HIBRIDS: Attention with Hierarchical Biases for Structure-aware Long Document Summarization

Shuyang Cao and **Lu Wang**
Computer Science and Engineering
University of Michigan
Ann Arbor, MI
{caoshuy, wangluxy}@umich.edu

Abstract

Document structure is critical for efficient information consumption. However, it is challenging to encode it efficiently into the modern Transformer architecture. In this work, we present HIBRIDS, which injects Hierarchical Biases foR Incorporating Document Structure into the calculation of attention scores. We further present a new task, hierarchical question-summary generation, for summarizing salient content in the source document into a hierarchy of questions and summaries, where each follow-up question inquires about the content of its parent question-summary pair. We also annotate a new dataset with $6,153$ question-summary hierarchies labeled on long government reports. Experiment results show that our model produces better question-summary hierarchies than comparisons on both hierarchy quality and content coverage, a finding also echoed by human judges. Additionally, our model improves the generation of long-form summaries from lengthy government reports and Wikipedia articles, as measured by ROUGE scores.

1 Introduction

Document structure facilitates information searching, reading comprehension, and knowledge acquisition by providing an informative overview of the content (Guthrie et al., 1991; Meyer et al., 1980; Taylor and Beach, 1984; Shavelson, 1974; Jonassen, 1988). Specifically, for summarization, its utility is twofold: (1) *Source* document structures, such as sections and paragraphs, can be instructive for summary generation (Cohan et al., 2018; Celikyilmaz et al., 2018; Zhang et al., 2019); (2) Structures in *output* summaries, e.g., timelines (Shahaf et al., 2012; Wang et al., 2015) or aspects (Angelidis and Lapata, 2018), can also ease content understanding.

Nonetheless, state-of-the-art abstractive summarization systems, all built on the Transformer architecture (Zhang et al., 2020; Lewis et al., 2020), use attentions to estimate relations between pairwise tokens and largely ignore document structures. While hierarchical encoding has been investigated (Zhang et al., 2019; Balachandran et al., 2021), its need for training large amounts of additional parameters leads to increased memory footprint and thus limits the allowed input length. As for the output, the structure of single document summaries remains largely "flat", such as a list of aspects (Meng et al., 2021). We argue that it is imperative to develop systems that can output summaries with rich structures to support knowledge acquisition, which is especially critical for long documents that cover numerous subjects with varying details (Huang et al., 2021; Kryściński et al., 2021).

This work consists of two main objectives: (1) effectively informing summarization models of the source document's structure, and (2) presenting a new summarization task that produces hierarchically organized question-summary pairs to facilitate information consumption. To this end, we propose **HIBRIDS** (Hierarchical Biases foR Incorporating Document Structure).[1] We design learnable *hierarchical biases*, as part of the Transformer attention calculation, to adjust attention weights based on tokens' relative positions with regard to the document structure, inspired by the relative position method that modifies attention calculation (Raffel et al., 2020). Concretely, we leverage the natural structure of a document, i.e., section levels, to construct a *document structure tree* (Figure 2). Each learnable bias corresponds to the relation between a pair of sections, based on the distance between them in the structure tree. Intuitively, hierarchical biases adjust attention weights between tokens based on how conceptually close/distant their corresponding sections are, and they also enable summarizers to capture long-range

[1] Our code and newly collected data can be found at https://shuyangcao.github.io/projects/structure_long_summ.

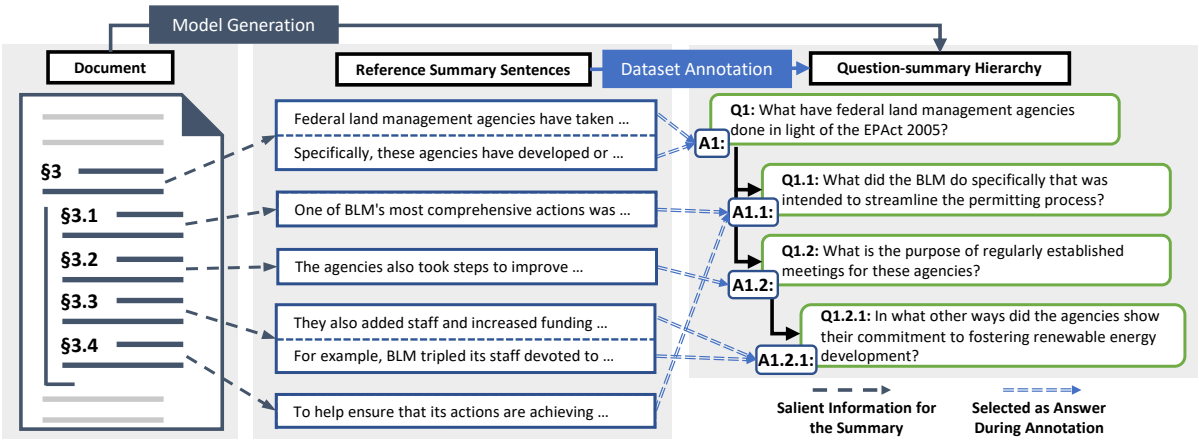

Figure 1: The question-summary hierarchy annotated for sentences in a reference summary paragraph. Summarization models are trained to generate the question-summary hierarchy *from the document*, which signifies the importance of encoding the document structure. For instance, to generate the follow-up question-summary pairs of Q1.1 and A1.1 from A1, it requires the understanding of both the content and the parent-child and sibling relations among §3, §3.1, and §3.4.

relatedness for better document understanding.

Furthermore, we design a new summarization task, **hierarchical question-summary generation**: Given a document, automatically generate questions and summaries that are organized hierarchically to lay out details for topics at different levels. As shown in Figure 1, each question asks about *salient* content of the document (to be summarized) and its child questions focus on content in the corresponding summary. This hierarchy not only exposes salient topics and their relations, but also allows readers to quickly identify aspects of interest to focus on. Our task design is inspired by the top-down knowledge learning process: People start by asking broad questions to acquire general knowledge, and then dive into details (Hintikka, 1981; Stede and Schlangen, 2004). Notably, as there is no available dataset with such annotations, we also label a new dataset, **GovReport-QS**, consisting of 6,153 question-summary (QS) hierarchies for summary paragraphs based on 1,714 reports from the GovReport dataset (Huang et al., 2021). Each summary paragraph contains 4.07 questions with an average QS hierarchy depth of 2.26 levels.

We first compare HIBRIDS with models that use structure-aware architectures (Rohde et al., 2021) and linear relative positions (Raffel et al., 2020). We conduct experiments on the hierarchical QS generation dataset using two setups: (1) generating a full hierarchy given the first question, and (2) generating follow-up questions given a QS pair. Automatic evaluation shows that our model produces better follow-up questions and summaries than comparisons, while also achieving better or comparable content coverage of full summaries, when compared with a hierarchical model (Rohde et al., 2021) that learns 2M more parameters. In human evaluation, HIBRIDS is considered to build better hierarchies that require fewer manual corrections with more relevant summaries. We further test on the long document summarization task to produce full summaries using GovReport and *a newly collected dataset* consisting of about 21k high-quality biographies with summaries from Wikipedia. Again, our system summaries obtain uniformly higher ROUGE scores than comparisons, demonstrating the generalizability of HIBRIDS.

