ET5: A Novel End-to-end Framework for Conversational Machine Reading Comprehension

Xiao Zhang[123], **Heyan Huang**[123]*, **Zewen Chi**[123], **Xian-Ling Mao**[123]
[1]School of Computer Science and Technology, Beijing Institute of Technology
[2]Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications
[3]Southeast Academy of Information Technology, Beijing Institute of Technology
{yotta,hhy63,czw,maoxl}@bit.edu.cn

Abstract

Conversational machine reading comprehension (CMRC) aims to assist computers to understand an natural language text and thereafter engage in a multi-turn conversation to answer questions related to the text. Existing methods typically require three steps: (1) decision making based on entailment reasoning; (2) span extraction if required by the above decision; (3) question rephrasing based on the extracted span. However, for nearly all these methods, the span extraction and question rephrasing steps cannot fully exploit the fine-grained entailment reasoning information in decision making step because of their relative independence, which will further enlarge the information gap between decision making and question phrasing. Thus, to tackle this problem, we propose a novel end-to-end framework for conversational machine reading comprehension based on shared parameter mechanism, called entailment reasoning T5 (ET5). Despite the lightweight of our proposed framework, experimental results show that the proposed ET5 achieves new state-of-the-art results on the ShARC leaderboard with the BLEU-4 score of 55.2. Our model and code are publicly available[1].

Rule Document: Taking more leave than the entitlement. If a worker has taken more leave than they're entitled to, their employer must not take money from their final pay unless it's been agreed beforehand in writing. The rules in this situation should be outlined in the employment contract, company handbook or intranet site.

User Question: Can my employer take money from my final pay?
User Scenario: I just looked at my paperwork and I think my boss took out too much of my money.
Follow-up Q: Did you take more leave than they're entitled to?
Follow-up A: Yes

Decision Making: Yes | No | **Inquire** | Irrelevant
Final Answer: Did you agree to it beforehand in writing?

Figure 1: An example in the CMRC dataset. Machine should first make the decision of `Yes/No/Inquire/Irrelevant`, and then generate the follow-up question if the decision is `Inquire`. The colored sentences show the reasoning process for the final answer.

1 Introduction

Conversational machine reading comprehension (CMRC) (Saeidi et al., 2018) aims to assist machines to understand an natural language text and thereafter engage in a multi-turn conversation to answer questions related to the text. Specifically, the machine needs to reason for decision making and question generation by interacting through rule document, user question, user scenario, and dialogue history. As an example shown in Figure 1, after fully interacting with complicated context information, the machine makes a decision of `Yes/No/Inquire/Irrelevant`, and then generates a question under the `Inquire` decision.

Existing researches (Saeidi et al., 2018; Verma et al., 2020; Lawrence et al., 2019; Zhong and Zettlemoyer, 2019; Gao et al., 2020a,b; Ouyang et al., 2021) mainly aim to capture the interactions among the complicated inputs, and achieve promising results by conducting various fine-

*Corresponding author.
[1]https://github.com/Yottaxx/ET5

Proceedings of the 29th International Conference on Computational Linguistics, pages 570–579
October 12–17, 2022.

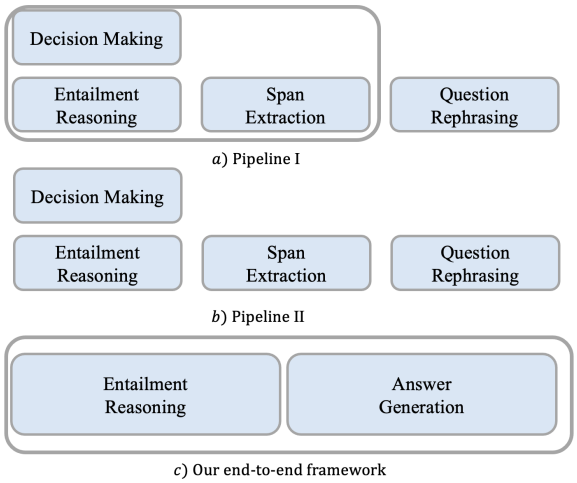

Figure 2: The overview of frameworks in CMRC. (a) For Pipeline I, decision making and span extraction models share the encoder but suffer from the problem of noisy span extraction. (b) For Pipeline II, the three stages are handled completely separately, and there is no information sharing among the three stages. (c) Our framework is an end-to-end framework with a shared encoder and a duplex decoder. The duplex decoder contains an entailment reasoning decoder and answer generation decoder, both the information of entailment reasoning and answer generation are shared through the common encoder. Both decisions and follow-up questions will be generated via answer generation decoder directly.

