Structure-Discourse Hierarchical Graph for Conditional Question Answering on Long Documents

Haowei Du[1,2,3], Yansong Feng[1], Chen Li [5], Yang Li[5], Yunshi Lan[6]
Dongyan Zhao[1,2,3,4,*]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Center for Data Science, Peking University
[3] Institute for Artificial Intelligence, Peking University
[4] State Key Laboratory of Media Convergence Production Technology and Systems
[5] Ant Group [6] East China Normal University
duhaowei@stu.pku.edu.cn, {fengyansong,zhaodongyan}@pku.edu.cn
wenyou.lc@antgroup.com, ly200170@alibaba-inc.com
yslan@dase.ecnu.edu.cn

Abstract

Conditional question answering on long documents aims to find probable answers and identify conditions that need to be satisfied to make the answers correct over long documents. Existing approaches solve this task by segmenting long documents into multiple sections, and attending information at global and local tokens to predict the answers and corresponding conditions. However, the natural structure of the document and discourse relations between sentences in each document section are ignored, which are crucial for condition retrieving across sections, as well as logical interaction over the question and conditions. To address this issue, this paper constructs a Structure-Discourse Hierarchical Graph (SDHG) and conducts bottom-up information propagation. Firstly we build the sentence-level discourse graphs for each section and encode the discourse relations by graph attention. Secondly, we construct a section-level structure graph based on natural structures, and conduct interactions over the question and contexts. Finally different levels of representations are integrated into jointly answer and condition decoding. The experiments on the benchmark ConditionalQA shows our approach gains over the prior state-of-the-art, by 3.0 EM score and 2.4 F1 score on answer measuring, as well as 2.2 EM score and 1.9 F1 score on jointly answer and condition measuring. Our code will be provided on https://github.com/yanmenxue/ConditionalQA.

1 Introduction

Conditional question answering (QA) aims to answer questions from contexts where conditions are used to distinguish answers as well as to provide additional information to support them (Saeidi et al., 2018; Gao et al., 2020; Ouyang et al., 2021a; Sun et al., 2022). Recently more interest of the community has been put on conditional QA over long documents like government policies which are close to reality scenes (Sun et al., 2022). These approaches based on transformer framework encode each document segment respectively, and proceed interaction between the question as well as different levels of contexts. The integrated token representations are used to predict the answer span and corresponding conditions. However, the natural document structure, i.e., section levels and discourse relations (Jia et al., 2018; Shi and Huang, 2019; Yu et al., 2022) between sentences within the document segment (section) are ignored, which are crucial for conditions retrieving across sections, as well as logical interaction over the question and conditions.

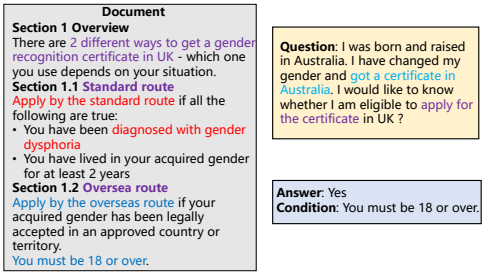


Figure 1: An example from ConditionalQA dataset.

We take an example from the ConditionalQA dataset in Figure 1. The document discusses the gender recognition certificate in UK and the question asks for the eligibility to apply. Section 1.1 and 1.2 are two child sections (subsections) of section 1, so they describe two parallel and relevant aspects about the contents in the parent section. We name the sections sharing the same parent section as sibling sections, like section 1.1 and 1.2. Section 1.1 and 1.2 elaborate two different routes to apply, each coupled with a group of conditions to satisfy. As long as the question satisfies one

*Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2023, pages 6282–6293
July 9-14, 2023

group of conditions, the answer will be “Yes". The structural relations among section 1, 1.1 and 1.2 enables the model to reason from “2 different ways to get a gender recognition certificate" to the “standard route" and “oversea route" in two subsections. Moreover, the discourse relation *condition* between “apply by the standard route if all $\cdots$ " and “you have been diagnosed $\cdots$ ", as well as “apply by the overseas route if $\cdots$ " and “you must be 18 or over" helps the model locate the relevant conditions. The question applies to the second route, satisfying the condition “gender has been legally accepted" and the unsatisfied condition “you must be 18 or over" needs to be outputted with the answer. It shows natural document structure and discourse information enhance the ability to retrieve relevant conditions across different sections and logically reason for the answer.

