018).
- BERT+HAE: a recent proposed ConvQA model with history modeling (Qu et al. 2019b).
- BERT+PosHAE: an improved method based on BERT+HAE (Qu et al. 2019b).
- HAM: the most recent study with history attention modeling from (Qu et al. 2019c).

Note that we omit the comparison of the methods that consider transfer learning or data augmentation here, as these studies used external data. We leave the study of enhancing our method with external data as the future work.

As shown in Table 6, our method obtains the best performance in F1, HEQD, and wins first place in the total score. Our method exceeds the recent history attended model HAM by a score of 0.4 in F1. Given the task is very challenging, this improvement is not small. This further shows our policy of selecting related history turns is superior to the other methods.

Case Study

We sample two examples from the development dataset in QuAC and visualize the actions made by the agent.

As shown in Table 7, it lists two consequent questions in one dialogue. The upper part shows the question "Is there anything else interesting in the article ?", the agent thinks all the history is relevant as the actions made by the agent are all ones. In the bottom part, given the question, "Are they close to solving it?", we find the term 'it' refers to the event mentioned in the last turn. Interestingly, the agent selects only the last turn as the most relevant part, which is reasonable. This is intuitive, as to answer the first question, the agent needs to attend to all the history, while for the second question, only the latest information is required. Despite the history of these two questions are almost the same, the RL agent is capable of making different choices according to the given questions. This shows our RL agent can selectively backtrack conversation history turns to help the ConvQA task.

Methods	F1	HEQQ	HEQD	Total
BiDAF++	51.8	45.3	2.0	99.1
BiDAF++ w/ 2-C	60.6	55.7	5.3	121,6
BERT+HAE	63.9	59.7	5.9	129.5
FlowQA	64.6	59.6	5.8	130.0
BERT+PosHAE	64.7	60.7	6.0	131.4
HAM	65.7	62.1	**7.3**	135.1
Our method	**66.1**	**62.2**	**7.3**	**135.6**

Table 6: Comparison between our method and the state-of-art ConvQA methods.

Related Work

Our task is related to machine reading comprehension, conversations, and reinforcement learning.

Machine Reading Comprehension (MRC) and Conversational Question Answering. MRC is at the central part of natural language understanding. Many high-quality challenges and datasets (Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018; Nguyen et al. 2016; Joshi et al. 2017; Kwiatkowski et al. 2019) have greatly boosted the research progress in this field, resulting in a wide range of model architectures (Seo et al. 2016; Hu et al. 2018; Wang et al. 2017; Huang et al. 2017; Clark and Gardner 2018). The MRC task is typically conducted in a single-turn QA manner. The goal is to answer the question by predicting an answer span in the given passage. The ConvQA task formulated in CoQA (Reddy, Chen, and Manning 2018b) and QuAC (Choi et al. 2018a) is closely related to the MRC task. A major difference is that the questions in ConvQA are organized in conversations. Thus we need to incorporate the conversation history to better understand the current question. Most methods seek to incorporate modeling the dialogue history into the process of the passage representation. FlowQA (Huang, Choi, and Yih 2018) adopts RNN to convert the passage representation from the past. FlowDelta (Yeh and Chen 2019) seeks to employ delta operation to model the change in relative turns. GraphFlow (Chen, Wu, and Zaki 2019) views each word in the passage as a node, uses the attention score as their connections, and then adopts a gating mechanism to fuse the representation of the past and the current. MC^2 (Zhang 2019) proposes to use CNN in multiple perspectives to capture the semantic changes based on FlowQA. On the other hand, methods that adopt history answer embedding is also competitive. HAE (Qu et al. 2019a) employs