grained entailment reasoning interaction strategies based on Pre-trained Language Models (PrLMs) (Devlin et al., 2019; Liu et al., 2020; Dong et al., 2019; Clark et al., 2020; Raffel et al., 2020). These methods (Zhong and Zettlemoyer, 2019; Gao et al., 2020a,b; Ouyang et al., 2021) typically adopt pipeline architectures, which are shown in Figure 2. These pipeline architectures typically require three steps : (1) decision making based on entailment reasoning; (2) span extraction if required by the above decision; (3) question rephrasing based on the extracted span. There are currently two types of pipeline structures: Pipeline I and Pipeline II. The Pipeline I make decisions and extract spans simultaneously, while the Pipeline II handles all three stages separately.

However, for nearly all these methods (Zhong and Zettlemoyer, 2019; Gao et al., 2020a,b; Ouyang et al., 2021), the span extraction and question rephrasing steps can't fully exploit the fine-grained entailment reasoning information in decision making step. For Pipeline II, these methods (Gao et al., 2020b; Ouyang et al., 2021) do not share entailment reasoning information among decision-making, span extraction, and question phrasing at all. For Pipeline I, these methods (Zhong and Zettlemoyer, 2019; Gao et al., 2020a) only approximate share the information through noisy span extraction. Both of them enlarge the information gap between decision making and question rephrasing, and seriously affect the performance of question generation.

To tackle this problem, we propose a novel end-to-end framework for conversational machine reading comprehension based on shared parameter mechanism, called entailment reasoning T5 (ET5). Specifically, the proposed framework consists of a text-to-text Transformer and an additional entailment reasoning decoder. The original decoder in the text-to-text Transformer will directly generate either decision or follow-up question based on the shared encoder enhanced by entailment reasoning. The entailment reasoning decoder can be configured with different entailment reasoning strategies. Despite the lightweight of our proposed framework, experimental results show that ET5 achieves new state-of-the-art results on the ShARC leaderboard with the BLEU-4 score of 55.2 and significantly improves the generalization performance of question generation.

Our contributions are summarized as follows:

- We propose a novel end-to-end framework, called ET5, to better capture the entailment information for question generation, and thus eliminate the information gap between decision making and question generation.
- Extensive experiments demonstrate the effectiveness of the proposed framework on ShARC benchmark, especially in the question generation sub-task.

2 Related Work

Conversation-based reading comprehension (Saeidi et al., 2018; Sun et al., 2019; Reddy et al., 2019; Choi et al., 2018; Cui et al., 2020; Gao et al., 2021) extends the context with dialogue history, which is formed to simulate the communication scene in real life. Most of them are ideal subtasks, either span-based QA tasks (Choi et al., 2018; Reddy et al., 2019) or multi-choice tasks (Sun et al., 2019; Cui et al., 2020). We focus on the task (Saeidi et al., 2018) that deal with real-world complexities, where the machine needs to make decisions or ask questions to keep the conversa-

tion going. This task (Saeidi et al., 2018) is called Conversational Machine Reading Comprehension (CMRC), which requires the machine to have the inference ability to capture the interactions among rule document, user question, user scenario, and dialogue history.

Recent studies (Zhong and Zettlemoyer, 2019; Gao et al., 2020a,b; Ouyang et al., 2021) in CMRC are generally utilized to match the relationship between the various information. E^3 (Zhong and Zettlemoyer, 2019) first investigates the importance of clarifying the different rule units for entailment reasoning. Different entailment reasoning strategies (Gao et al., 2020a,b, 2021) with fine-grained reasoning units are further proposed to improve the abilities of entailment reasoning. In addition, discourse relationships between fine-grained reasoning units are utilized to model the discourse graph (Ouyang et al., 2021; Zhang et al., 2021). These methods typically adopt pipeline architectures, DISCERN (Gao et al., 2020b) first discovers the unbalance and noisy problems of Pipeline I conducted by E^3 (Zhong and Zettlemoyer, 2019) and EMT (Gao et al., 2020a), then solves them by utilizing Pipeline II to process the three stages separately. However, due to the pipeline's inability to make full use of the entailment information, both of the above pipeline structures have the problem of information gap (Zhang et al., 2021) between decision making and question generation.

To better capture the entailment information for question generation and eliminate the information gap, we propose a novel end-to-end framework for conversational machine reading comprehension based on shared parameter mechanism, called ET5, which will be introduced in the next section.