To capture the natural structure among sections and the discourse relations between sentences, we propose our structure-discourse hierarchical graph (SDHG). We design a hierarchical and heterogeneous graph, which includes a section-level structure graph and a set of sentence-level discourse graphs. In the structure graph, each node denotes a section in the document, where the parent-child and sibling relations between sections are used to build the edges. We utilize GAT (Veličković et al., 2017) to propagate information on the structure graph to encode the information that the child sections elaborate parallel and relevant aspects about the contents in their parent section. Each section has its corresponding sentence-level discourse graph, where each node denotes a sentence in this section or a discourse relation between 2 text spans. Similarly we apply GAT to incorporate the logical discourse relations among the sentences. We apply bottom-up encoding process in our hierarchical framework, where the sentence representations from pretrained language model (PLM) (Raffel et al., 2020; Lewis, 2022) pass through respective sentence-level discourse graph to introduce the discourse relations, and the integrated representations go through the section-level structure graph to enhance the document structural information. We conduct the experiments on the benchmark dataset ConditionalQA, and significantly outperforms the existing approaches by 3.0 EM score and 2.4 F1 score for answer evaluation, and 2.2 EM score and 1.9 F1 score for jointly answer-condition evaluation.

Our contributions can be summarized as:

1. We are the first to incorporate natural document structure information and discourse relations between sentences to enhance the answer and condition retrieving across sections, as well as logical reasoning over the question and conditions for conditional QA on long documents.

2. Our approach outperforms existing methods on the benchmark dataset of this field, becoming the new state-of-the-art.

2 Related Work

Conditional QA requires finding the probable answers and identifying their unsatisfied conditions (Sun et al., 2022). E^3 (Zhong and Zettlemoyer, 2019) extracts a set of decision rules from the context and reasons about the entailment. DISCERN (Gao et al., 2020) splits the document into elementary discourse units (EDU) (Schauer, 2000) and predicts whether each EDU is entailed. DGM (Ouyang et al., 2021b) constructs the explicit and implicit graphs of EDU to capture the interactions among contexts and questions with the support of tagged discourse relationship. However, these models ignore the natural structure of documents, and the EDU-based discourse graph undermines the informational continuity of sentences. Moreover, simply concatenating the question with full context into a single input and encoding it with a Transformer model with $O(N^2)$ complexity make it not scalable to longer contexts.

ETC (Ainslie et al., 2020) introduces attention mechanism between global tokens and regular input tokens to scale input length and encode structured inputs. DocHopper (Sun et al., 2021) utilizes the structural information that paragraphs and sentences contain different levels of information, and perform evidence retrieval at both sentence and section levels. To efficiently aggregate and combine long documents information, FID (Izacard and Grave, 2021) concatenates the representations of different document sections produced by the encoder independently and performs fusion in the decoder only. To enhance interaction between different levels of text segments, CGSN (Nie et al., 2022) propagates information on the global and local graph composed of nodes for tokens, sentences as well as document sections. However, these models ignore the hierarchical structure of the document and discourse relations between sen-

tences within each document section, which brings difficulty to condition locating across sections and logical reasoning for answers.

HIBRIDS (Cao and Wang, 2022) injects learned biases in attention weights calculation to incorporate hierarchical document structure and produces better summaries for long documents. It shows the importance of hierarchical document structure for long document understanding. However, we highlight the section-level structural relations such as parent-child and sibling, instead of token-level path lengths and level differences on the document structure graph.

3 Preliminary

We study the task of conditional QA over long documents (LDCQA), where the answers are only applicable when certain conditions apply. The model learns to find answers to the question from the long context and additionally performs logical reasoning over the conditions to check whether the answers are eligible. If the answers require additional conditions to be satisfied, the model identifies these unsatisfied conditions as well. Formally, the input to the model includes a question $q = [q_1; q_2; \cdots; q_m]$ coupled with a document $d = [d_1; d_2; \cdots; d_n]$, where m and n denotes the length of the question and context. In our LDCQA setting, the length n can be larger than 10K. The model outputs a list of answers coupled with corresponding conditions $\{(a_1, \{c_1^{(1)}; \cdots; c_{k_1}^{(1)}\}); \cdots; (a_i, \{c_1^{(i)}; \cdots; c_{k_i}^{(i)}\}); \cdots; (a_L, \{c_1^{(L)}; \cdots; c_{k_L}^{(L)}\})\}$, where $L \geq 0$ denotes the number of answers and $k_i \geq 0$ denotes the number of conditions for i-th answer.

4 Methodology

As shown in Figure 2, our approach includes 4 modules: PLM based contextual encoder, sentence-level discourse graph encoder, section-level structure graph encoder, fusion and decoding. First we encode each document section respectively to obtain contextual representations. Then we proceed sentence interaction using parsed discourse relations for each section. Then we conduct information propagation on the structure graph. Finally, we integrate 3 levels of section representations with token representations to jointly generate the answers and conditions.

