

Towards Interpretable and Reliable Reading Comprehension: A Pipeline Model with Unanswerability Prediction

1st Kosuke Nishida
NTT Media Intelligence Laboratories
NTT Corporation
Yokosuka, Kanagawa, Japan
kosuke.nishida.ap@hco.ntt.co.jp

2nd Kyosuke Nishida
NTT Media Intelligence Laboratories
NTT Corporation
Yokosuka, Kanagawa, Japan
kyosuke.nishida.rx@hco.ntt.co.jp

3rd Itsumi Saito
NTT Media Intelligence Laboratories
NTT Corporation
Yokosuka, Kanagawa, Japan
itumi.saito.df@hco.ntt.co.jp

4th Sen Yoshida
NTT Media Intelligence Laboratories
NTT Corporation
Yokosuka, Kanagawa, Japan
sen.yoshida.tu@hco.ntt.co.jp

Abstract—Multi-hop QA with annotated supporting facts, which is the task of reading comprehension (RC) considering the interpretability of the answer, has been extensively studied. In this study, we define an *interpretable reading comprehension* (IRC) model as a pipeline model with the capability of predicting unanswerable queries. The IRC model justifies the answer prediction by establishing consistency between the predicted supporting facts and the actual rationale for interpretability. The IRC model detects unanswerable questions, instead of outputting the answer forcibly based on the insufficient information, to ensure the reliability of the answer. We also propose an end-to-end training method for the pipeline RC model. To evaluate the interpretability and the reliability, we conducted the experiments considering unanswerability in a multi-hop question for a given passage. We show that our end-to-end trainable pipeline model outperformed a non-interpretability model on our modified HotpotQA dataset. Experimental results also show that the IRC model achieves comparable results to the previous non-interpretability models in spite of the trade-off between prediction performance and interpretability.

Index Terms—interpretability, reading comprehension, question answering

I. INTRODUCTION

There is increasing demand for automated decision-making by using artificial intelligence (AI) [1], [2]. Moreover, reading comprehension (RC), a task to answer a question with textual sources, is an important topic in AI research. It is tackled with deep neural models such as BERT [3]. However, it is difficult for a neural network black box to provide reasons for its predictions. This problem affects the perceived reliability and interpretability of RC models used in social contexts.

Multi-hop QA datasets with annotated supporting facts (SFs) [4], [5] have been proposed for the RC task in order to develop a way to interpret the model’s predictions. Fig. 1 shows an example in the HotpotQA dataset. In HotpotQA, the model outputs an answer A and SFs R in response to a query Q on a

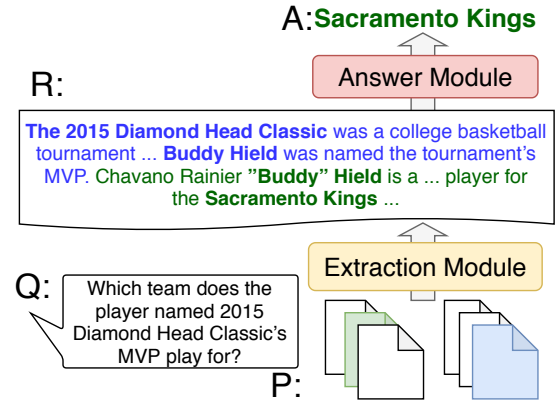


Fig. 1. Concept of interpretable reading comprehension. The QA module predicts "Can Not Answer" if the output of the extraction module does not include both ground-truth SFs.

given passage P . The SFs are a set of sentences that describe the reasoning behind the answer.

Our goal is to enable the model to predict the answer and give a corresponding rationale for the sake of *interpretability* and *reliability* in RC. First, to increase interpretability, the model should extract rationale sentences from the passage that are truly required for reasoning; these sentences are called SFs in HotpotQA. The previous models can predict SFs independently of the predicted answer [6], [7]. Therefore, the models can predict sentences that are not actually used for reasoning in the models as SFs. In this study, we propose to use a pipeline model for RC that first extracts rationales from the passage and then predicts the answer; this model has been used in the past for interpretation of text classification tasks [8]–[10]. The proposed interpretable reading comprehension (IRC) model can

justify the model prediction by ensuring consistency between the predicted SFs and the actual rationale used by the RC module. Fig. 1 shows the concept of IRC.

Next, to increase the reliability of the justified answer prediction, we consider that the model should answer with valid reasoning based on sufficient information. It is known that HotpotQA has a reasoning shortcut problem [11]. This is a phenomenon in which the model can directly locate the answer from the only one sentence (the green sentence in the example), despite the fact that there are multiple SFs (both statements). Since the HotpotQA dataset will always have an answer for each query, the model learns to answer with dishonest reasoning.

We prohibit the model from answering with the invalid reasoning by training it to detect unanswerable queries. We introduce a ‘Can Not Answer’ (CNA) label as an answer candidate. The model can avoid answering with invalid reasoning by outputting a CNA label.

We analyzed the reliability and the interpretability of RC models from two points of view. Firstly, the unanswerability detection has the effect of preventing reasoning shortcuts. To show this, we made simple modifications to the evaluation setting of HotpotQA. In this setting, the model must output a CNA label if the passage does not have sufficient information. Secondly, the end-to-end learning of the pipeline model with the CNA label enables the predicted rationale to explain the valid reasoning behind the answer. To show how the answer module finds an answer and determines that the reasoning is valid, we conducted a qualitative analysis of the rationale prediction.