3 Method

3.1 Settings of ET5

Each example of CMRC is formed as the tuple $\{C, R, A, S\}$. C donates the context, which is a concentrated sentence of rule document, user scenario, user question, and dialogue history. Especially, $C = \{e_1, e_2, ..., e_k, s, q, d_1, d_2, ..., d_n\}$, where e donates the elementary discourse unit (EDU) segmented from by rule documents. s and q are user scenario and user question, d represents the dialogues. Each item of C is prefixed with a special token to represents the following sentence, the details of the prefix are written in Section 3.2. R represents the discourse relations among EDUs, the parsed details are reported in Section 4.1. A is the final answer, including the decision or follow-up question. S donates the entailment reasoning state of each EDU in `ENTAILMENT`, `CONTRADICTION`, or `NEUTRAL`. To get the noisy supervision signals of entailment states, we adopt a heuristic approach[2] following the previous study (Gao et al., 2020a). Given inputs C, R, ET5 needs reasoning entailment states S and final answer A including the decision and follow-up question. As illustrated in Figure 3, we conduct duplex decoder to process answer generation and entailment reasoning simultaneously in a multi-task training approach with the shared encoder. The training procedure and evaluating procedure are illustrated in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 Training procedure of ET5

Input: Concentrated context C, discourse relations R, learning rate τ, discourse relations R

Output: Final answer A, entailment reasoning state S, ET5 encoder parameters $\boldsymbol{\theta}_e$, ET5 answer generation decoder parameters $\boldsymbol{\theta}_a$, ET5 entailment reasoning decoder parameters $\boldsymbol{\theta}_d$

Initialize $\boldsymbol{\theta}_e$,$\boldsymbol{\theta}_a$,$\boldsymbol{\theta}_d$
while not converged **do**
  for $i = 1, 2, \ldots, N$ **do**
    $\boldsymbol{e}_i = f(c_i, \boldsymbol{\theta}_e)$ s.t. $\forall c \in \mathcal{C}$
    $\boldsymbol{s}_i = f(e_i, r_i, \boldsymbol{\theta}_d)$ s.t. $\forall r \in \mathcal{R}$
    $\boldsymbol{a}_i = f(e_i, \boldsymbol{\theta}_a)$
  end for
  $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}$
  $\boldsymbol{\theta}_e \leftarrow \boldsymbol{\theta}_e - \tau \boldsymbol{g}$
  $\boldsymbol{\theta}_d \leftarrow \boldsymbol{\theta}_d - \tau \boldsymbol{g}$
  $\boldsymbol{\theta}_a \leftarrow \boldsymbol{\theta}_a - \tau \boldsymbol{g}$
end while

3.2 Encoding

Fine-grained Prefix Prompt We investigate and propose a fine-grained prefix strategy, to prompt the interactions among different components of the input. As shown in Figure 3, the concatenate input is prefixed with a text-form task prefix. Furthermore, given relationship tagged EDUs, user question, user scenario, dialogue history as inputs, each of them is prefixed with a fine-grained

[2]The noisy supervision signal is a heuristic label obtained by the minimum edit distance.