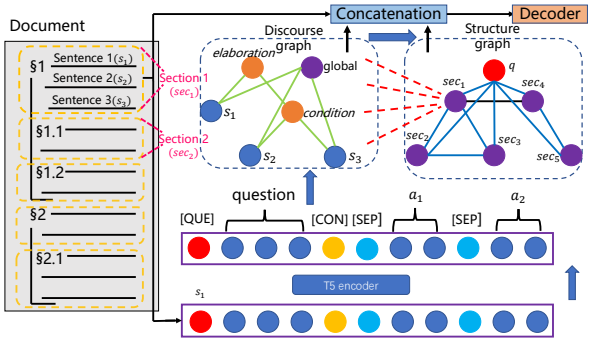
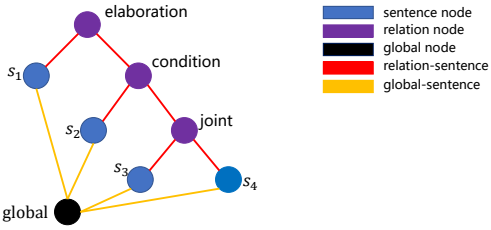


Figure 2: Method Overview. Our approach consists four main components: contextual encoder, sentence-level discourse graph encoder, section-level structure graph encoder, fusion and decoding. The hierarchical graph does information propagation from bottom to up. The "*elaboration*" and "*condition*" denote two types of discourse relations defined in (Carlson and Marcu, 2001).



Context: s_1[Standard route] s_2[Apply by the standard route if all the following are true:] s_3[You have been diagnosed with gender dysphoria] s_4[You have lived in your acquired gender for at least 2 years]

Figure 3: Discourse graph of section 1.1 in Figure 1. There are 3 types of nodes: sentence node, relation node and question node.

4.1 Pre-processing

Document Segmentation We segment the document into different levels of sections by the heading tags in the document. Each pair of headings such as "<h1>" and "</h1>", embraces the section title and is followed by a continuous chunk of contexts until the next pair of headings. We concatenate the title and the context chunk as the contents of one section. The hierarchy of headings is applied to build the document structure graph. Specifically, we add edges between parent sections and their child sections, as well as between sibling sections.

Discourse Parsing Considering the ground-truth discourse tree is not provided, we utilize a pre-trained discourse parser (Yu et al., 2022) for each section to decide the dependencies between sentences and the corresponding relation types. The parser does discourse parsing based on rhetorical structure theory (RST) (Mann and Thompson, 1988; Taboada and Mann, 2006) and utilizes 18 simplified coarse-grained relations such as *elabora-*

tion, *circumstance*, *condition*, and etc (Carlson and Marcu, 2001; Zhang et al., 2021; Yu et al., 2022). The discourse tree contains 2 types of nodes: relation nodes and leaf nodes. In our setting, the leaf node denotes a sentence in the section and the relation node identifies the relation type between two continuous text spans. We add a global node connected with every sentence node, so the discourse tree is converted into a discourse graph for each document section. The discourse graph of section 1.1 in Figure 1 is shown in Figure 3, which includes 4 relation types: *elaboration*, *condition*, *joint*.

4.2 Contextual Encoder

Our generative model for conditional QA is based on a sequence-to-sequence pretrained language model such as T5 or BART. The model takes the concatenation of question and context as the input and derives the contextual representations. Specifically, each document section is concatenated with the question and processed independently from other sections by the encoder. We add special tokens "[QUE]“ and "[CON]" before the question and context, as well as "[SEP]" to separate each sentence in the document section. Because we encode one section at a time, our approach is scalable to long documents with many sections.

4.3 Sentence-level Discourse Graph Encoder

For each document section, we build a discourse relation graph to incorporate discourse relational information between sentences within the section. We utilize RST discourse parser to derive the discourse graph and add a global node to represent the full section. We add edges between the global node and the leaf nodes, i.e., sentence nodes to increment the information flow among sentences. For a section composing n sentences, we initial the representation of global node as the hidden state h_0 corresponding to “[CON]" from contextual encoder, and the representation of t-th sentence node as the hidden state corresponding to the t-th “[SEP]" token h_t, $1 \leq t \leq n$. We apply GAT to do information propagation and derive the discourse relation enhanced representations:

$$\mathbf{a_{ij}} = \mathbf{MLP}([\mathbf{h_i} : \mathbf{h_j}]) \quad (1)$$

$$\alpha_{ij} = \frac{\exp(\mathbf{LeakyRelu}(\mathbf{a_{ij}}))}{\sum_{j' \in N(i)} \exp(\mathbf{LeakyRelu}(\mathbf{a_{ij'}}))} \quad (2)$$

$$\hat{\mathbf{h}}_\mathbf{i} = \sigma(\sum_{j \in N(i)} \alpha_{ij} \mathbf{W} \mathbf{h_i}) \quad (3)$$

where $N(i)$ denotes the neighbour nodes of node i, $1 \leq i \leq n$ and σ denotes activating function. We take the final representation of global node as the section representation that incorporates the discourse relational information.