Our main contributions are as follows.

- We define IRC as an end-to-end trainable pipeline model with a ‘Can Not Answer’ prediction. For interpretability, the model justifies its prediction by ensuring consistency between the predicted SFs and the actual rationale. For reliability, the model can detect unanswerable queries instead of outputting an answer forcibly with invalid reasoning.
- We conducted experiments on the modified HotpotQA dataset that contains a CNA label as an answer candidate to evaluate the model’s reliability. We show that the IRC model outperforms a non-interpretable model.
- We show that the IRC model achieves comparable results to the previous non-interpretable models in the original setting of HotpotQA. Although there is a trade-off between prediction performance and interpretability [2], the proposed IRC model maintains prediction performance while increasing interpretability.
- Through a qualitative analysis, we discuss how the answer module locates an answer and determines that the reasoning is valid. Even if the predicted rationales include non-gold SFs, they often play a role for reasoning in the model, such as by being supplementary explanations of entities.

II. PRELIMINARIES

A. Task Definition

We define the interpretable reading comprehension to provide consistency to the predicted SFs and rationales behind the model’s prediction.

Def. 1 (Interpretable Reading Comprehension). We say that an RC model is interpretable if it has two modules with the following inputs and outputs.

- **Extraction Module:** The inputs are a passage P and query Q . The output is the rationale R .
- **Answer Module:** The inputs are a rationale R and query Q . The output is the answer A .

The rationale is a set of sentences. The answer is the label or the span in the passage. The candidate answers include the CNA label, which represents that the passage is insufficient as a reason. In what follows, we define A^* to be the ground-truth answer and \hat{A} to be the predicted answer.

The IRC model can justify the model prediction. That is, because the answer module only uses the information in the rationale R , we can avoid a situation where the answer module implicitly depends on other information. Here, [12] divided the explainable models to hard and soft approaches. The IRC model is a hard approach, and this approach is faithful because of the discrete extraction of the rationale. In comparison, the soft approach may still use all sentences in the passage to predict particular answer independently on the SFs prediction. We call such model a one-stage model.

B. Related Work

1) *Interpretable NLP:* One of the goals of explainable AI is to ‘produce more explainable models, while maintaining a high level of learning performance’ [2]. We contributed to this goal because the IRC model justifies the answer without affecting answer performance.

There are various approaches to making interpretable models [13]; we focused on hard selection of the input for the justification and avoidance of the invalid reasoning. The hard approach of the pipeline models has been used for text classification [9], [10]. There is a limitation when it comes to applying the pipeline models used in text classification tasks to the RC tasks because the rationale of the text classification tasks is a few words such as positive words. [14], [15] provided a new task and dataset for medical and fact-checking usages focusing on interpretability. They also used hard selection of the rationale with a pipeline model.

2) *Reading Comprehension:* Multi-hop QA was proposed in order to verify the ability to reason over multiple text [4], [16]. Multi-hop reasoning is essential to evaluating the interpretability of RC, because in traditional RC datasets such as SQuAD [17], the query can be answered with the single sentence that has the answer [18]. SQuAD2.0 [19] has the no-answer option, but reasoning on the basis of only one sentence is not suitable for an evaluation of interpretability.

We chose the HotpotQA dataset for our study, although there are a few multi-hop QA datasets with manually annotated SFs. In particular, MultiRC [5] is a multiple-choice dataset, where, unlike HotpotQA, the number of correct answer options for each question is not pre-specified. In this dataset, the SFs (used sentences) for answering the question (choosing all correct answer options) are provided. However, the actual SFs required for each answer option are different, but are not annotated individually. A future challenge will be to extend our model to make it able to answer questions from several different perspectives and to explain these perspectives as its rationale.

We should mention the work using pipeline models for RC. Some studies have used pipeline models for efficient computation [20], [21]. [22] proposed a pipeline model for multi-hop QA. Their pipeline model extracts the relevant sentences from the passage and then predicts the answer and the SFs from the sentences. They showed that the sentence-level extraction is a strong approach to multi-hop QA. However, their model does not have our interpretable structure, because their answer module relies on information other than the predicted SFs. Neither of these studies proposed any end-to-end training method.

[11] pointed out the reasoning shortcut and created an adversarial dataset by generating fake answers. The reasoning shortcut was tackled by decomposing the query to sub-queries [23]–[25]. They used complex models to combine the single-hop sub-queries. By comparison, the IRC model is characterized by the detection of CNA label and justification of the prediction.

III. PROPOSED METHOD

A. Model Architecture

Our pipeline model consists of an extraction module and answer module. Each module has a language understanding layer and a task-specific linear layer. The language understanding layer is a pre-trained language model (LM), and we used BERT_{base}. Fig. 2 shows the model architecture.