2 Related Work

Document Structure-aware Summarization. Structural information has long been leveraged for identifying summary-worthy content, including discourse structures labeled by experts (Marcu, 1997) or automatic parsers (Hirao et al., 2013; Durrett et al., 2016; Xu et al., 2020), and topical structures derived from lexical chains (Barzilay and Elhadad, 1999) or probabilistic models (Barzilay and Lee, 2004; Daumé III and Marcu, 2006). Natural structures of documents, such as sentences, have been used for pre-training a sentence-level encoder (Zhang et al., 2019) or inducing dependencies among them (Liu et al., 2019) for building extractive summarization systems. Based on separately encoded paragraphs, deep communica-

tion agents (Celikyilmaz et al., 2018) and inter-paragraph attentions (Liu and Lapata, 2019) are employed to build abstractive summarization models by exchanging information from different paragraphs. Using section structures, Cohan et al. (2018) design a section-level encoder based on the output of a word-level encoder for long document summarization. Nevertheless, multi-level encoders are more expensive since they introduce a significant amount of parameters and add extra padding at multiple levels of model design. By contrast, HIBRIDS effectively informs models of document structure by introducing a novel bias term in attention calculation among tokens, which only introduces a small number of learnable parameters.

Long Document Summarization also benefits from the inclusion of document structure information. For example, extractive summarization methods are developed to combine section-level and sentence-level information encoded by multi-level encoders (Xiao and Carenini, 2019) and include longer context via sliding encoding over sections (Cui and Hu, 2021). Recent work on summarizing long documents focuses on designing efficient Transformers with sparse attentions to produce abstractive summaries for long documents in an end-to-end fashion (Beltagy et al., 2020; Zaheer et al., 2020; Huang et al., 2021). However, they all ignore the natural structure of long documents, such as sections and subsections. Based on a simple design, HIBRIDS can be integrated into any efficient Transformer seamlessly for incorporating document structure information.

Generating question-answer (QA) pairs has been studied to facilitate information seeking within documents, mainly for producing questions that can be addressed by short phrases (Du and Cardie, 2018; Liu et al., 2020). Prior work mostly focuses on improving QA pair relevance by leveraging additional QA systems (Sachan and Xing, 2018), measuring roundtrip consistency (Alberti et al., 2019), or refining questions iteratively (Qu et al., 2021). Generating a two-level hierarchy of QA pairs from a given paragraph is investigated by Krishna and Iyyer (2019). Our work is different in at least three aspects. First, our goal is to provide a structured summary that focuses on the *salient content* of the given document, rather than creating questions about any generic information, as done in most QA data construction (Rajpurkar et al., 2016; Choi et al., 2018). Second, our GovReport-QS data

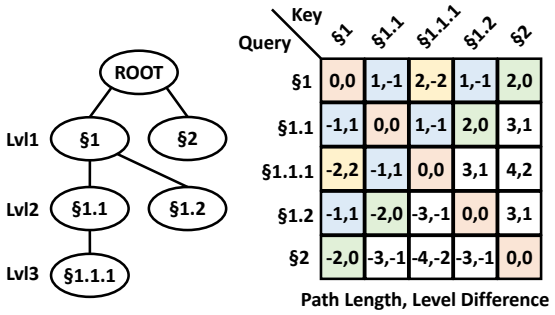

Figure 2: Example path lengths and level differences (right) that encode the relative positions with regard to the document tree structure (left). Each query/key represents a block of tokens that belong to the same section. We highlight important section relations: self, parent-child, ancestor-descendant (other than parent-child), and sibling. From §1 (level 1) to §1.2 (level 2), the level difference is **-1** and path length is **1** since §1 occurs *before* §1.2. When looking back from §1.2 to §1, both numbers' signs are flipped, i.e, (**1**, **-1**).

concerns richer hierarchies for presenting content in long documents, e.g., 23.6% of our hierarchies contain at least *three levels*. Our parent-child pairs also cover diverse relations, e.g., adding explanations or expanding the topics, beyond asking about specific details as done in Krishna and Iyyer (2019). Third, our questions are designed to be open-ended and grounded in the given document, so our new task is more suitable for summarization models.

3 HIBRIDS with Hierarchical Biases

In this section, we first introduce how relative positions are defined over the document structure tree. Then we present HIBRIDS, which can be included in encoder self-attentions or decoder cross-attentions to adjust the attention scores based on tokens' relative positions.

3.1 Document Structure Tree and Tree-based Relative Positions

We first construct a *document structure tree* (Figure 2, left), by leveraging the natural structure of sections and subsections (henceforth sections) in documents, which is available in our experiment data extracted from government reports and Wikipedia articles. We then capture the relative position between pairwise tokens x and y in two different sections, e.g., S_x and S_y, with two tree-based measures. (1) $\texttt{PathLen}(x, y)$: the length of the shortest path from S_x to S_y; (2) $\texttt{LvlDiff}(x, y)$: the level difference from S_x to S_y. $\texttt{PathLen}$ is

designed to be *asymmetric* to capture content ordering, i.e., its value is positive if S_x appears before S_y in the document, and vice versa. Examples are displayed in Figure 2.

3.2 Attentions with Hierarchical Biases

The design of HIBRIDS is based on a lookup table $B[\cdot,\cdot]$: Each item in it corresponds to a learnable hierarchical bias defined by path length and level difference, which is then used to bias the attention calculation for tokens in different sections. Each head maintains its own lookup table B.

We first apply HIBRIDS to Transformer encoder self-attention computation, which is called **HIBRIDS-ENC**. Given the i-th query $\boldsymbol{q}_i$ and the matrix $\boldsymbol{K}$ formed by n keys for all input tokens, HIBRIDS adds a bias for each key, with respect to the i-th query, to attention calculation:

$$a_{ij} = \text{softmax}(\boldsymbol{q}_i \boldsymbol{K}^T + \boldsymbol{b}_i)_j \quad (1)$$

where the vector $\boldsymbol{b}_i = [b_{i1}, \ldots, b_{ij}, \ldots, b_{in}]$ contains the bias terms derived from our hierarchical biases as follows:

$$b_{ij} = B[\texttt{PathLen}(i,j), \texttt{LvlDiff}(i,j)] \quad (2)$$

where $\texttt{PathLen}(i,j)$ and $\texttt{LvlDiff}(i,j)$ are the path length and level difference between the sections that tokens i and j belong to. Note that b_{ij} varies among different heads. HIBRIDS-ENC guides tokens to attend to structurally related tokens during encoding.