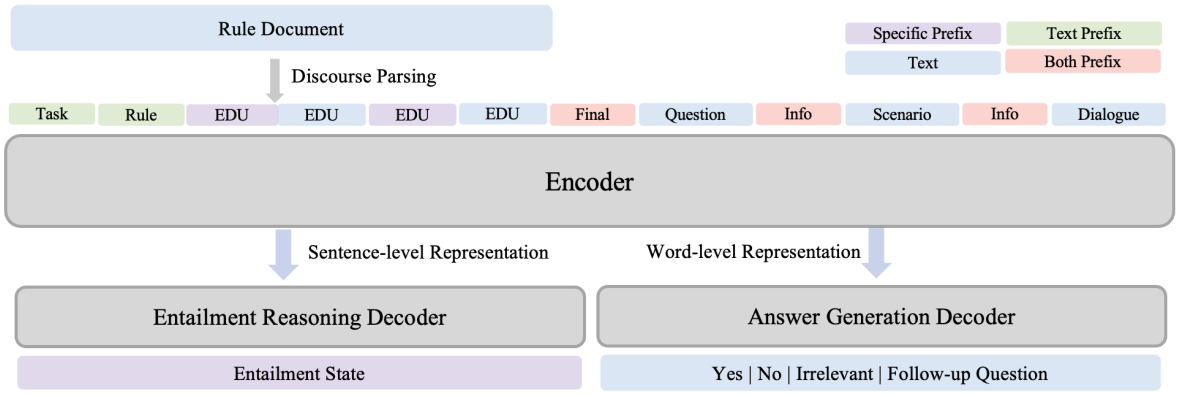

Figure 3: The architecture of ET5. Our proposed framework is an end-to-end framework based on a single text-to-text Transformer. The decoder of our proposed framework is a duplex decoder, including entailment reasoning decoder and answer generation decoder. The answer generation decoder will generate the final answer directly, either of the decision or the follow-up question. The entailment reasoning decoder is utilized to reason the fine-grained entailment states, which is only activated in the training stage. Red boxes indicate the fine-grained prefixes, including special prefixes and text prefixes, which are represented by purple boxes and green boxes, respectively. Special prefixes refer to special tokens that aim to get the sentence-level representations. Text prefixes refer to the component-specific text prefixes that aim to differentiate among different input types.

Algorithm 2 Evaluating procedure of ET5

Input: Concentrated context C, ET5 encoder parameters $\boldsymbol{\theta}_e$, ET5 answer generation decoder parameters $\boldsymbol{\theta}_a$

Output: Final answer A

1: Initialize $\boldsymbol{\theta}_e$,$\boldsymbol{\theta}_a$
2: **for** $i = 1, 2, \ldots, N$ **do**
3: $\quad \boldsymbol{e}_i = f(c_i, \boldsymbol{\theta}_e)$ s.t. $\forall c \in \mathcal{C}$
4: $\quad \boldsymbol{a}_i = f(e_i, \boldsymbol{\theta}_a)$
5: **end while**

prefix.The fine-grained prefix consists of a text prefix and a special prefix. The text prefix is used to differentiate between different types of input as an addition information prefix. The special prefix is used to obtain sentence-level representations required for entailment reasoning. In the case of user scenario and each dialogue history usually play a similar role as an information provider in CMRC tasks, user scenario and each dialogue history share the same text prefix. Meanwhile, each EDU has a special token [EDU]. Both user question and final answer are prefixed with the same text FINAL. We concatenate the fine-grained prefixed EDUs, user question, user scenario, dialogue history to get the encoding representations.

Fine-grained Prefix Encoding We concatenate the fine-grained prefixed EDUs, user question, user scenario, dialogue history as the input. Encoder representation H_e is encoded with the input by conducting T5 encoder (Raffel et al., 2020) as the encoder. Let $H_s = [h_{e_1}, h_{e_2}, ..., h_{e_k}, h_{f_i}, h_{s_i}, h_{d_1}, ..., h_{d_n}]$, H_s to donate the sentence-level representations. h_e, h_f, h_s, h_d represent the fine-grained special prefix token representation of EDU, user question, user scenario, and dialogue history, respectively.

3.3 Decoding

Our decoding is duplex decoding, including entailment reasoning and answer generation. Especially, both answer generation and entailment reasoning are activated in the training stage. During the inference stage, the answer generation decoder will directly generate either the decision or the follow-up question while the entailment reasoning will be dropped. We conduct entailment reasoning decoder with various entailment reasoning strategies in the experiments, including inter attention reasoning (Gao et al., 2020b) and dialogue graph modeling (Ouyang et al., 2021). We mainly introduce dialogue graph reasoning here, because dialogue graph modeling only has one more graph reasoning block than inter attention reasoning, the other structures are the same.

Entailment Reasoning We utilize dialogue graph modeling for entailment reasoning decoding. Dialogue graph consists of the explicit discourse graph, the implicit discourse graph, and the

inter attention reasoning. The details are shown in the following.

Given H_s and R, we construct the explicit discourse graph G to explicitly model the complex logical structures between the various information in CMRC by introducing discourse relationships among the rule conditions. Following previous (Ouyang et al., 2021), the graph is formed as a Levi graph (Levi, 1942).