Extraction Module: We input the token sequence $['[CLS^Q]'; \text{query}; '[SEP^Q]'; '[CLS^S]'; \text{sentence 1}; '[SEP^S]'; \dots; '[CLS^S]'; \text{sentence } N^s; '[SEP^S]']$ to BERT, where $['[CLS^Q]'; '[SEP^Q]'; '[CLS^S]'; \text{and } '[SEP^S]'$ are special tokens and N^s is the number of sentences in the passage. The i -th $['[CLS^S]'$ token output is the i -th sentence representation $s_i \in \mathbb{R}^d$, where d is the embedding dimension. We obtain the i -th sentence score from the linear layer output:

$$p_i = \text{sigmoid}(W^s s_i + b^s), \quad (1)$$

where $W^s \in \mathbb{R}^d, b^s \in \mathbb{R}$ are trainable parameters.

Answer Module: We input the token sequence $['[CLS]'; \text{query}; '[SEP]'; \text{rationale}; '[SEP]']$ to BERT, where $['[CLS]'$ and $['[SEP]'$ are special tokens. The output sequence is denoted as H^a . The answer layer has two linear transformations

$$c = W^c h_0^a + b^c \in \mathbb{R}^{N^c}, \quad (2)$$

$$[a_i^s; a_i^e]^\top = W^a h_i^a + b^a \in \mathbb{R}^2. \quad (3)$$

Equation (2) is the answer label classification. The number of answer labels N^c depends on the task. The candidates of the

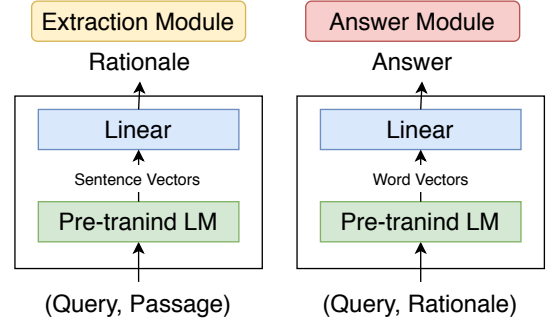


Fig. 2. Proposed IRC model.

answer labels include ‘Span’ and CNA. The ‘Span’ label means that the query should be answered by span extraction. Equation (3) is the answer span extraction. Each dimension is the score at the start or end of the answer span. $W^c \in \mathbb{R}^{N^c \times d}, b^c \in \mathbb{R}^{N^c}, W^a \in \mathbb{R}^{2 \times d}, b^a \in \mathbb{R}^2$ are trainable parameters.

B. Inference

In order to extract the actually required sentences for the reasoning, we use the CNA classification score to determine whether the extracted sentences are sufficient for reasoning. Firstly, the extraction module outputs the sentences with $p_i > \alpha$ as a rationale \hat{R} . Then, if the answer module predicts CNA, we add sentences determined by $\text{argmax}_{i \notin \hat{R}} p_i$ to the rationale \hat{R} . This operation continues until a stop criterion, the max number of rationales N^r , is met. N^r, α are hyperparameters.

C. Training

Loss Function: The loss of the extraction module L^R is the binary cross entropy loss between the sentence extraction probabilities $\{p_i\}$ and the ground-truth rationale $R^* \in 2^{N^s}$. The loss of the answer module L^A is the sum of the cross entropy losses of the answer label classification, the start token of the answer span, and the end token.

Pre-Training: We pre-train the extraction module and the answer module separately. The extraction module is trained with L^R , and the answer module is trained with L^A . The input for the answer module is the ground-truth rationale R^* .

End-to-End Training: We propose an end-to-end training algorithm for learning the interactions between the two modules. Here, we consider three different losses. The end-to-end answer loss L^{E2E} is the loss to locate the answer. The rationale extraction loss L^R is the same as in the pre-training. The no-answer penalty L^{NA} helps the extraction module not to miss the sentence with the answer span.

Beforehand, the ground-truth answer is replaced by a CNA label if the result of the sampling \hat{R} does not include the whole ground-truth rationale R^* . This is because that the answer module must learn how to determine if the query cannot be answered the sentences extracted from the passage. We also augment one CNA sample for each query by performing negative sampling on the passage.

First, we explain the end-to-end answer loss, L^{E2E} . Ideally, we want to apply the loss on the probability

$$\Pr(A^*|P, Q) = \sum_{R \in 2^{N^s}} \Pr(A^*|R, Q) \Pr(R|P, Q) \quad (4)$$

to L^{E2E} . However, for computational reasons, we use the predicted rationale \hat{R} to calculate L^{E2E} .

The operation of extracting the rationale \hat{R} is not differentiable, so the gradient obtained in the answer module does not backpropagate to the extraction module. Therefore, we use a straight-through Gumbel-softmax estimator [26], [27]. The sentences are extracted by sampling in accordance with a discrete distribution. Let g_i, g'_i ($i = 1, \dots, N^s$) be i.i.d. samples from the Gumbel distribution¹. We sample a set of sentences by using the Gumbel-softmax trick [28], [29]. The i -th sentence is extracted if $g_i + \log p_i > g'_i + \log(1 - p_i)$. The continuous relaxation is

$$z_i = \frac{\exp(\frac{g_i + \log p_i}{\tau})}{\exp(\frac{g_i + \log p_i}{\tau}) + \exp(\frac{g'_i + \log(1 - p_i)}{\tau})}, \quad (5)$$

where τ is the hyperparameter of the temperature. Let $\mathbb{I}_{\hat{R}}(i)$ be the indicator function that returns 1 if the i -th sentence is extracted. On the backward path, we use the straight-through Gumbel-softmax estimator $\nabla \mathbb{I}_{\hat{R}}(i) \approx \nabla z_i$ as the approximation.