We then apply HIBRIDS to decoder cross-attention calculation, named as **HIBRIDS-DEC**, to encourage more coherent generation by establishing better alignment with the source document. At the generation step t, the cross-attention weight to the j-th input token adjusted by bias b_{tj} is obtained similarly as in Eq. 1 with the following modification. We calculate b_{tj} as the weighted sum of the hierarchical biases for all input tokens (indexed with l) to the j-th token. The weight is chosen as the decoder's second last layer's cross-attention score between the t-th generated token and the l-th input token, which is shown to better capture word alignment (Garg et al., 2019; Cao and Wang, 2021a). b_{tj} is only applied to the decoder's last layer with the following formulation:

$$b_{tj} = \sum_l a^{crs}_{tl} \cdot B[\texttt{PathLen}(l,j), \texttt{LvlDiff}(l,j)] \quad (3)$$

where a^{crs}_{tl} is the decoder's second last layer's cross-attention weight for the generation step t to the l-th input token.

HIBRIDS$_S$ with Selected Relations. We further consider only keeping salient relations from the tree to reduce the number of parameters to learn, including *self* (same section), *parent-child*, *ancestor-descendant*, *sibling*, *neighboring in text*, and *within the same top-level section* (e.g., §1.1.1 and §1.2 are both in §1). In total, they account for 21.6% of all relation occurrences. The modified HIBRIDS$_S$ can also be applied to both encoder and decoder.

4 A New Task: Hierarchical Question-summary Generation

We introduce a new summarization task in this section: Given a document or several sections of a document, we aim to generate question-summary (QS) pairs that are organized hierarchically. As shown in Figure 1, this QS hierarchy lays out details for topics at multiple levels, with each child QS pair expanding the content of its parent. Our task is motivated by how human learns knowledge in a top-down fashion, where general knowledge is acquired first and details and in-depth content are explored later (Hintikka, 1981). This hierarchy proactively highlights the document structure, to further promote content engagement and comprehension (McKeown et al., 2009).

4.1 Question-summary Hierarchy Annotation Procedure

We first annotate a new dataset, GOVREPORT-QS, with hierarchical QS pairs, based on articles and corresponding summaries selected from the GOVREPORT dataset (Huang et al., 2021). As these documents and summaries have 9,409 and 553 words on average respectively, directly annotating full documents with a QS hierarchy presents a challenge. To address this, we ask annotators to create hierarchical questions for a selected summary paragraph and only allow them to select complete sentences from the summary paragraph as the corresponding answers. Each question created should be fully addressed by its answer and the answer should not contain information irrelevant to the question. For follow-up questions, they are encouraged to ask about specific details or issue questions that can yield summaries that elaborate from their parents. Annotators are also instructed to construct

hierarchies of as many levels as possible. Figure 1 demonstrates how hierarchical questions are created and how answer sentences are selected when annotating a report on the development of renewable energy.

To cover more documents and avoid collecting shallow hierarchies, each summary paragraph is annotated by one annotator and we select high-quality summary paragraphs for annotation based on heuristic rules, e.g., each paragraph should have at least 3 sentences and 70 words and an adequate level of abstractiveness as measured by normalized density of extractive fragments (Grusky et al., 2018) (with a threshold of < 0.15). Annotation instructions and details of paragraph selection are in Appendix A.

We hired 11 college students who are native English speakers to carry out the annotation tasks in multiple rounds. Feedback was provided to each annotator after each round. A finalization stage was conducted after collecting all annotations, where 4 high-quality annotators were asked to correct typos, remove factoid questions, and make minor adjustment to the hierarchies when errors were detected.

GovReport-QS Statistics. In total, 6,153 summary paragraphs are annotated with 25,055 QS pairs. On average, 4.07 QS pairs are created per summary paragraph, spanning 2.26 levels. 70.5% and 23.6% of paragraphs are annotated with *two* and *three* levels of questions, making our dataset a valuable benchmark for studying QS hierarchy generation, query-focused summarization, and question generation.

4.2 Aligning Summary Paragraphs with Document Sections

The QS hierarchies then become the target generation, and we construct inputs to our QS hierarchy generation system by mapping annotated summary paragraphs back to *sections* in source documents.

Concretely, we match each summary sentence to a document paragraph based on a combination of BERT-based, word overlap-based, and entity overlap-based similarities (details in Appendix A). All *sections* where matched paragraphs belong, along with the titles of their ancestor sections, are combined together to serve as the system input for generating the corresponding QS hierarchy, as demonstrated in Figure 1. The paired sections have an average length of 2,029, longer than documents in many standard summarization benchmarks.

5 Experiment Setups

5.1 Datasets and Tasks

We evaluate HIBRIDS on three different tasks with outputs of varying structures.

Task I: QSGen-Hier. Based on GovReport-QS, we first experiment with a setup where, given the aligned document sections and a root question, the model is expected to produce a summary that addresses the question as well as the rest of the hierarchy. To linearize a QS hierarchy for the Transformer sequential decoder, we concatenate its QS pairs following a depth-first traversal. Special tokens are inserted before each QS pair to indicate the change of its level from the previous QS pair: `[L↓]`, `[L↑]`, and `[L−]` indicate that the level has incremented, decremented, and not changed, respectively. For example, the sample hierarchy in Figure 1 can be formulated as: "*A1* `[L↓]` *Q1.1 A1.1* `[L−]` *Q1.2 A1.2* `[L↓]` *Q1.2.1 A1.2.1*". On this task, we divide our samples into train/dev/test splits with sizes of 4,878/644/631.

Task II: QSGen-ChildQ. Next, we leverage GovReport-QS for follow-up question generation: Given a QS pair and the aligned document sections, we aim to generate all child questions. With this setup, two samples can be created from the example in Figure 1. The first one takes as input *"Q1 A1"* and the aligned sections to generate *"Q1.1 Q1.2"*, whereas the other reads in *"Q1.2 A1.2"* and the aligned sections to produce *"Q1.2.1"*. Here we construct train/dev/test splits with sizes of 7,157/958/942.

Task III: Full Summary Generation. We also conduct experiments on GovReport to test HIBRIDS on generating long-form summaries for long inputs. We use the original data splits with 17,516/974/973 samples in train/dev/test sets. We further collect *a new dataset* from WikiProject Biography[2] (WikiBioSum) to perform biography summarization. After collecting all available biographies, we keep the ones with at least two levels of section hierarchy and preserve section structures of all levels. For each article, the paragraph before the first section is treated as the target summary, and the rest becomes the input. The finalized dataset has 20,833 pairs, divided into 18,751/1,041/1,041 samples for train/dev/test sets.

[2] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Biography

The average lengths of the input and output for WIKIBIOSUM are 3,478 and 1,266. Details of WIKIBIOSUM data collection and filtering procedures are in Appendix B.

We set the maximum input length to 5,120 for QSGen-Hier, QSGen-ChildQ, and full document summarization on WIKIBIOSUM. On GOVREPORT, the limit is set to 16,384.

5.2 Evaluation and Comparisons

Evaluation Metrics. We use ROUGE (Lin, 2004) for summarization evaluation and additionally report BLEU up to 4-gram (Papineni et al., 2002) for evaluating the generated questions.