There are three types of vertices in the graph: EDUs, discourse relationships, and user scenarios. Each EDU duplex connects with the tagged relationship. The user scenario connects all the other vertices as a global vertex. All the types R_L of the possible edges between vertices are six, each of them is named as *default-in*, *default-out*, *reverse-in*,*reverse-out*, *self*, and *global*. The EDUs vertices and user scenario vertex are initialized with the contextualized representation in H_s. And the discourse relationships vertices are initialized with a conventional embedding layer. Then the representation h_p of each node v_p is initialized. To handle the multi-relation graphs and dynamically weight the different relations, we use a relational graph convolution network (Schlichtkrull et al., 2018) with a gating mechanism. the graph-based information processing can be written as:

$$g_p^{(l)} = \text{Sigmoid}(h_p^{(l)} w_{r,g}^{l}), \tag{1}$$

$$h_p^{(l+1)} = \text{ReLU}(\sum_{r \in R_L} \sum_{v_p \in \mathcal{N}_r(v_p)} g_p^{(l)} \frac{1}{c_{p,r}} w_r^{(l)} h_q^{(l)}), \tag{2}$$

where $w_r^{(l)}$ is the trainable parameters of layer l. $w_{r,g}^{(l)}$ is trainable parameters under relation type r of layer l. $c_{p,r}$ is the number of the neighbors of node v_p with relationship r. $\mathcal{N}_r(v_p)$ refers to those neighbors. Let $H_p = [h_{p_1}^{(l+1)}), h_{p_2}^{(l+1)}, ..., h_{f_i}, h_{s_i}, h_{d_1}, ..., h_{d_n}]$, l is the last layer, H_p donate the explicit discourse graph representation.

Given the EDUs tokens hidden representation E from H_e. We decouple and fuse the local information and the contextualized information by conducting the implicit discourse graph. Considering each token i of EDU as a vertex in the graph, the adjacent matrices can express the implicit discourse graph. We use I_i donate the index of token i in EDU, the information decoupling adjacent matrices M can be written as:

$$M_l[i,j] = \begin{cases} 0, & I_i = I_j \\ -\infty, & otherwise \end{cases} \tag{3}$$

$$M_c[i,j] = \begin{cases} 0, & I_i \neq I_j \\ -\infty, & otherwise \end{cases}, \tag{4}$$

where M_l and M_c are conducted to express the local and contextualized information. We use multi-head-self-attention (MHSA) (Vaswani et al., 2017) to process decoupling:

$$G_i = \text{MHSA}(E, M_i), \quad i \in \{l, c\}, \tag{5}$$

after exploring the potential textual relations in the rule document, we apply a fusion layer to fuse the information by considering the encoder encoding and the attention hidden states of EDUs:

$$\tilde{E}_1 = \text{ReLU}(f([E, G_l, E - G_l, E \odot G_l])), \tag{6}$$

$$\tilde{E}_2 = \text{ReLU}(f([E, G_c, E - G_l, E \odot G_c])), \tag{7}$$

$$g = \text{Sigmoid}(f([\tilde{E}_1, \tilde{E}_2])), \tag{8}$$

$$C = g \odot G_l + (1 - g) \odot G_c, \tag{9}$$

where f is the fully-connected layer. Let $H_i = [h_{c_1}, h_{c_2}, ..., h_{f_i}, h_{s_i}, h_{d_1}, ..., h_{d_n}]$, H_i donate the explicit discourse graph representation. h_{c_i} is updated by the representation of [EDU] in C.

Given the sentence-level representation H_e, H_p, H_i, inter attention reasoning aims to fully interact with various information, including EDUs, user question, user scenario, dialogue history. We utilize an inter-sentence Transformer (Vaswani et al., 2017) to reason the entailment states. Let $\tilde{H}_e$, $\tilde{H}_p$, $\tilde{H}_i$ donate the inter-sentence Transformer encoding representation, $\tilde{H}_s$ donate the average encoding, namely, $\tilde{H}_s = [\tilde{h}_{e_1}, \tilde{h}_{e_2}, ..., \tilde{h}_{e_k}, \tilde{h}_{f_i}, \tilde{h}_{s_i}, \tilde{h}_{d_1}, ..., \tilde{h}_{d_n}]$. All the vectored representations are in the same dimension. Following previous studies (Gao et al., 2021), we utilize a linear transformation to track the entailment reasoning state of each EDU:

$$c_i = W_c \tilde{h}_{e_i} + b_c \in \mathcal{R}^3, \tag{10}$$

where the W_c is trainable parameters, c_i is the predicted score for the three labels of the i-th states.

Answer Generation Answer generation is utilized to generate either the decision or the follow-up question. We employ T5 decoder as our answer generation decoder. Given encoder hidden

representation H_e, and the set of final answer $(a_1, a_2, ..., a_n)$, including decision or follow-up question, each of the answers is composed of the variable-length tokens $(x_1, x_2, ..., x_m)$, the probabilities over the tokens are shown in the blow:

$$p(a) = \prod_{1}^{m} p(x_i|x_{<i}, H_e; \theta), \quad (11)$$

where θ donates the trainable parameters of our decoder.