Turns	Question	Answer	Action
0	Did they have any clues?	probably FSB) are known to have targeted the webmail account of the murdered Russian journalist Anna Politkovskaya	1
1	How did they target her email?	On 5 December 2005, RFIS initiated an attack against the account annapolitovskaya@US Provider1, by deploying malicious software	1
2	Did they get into trouble for that?	CANNOTANSWER	1
3	Did they have any murder suspects?	After the three Makhmudov brothers, Khadjikurbanov and Lom-Ali Gaitukayev were convicted in 2014')	1
4	Did they go to jail?	CANNOTANSWER	1
5	Is there anything else interesting in the article?		1
0	Did they have any clues?	probably FSB) are known to have targeted the webmail account of the murdered Russian journalist Anna Politkovskaya	0
1	How did they target her email?	On 5 December 2005, RFIS initiated an attack against the account annapolitovskaya@US Provider1, by deploying malicious software	0
2	Did they get into trouble for that?	CANNOTANSWER	0
3	Did they have any murder suspects?	After the three Makhmudov brothers, Khadjikurbanov and Lom-Ali Gaitukayev were convicted in 2014')	0
4	Did they go to jail?	CANNOTANSWER	0
5	Is there anything else interesting in the article?	In accordance with Russian law, there is a 15-year statute of limitation for the p̈articularly gravec̈rime of first degree murder.	1
6	"Are they close to solving it?		1

Table 7: Two sample examples from development dataset in QuAC. Although with similar conversation history (the same history turns from 0 to 4), the RL agent acts differently for each dialogue turn.

answer embedding to indicate the position the history answers. HAM (Qu et al. 2019b) further adopts an attention mechanism to select related history questions.

Reinforcement Learning (RL). RL seeks to train agents with implicit feedback, and it comprises a series of goal-oriented algorithms that have been studied for many decades in many disciplines (Sutton and Barto 1998; Arulkumaran et al. 2017; Li 2017). The recent development in deep learning has greatly contributed to this area and has delivered amazing achievements in many domains, such as playing games against humans (Mnih et al. 2013; Silver et al. 2017).

There are generally two lines of work in RL: value-based methods and policy-based methods. Value-based methods, including SARSA (Rummery and Niranjan 1994) and the Deep Q Network (Mnih et al. 2015), take actions based on estimations of expected long-term return. On the other hand, policy-based methods optimize for a strategy that can map states to actions that promise the highest reward. Finally, hybrid methods, such as the actor-critic algorithm (Konda and Tsitsiklis 2003), integrate a trained value estimator into policy-based methods to reduce variance in rewards and gradients. In this work, we mainly experiment with hybrid methods.

The nature of RL problems is making a sequence of actions based on certain observations in order to achieve a long-term goal. This nature has made RL suitable to deal with data selection problems in many areas (Fang, Li, and Cohn 2017; Wu, Li, and Wang 2018; Fan et al. 2017; Patel, Chitta, and Jasani 2018; Wang et al. 2018; Feng et al. 2018). The study in (Takanobu et al. 2018) adopts reinforcement learning in the topic segmentation task. The agent is responsible for assigning a label for the segment in the dialogue. The study in (Buck et al. 2018) adopts reinforcement learning to generate questions of better quality. It freezes the QA model and regards the seq2seq model as the agent.

Our proposed method seeks to identify helpful conversation history to construct better training data. To the best of our knowledge, our work is the first to study reinforcement learning to backtrack history turns in the ConvQA setting. Our proposed method is an end-to-end trainable approach that shows better results than the competitive baselines.

Conclusion

In this study, we proposed a reinforcement learning method to automatically select related history turns for multi-turn machine reading comprehension. Compared with the recent history modeling approaches, our method can select helpful history turns to boost the performance of the MRC model. For a question in the dialogue, the actions made by the learned policy are shown to be helpful to select the related history turns and achieves better performance than the competing methods. Extensive experiments on public datasets show our method yields consistently better performance than the competing methods. Quantitative analysis also shows the selection behaviors made by the learned policy are insightful.

As for future work, we seek to examine the generalization capability of our RL method for more context-aware applications in dialogue modeling. We also seek to leverage the recent advance of RL to further boost model efficiency.