4.4 Section-level Document Graph Encoder

We construct a node for each document section and add the question node in the structure graph. Section nodes are connected with their parent section and child sections. These parent-child edges encode the information that the child section depicts a specific aspect about the parent section. Section nodes at the same level which share the same parent section are connected and these sibling edges incorporate the information that these sections elaborate parallel and relevant aspects of the parent. Additionally, we connect the question node with each section node to enhance the information flow between the question and contexts. We initial the question representation q as the hidden state corresponding to the “[QUE]" obtained in the contextual encoder module. The initialization of section nodes comes from the representations of global nodes in corresponding discourse graphs. Similarly, we produce information transmission on the structure graph by GAT network and obtain the structure-aware section representations: $[\mathbf{q}'; \mathbf{h}'_\mathbf{1}; \mathbf{h}'_\mathbf{2}; \cdots; \mathbf{h}'_\mathbf{N}] = \mathbf{GAT}([\mathbf{q}; \hat{\mathbf{h}}_\mathbf{1}; \hat{\mathbf{h}}_\mathbf{2}; \cdots; \hat{\mathbf{h}}_\mathbf{N}])$, where N denotes the number of sections in the document.

4.5 Fusion and Decoding

Considering that $\{\mathbf{h_i}, 1 \leq i \leq n\}$, $\{\hat{\mathbf{h}}_\mathbf{i}, 1 \leq i \leq n\}$, $\{\mathbf{h}'_\mathbf{i}, 1 \leq i \leq n\}$ respectively contain the contextual information, discourse relational information and document structure information, we concatenate the token representations of question and contexts, as well as 3 levels of section representations sequentially as follows: $[\mathbf{t}^\mathbf{q}_\mathbf{1}; \mathbf{t}^\mathbf{q}_\mathbf{2}; \cdots; \mathbf{t}^\mathbf{q}_\mathbf{Q}; \mathbf{t}^\mathbf{c}_\mathbf{1}; \mathbf{t}^\mathbf{c}_\mathbf{2}; \cdots; \mathbf{t}^\mathbf{c}_\mathbf{C}; \mathbf{h_1}; \mathbf{h_2}; \cdots; \mathbf{h_N}; \hat{\mathbf{h}}_\mathbf{1}; \hat{\mathbf{h}}_\mathbf{2}; \cdots; \hat{\mathbf{h}}_\mathbf{N}; \mathbf{h}'_\mathbf{1}; \mathbf{h}'_\mathbf{2}; \cdots; \mathbf{h}'_\mathbf{N}]$, where Q, C, N denote the number of question tokens, context tokens and document sections. Then we pass them into PLM decoder to generate the sequence shaped as “$\cdots$ [ANS] a_i [CON] $c^{(i)}_1$ $\cdots$ [CON] $c^{(i)}_{N(i)}$ $\cdots$" where a_i and $c^{(i)}_j$ denote the i-th answer and the j-th condition of the i-th answer, “[ANS]" and “[CON]" are special tokens added into PLM tokenizer. Our model is optimized by the cross-entropy loss between the predicted sequence and

	train	dev	test	all
documents	436	59	139	652
questions	2338	285	804	3427
length	2050	2142	2324	2093

Table 1: Statistics about the ConditionalQA dataset. The row "documents" denotes the number of documents in different parts of the dataset, and the row "length" denotes the average document lengths.

	Type	Number
Answer type	yes / no	1751
	extractive	1527
Condition type	deterministic	2475
	conditional	803
Answer number	single	2526
	multiple	752
	not answerable	149

Table 2: Statistics about the question types in ConditionalQA dataset. The row "deterministic" denotes the number of cases which have no unsatisfied conditions. The row "conditional" denotes the number of cases which have unsatisfied conditions to identify.

ground truth:

$$L = -\log p(r|q, C) \tag{4}$$

$$= -\sum_{i=1}^{L} \log p(r_i|q, C, r_{<i}) \tag{5}$$

where $r = \{(a_i, c_i)\}_{i=1}^{L}$, a_i and c_i denote the i-th answer and condition.

5 Experiments

5.1 Dataset

ConditionalQA dataset is a challenging benchmark on conditional QA over long documents (Sun et al., 2022). There are 3427 questions in ConditionalQA and the average length of documents is larger than 2K by Table 1. Table 2 shows it contains different types of questions such as yes/no questions, free-form extractive questions, questions with multiple answers and not-answerable questions. Many questions in ConditionalQA are deterministic where the conditions needed have been satisfied in the question. It poses difficulties for the model to locate the conditions needed to answer the question and check the satisfaction of these conditions.