3.4 Training Objective

Entailment Reasoning Given the entailment fulfillment states c_i, the entailment reasoning is supervised by cross-entropy loss:

$$\mathcal{L}_{enatil} = -\frac{1}{N}\sum_{i=1}^{N} log\,\mathrm{softmax}(c_i)_r, \quad (12)$$

where r is the ground truth of entailment state.

Answer Generation Given the encoder representation H_e, the answer generation training objective is computed by:

$$\mathcal{L}_{answer} = -\sum_{i=1}^{M} log\, p(x_i|x_{<i}, H_e; \theta), \quad (13)$$

The overall loss function is:

$$\mathcal{L} = \mathcal{L}_{answer} + \lambda\mathcal{L}_{entail}. \quad (14)$$

4 Experiment and Analysis

4.1 Data

Dataset The experimental dataset is ShARC, the current CMRC benchmark, which is built up from 948 rule text. The corpus is clawed from the government website. The utterances size of ShARC is 32,436, each of the utterances related to a dialog tree, the utterances with the same rule text refer to the same dialog tree. Each dialog tree contains all possible fulfillment combinations of conditions. The train, dev, test size is 21,890, 2,270, 8,276, respectively. Each item consists of utterance id, tree id, rule document, initial question, user scenario, dialog history, evidence, and the decision. Evidence is only used to support the answer, and can't be treated as input.

Preprocess Following previous methods (Ouyang et al., 2021; Gao et al., 2020b), we first split rule documents into elementary discourse units (EDUs), and then tag the discourse relationship among EDUs. For discourse segmentation, the rule documents are split into EDUs by using a pre-trained discourse parser (Li et al., 2018). For discourse relation extraction, we utilize a pre-trained discourse relation parser[3] to tag the structural relations among EDUs.

4.2 Setup

Evaluation Evaluation in ShARC is divided into two parts. First is decision classification: Micro-Acc and Macro-Acc scores are used for the evaluation in classification. Then question generation part is evaluated with BLEU (Papineni et al., 2002) score only if the prediction and ground truth in classification are both inquired.

Implementation Details We implement ET5 by configuring entailment reasoning decoder with two different methods: inter attention reasoning (Gao et al., 2020b) and dialogue graph reasoning (Ouyang et al., 2021), named ET5-Discern and ET5 respectively. The parameters of entailment reasoning decoder are randomly initialized, the remain parameters are initialized with official T5 (Raffel et al., 2020). ET5 and ET5-Discern are fine-tuned with AdamW (Loshchilov and Hutter, 2018) in 16 epochs, and the batch sizes are 32 and 16 respectively. We use hierarchical learning rates, the learning rate of T5 is 2e-4, the learning rate of other parameters are 2e-5. We've tried 1.0, 1.5, 2.0, 3.0 for λ, and find 1.0 is the best base on the results in the dev set. During inference decoding, the beam search number is set to 5. All results are conducted in two 3090 GPU (24GB memory)

4.3 Results

All results in the blind held-out test set of the ShARC benchmark are illustrated in Table 1. There are two different implementations here. ET5-Discern is configured with a DISCERN-formed entailment reasoning decoder by using the base-size model as the backbone. ET5 is configured with a DGM-formed entailment reasoning decoder by using the large-size model as the backbone.

[3] https://github.com/shizhouxing/DialogueDiscourseParsing

Models	Micro	Macro	BLEU-1	BLEU-4
Seq2Seq (Saeidi et al., 2018)	44.8	42.8	34.0	7.8
Pipeline (Saeidi et al., 2018)	61.9	68.9	54.4	34.4
BERTQA (Zhong and Zettlemoyer, 2019)	63.6	70.8	46.2	36.3
UrcaNet (Verma et al., 2020)	65.1	71.2	60.5	46.1
BiSon (Lawrence et al., 2019)	66.9	71.6	58.8	44.3
E^3 (Zhong and Zettlemoyer, 2019)	67.6	73.3	54.1	38.7
EMT (Gao et al., 2020a)	69.1	74.6	63.9	49.5
DISCERN (Gao et al., 2020b)	73.2	78.3	64.0	49.1
DGM (Ouyang et al., 2021)	**77.4**	**81.2**	63.3	48.4
ET5-Discern (ours)	**74.4**	**78.7**	**66.4**	**51.6**
ET5 (ours)	76.3	80.5	**69.6**	**55.2**

Table 1: Performance on the blind held-out test set of ShARC benchmark.