We propose to evaluate the generated QS hierarchy against the reference hierarchy with F1 scores calculated as follows, inspired by labeled attachment score in dependency parsing (Zeman et al., 2017): We first map each generated QS pair to a reference QS pair following the highest sum of ROUGE-1 and ROUGE-2 scores between their summaries. After that, we consider two QS pairs with parent-child relation in the generated hierarchy. A *match* is established only when their mapped QS pairs have a parent-child or ancestor-descendant relation in the reference hierarchy. Precision can then be calculated based on the matching results. We further weight each match based on the sum of the ROUGE-1 and ROUGE-2 scores calculated over both parent and child summaries. Weighted recall and F1 are calculated similarly.

Comparisons. All tasks in this work involve long inputs. To allow efficient encoding, we use **LONGFORMER** (Beltagy et al., 2020) with a window size of 1024 as the base model, and fine-tune it for all systems and comparisons.

We first consider comparisons by adding special tokens to encode document structure: (1) **SECTOK** inserts a special token `[SEC]` at the start of each section. (2) **LVLSECTOK** further differentiates sections at varying levels using different tokens (e.g., `[SEC-L1]` for §1, `[SEC-L2]` for §1.1).

Based on LVLSECTOK, we build all HIBRIDS variants and other comparisons listed below:

• **HIERENC**: We implement the hierarchical model by Rohde et al. (2021), where we replace its sentence encoder with a section encoder of 12 layers to maintain section structures. Among all models, HIERENC requires the most architecture change and adds the most parameters to learn.

• **MULTITASK**: We also consider predicting the selected relations used by HIBRIDS$_S$ (§3) in a multi-task prediction setup with a bilinear classifier, operating on the representations of section tokens. We use equal weights for prediction loss and summarization loss.

• **TOKBIAS** uses linear relative position biases as in T5 (Raffel et al., 2020), which changes Eq. 2 to $b_{ij} = R[i-j]$ where $R[\cdot]$ is a lookup table with each item corresponding to a learnable bias for a given relative distance.

• **SECBIAS** replaces token-level linear distance in TOKBIAS with section-level linear distance.

Notably, LONGFORMER and models using special tokens have 4.59M parameters. HIBRIDS and models with linear relative position biases use about 4.60M parameters in total. On the other hand, HIERENC and MULTITASK modify the architecture and have 6.62M and 4.66M parameters, which is less efficient for learning compared with models that use bias terms to adjust attention calculation.

Model	Hier F1	Summary R1	Summary R2	Summary RL	Ques B4
LONGFORMER	12.67	42.34	16.18	37.60	10.00
SECTOK	12.86	42.67	16.34	38.01	10.02
LVLSECTOK	12.74	42.34	16.31	37.61	10.09
Structure-aware Comparisons					
HIERENC	11.77	**42.82**	16.32	**38.06**	9.89
MULTITASK	12.64	41.19	15.49	36.58	9.66
Models with Linear Bias					
TOKBIAS	12.43	42.58	16.41	37.71	10.06
SECBIAS	12.54	42.54	16.39	37.80	10.00
Our Models					
HIBRIDS-ENC	**13.26**	42.74	**16.55**	38.03	**10.16**
HIBRIDS$_S$-ENC	13.16	42.50	16.16	37.69	10.09
HIBRIDS-DEC	12.68	42.31	16.17	37.58	9.75
HIBRIDS$_S$-DEC	12.71	42.44	16.42	37.82	9.84

Table 1: Results for QSGen-Hier on GOVREPORT-QS. The best result per metric is **bolded**. Applying HIBRIDS on the encoder produces better QS hierarchies (higher F1) and questions (higher BLEU). Our models also yield better or comparable ROUGE scores, especially compared with HIERENC which requires 43% more parameters and extra engineering efforts for architecture change. **Ques**: question; **Hier**: hierarchy.

6 Experiment Results

6.1 Hierarchical Question-summary Generation

Results on QSGen-Hier. We report results on the task of generating QS hierarchies in Table 1. HIBRIDS-ENC *uniformly outperforms other variants and all comparisons on all metrics,* except

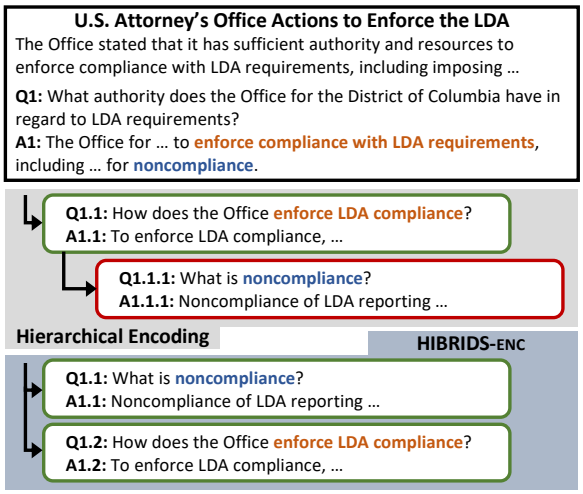

Figure 3: Sample output by the hierarchical encoding model (HIERENC) and HIBRIDS-ENC. Our generated structure makes more sense with the constructed follow-up questions to Q1, highlighted in **green**, than the comparison model HIERENC.

Model	R1	R2	RL	B4
LONGFORMER	26.90	8.69	25.57	14.44
SECTOK	26.76	8.82	25.42	14.51
LVLSECTOK	26.80	8.75	25.52	14.33
Structure-aware Comparisons				
HIERENC	26.38	8.81	24.99	14.54
MULTITASK	26.84	8.46	25.41	14.59
Models with Linear Bias				
TOKBIAS	26.73	8.69	25.38	14.43
SECBIAS	27.25	9.07	25.92	**14.76**
Our Models				
HIBRIDS-ENC	**27.33**	**9.46**	**26.00**	14.73
$\text{HIBRIDS}_{\text{S}}$-ENC	26.41	8.74	24.99	14.44
HIBRIDS-DEC	27.17	8.67	25.71	14.36
$\text{HIBRIDS}_{\text{S}}$-DEC	26.29	8.50	25.09	14.30

Table 2: Results for QSGen-ChildQ. The best result per metric is **bolded**. Using HIBRIDS on encoder generates better follow-up questions according to ROUGE scores.

for ROUGE-1 and ROUGE-L scores by HIERENC. Note that HIERENC learns 2M more new parameters than our models, and it produces QS hierarchies of lower quality despite its competitive ROUGE scores (Figure 3). This signifies the effectiveness of our design that directly injects structural information into word-level relation computation. Meanwhile, HIBRIDS *on encoder is better at hierarchy quality than its variant on decoder*, suggesting the importance of resolving section relations during encoding.

Though not reported here, we experiment with HIBRIDS on both the encoder and the decoder, and it results in degraded performance. One possible cause is that HIBRIDS functions differently in these two setups (discussed in §7). We will explore better fusion techniques in future work.

Results on QSGen-ChildQ. Results on generating follow-up questions further validate the usefulness of hierarchical biases as shown in Table 2, where *questions generated by* HIBRIDS-ENC *have the best quality* as measured by all metrics except for BLEU. SECBIAS, which is aware of section-level linear distance, also obtains outstanding performance, since it focuses on intra-section information and thus better determines what child questions should be asked for better relevance.