By Jensen's inequality, the loss of the answer module L^A is an upper bound of intractable L^{E2E} ,

$$\begin{aligned} L^{\text{E2E}} &= -\log \Pr(A^*|P, Q) \\ &= -\log \sum_R \Pr(A^*|R, Q) \Pr(R|P, Q) \\ &\leq -\sum_R \Pr(R|P, Q) \log \Pr(A^*|R, Q) \\ &= -\mathbb{E}_{R \sim \Pr(R|P, Q)} [\log \Pr(A^*|R, Q)] \\ &\approx -\log \Pr(A^*|\hat{R}, Q) = L^A. \end{aligned} \quad (6)$$

The last approximation uses the Gumbel-softmax trick. Therefore, we can learn the model in the end-to-end fashion with L^A instead of L^{E2E} .

Then, we introduce a no answer penalty L^{NA} . Because the extracted sentences are sampled independently of the ground-truth rationale R^* , we emphasize the sentences including the answer. The penalty is defined as

$$L^{\text{NA}} = \max(0, \max_{\hat{r} \in \hat{R}} W^s s_{\hat{r}} - \max_{i \in S_A} W^s s_i), \quad (7)$$

where S_A is the set of sentences with the ground-truth answer span. L^{NA} is zero if one of the sentences with the ground-truth answer has a higher score than that of any of the extracted sentences.

As a result, the loss function is $L^A + \lambda^R L^R + \lambda^{\text{NA}} L^{\text{NA}}$, where λ are hyperparameters. L^R , and L^{NA} help to stabilize the training which is otherwise affected by the approximation.

¹ $g = -\log(-\log(u))$, $u \sim \text{Uniform}(0, 1)$

IV. EVALUATION

A. Dataset and Metrics

We used HotpotQA, which is a multi-hop QA dataset with manually annotated SFs consisting of multiple sentences. In HotpotQA, the query Q refers to the content of two paragraphs from two Wikipedia articles. Each passage P consists of ten paragraphs. HotpotQA has two settings. In the distractor setting, the passage has two gold paragraphs, and the other eight paragraphs are selected in accordance with the TF-IDF similarity scores. The fullwiki setting is the RC task including the retrievals from a preprocessed Wikipedia dump. This setting also provides ten paragraphs that are retrieved in accordance with the TF-IDF similarity for convenience. The outputs are the answer A and the SFs, which are the rationale R in our IRC definition. The ground-truth answer A^* consists of the answer labels {'Yes', 'No', 'Span'} and a span in the passage. The answer span exists only if the ground-truth answer label is 'Span'. We added CNA to the answer labels. The rationale R is a set of sentence IDs.

The answers were evaluated in terms of exact matching (EM) and partial matching (F1) on the string. The rationales were evaluated in terms of EM and F1 on the set of sentence IDs. The leaderboard were evaluated on the test set; the others were evaluated on the development set.

B. Model Implementations

We used two extra modules to adapt our model to HotpotQA.

Paragraph Ranker: To select the input of the extraction module from the whole passage consisting of N^p paragraphs, we used the paragraph ranker based on the SAE paragraph ranker [7]. The paragraph ranker aims to retrieve a paragraph pair including the content referred to by the query. The input is the query and a paragraph. The output is the score of the i -th paragraph. It was calculated as S_i^{SAE} , and we ranked the paragraph pairs $\{(i, j)\}$ according to $S_i^{\text{SAE}} + S_j^{\text{SAE}}$.

To train the main model, we inputted gold paragraph pairs without the paragraph ranker. Moreover, we used negative sampling to make the CNA samples. Here, a randomly selected paragraph of the gold paragraph pairs was replaced with a non-gold paragraph including the sentence with the highest TF-IDF similarity to the query.

Answer Re-ranking: For inference, we used the paragraph ranker to extract the top- K paragraph pairs, which were then used as the inputs of the extraction module. For each pair, we used the extraction module and the answer module as above. Finally, we reranked the K pairs according to

$$\frac{1}{2}(S_i^{\text{SAE}} + S_j^{\text{SAE}}) - \frac{\exp(c_{\text{CNA}})}{\sum_l \exp(c_l)}, \quad (8)$$

where the answer label score c is calculated using (2).

Algorithm 1 shows the pseudo-code of the model for inference in HotpotQA.