5.2 Evaluation Metrics

The predictions are evaluated using two sets of metrics: EM/F1 and conditional EM/F1. EM/F1 are the traditional metrics that measure the predicted answer spans. The ConditionalQA dataset introduced another metric, conditional EM/F1, that jointly measures the accuracy of the answer span and the unsatisfied conditions. As defined in the original paper (Sun et al., 2022), the conditional EM/F1 is the product of the original answer EM/F1 and the EM/F1 of the predicted unsatisfied conditions. The conditional EM/F1 is 1.0 if and only if the predicted answer span is correct and all unsatisfied conditions are found. If there is no unsatisfied condition, the model should predict an empty set.

5.3 Baselines

We compare our approach with 3 strong baselines on LDCQA.

ETC (Ainslie et al., 2020) applies global-local attention mechanism between global and local tokens, and enables the model scale to long inputs. However, the fully connected topology of token graphs cannot capture the natural structure of the document.

DocHopper (Sun et al., 2021) highlights the structural information that a passage contains consecutive and relevant information, and retrieves information by jointly sentence and passage level. However, the natural structural information between passages is ignored,

FID (Izacard and Grave, 2021) independently encodes different passages and concatenates the representations in the decoder only, which decreases calculation cost and improves performance for QA on long documents. However, the natural structure of documents and discourse information in each section are neglected.

5.4 Experimental details

Following FID (Izacard and Grave, 2021), we utilize pretrained model T5-base as our backbone. The information propagation step for discourse graphs and the document structure graph are set to 2. We optimize all models with Adam optimizer, where the initial learning rate is set to 1e-4 and the dropout rate is set to 0.1. The nuclearity of discourse relations distinguishes the different logical roles of two spans (Carlson and Marcu, 2001), so we add the nuclearity label produced by our discourse parser to each relation node in the discourse graphs. We focus more on formal texts on websites such as news, policies and articles (Huang

	Yes/No		Extractive		Conditional		Overall	
	EM / F1	w/ conds	EM / F1	w/ conds	EM / F1	w/ conds	EM / F1	w/ conds
majority	62.2 / 62.2	42.8 / 42.8	– / –	– / –	– / –	– / –	– / –	– / –
ETC	63.1 / 63.1	47.5 / 47.5	8.9 / 17.3	6.9 / 14.6	39.4 / 41.8	2.5 / 3.4	35.6 / 39.8	26.9 / 30.8
DocHopper	64.9 / 64.9	49.1 / 49.1	17.8 / 26.7	15.5 / 23.6	42.0 / 46.4	3.1 / 3.8	40.6 / 45.2	31.9 / 36.0
FID	64.2 / 64.2	48.0 / 48.0	25.2 / 37.8	22.5 / 33.4	45.2 / 49.7	4.7 / 5.8	44.4 / 50.8	35.0 / 40.6
SDHG	**67.4 / 67.4**	**50.2 / 50.2**	**29.2 / 42.0**	**25.4 / 37.0**	**48.3 / 52.3**	**5.9 / 7.6**	**47.4 / 53.2**	**37.2 / 42.5**

Table 3: Experimental results on ConditionalQA test set. The "EM/F1" columns report the original EM/F1 metrics that are only evaluated on the answer span. The "w/ conds" column denotes the conditional EM/F1 metric discussed in §5.2. The baseline results are obtained from (Sun et al., 2022). The row "majority" denotes always predicting "yes" without conditions. Our approach significantly outperforms FID, where p-values of EM and F1 are smaller than 0.001.

et al., 2021), which are abundant in real-world scenarios. It also provides a direction for unstructured texts to first segment into different sections (Cho et al., 2022) and apply our structure-discourse aware model.

5.5 Results

The results of different approaches are presented in Table 3. Our approach outperforms all the existing methods on ConditionalQA, achieving the new state-of-the-art. It is efficient to introduce natural document structure and discourse relations into conditional QA on long documents. We outperform the strong baseline FID by 3.0 EM score and 2.4 F1 score in answer measuring, 2.2 EM score and 1.9 F1 score in joint answer-condition measuring. On different types of questions, such as yes/no questions and free-form extractive questions, our model outperforms FID by over 3.2 EM and F1 score in answer measuring, as well as over 2.2 EM and F1 score in jointly answer and condition measuring. It demonstrates the robust improvement of our structure and discourse aware framework in different types of questions on both answer and condition measuring.

6 Analysis

In this part, we do 3 ablation studies to evaluate the efficiency of 3 levels of section representations in section 4.5. Then we probe our performance on long and complex documents. Moreover, we explore the role of accurate document structures and discourse relations in document sections. Finally, we take an example from ConditionalQA dataset to show the efficiency of our structure and discourse aware hierarchical framework.