Acknowledgments

This work was partially sponsored by the NSFC projects (No. 61402403, No. 62072399), Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Chinese Knowledge Center for Engineering Sciences and Technology, MoE Engineering Research Center of Digital Library, and the Fundamental Research Funds for the Central Universities. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Arulkumaran, K.; Deisenroth, M. P.; Brundage, M.; and Bharath, A. A. 2017. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine* .

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 41–48. New York, NY, USA. doi:10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Buck, C.; Bulian, J.; Ciaramita, M.; Gajewski, W.; Gesmundo, A.; Houlsby, N.; and Wang, W. 2018. Ask the Right Questions: Active Question Reformulation with Reinforcement Learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. URL https://openreview.net/forum?id=S1CChZ-CZ.

Chen, Y.; Wu, L.; and Zaki, M. J. 2019. GraphFlow: Exploiting Conversation Flow with Graph Neural Networks for Conversational Machine Comprehension. *CoRR* abs/1908.00059. URL http://arxiv.org/abs/1908.00059.

Choi, E.; He, H.; Iyyer, M.; Yatskar, M.; tau Yih, W.; Choi, Y.; Liang, P.; and Zettlemoyer, L. S. 2018a. QuAC: Question Answering in Context. In *EMNLP*.

Choi, E.; He, H.; Iyyer, M.; Yatskar, M.; Yih, W.-t.; Choi, Y.; Liang, P.; and Zettlemoyer, L. 2018b. QuAC: Question Answering in Context. In *EMNLP*, 2174–2184.

Clark, C.; and Gardner, M. 2018. Simple and Effective Multi-Paragraph Reading Comprehension. In *ACL*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805. URL http://arxiv.org/abs/1810.04805.

Fan, Y.; Tian, F.; Qin, T.; Bian, J.; and Liu, T. 2017. Learning What Data to Learn. *Arxiv Preprint, CoRR* .

Fang, M.; Li, Y.; and Cohn, T. 2017. Learning how to Active Learn: A Deep Reinforcement Learning Approach. In *EMNLP*.

Feng, J.; Huang, M.; Zhao, L.; Yang, Y.; and Zhu, X. 2018. Reinforcement Learning for Relation Classification From Noisy Data. In *AAAI*.

Hu, M.; Peng, Y.; Huang, Z.; Qiu, X.; Wei, F.; and Zhou, M. 2018. Reinforced Mnemonic Reader for Machine Reading Comprehension. In *IJCAI*.

Huang, H.-Y.; Choi, E.; and tau Yih, W. 2018. FlowQA: Grasping Flow in History for Conversational Machine Comprehension. *CoRR* .

Huang, H.-Y.; Choi, E.; and Yih, W.-t. 2018. FlowQA: Grasping Flow in History for Conversational Machine Comprehension. *CoRR* abs/1810.06683. URL http://arxiv.org/abs/1810.06683.

Huang, H.-Y.; Zhu, C.; Shen, Y.; and Chen, W. 2017. FusionNet: Fusing via Fully-Aware Attention with Application to Machine Comprehension. *CoRR* abs/1711.07341.

Joshi, M. S.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. S. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *ACL*.

Ju, Y.; Zhao, F.; Chen, S.; Zheng, B.; Yang, X.; and Liu, Y. 2019. Technical report on Conversational Question Answering. *CoRR* abs/1909.10772. URL http://arxiv.org/abs/1909.10772.

Konda, V. R.; and Tsitsiklis, J. N. 2003. On Actor-Critic Algorithms. *SIAM J. Control Optim.* .

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Kelcey, M.; Devlin, J.; Lee, K.; Toutanova, K. N.; Jones, L.; Chang, M.-W.; Dai, A.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics* .

Li, Y. 2017. Deep Reinforcement Learning: An Overview. *CoRR* .

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. A. 2013. Playing Atari with Deep Reinforcement Learning. *CoRR* .