Human evaluation is conducted on QSGen-Hier, for five models with the highest automatic scores, to help understand how well the generated hierarchies are structured. We hire three judges who have *extensive experience* in summarization annotation and evaluation tasks to assess 50 groups of question-summary hierarchies. Human inspection on randomly selected outputs shows that most system generations have an appropriate coverage of the salient content in the source. Therefore, we focus on evaluating both global coherence and local coherence of the QS hierarchies based on the following two aspects. First, we ask evaluators to correct each generated hierarchy by rearranging the QS pairs so that each pair is attached to the parent that forms the best follow-up relation in steps. For each step, they are only allowed to attach a pair to its grandparent or sibling (i.e., the parent or child of its current parent). They then report the **number of edits** conducted for the rearrangement. Second, for each QS pair, we ask them to determine if the question can be **answered** by the summary. Details of human evaluation are in Appendix C.

As can be seen from Table 3, QS hierarchies generated by HIBRIDS-ENC model contain the best structured summaries as they require the fewest number of corrections and the generated questions are also more likely to be addressed by the corresponding summaries. Despite being competitive on automatic metrics, SECTOK generates hierarchies that require the most corrections. Upon additional inspection, we find that HIBRIDS's outputs often have better local coherence than the comparisons. Additionally, all models struggle to generate more engaging questions, which poses another challenge to future studies.

Model	# of Edits (↓)	Answerable Qs (↑)
SECTOK	4.73	81.8%
LVLSECTOK	4.62	78.6%
HIERENC	4.17	81.4%
TOKBIAS	3.77	82.8%
HIBRIDS-ENC	**3.67**	**84.1%**

Table 3: Human evaluation results on QSGen-Hier. Hierarchies produced by HIBRIDS-ENC require fewer correction edits by human and contain more answerable questions by the generated summaries. Krippendorff's α: 0.55, 0.44.

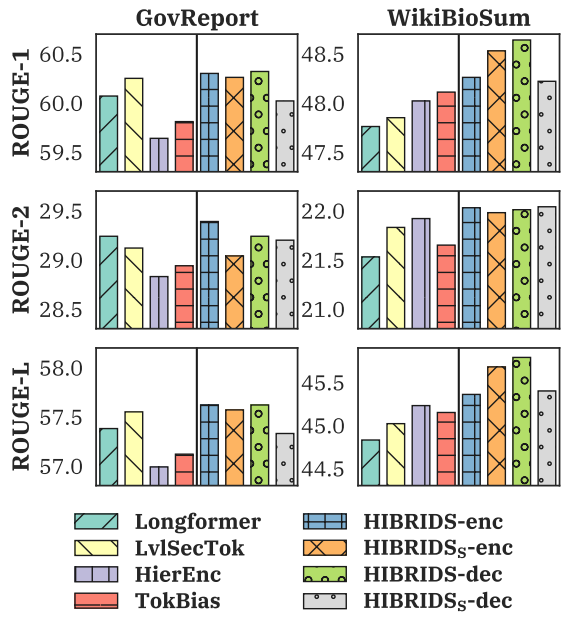

Figure 4: Results on full summary generation. In each subfigure, the left panel includes models for comparisons and the right panel shows our models. HIBRIDS on either encoder and decoder uniformly outperforms the comparisons on both datasets.

6.2 Full Summary Generation

As demonstrated in Figure 4, HIBRIDS *with full hierarchical biases outperform all comparisons* on both datasets, suggesting that our design of including structural relations in bias terms can generalize to other tasks. Compared to the results on QS hierarchy generation, using HIBRIDS on the decoder yields greater improvement on full summary generation, especially in the biography domain where HIBRIDS-DEC obtains the best performance. It is likely that the longer summary length and higher compression ratio on WIKIBIOSUM (1,266 and 0.45) makes generation coherence more important by using better alignment. This highlights how hierarchical biases can aid long text generation.

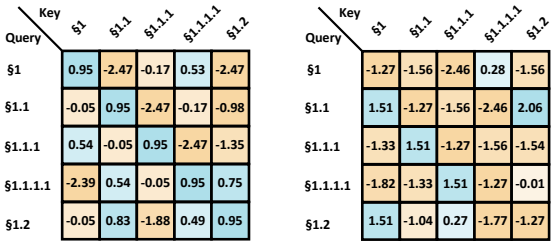

Figure 5: Visualization of hierarchical biases in HIBRIDS-ENC (left) and HIBRIDS-DEC (right) on QSGen-Hier. Positive and negative values are shaded in **blue** and **orange**. Displayed values are 100X of actual values. HIBRIDS-ENC biases towards current, grandparent and preceding sibling sections, while HIBRIDS-DEC focuses on parent and succeeding sibling sections.

7 Further Analyses

7.1 Visualizing the Learned Biases

Here we aim to understand what is learned by our hierarchical biases. For HIBRIDS-ENC and HIBRIDS-DEC trained on QSGen-Hier, we visualize the values of their *learned* hierarchical biases averaged over all heads at all layers for each (path length, level difference) pair on an example structure. Additional visualization is in Appendix D.

From Figure 5 we see that using HIBRIDS on the encoder encourages models to encode various relations, e.g., by upweighing grandparent (§1.1.1 to §1, §1.1.1.1 to §1.1) and preceding sibling (§1.2 to §1.1), and downweighing children (§1 to §1.1 and §1.2, §1.1 to §1.1.1). This highlights the need of learning heterogeneous relations among sections beyond token distances. By contrast, HIBRIDS on the decoder consistently biases towards parent and sibling contexts. It might be because that the generation of fluent and coherent question-summary pairs relies on being aware of the scope of sections at the same or higher levels.

7.2 Ablation Study for HIBRIDS

We examine which design choices contribute the most to the performance gain by HIBRIDS, by carrying out ablation studies on QSGen-Hier with HIBRIDS-ENC. We consider taking out (1) level difference, (2) path length, and (3) asymmetry of path length. As shown in Table 4, removing any component reduces summaries' content coverage and hierarchy quality, underscoring their contributions in more precisely representing structural relations for better document encoding. Level difference adds the most to hierarchy quality, as levels

Model	Summary RL	Question B4	Hierarchy F1
HIBRIDS-ENC	38.03	10.16	13.26
w/o Level Difference	–0.50	–0.08	–0.51
w/o Path Length	–0.43	+0.05	–0.18
w/o Asymmetric Path	–0.15	–0.12	–0.18

Table 4: Ablation study results. Performance change compared to the full model are reported. Larger decreases of metrics are shaded with darker orange. Removing level difference hurts the hierarchy quality substantially.

Model	QSGen-Hier R-2	QSGen-Hier R-L	QSGen-Hier Hier F1	GovReport R-2	GovReport R-L
HierEnc	16.32	38.06	11.77	28.83	56.99
w/ HIBRIDS	+0.44	+0.37	+0.22	+0.15	+0.22

Table 5: Effects of applying HIBRIDS to the extra section-level encoders of HierEnc on two tasks. HIBRIDS improves the performance of HierEnc on all metrics.

directly signal when to generate follow-up questions.