6.1 Ablation Study

In our fusion module, we concatenate three levels of section representations: original contextual representations, discourse-aware representations, and document structure-aware representations with the token representations. In this part, we conduct 3 ablation experiments on the development set of ConditionalQA to evaluate their respective efficiency.

	Overall	
	EM / F1	w/ conds
-contextual	**48.2** / 55.6	37.9 / 45.3
-discourse	45.0 / 53.7	36.4 / 44.5
-structure	47.8 / 53.6	**40.2** / 45.6
SDHG	47.9 / **56.6**	38.3 / **46.6**

Table 4: Ablation Results on development set of ConditionalQA by overall EM and F1 metrics for answer and condition prediction.

Do contextual representations of sections matter? To evaluate the efficiency of contextual representations in SDHG, we remove the list of section representations $\{h_i, 1 \leq i \leq N\}$ from section 4.5. By Table 4, the performance of this ablation will gain 0.3 EM score and drop 1.0 F1 score on measuring answers, meanwhile drop 0.4 EM score and 1.3 F1 score on jointly measuring answers and conditions. It shows the original representations of document sections from PLM contain contextual section-level information which are important to our model for conditional QA on long documents.

Do discourse relations between sentences in each section matter? In this ablation, we remove the discourse graph for each document section from our hierarchical framework. Specifically, we take the original contextual representations of sections from PLM to initial the normal node in document structure graph, and concatenate the contextual and document structural representations to the decoder. As shown in Table 4, this ablation drops 2.9 EM score and 2.9 F1 score on answer measuring, as well as 1.9 EM score and 2.1 F1 score on jointly

	Conditional w/ conds	
Group	1	2
Avg. len.	1182	3094
FID	6.1 / 8.4	0.7 / 5.2
SDHG	6.5 / 7.6	5.2 / 7.1

Table 5: Performance on 2 groups of cases in ConditionalQA development set classified by document length , the row "Avg. len." denotes the average length of document for different case groups.

	Conditional w/ conds	
Group	1	2
Avg. # sect.	12	30
FID	6.4 / 8.7	0.7 / 5.0
SDHG	6.7 / 7.8	5.1 / 6.7

Table 6: Performance on 2 groups of cases in ConditionalQA development set classified by section number , the row "Avg. # sect." denotes the average number of document sections for different case groups.

answer and condition measuring. It demonstrates the discourse relations information between sentences enhance the logical interaction between the question and relevant conditions for our model.

Does document structural information matter? In this ablation, we remove the document structure graph from our model to probe the efficiency of natural structural information. Concretely, we only concatenate the contextual representations and discourse-aware representations with token representations to the decoder. As shown in Table 4, this ablation drops 0.1 EM score and 3.0 F1 score on answer measuring, as well as gains 1.9 EM score and drops 1.0 F1 score on jointly answer and condition measuring. The natural structural information that sibling sections elaborate parallel and relevant aspects of the parent section helps our model locate relevant conditions across different document sections, thus improving the prediction of answers and unsatisfied conditions.

6.2 Capacity for Long and Complex Documents

In this part, we classify the cases of ConditionalQA development set into 2 groups respectively based on the quantile of 3 metrics: the length of the document, the number of document sections, the number of sentences in the document. The larger number of document sections reflects the larger size and complexity of document structure graph, while the larger number of sentences in the document embodies the larger size and complexity of discourse graphs. We focus on the cases with unsatisfied conditions, so we choose to evaluate by conditional jointly answer and condition measuring.

As shown in Table 6, our model gains 0.3 EM score on group 1 cases and gains 4.4 EM score on group 2 cases compared with baseline FID. It shows with more document sections, the structure graph contains more structural information between document sections, which enhances our capacity to retrieve the answers and conditions across sections. As shown in Table 7, our gains 0.3 EM score on group 1 cases and gains 4.2 EM score on group 2 cases compared with FID. With larger size of discourse graph, the more abundant discourse relations between sentences stimulate the logical interaction between the question and conditions, which helps our model understand the context and predict the unsatisfied conditions. As shown in Table 5, our model gains 0.4 EM score on group 1 and gains 4.5 EM score on group 2. With longer documents, our model incorporates richer information from the document structure and discourse relations into conditional QA. It demonstrates the capacity of our model for long and complex documents.

6.3 Role of Accurate Structure Graph and Discourse Relations

In this part, we explore if our model architecture can truly distinguish the information of document natural structure and discourse relations in each section. In exploration 1, we flatten the hierarchical document structure and consider all the sections at the same level. In this way, the structure graph is fully connected and all the nodes propagate information with each other. In exploration 2, we disrupt the discourse relations between sentences in each document section. Considering "*elaboration*" is the most discourse relation in the dataset, we disturb the discourse graph by assigning all relation nodes to be "*elaboration*". In this way, the model treats all the sentences as progressive elaborations and ignores the original logical relations between sentences.