Algorithm 1 Pipeline Model in Inference

Require: Passage P , Query Q , Hyperparameter K, α, N^r

- 1: Retrieve the top- K paragraph pairs with the paragraph ranker
 - 2: **for** Select the k -th paragraph pair from the top- K paragraph pairs **do**
 - 3: Obtain the sentence scores p_i with the extraction module from the k -th paragraph pair
 - 4: Select the sentences i with $p_i > \alpha$ as rationale \hat{R}
 - 5: Select the answer \hat{A} with the answer module from rationale \hat{R}
 - 6: **while** Answer \hat{A} is CNA and $|\hat{R}| < N^r$ **do**
 - 7: Add sentence $\text{argmax}_{i \notin \hat{R}} p_i$ to the rationale \hat{R}
 - 8: Select the answer \hat{A} with the answer module from rationale \hat{R}
 - 9: **end while**
 - 10: Add the answer \hat{A} and the rationale \hat{R} of the k -th paragraph pair to the prediction candidates
 - 11: **end for**
 - 12: Rerank the answer and the rationale in the K prediction candidates with the answer reranking (8)
 - 13: Output the answer and the rationale
-

C. Evaluated Models

We evaluated the prediction of the IRC model and the one-stage baseline model. The one-stage model simultaneously outputs the answer and the SFs with a shared module from the passage. Therefore, in comparison with the IRC model, the predicted SFs of the one-stage model can not justify the answer prediction.

As in the IRC model, the input passage of the one-stage model was a paragraph pair retrieved with the paragraph ranker. The model predicted the answer and the SFs from the output representations of BERT and the linear layers. The answer prediction and the re-ranking of the paragraphs were the same as those in the IRC model. For training, the loss function was $L^R + L^A$.

We trained the models on one NVIDIA Tesla P100 GPU (16GB). The training took less than one day with gradient accumulation. We used the PyTorch implementation of BERT².

The hyperparameter settings are in Table I. The threshold values for the rationale extraction, α , were determined from 0 to 0.9 by 0.1 to maximize the answer’s F1 score for the IRC model and to maximize the SFs’ F1 score for the one-stage model, because the answer performance of the one-stage model does not depend on the SFs’ prediction. The rest of the implementation including that of the optimizer followed the Pytorch BERT implementation.

D. Evaluation in HotpotQA with considering ‘Can Not Answer’

First, we evaluated the ability of the model to avoid answering with reasoning based on insufficient information. We conducted experiments in the fullwiki and CNA setting (‘Fullwiki+CNA’).

1) *Experimental Setup:* We used the ten paragraphs published as the fullwiki setting for the passage. However, a passage fully based on TF-IDF retrieval might not have the ground-truth answer or ground-truth SFs, and in such cases the model cannot provide any correct reasoning. Therefore, we replaced the ground-truth answer label with CNA if the

TABLE I
HYPERPARAMETERS.

λ^R in loss function	0.1
λ^{NA} in loss function	1
max number of rationales N^r	5
number of paragraph pairs K	3
batch size	72
epochs in pre-training	5
epochs in end-to-end training	2
max sequence length	512
max sentence length	160
max number of sentences	20
max query length	64
temperature in Gumbel softmax τ	0.5
learning rate	5e-5
weight decay	0

TABLE II
DATA STATISTICS OF THE FULLWIKI+CNA SETTING.

Can Not Answer	# Absent Gold SFs	# Examples
	0	2089
✓	1	3418
✓	2	1504
✓	3+	374

passage did not have both gold paragraphs. We removed the sentences not included in the passage from the ground-truth SFs. The data statistics of Fullwiki+CNA setting are listed in Table II. This is a different evaluation from the original HotpotQA fullwiki setting. If any of the gold SFs are not included in the passage, the gold answer is replaced with a CNA label.

The implementations of the IRC and one-stage model followed the algorithms described in Section III-A and IV-B. Finally, if the CNA score was higher than a hyperparameter β , the model outputted the CNA label. β , was determined from 0 to 0.9 by 0.1 to maximize the answer’s F1 score for each model.

2) Results and Discussion:

²<https://github.com/huggingface/pytorch-transformers>

TABLE III
PERFORMANCE OF OUR MODELS IN THE FULLWIKI+CNA SETTING.

	Answer		SFs			
	EM	F1	EM	Precision	Recall	F1
One-Stage Model	75.7	79.2	33.5	53.2	56.1	52.7
IRC model	80.2	83.3	12.5	36.1	71.4	43.2
– End2End Training	66.6	69.9	8.43	35.4	76.9	45.1
– L^R in End2End Training	74.6	77.0	0.662	21.2	58.7	29.8
– L^{NA} in End2End Training	73.5	76.8	10.9	38.1	75.6	47.1

a) Does the IRC model improve answer performance considering unanswerability?: As shown in Table III, the IRC model predicted the answer more accurately than the one-stage model did. We consider that the IRC model effectively recognized the irrelevance of the rationale to the query in the answer module. This is because that the IRC model determines the CNA prediction against the rationale for the sake of the consistency. In contrast, the one-stage model was not good at CNA detection because its input always had information that was unnecessary for answering the query.

The IRC model extracts the rationale by focusing on recall because the rationale must cover the text necessary for answering the query. Under-extraction of SFs with the correct answer, which results in a high precision score, causes the reasoning shortcut problem. We can explain the low EM score of the IRC model similarly. If the model aims for a high EM score, it is inevitable that under-extraction will occur as much as over-extraction. This is not an acceptable situation for interpretable and reliable RC.

The IRC model outperformed the ablated models. This indicated that our end-to-end training enabled the rationale module to consider ease of reasoning.

b) Can the IRC model avoid the reasoning shortcut?: Then, we discuss the performance in terms of the CNA detection task. Table IV shows the results of the evaluation of whether the answer is CNA or not. Each example is positive if the answer is CNA. The high performance in this evaluation suggests that the model avoids the reasoning shortcut, where the model answers by force from insufficient text.