7.3 Can HIBRIDS Improve Hierarchical Encoding?

We further study if HIBRIDS can boost the section encoder of HierEnc. Table 5 shows that HierEnc with HIBRIDS gains further improvements on generating QS hierarchies and full document summarization on GovReport. This points to promising future adoptions of HIBRIDS by existing models that would benefit from encoding document structure.

8 Conclusion

We present HIBRIDS, which effectively and efficiently injects document structure information into abstractive summarization models via hierarchical learnable biases that adjust the attention score matrix. A new task, hierarchical question-summary generation, is then introduced for generating hierarchically organized question-summary pairs, to expose document structure and salient content to readers. We annotate a new dataset consisting of 6,153 summary paragraphs with question-summary hierarchies to facilitate our study, and it can also be used for query-focused summarization and question generation. Experiments on hierarchical question-summary generation and full summary generation show that HIBRIDS produces question-summary hierarchies of higher quality as measured by both automatic metrics and human judges, and achieves higher content coverage of summaries than competitive comparisons as reported by ROUGE.

Acknowledgements

This work is supported in part by National Science Foundation through grant IIS-2046016, Oracle Cloud credits and related resources provided by the Oracle for Research program. We thank the anonymous reviewers for their valuable suggestions.

Ethical Consideration

Collection of GovReport-QS and WikiBioSum. We comply with the terms of use and copyright policies of all data sources during the collection of GovReport-QS and WikiBioSum. Personal and other sensitive information is not collected to ensure the privacy of content creators. Before annotating GovReport-QS, we obtain consents from the annotators and inform them of their rights to temporarily suspend or quit the annotation process. During annotation, annotators are fairly compensated ($\approx$ $15 per hour).

Limitations and Potential Risks of HIBRIDS and GovReport-QS. While our experiments focus on datasets consisting of formal long documents, we recognize that long documents could be written in informal languages where our model might not perform reasonably and could generate degraded or even incorrect outputs. Despite recent advancement in improving summary factuality along with its evaluation (Kryscinski et al., 2020; Goyal and Durrett, 2020; Scialom et al., 2021; Cao and Wang, 2021b), the accuracy of existing factuality evaluation metrics has not been verified on long documents, which further increases the risk of incorrect outputs by our model.

As our GovReport-QS is based on reports from the United States (US) Government, the topics covered by the dataset are mostly relevant to the national interest of US. Therefore, models trained on our dataset might not be suitable for producing structured summaries for documents published by other countries that focus on other topics. Moreover, our GovReport-QS might bias the model towards a pro-US perspective, which could produce outputs that are harmful to certain populations.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Vidhisha Balachandran, Artidoro Pagnoni, Jay Yoon Lee, Dheeraj Rajagopal, Jaime Carbonell, and Yulia Tsvetkov. 2021. StructSum: Summarization via structured representations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2575–2585, Online. Association for Computational Linguistics.

Regina Barzilay and Michael Elhadad. 1999. Using lexical chains for text summarization. *Advances in automatic text summarization*, pages 111–121.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 113–120, Boston, Massachusetts, USA. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Shuyang Cao and Lu Wang. 2021a. Attention head masking for inference time content selection in abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5008–5016, Online. Association for Computational Linguistics.

Shuyang Cao and Lu Wang. 2021b. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana. Association for Computational Linguistics.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost.

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.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Peng Cui and Le Hu. 2021. Sliding selector network with dynamic memory for extractive summarization of long documents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5881–5891, Online. Association for Computational Linguistics.

Hal Daumé III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 305–312.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917, Melbourne, Australia. Association for Computational Linguistics.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1998–2008, Berlin, Germany. Association for Computational Linguistics.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. 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 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

John T. Guthrie, Tracy Britten, and K. Georgene Barker. 1991. Roles of document structure, cognitive strategy, and awareness in searching for information. *Reading Research Quarterly*, 26(3):300–324.

Jaakko Hintikka. 1981. The logic of information-seeking dialogues: A model.

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1515–1520, Seattle, Washington, USA. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

David H. Jonassen. 1988. Designing structured hypertext and structuring access to hypertext. *Educational Technology*, 28(11):13–16.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kalpesh Krishna and Mohit Iyyer. 2019. Generating question-answer hierarchies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2321–2334, Florence, Italy. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2021. Booksum: A collection of datasets for long-form narrative summarization.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020. *Asking Questions the Human Way: Scalable Question-Answer Generation from Text Corpus*, page 2032–2043. Association for Computing Machinery, New York, NY, USA.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Yang Liu, Ivan Titov, and Mirella Lapata. 2019. Single document summarization as tree induction. 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 1745–1755, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Marcu. 1997. From discourse structures to text summaries. In *Intelligent Scalable Text Summarization*.

Margaret G. McKeown, Isabel L. Beck, and Ronette G.K. Blake. 2009. Rethinking reading comprehension instruction: A comparison of instruction for strategies and content approaches. *Reading Research Quarterly*, 44(3):218–253.

Rui Meng, Khushboo Thaker, Lei Zhang, Yue Dong, Xingdi Yuan, Tong Wang, and Daqing He. 2021. Bringing structure into summaries: a faceted summarization dataset for long scientific documents. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th*

International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 1080–1089, Online. Association for Computational Linguistics.

Bonnie J. F. Meyer, David M. Brandt, and George J. Bluth. 1980. Use of top-level structure in text: Key for reading comprehension of ninth-grade students. *Reading Research Quarterly*, 16(1):72–103.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Fanyi Qu, Xin Jia, and Yunfang Wu. 2021. Asking questions like educational experts: Automatically generating question-answer pairs on real-world examination data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2583–2593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Tobias Rohde, Xiaoxia Wu, and Yinhan Liu. 2021. Hierarchical learning for generation with long source sequences.

Mrinmaya Sachan and Eric Xing. 2018. Self-training for jointly learning to ask and answer questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 629–640, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. 2012. Trains of thought: Generating information maps. In *Proceedings of the 21st international conference on World Wide Web*, pages 899–908.

Richard J. Shavelson. 1974. Methods for examining representations of a subject-matter structure in a student's memory. *Journal of Research in Science Teaching*, 11(3):231–249.

Manfred Stede and David Schlangen. 2004. Information-seeking chat: Dialogues driven by topic-structure. In *Proceedings of Catalog (the 8th workshop on the semantics and pragmatics of dialogue; SemDial04)*. Citeseer.

Barbara M. Taylor and Richard W. Beach. 1984. The effects of text structure instruction on middle-grade students' comprehension and production of expository text. *Reading Research Quarterly*, 19(2):134–146.