Models	Micro	Macro	BLEU-1	BLEU-4	Params
Discern	74.9	**79.8**	65.7	52.4	330M
ET5-Discern	**75.4**	79.7	65.2	51.1	220M
DGM	**78.6**	82.2	**71.8**	**60.2**	1020M
ET5	**78.6**	**82.5**	65.3	53.3	770M

Table 2: Performance on the dev set of the ShARC benchmark. Params are the parameter numbers of PrLMs used in the framework.

Models	Dev Set		Test Set	
	BLEU-1	BLEU-4	BLEU-1	BLEU-4
E^3	67.1	53.7	54.1(-13.0)	38.7(-15.0)
EMT	67.5	53.2	63.9(-3.6)	49.5(-3.7)
DISCERN	65.7	52.4	64.0(-1.7)	49.1(-3.3)
DGM	71.8	60.2	63.3(-8.5)	48.4(-11.8)
ET5-Discern	65.2	51.1	**66.4(+1.2)**	**51.6(+0.5)**
ET5	65.3	53.3	**69.6(+4.3)**	**55.2(+1.9)**

Table 3: Performance of BLEU scores on the dev set and test set of the ShARC benchmark.

Experimental results demonstrate that the proposed framework achieves new SOTA with considerable improvement in terms of BLEU scores. ET5-Discern outperforms DISCERN by 2.4 in BLEU-1, 2.5 in BLEU-4, 1.2 in micro-averaged accuracy, and 0.4 in macro-averaged accuracy. ET5 outperforms DGM by 6.3 in BLEU-1, 6.8 in BLEU-4. We further analyze the results in the dev set shown in Table 2. Compared to the existing pipeline framework, our framework reduces the number of parameters by 32.5% and 24.5% for the base-size model and large-size model, respectively.

Particularly, the BLEU scores of our ET5 framework outperform DISCERN and DGM with a considerable improvement in the test set. Compared with the previous SOTA, the results have increased by 5.6 and 5.7 respectively in BLEU-1 and BLEU-4. Moreover, as shown in Table 3, the existing pipeline frameworks have a certain degree of decline on the test set with BLEU scores, which indicates the drawback of the existing pipeline architectures. In the contract, the BLEU scores of ET5 and ET5-Discern continue to improve on the test set, which demonstrates the better generalization of our framework ET5 in question generation. The above results prove that our proposed framework takes better advantage of the fine-grained entailment reasoning information and eliminate the information gap between decision making and question generation.

Additionally, in the decision making evaluation, we achieve the best performance in the dev set, but there is a slight drop in the test set. However, a good classification result must be an inference based on an existing fact. Intuitively, the correctness of reasoning can be analyzed by the performance of the question generation. Correct reasoning will make the model ask the right questions. Correct classification, but asking the wrong question, does not mean that the model has learned the reasoning ability correctly, and the phenomena such as statistical bias may also cause this problem.

4.4 Ablation Studies

The existing generation question evaluation metrics suffer from randomness[4] on the small dev set

[4]The generated questions are evaluated with BLEU scores only if the prediction and ground truth in classification are

Models	Micro	Macro	ABLEU-1	ABLEU-4
ET5-Base	**75.9**	**80.4**	**54.7**	**43.6**
ET5-Base-wo/g	75.4	79.7	45.0	36.4
ET5-Base-wo/g+f	73.4	78.0	49.4	40.3
ET5-Base-wo/e+f	72.9	77.3	42.1	35.0

Table 4: Ablation study of our base-size model on the dev set of ShARC.

(2,270). To better evaluate the question generation abilities of models on the dev set, we utilized ALL-BLEU (ABLEU) to evaluate all the examples that ground truth is inquired to generate a question in ablation studies. All the other settings remain the same with official evaluation.

The ablation studies of ET5 on the dev set on ShARC benchmark are shown in Table 4. We use the base-size model to investigate the impacts of different components, there are three ablations of our ET5-Base is considered:

- **ET5-Base-wo/g** trains the model without graph reasoning block, the setting is the same as ET5-Discern.
- **ET5-Base-wo/g+f** trains the model without graph reasoning block and fine-grained prefix.
- **ET5-Base-wo/e+f** trains the model without entailment reasoning decoder and fine-grained prefix, which can be considered as the original T5 model.