As shown in Table 8, with flattened document structure, the structural information that child sections describe parallel and relevant aspects of the parent section is lost. As a result, this exploration drops 2.6 EM score and 4.1 F1 score on answer measuring, as well as 0.1 EM score and 1.8

	Conditional w/ conds	
Group	1	2
Avg. # sent.	60	156
FID	6.6 / 9.0	0.7 / 4.8
SDHG	6.9 / 8.0	4.9 / 6.6

Table 7: Performance on 2 groups of cases in ConditionalQA development set classified by sentence number , the row "Avg. # sent." denotes the average number of sentences in the document for different case groups.

	Overall	
	EM / F1	w/ conds
flatten structure	45.3 / 52.5	38.2 / 44.8
all elaboration	46.4 / 53.9	38.3 / 45.3
SDHG	**47.9 / 56.6**	**38.3 / 46.6**

Table 8: Two explorations for our perceptual ability to document structure and discourse relations.

F1 score on jointly answer and condition measuring. Furthermore, compared with ablation model 3, which abandons the whole document structure information, this exploration drops 1.4 EM score and 1.1 F1 score on answer measuring, as well as 2.0 EM score and 0.8 F1 score on jointly answer and condition measuring. It demonstrates the fully connected structure graph (Nie et al., 2022) connect many irrelevant document sections, which introduces noisy information chaos into the model and undermines the overall performance.

As shown in Table 8, with all the discourse relations between sentences set to "*elaboration*", the logical information of other discourse relations such as "*condition*" and "*joint*" are abandoned, this exploration drops 1.5 EM score and 2.7 F1 score on answer measuring, as well as 1.3 F1 score on jointly answer and condition measuring. The comprehensive discourse relations contain abundant logical information between sentences, which improves the condition locating and reasoning for our model.

Moreover, compared with ablation model 2, which removes all the discourse information, this exploration gains 1.4 EM score and 0.2 F1 score on answer measuring, as well as 1.9 EM score and 0.8 F1 score in jointly answer and condition measuring. Because "elaboration" accounts for the largest proportion in discourse relations, the discourse graph encoder helps this exploration better understand progressive sentences, improving the prediction for answers and conditions. It demonstrates that our model has the ability to capture correct discourse relational information into answer and condition prediction. Considering the pretrained discourse parser we used does not provide the golden parsing result, our model shows promising better performance with more efficient parsing techniques.

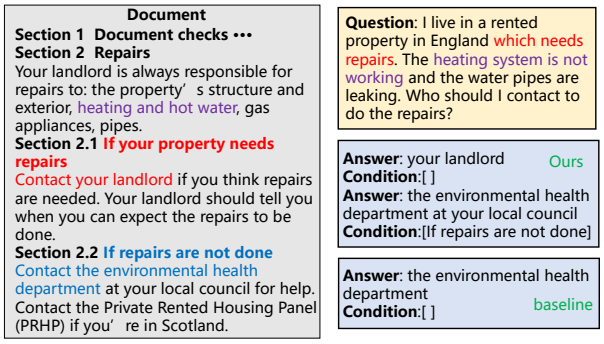


Figure 4: An example from ConditionalQA dataset, where we obtain the correct answers and conditions but the baseline FID fails.

6.4 Case Study

In this part, we take an example from ConditionalQA dataset to show the efficiency to incorporate natural document structure and discourse relations between sentences. As shown in Figure 4, the document discusses the private renting in UK and the question asks for the approach to do the repairs. Section 2.1 and 2.2 are two child sections of section 2, and they describe two different but relevant ways to ask for repairs mentioned in section 2. The structural relations among section 2, 2.1, 2.2 allow the model to reason from "heating and hot water" in section 2 to the two routes to ask for repairs in section 2.1 and 2.2, retrieving different answers and corresponding conditions across sections. Moreover, the discourse relation "*condition*" between "your property needs repairs" and "contact your landlord", as well as "repairs are not done" and "contact the environmental health department", enable our model to locate different conditions corresponding to each answer. Because the question satisfies the condition "property needs repairs", the answer "contact your landlord" has no unsatisfied conditions, but the answer "contact the environmental health department" has to be outputted with its corresponding condition. However, without document structure information, the baseline FID only retrieves section 2.2, ignoring the parallel section 2.1; without discourse relations, FID neglects the condition corresponding to the answer "contact the environmental health department". Therefore, the above demonstrates the efficiency of our structure-discourse hierarchical graph reasoning framework.

7 Conclusion

In this paper, we propose a novel and efficient framework with hierarchical section-level structure graph and sentence-level discourse graph for conditional QA on long documents. We incorporate the natural document structure and logical discourse relations to locate answers as well as unsatisfied conditions by cross-sections retrieving and logical reasoning. We conduct experiments on the benchmark dataset in this field and our approach outperforms all the existing methods.