Lu Wang, Claire Cardie, and Galen Marchetti. 2015. Socially-informed timeline generation for complex events. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1055–1065, Denver, Colorado. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. 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 3011–3021, Hong Kong, China. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

A Details of GovReport-QS

Dataset Choice. We choose GovReport dataset (Huang et al., 2021) for our annotation because it contains long documents (9409 tokens) and summaries (553 tokens) with key information spread throughout documents, which ensures the building of rich question-summary hierarchies. Moreover, the documents in GovReport are organized into multiple levels of sections, which justifies our decision to present salient document information with question-summary hierarchies.

Summary Paragraph Selection. Documents that are short or contain very few sections are less likely to yield rich QS hierarchies. To select high-quality paragraphs for annotation, we first consider using summary paragraphs associated with documents that have at least 3 sections. Moreover, the average number of paragraphs in each section should be at least 5. We then discard summaries that have less than 3 paragraphs. Among the paragraphs of the remaining summaries, we select those with at least 3 sentences and 70 words. To incorporate more abstractive summaries in the question-summary pairs, we further calculate the normalized density (Grusky et al., 2018) between each summary paragraph and its corresponding document, and then keep the paragraphs with a normalized density less than 0.15. The selection process results in 25,063 summary paragraphs which are then randomly sampled for annotation.

Annotation Process. We hire 11 college students who are native English speakers as annotators. They are informed of the job opportunity through email lists that advertise on-campus jobs. They sign up for the annotation job by filling a Google Form containing a detailed job description and consent form. The employment process is handled through the school employment system. Before annotating, they read the annotation instruction and examples with annotated question summary hierarchies. In each round of the annotation, each annotator is given 28–33 summary paragraphs, which takes about 2 hours to finish. We pay each annotator $30 (≈ $15 per hour) for each round. Appen[3] is used for building the annotation interface and collecting annotations. The annotation instruction is shown in Figure 7–10.

Section Alignment. We align each annotated summary paragraph with sections in the source document (§ 4) in the following way. Three similarity scores are computed for each pair of summary sentence and document paragraph: (1) cosine similarity between the representations computed by Sentence BERT (Reimers and Gurevych, 2019) for the summary sentence and the document paragraph; (2) the percentage of unique bigrams in the summary sentence that occur in the document paragraph; and (3) the percentage of unique named entities[4] that

[3] https://appen.com

[4] We use SpaCy 3.0.3 (Honnibal and Montani, 2017) with `en_core_web_sm` for named entity recognition.

occur in the document paragraph. The final similarity score is the sum of these three scores, with weights 0.4, 1.0, and 0.2, respectively. We tune the weights based on the manual alignment for 836 summary paragraphs associated with 42 report documents. Finally, each summary sentence is mapped to the document paragraph with the highest similarity score.

Copyright Policy. Documents and summaries in GovReport dataset are published by Government Accountability Office (GAO)[5] and Congressional Research Service (CRS)[6]. The original publications are not protected by copyright law and Huang et al. (2021) make GovReport publicly available. We release the new annotations under the CC BY 4.0 license[7]. Users of the data must also acknowledge GAO and CRS as the sources of the original publications.

B Details of WikiBioSum

Data Collection. To collect biographies from WikiProject Biography[8], we first use Scrapy[9] to get the names of articles curated by the project. We then extract article content with WikiExtractor[10] from the English Wikipedia dump[11] at 2021/08/01 using the article names.

Data Filtering. In addition to keeping biographies with at least two levels of section hierarchy, we discard biographies that have a quality class that lower than C.[12] The quality class of each biography is assessed by the members of WikiProject Biography. To get rid of samples where summaries can be generated by reading the first half of the documents only, we check the occurrences of summary bigrams in the documents and keep the samples where the second half of the documents contain at least 9% of new summary bigrams that do not occur in the first half.

[5] https://www.gao.gov/

[6] https://crsreports.congress.gov/

[7] https://creativecommons.org/licenses/by/4.0/

[8] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Biography

[9] https://scrapy.org

[10] https://github.com/attardi/wikiextractor. We modify the original code so that full section structures can be preserved.

[11] https://dumps.wikimedia.org

[12] Quality classes include FA, A, GA, B, C, Start, and Stub, sorted from best to worst.

model	Avg QS Pairs / Hier
SecTok	5.29
LvlSecTok	5.10
HierEnc	5.29
TokBias	4.95
HIBRIDS-enc	5.17

Table 6: Average numbers of QS pairs generated for each hierarchy by models in our human evaluation.

Statistics. As reported in the main paper, the average lengths of the input and output are 3,478 and 1,266. The average number of sections in the input is 11.65, with an average depth of 2.22 levels. Moreover, each document has 32.19 paragraphs.

Copyright Policy. We follow the Wikipedia copyright policy[13] to collect the WikiBioSum dataset. The WikiBioSum dataset will be released under the CC BY-SA 3.0 license[14]. Usage of the WikiBioSum dataset is limited by the copyright policy of Wikipedia.

C Details of Human Evaluation

We conduct human evaluation for question-summary hierarchies generated by five models. Human evaluation instructions are shown in Figure 11. The annotators use an HTML interface (Figure 12). Model names are not displayed, and their outputs in each group are randomly shuffled. The interface displays all the annotations made by the same annotator, which helps human subjects achieve better annotation consistency across different model outputs. Finally, we report the average numbers of QS pairs per hierarchy for each model in Table 6.

D Additional Visualization

We show the biases learned by HIBRIDS for full document summarization on GovReport in Figure 6. Behaviors of HIBRIDS on GovReport are different from those observed on QSGen-Hier in §7. On GovReport, using HIBRIDS on the encoder encourages each token to attend to other tokens within the same section, highlighting its focus on recency. By contrast, HIBRIDS on the decoder biases towards short-term contexts before a given token and strongly discourages attentions to long-range contexts. It might be because that

[13] https://en.wikipedia.org/wiki/Wikipedia:Copyrights

[14] https://creativecommons.org/licenses/by-sa/3.0/

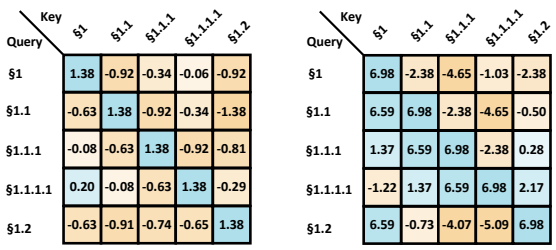

Figure 6: Visualization of hierarchical biases in HIBRIDS-ENC (left) and HIBRIDS-DEC (right) on GOVREPORT. Positive and negative values are shaded in **blue** and **orange**. Displayed values are 100X of actual values. HIBRIDS-ENC biases towards recency, while HIBRIDS-DEC focuses on parent sections.

the generation of fluent and coherent summaries mainly depends on local and past contexts.

E Sample Output

We show more outputs by HIBRIDS-ENC on QSGen-Hier in Table 7.

F Details of Implementation

We take the implementation of Longformer from Huggingface 4.8.1 (Wolf et al., 2020), which is licensed under the Apache License 2.0[15]. The model configuration and pre-trained weights of `allenai/led-large-16384`[16] are used. For model training, we use Fairseq (commit `f34abcf2`) (Ott et al., 2019) that adopts MIT License[17]. Both model training and decoding are performed on the A6000 GPU with 48GB memory and the A100 GPU with 40GB memory.