The IRC model outperformed the one-stage model. This is the same tendency as in Table III. Both the IRC and one-stage models detected the CNA with an EM score of more than 85.6%. This indicates that a label classification including CNA and negative sampling of the paragraphs are effective at avoiding the reasoning shortcut.

We consider that the consistency between the SFs and answer prediction contributed to the CNA detection performance of the IRC model. In comparison, the one-stage model that ignores consistency may not detect that the query is unanswerable because it predicts the answer without recognizing that it may have missed the necessary information in the passage.

c) In what examples do the models detect unanswerable queries?: We classified the queries with respect to the number of absent gold SFs in the passage. Table V lists the CNA prediction ratio for each class. Lower CNA prediction ratio is better if all the gold SFs are extracted. Otherwise, higher is

TABLE IV
PERFORMANCE OF OUR MODELS AS THE CNA DETECTION MODEL.

	Acc.	Precision	Recall	F1
One-Stage Model	85.6	94.2	85.4	89.6
IRC model	88.7	90.5	94.3	92.4
– End2End Training	77.9	90.9	77.1	83.5
– L^R in End2End Training	83.2	87.8	89.2	88.5
– L^{NA} in End2End Training	83.5	93.9	82.6	87.9

TABLE V
CNA PREDICTION RATIO FOR EACH CLASS.

# Absent Gold SFs	# Examples	IRC	One-Stage
0 (↓)	2089	25.3	13.3
1 (↑)	3418	92.8	80.0
2 (↑)	1504	96.8	94.8
3+ (↑)	374	98.4	96.8

better. 2089 examples included all the gold SFs, so the model should predict the gold answer in the examples. The others had insufficient information. The IRC model predicted CNA correctly for more than 92.8% of the insufficient passages. The one-stage model performed worse in the examples with one absent sentence. This is the most typical case of the reasoning shortcut, where the model outputs the answer from one sentence including the answer while ignoring the sentence linking the query to the answer sentence.

We further classified the 2089 examples according to whether the predicted SFs are sufficient or not (i.e., whether the model predicted the SFs including all the gold SFs or not). Table VI shows the results. We found that, in 837 examples, the extraction module of the IRC model failed to extract gold SFs, and one-stage model had more failed extractions. In such insufficient cases, the answer module of the IRC model outperformed the one-stage model in the CNA prediction performance by 16.3% (higher is better). In sufficient cases, the one-stage model outperformed the answer module of the IRC model by 10.3 % (lower is better). We observed that the answer module predicted the CNA label conservatively in addition to the conservative prediction of the extraction module. This is important for reliability and interpretability.

E. Evaluation in HotpotQA without considering 'Can Not Answer'

We compared the IRC and one-stage model with the previous models in the distractor setting.

1) Experimental Setup: The answer labels were {'Yes', 'No', 'Span'}. For training, we used the CNA label. Except for CNA, we regarded the label with the highest score to be the predicted label.

2) Results and Discussion:

a) Does the IRC model improve the performance even when ignoring the unanswerability?: Table VII shows the test results. The previous models have sophisticated structures, such as graph neural networks using entity linkage. However, the IRC model achieved comparable results to these models. In particular, our simple pipeline model consisting of a BERT

TABLE VI
CNA PREDICTION RATIO IN EXAMPLES INCLUDING ALL GOLD SFs.

Is Sufficient	IRC		One-Stage	
	# Examples	CNA	# Examples	CNA
✓	1252	14.1	1178	3.82
	837	41.9	911	25.6

TABLE VII
PERFORMANCE OF MODELS ON THE HOTPOTQA DISTRACTOR SETTING LEADERBOARD³

	Answer		SFs	
	EM	F1	EM	F1
Baseline [4]	45.6	59.0	20.3	64.5
KGNN [30]	50.8	65.8	38.7	76.8
QFE [6]	53.9	68.1	57.8	84.5
DFGN (base) [31]	56.3	69.7	51.5	81.6
SAE (base) [7]	60.4	73.6	56.9	84.6
HGN (large) [32]	66.1	79.4	60.3	87.3
SAE (large) [7]	66.9	79.6	61.5	86.9
C2F Reader (large) [33]	68.0	81.3	60.8	87.6
IRC model (base)	58.6	72.5	36.7	79.4

³The unpublished models are not listed. The IRC model used BERT_{base}.

layer and a linear layer had an F1 score that was 2.8 points higher than that of DFGN and 1.2 points lower than that of SAE, where both DFGN and SAE consisted of the BERT base model and the graph neural networks.

Table VIII shows the results of the evaluation on the development set. The IRC model performed comparably to the one-stage model. We consider that the IRC model has an effect on unanswerability detection, but has little effect on locating the answer. In addition, the IRC model could justify the model prediction. This shows that the IRC model can maintain prediction performance while providing a higher level of interpretability.