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540): 529–533. ISSN 00280836.

Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *CoRR* abs/1611.09268.

Patel, Y.; Chitta, K.; and Jasani, B. 2018. Learning Sampling Policies for Domain Adaptation. *CoRR* .

Qu, C.; Yang, L.; Qiu, M.; Croft, W. B.; Zhang, Y.; and Iyyer, M. 2019a. BERT with History Answer Embedding for Conversational Question Answering. In *SIGIR*, 1133–1136. doi:10.1145/3331184.3331341. URL https://doi.org/10.1145/3331184.3331341.

Qu, C.; Yang, L.; Qiu, M.; Zhang, Y.; Chen, C.; Croft, W. B.; and Iyyer, M. 2019b. Attentive History Selection for Conversational Question Answering. In *CIKM*, 1391–1400. doi:10.1145/3357384.3357905. URL https://doi.org/10.1145/3357384.3357905.

Qu, C.; Yang, L.; Qiu, M.; Zhang, Y.; Chen, C.; Croft, W. B.; and Iyyer, M. 2019c. Attentive History Selection for Conversational Question Answering. *CoRR* abs/1908.09456. URL http://arxiv.org/abs/1908.09456.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *ACL*.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *EMNLP*.

Reddy, S.; Chen, D.; and Manning, C. D. 2018a. CoQA: A Conversational Question Answering Challenge. *CoRR* abs/1808.07042. URL http://arxiv.org/abs/1808.07042.

Reddy, S.; Chen, D.; and Manning, C. D. 2018b. CoQA: A Conversational Question Answering Challenge. *CoRR* abs/1808.07042.

Rummery, G. A.; and Niranjan, M. 1994. On-Line Q-Learning Using Connectionist Systems. Technical report, University of Cambridge.

Seo, M. J.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional Attention Flow for Machine Comprehension. *CoRR* abs/1611.01603.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; Chen, Y.; Lillicrap, T.; Hui, F.; Sifre, L.; van den Driessche, G.; Graepel, T.; and Hassabis, D. 2017. Mastering the game of Go without human knowledge. *Nature* 550(7676): 354–359. ISSN 1476-4687. doi:10.1038/nature24270.

Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement Learning - An Introduction*. Adaptive Computation and Machine Learning. MIT Press.

Takanobu, R.; Huang, M.; Zhao, Z.; Li, F.-L.; Chen, H.; Zhu, X.; and Nie, L. 2018. A Weakly Supervised Method for Topic Segmentation and Labeling in Goal-oriented Dialogues via Reinforcement Learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 4403–4410. URL https://doi.org/10.24963/ijcai.2018/612.

Wang, S.; Yu, M.; Guo, X.; Wang, Z.; Klinger, T.; Zhang, W.; Chang, S.; Tesauro, G.; Zhou, B.; and Jiang, J. 2018. R^3: Reinforced Ranker-Reader for Open-Domain Question Answering. In *AAAI*.

Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated Self-Matching Networks for Reading Comprehension and Question Answering. In *ACL*.

Williams, R. J. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning* 8: 229–256.

Wu, J.; Li, L.; and Wang, W. Y. 2018. Reinforced Co-Training. In *NAACL*.

Yatskar, M. 2019. A Qualitative Comparison of CoQA, SQuAD 2.0 and QuAC. In *NAACL-HLT*, 2318–2323.

Yeh, Y. T.; and Chen, Y.-N. 2019. FlowDelta: Modeling Flow Information Gain in Reasoning for Conversational Machine Comprehension. *CoRR* abs/1908.05117. URL http://arxiv.org/abs/1908.05117.

Zhang, X. 2019. MCˆ2: Multi-perspective Convolutional Cube for Conversational Machine Reading Comprehension. In *ACL*, 6185–6190.

Zhu, C.; Zeng, M.; and Huang, X. 2018. SDNet: Contextualized Attention-based Deep Network for Conversational Question Answering. *CoRR* .