4.4.1 Analysis of Graph Reasoning

Graph Reasoning consists of explicit discourse graph reasoning and implicit discourse graph reasoning, each of them introducing discourse relations among EDUs and decoupling-fusion mechanism into ET5, respectively. This setting is the same as ET5-Discern. Both accuracy scores and ABLEU scores are improved by introducing graph reasoning. In addition, we observe a significant reduction in the ABLEU scores if removing graph reasoning. ABLEU is used to measure whether the model answers due to the correct reasoning of the missing knowledge, the results show ET5-Base correctly reasoned out the missing knowledge, which suggests the necessity of graph reasoning block.

both 'inquire'.

4.4.2 Analysis of Fine-grained Text Prefix

We investigate the necessity of the fine-grained text prefix by additional removing the fine-grained text prefix in ET5-Discern, while it's hard to reason for the entailment of EDUs without the fine-grained special prefix. We feed fine-grained special tokens prefixed text into ET5 directly. As shown in the results, compared with ET5-Base-wo/g and ET5-Base-wo/g+f, the accuracy will be significantly improved by introducing the fine-grained text prefix, which indicates that directly using special token prefixes will cause noise disturbance for semantic learning. As illustrated in 4, the ABLEU-1 is decreased by 4.4, and the ABLEU-4 is decreased by 3.9. The above results show the importance of the fine-grained text prefix.

4.4.3 Analysis of Entailment Reasoning

ET5-Base-wo/e+f can be considered as the official T5 model. As shown in Table 4, the lack of fine-grained entailment reasoning information will seriously affect the performance of decision making and question generation. Compared with the performance of ET5-Base, the ABLEU-1 and ABLEU-4 of ET5-Base-wo/e+f decreased by 7.3 and 5.3 after removing entailment reasoning decoder, which indicates the importance of entailment reasoning, especially for reasoning of question generation.

5 Conclusion

In this paper, we propose a novel end-to-end framework, called ET5, to better capture the entailment information for question generation in CMRC, and thus eliminate the information gap between decision making and question generation. By conducting a parameter shared encoder between answer generation decoder and entailment reasoning decoder, the answer generation decoder can utilize the fine-grained entailment reasoning information to enhance the performance of question generation. Experimental results suggest that the proposed framework ET5 achieves the new state-of-the-art results on the ShARC benchmark.

Acknowledgements

The work is supported by National Key R&D Plan (No.2020AAA0106600), National Natural Science Foundation of China (No.U21B2009, 62172039 and L1924068).

References

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. MuTual: A dataset for multi-turn dialogue reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1406–1416, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32.

Yifan Gao, Jingjing Li, Michael R Lyu, and Irwin King. 2021. Open-retrieval conversational machine reading. *arXiv preprint arXiv:2102.08633*.

Yifan Gao, Chien-Sheng Wu, Shafiq Joty, Caiming Xiong, Richard Socher, Irwin King, Michael Lyu, and Steven C.H. Hoi. 2020a. Explicit memory tracker with coarse-to-fine reasoning for conversational machine reading. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Yifan Gao, Chien-Sheng Wu, Jingjing Li, Shafiq Joty, Steven C.H. Hoi, Caiming Xiong, Irwin King, and Michael Lyu. 2020b. Discern: Discourse-aware entailment reasoning network for conversational machine reading. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2439–2449.

Carolin Lawrence, Bhushan Kotnis, and Mathias Niepert. 2019. Attending to future tokens for bidirectional sequence generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1–10, Hong Kong, China. Association for Computational Linguistics.

Friedrich Wilhelm Levi. 1942. *Finite geometrical systems: six public lectues delivered in February, 1940, at the University of Calcutta*. University of Calcutta.

Jing Li, Aixin Sun, and Shafiq Joty. 2018. Segbot: a generic neural text segmentation model with pointer network. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4166–4172.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam. In *International Conference on Learning Representations*.

Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2021. Dialogue graph modeling for conversational machine reading. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3158–3169, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7.

Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Nikhil Verma, Abhishek Sharma, Dhiraj Madan, Danish Contractor, Harshit Kumar, and Sachindra Joshi. 2020. Neural conversational QA: Learning to reason vs exploiting patterns. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7263–7269, Online. Association for Computational Linguistics.

Zhuosheng Zhang, Siru Ouyang, Hai Zhao, Masao Utiyama, and Eiichiro Sumita. 2021. Smoothing dialogue states for open conversational machine reading. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Victor Zhong and Luke Zettlemoyer. 2019. E3: Entailment-driven extracting and editing for conversational machine reading. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.