Similarly to the Fullwiki+CNA setting, the IRC model outperformed the one-stage model in terms of recall. The IRC model focuses on the recall score for interpretability. By contrast, the one-stage model extracts sentences with high precision scores. We consider that this is due to the difficulty in judging the irrelevance of the sentence to the query because of the relations among the sentences in the same paragraph. The CNA output seems to be useful to determine requirements conservatively.

b) Qualitative Analysis of Extracted Rationale in case of sufficient extraction: The high recall prediction of the IRC model means that it redundantly extracts rationales. However, we found that the predicted rationale often included a sentence useful for the answer prediction in addition to the ground-truth SFs. Table IX shows some examples. Double quotations represent the title of the article.

The first example is a case where the manual annotation is insufficient. The reasoning is incomplete without the second sentence. The second example is a case where we feel that the ground-truth SFs are sufficient for reasoning because we know ‘disir’ and ‘Idisi’ are synonyms from the third sentence. However, we consider that the second sentence mentioning

TABLE VIII
PERFORMANCE OF THE MODELS ON THE DEVELOPMENT SET.

	Answer		SFs			
	EM	F1	EM	Precision	Recall	F1
One-Stage Model	58.4	72.8	54.3	87.3	85.7	85.0
IRC model	58.6	72.9	36.6	76.0	89.1	79.8

‘ghost’ contributes to the reasoning of the model. From another point of view, the infrequent words, ‘disir’ and ‘Idisi’, are not well represented in the embedding space, so the second sentence plays the role of prior knowledge.

c) Qualitative Analysis of Extracted Rationale in case of insufficient extraction: Though the IRC model extracts rationales conservatively, there are some examples where the model extracts insufficient sentences but predicts a non-CNA answer. However, we observed that such extractions also have enough information for reasoning.

[34] classifies the examples in the distractor setting in four categories: multi-hop, weak-distractors, redundant evidence, and non-compositional single-hop. Redundant evidence means that the query has a redundant entity. We found that sentence not included in the gold SFs usually serve as substitutes for an absent gold SF or had the effect of increasing the confidence of the answer.

Table X shows an example of redundant evidence examples with the redundant entity of Gary Sinese because the query is valid without mentioning him. The only one gold SF is sufficient for reasoning. However, the non-gold sentence gave basic knowledge about Gary Sinese. We consider that the supplementary information has an effect on confidence in the model even if the information is on a redundant entity.

V. CONCLUSION

We defined interpretable reading comprehension (IRC) as the ability of a model to justify an answer behind valid reasoning based on sufficient information. It is implemented as a pipelined model with a ‘Can Not Answer’ label. The IRC model regards the SFs as information passed between the modules in order to provide the actual rationale of the answer. The CNA output improves the reliability of the answer prediction by detecting reasoning with insufficient rationale. We believe that our study advances the interpretability of RC to the next level and the IRC model resolves the issues of reading comprehension, such as fact checking.

REFERENCES

- [1] D. Aha, Ed., *Proceedings of the IJCAI Workshop on Explainable Artificial Intelligence*, 2018.
- [2] D. Gunning, *Explainable artificial intelligence (XAI)*. Defense Advanced Research Projects Agency (DARPA), 2017.
- [3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” *NAACL-HLT*, pp. 4171–4186, 2019.
- [4] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, “HotpotQA: A dataset for diverse, explainable multi-hop question answering,” in *EMNLP*, 2018, pp. 2369–2380.
- [5] D. Khashabi, S. Chaturvedi, M. Roth, S. Upadhyay, and D. Roth, “Looking beyond the surface: A challenge set for reading comprehension over multiple sentences,” in *NAACL-HLT*, 2018, pp. 252–262.

TABLE IX
OUTPUTS OF OUR MODEL WITH SUFFICIENT RATIONALE.

Ex.1 Q: What was the sequel of the game that e was published by U.S. Gold in 1992? A: Fade to Black		
ground-truth	extracted	text
✓	✓	"Fade to Black (video game)" Fade to Black is an action-adventure game ... published by Electronic Arts.
	✓	"Fade to Black (video game)" It is the sequel to the 1992 video game 'Flashback'.
✓	✓	"Flashback (1992 video game)" Flashback ... is a 1992 science fiction cinematic platform game ... published by U.S. Gold ...
Ex.2 Q: Besides dísir, what is another Nordic term for a ghost? A: Idisi		
ground-truth	extracted	text
✓	✓	"North Germanic languages" ... 'Nordic languages', a direct translation of the most common term used among Danish, Swedish and Norwegian scholars and laypeople.
	✓	"Dís" In Norse mythology, a dís ('lady', plural dísir) is a ghost, spirit or deity associated ...
✓	✓	"Dis" The North Germanic dísir and West Germanic Idisi are believed by some scholars to be related due to linguistic and mythological similarities ...

TABLE X
OUTPUTS OF OUR MODEL WITH INSUFFICIENT RATIONALE.

Ex.1 Q: Multiple award-winning actor Gary Sinise appeared in The Stand in 1994 - a miniseries based on a novel and screenplay by which noted author? A: Stephen King		
ground-truth	extracted	text
✓	✓	"The Stand (miniseries)" The Stand is a 1994 American television miniseries based on the novel of the same name by Stephen King.
	✓	"Gary Sinise" Gary Alan Sinise (...) is an American actor, director, and musician.
✓		"Gary Sinise" Among other awards, he has won an Emmy Award, a Golden Globe Award, a star on Hollywood Walk of Fame and has been nominated for an Academy Award.