Limitations

We showed that our model is efficient in handling conditional QA on long documents with hierarchical reasoning framework. However, our discourse graphs for each document section are constructed based on the prediction of the pretrained discourse parser. There is promising improvement for our approach by use of more efficient discourse parsers.

References

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. Etc: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284.

Shuyang Cao and Lu Wang. 2022. Hibrids: Attention with hierarchical biases for structure-aware long document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 786–807.

Lynn Carlson and Daniel Marcu. 2001. Discourse tagging reference manual. *ISI Technical Report ISI-TR-545*, 54(2001):56.

Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Fei Liu, and Dong Yu. 2022. Toward unifying text segmentation and long document summarization. *arXiv preprint arXiv:2210.16422*.

Yifan Gao, Chien-Sheng Wu, Jingjing Li, Shafiq Joty, Steven CH Hoi, Caiming Xiong, Irwin King, and Michael Lyu. 2020. 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.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. *arXiv preprint arXiv:2104.02112*.

Gautier Izacard and Édouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880.

Yanyan Jia, Yuan Ye, Yansong Feng, Yuxuan Lai, Rui Yan, and Dongyan Zhao. 2018. Modeling discourse cohesion for discourse parsing via memory network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 438–443.

Armanda Lewis. 2022. Multimodal large language models for inclusive collaboration learning tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 202–210, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Yuxiang Nie, Heyan Huang, Wei Wei, and Xian-Ling Mao. 2022. Capturing global structural information in long document question answering with compressive graph selector network. *arXiv preprint arXiv:2210.05499*.

Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2021a. 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.

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

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

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. *arXiv preprint arXiv:1809.01494*.

Holger Schauer. 2000. From elementary discourse units to complex ones. In *1st SIGdial Workshop on Discourse and Dialogue*, pages 46–55.

Zhouxing Shi and Minlie Huang. 2019. A deep sequential model for discourse parsing on multi-party dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7007–7014.

Haitian Sun, William Cohen, and Ruslan Salakhutdinov. 2022. Conditionalqa: A complex reading comprehension dataset with conditional answers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3627–3637.

Haitian Sun, William W Cohen, and Ruslan Salakhutdinov. 2021. End-to-end multihop retrieval for compositional question answering over long documents.

Maite Taboada and William C Mann. 2006. Applications of rhetorical structure theory. *Discourse studies*, 8(4):567–588.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Nan Yu, Meishan Zhang, Guohong Fu, and Min Zhang. 2022. Rst discourse parsing with second-stage edu-level pre-training. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4269–4280.

Longyin Zhang, Fang Kong, and Guodong Zhou. 2021. Adversarial learning for discourse rhetorical structure parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3946–3957.

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*, pages 2310–2320.

A Appendix

A.1 Reproducibility Checklist

Did you discuss any potential risks of your work? The methods in this work do not pose any ethical or security related risks.

Did you discuss the license or terms for use and/or distribution of any artifacts? ConditionalQA is distributed under a CC BY-SA 4.0 License (https://creativecommons.org/licenses/by-sa/4.0/).

Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? We use conditionalQA following the instructions of its creator (https://haitian-sun.github.io/conditionalqa/).

Did you discuss the steps taken to check whether the data that was collected/used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it? By the creator of the dataset, this dataset does not include any privacy information.

Did you provide documentation of the artifacts? The document of the dataset can be found in https://haitian-sun.github.io/conditionalqa/.

Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used? There are about 252 million parameters in our model. We run experiments on one Tesla v100 gpu and the training time is about 5 hours.

Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values? We use F1 for jointly answer and condition measuring on the development set to choose the hyperparameter. The specific values are in section 5.4 Experimental Details.

If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)? We the existing packages Pytorch and NLTK to implement our model.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
In section Limitations.

☑ A2. Did you discuss any potential risks of your work?
In the subsection Reproducibility Checklist of section Appendix.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
In section Abstract and Introduction.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

In section 5.1 Dataset.

☑ B1. Did you cite the creators of artifacts you used?
In section 5.1 Dataset.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
In the subsection Reproducibility Checklist of section Appendix.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
In the subsection Reproducibility Checklist of section Appendix.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
In the subsection Reproducibility Checklist of section Appendix.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
In the subsection Reproducibility Checklist of section Appendix.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
We report the relevant statistics in section 5.1 Dataset.

C ☑ Did you run computational experiments?

In section 5 Experiments.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
In the subsection Reproducibility Checklist of section Appendix.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
In the subsection Reproducibility Checklist of section Appendix.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
In section 5 Experiments.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
In the subsection Reproducibility Checklist of section Appendix.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.