- [6] K. Nishida, K. Nishida, M. Nagata, A. Otsuka, I. Saito, H. Asano, and J. Tomita, "Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction," in *ACL*, 2019, pp. 2335–2345.
- [7] M. Tu, K. Huang, G. Wang, J. Huang, X. He, and B. Zhou, "Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents," in *AAAI*, 2020.
- [8] T. Lei, R. Barzilay, and T. Jaakkola, "Rationalizing neural predictions," in *EMNLP*, 2016, pp. 107–117.
- [9] J. Bastings, W. Aziz, and I. Titov, "Interpretable neural predictions with differentiable binary variables," in *ACL*, 2019, pp. 2963–2977.
- [10] M. Yu, S. Chang, Y. Zhang, and T. Jaakkola, "Rethinking cooperative rationalization: Introspective extraction and complement control," in *EMNLP-IJCNLP*, 2019, pp. 4094–4103.
- [11] Y. Jiang and M. Bansal, "Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA," in *ACL*, Jul. 2019, pp. 2726–2736.
- [12] J. DeYoung, S. Jain, N. F. Rajani, E. Lehman, C. Xiong, R. Socher, and B. C. Wallace, "ERASER: A benchmark to evaluate rationalized NLP models," in *ACL*, 2020, pp. 4443–4458.
- [13] Z. C. Lipton, "The mythos of model interpretability," in *WHI@ICML*, 2016, pp. 96–100.
- [14] E. Lehman, J. DeYoung, R. Barzilay, and B. C. Wallace, "Inferring which medical treatments work from reports of clinical trials," in *NAACL-HLT*, 2019, pp. 3705–3717.
- [15] S. Chen, D. Khoshabi, W. Yin, C. Callison-Burch, and D. Roth, "Seeing things from a different angle: Discovering diverse perspectives about claims," in *NAACL-HLT*, 2019, pp. 542–557.
- [16] J. Welbl, P. Stenetorp, and S. Riedel, "Constructing datasets for multi-hop reading comprehension across documents," *TACL*, vol. 6, pp. 287–302, 2018.
- [17] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *EMNLP*, 2016, pp. 2383–2392.
- [18] S. Sugawara, K. Inui, S. Sekine, and A. Aizawa, "What makes reading comprehension questions easier?" in *EMNLP*, 2018, pp. 4208–4219.
- [19] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for SQuAD," in *ACL*, 2018, pp. 784–789.
- [20] E. Choi, D. Hewlett, J. Uszkoreit, I. Polosukhin, A. Lacoste, and J. Berant, "Coarse-to-fine question answering for long documents," in *ACL*, 2017, pp. 209–220.
- [21] S. Min, V. Zhong, R. Socher, and C. Xiong, "Efficient and robust question answering from minimal context over documents," in *ACL*, 2018, pp. 1725–1735.
- [22] D. Groeneveld, T. Khot, A. Sabharwal *et al.*, "A simple yet strong pipeline for hotpotqa," *arXiv preprint arXiv:2004.06753*, 2020.
- [23] Y. Jiang and M. Bansal, "Self-assembling modular networks for interpretable multi-hop reasoning," in *EMNLP-IJCNLP*, 2019, pp. 4474–4484.
- [24] E. Perez, P. Lewis, W.-t. Yih, K. Cho, and D. Kiela, "Unsupervised question decomposition for question answering," in *EMNLP*, 2020, pp. 8864–8880.
- [25] Y. Tang, H. T. Ng, and A. K. Tung, "Do multi-hop question answering systems know how to answer the single-hop sub-questions?" *arXiv preprint arXiv:2002.09919*, 2020.
- [26] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *ICLR*, 2017.
- [27] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.
- [28] E. J. Gumbel, *Statistical theory of extreme values and some practical applications: a series of lectures*. US Government Printing Office, 1948, vol. 33.
- [29] C. J. Maddison, D. Tarlow, and T. Minka, "A* sampling," in *NIPS*, 2014, pp. 3086–3094.
- [30] D. Ye, Y. Lin, Z. Liu, Z. Liu, and M. Sun, "Multi-paragraph reasoning with knowledge-enhanced graph neural network," *arXiv preprint arXiv:1911.02170*, 2019.
- [31] Y. Xiao, Y. Qu, L. Qiu, H. Zhou, L. Li, W. Zhang, and Y. Yu, "Dynamically fused graph network for multi-hop reasoning," in *ACL*, 2019.
- [32] Y. Fang, S. Sun, Z. Gan, R. Pillai, S. Wang, and J. Liu, "Hierarchical graph network for multi-hop question answering," in *EMNLP*, 2020.
- [33] N. Shao, Y. Cui, T. Liu, S. Wang, and G. Hu, "Is graph structure necessary for multi-hop reasoning?" *arXiv preprint arXiv:2004.03096*, 2020.
- [34] S. Min, E. Wallace, S. Singh, M. Gardner, H. Hajishirzi, and L. Zettlemoyer, "Compositional questions do not necessitate multi-hop reasoning," in *ACL*, 2019, pp. 4249–4257.