Training Settings. During training, we set the number of tokens in each batch to 10,240 for QSGen-Hier, QSGen-ChildQ, and full summary generation on WIKIBIOSUM. On GOVREPORT, each batch contains 16,384 tokens. As limited by the design of Longformer, the maximum output length for all tasks is set to 1,024. We use Adam (Kingma and Ba, 2015) as the optimizer, with a maximum learning rate of 5×10^{-5}. The optimizer updates the model parameters every 8 batches. We set the maximum numbers of update steps to 500, 700, 2,400, and 5,000 respectively for QSGen-Hier, QSGen-ChildQ, WIKIBIOSUM, and GOVREPORT. Importantly, we adopt gradient checkpointing (Chen et al., 2016) to reduce the memory consumption of back propagation.

Decoding Settings. A beam search with a beam size of 4 is used for decoding. The maximum decoding length is 1,024. We also disable the generation of repeated 5-grams.

Running Time. HIBRIDS takes 2, 2, 5, and 12 hours for training on QSGen-Hier, QSGen-ChildQ, WIKIBIOSUM, and GOVREPORT with 4 GPUs. Decoding on QSGen-Hier and QSGen-ChildQ takes one hour. For decoding on WIKIBIOSUM, and GOVREPORT, it uses 3 and 4 hours.

Evaluation. We compute ROUGE scores (Lin, 2004) using the implementation by Google Research[18]. For BLEU scores, we use NLTK 3.5 (Bird et al., 2009).

[15] https://www.apache.org/licenses/LICENSE-2.0

[16] https://huggingface.co/allenai/led-large-16384

[17] https://opensource.org/licenses/MIT

[18] https://github.com/google-research/google-research/tree/master/rouge

Example 1

Q1: What incited the start of the FY2009 appropriation process?
A1: On February 4, 2008, President Bush sent his FY2009 budget to Congress, which included a request for $39 billion for the Department of Housing and Urban Development (HUD).

Q1.1: How did Congress respond to this request?

A1.1: On June 4, 2008, the Senate passed the FY2009 budget resolution conference agreement (H.Rept. 110-659) and the House passed it the following day.

Q1.2: What was the result of the FY2009 appropriations process?

A1.2: On March 11, 2009, a FY2009 omnibus appropriations bill was signed into law, funding HUD for the remainder of the fiscal year (P.L. 111-8). It provides a more than 10% increase in regular, non-emergency appropriations over the FY2008 level.

Q1.2.1: How did the omnibus appropriations bill affect HUD?

A1.2.1: It provided nearly $13.7 billion for HUD programs.

Example 2

Q1: To what extent is democracy promotion an element of U.S. foreign policy?
A1: For decades U.S. policymakers have connected U.S. national security and other core interests with the spread of democracy around the world. Reflecting this, the promotion of democracy has been a longstanding and multifaceted element of U.S. foreign policy, and one often interrelated with U.S. efforts to promote human rights.

Q1.1: How has the promotion of democracy promotion been supported by Congress?

A1.1: Congress has often played an important role in supporting and institutionalizing U.S. democracy promotion by passing key legislation, appropriating funds for foreign assistance programs and other democracy promoting activities, and conducting oversight of aspects of U.S.-led foreign policy relevant to democracy promotion.

Q1.2: What is the current state of democracy promotion?

A1.2: Widespread concerns exist among analysts and policymakers over the current trajectory of democracy around theworld and multiple hearings in the 115th Congress reflected bipartisan concern over this issue.

Q1.2.1: What are some of these concerns?

A1.2.1: Frequently cited concerns include the rise of authoritarian populist and nationalist leaders, the potential negative influence on democracy from internationally assertive authoritarian states, questions over the enduring appeal of democracy as a political system, new tools nondemocratic governments are using to stifle potential democratizing forces, and others.

Example 3

Q1: How should GA strategies be approached?
A1: GA security poses significant challenges for policymakers and security experts because GA is highly diverse, geographically dispersed, and relatively open compared to commercial airports servicing passenger airlines and other protected infrastructure such as nuclear reactors and chemical plants.
Q2: What is the primary threat posed by GA aircraft?
A2: The primary threat posed to GA aircraft is not so much to GA assets themselves, but rather, from terrorists seeking to exploit GA assets to attack critical infrastructure or high-profile targets.

Q2.1: What is a secondary threat to GA aircraft?

A2.1: A secondary threat is that terrorists may infiltrate or otherwise exploit GA to gain knowledge and/or access to the airspace system in the United States.

Q2.1.1: What are some examples of this threat?

A2.1.1: For example, some corporate aviation operators have expressed concern that aircraft carrying high-profile business leaders and executives, such as presidents of major U.S. corporations, could be targeted, particularly when operating overseas in areas where security concerns exist.

Table 7: Example outputs by HIBRIDS-ENC on QSGen-Hier. Indentation indicates the levels of question-summary pairs.

Task description and guidelines

We are building a dataset of Question-Answer pairs based on given texts. In this task, you will be shown a paragraph from a summary of a US government report. You are expected to (1) read the paragraph, (2) create questions, and (3) provide their respective answers using the given paragraph such that your question-answer pairs cover the whole essence of the given paragraph. More details are given below.

We have compiled a list of FAQ below regarding the details and rules for the task. Please read them carefully before you start.

Q1:How many paragraphs do I annotate?

A1:The annotation task given to you contains **about 30 summary paragraphs**. Notice that each page contains only two paragraphs, you need to click the **"Submit and Continue"** button to see the next two paragraphs. The time limit for each page is **30 minutes**. You're expected to finish each page within 30 minutes. If you exceed the time limit, you need to re-open the given link and continue your task.

These summary paragraphs are extracted from summaries of US government reports (CRS and GAO). Summary paragraphs belonging to the same report occur in order (as in the original report), but some paragraphs in the original report might not be included and thus the paragraphs you are annotating might not be consecutive. We recommend you **finish annotating paragraphs of the same report in one sitting** so that you can have better context for the paragraphs. Each summary paragraph comes with a title (from the paragraph's corresponding report). You will generate your question-answer pairs based on this summary paragraph.

Q2:What types of questions should I create?

A2:In short, you are expected to write **complex (narrative) questions**, the answers of which usually consist of one or more sentences and function as reasoning or explaining a concept.

We are trying to build a dataset with narrative/complex questions and their answer pairs. The answers to such complex questions are more than just a few words -- they should be one or more complete sentences. You may want to ask questions starting with **"why", "how", or "what"**. These tend to create complex questions whose corresponding answers try to reason or explain a concept. Please **REFRAIN** from asking questions that start with "who", "which", "when", "how many", "how much", etc. Make sure that your questions are complete and grammatical. See EXAMPLES.

In addition to this, you can also ask another ("why", "how", or "what") question as a follow-up to one or more of your questions if possible for the given summary paragraph. You can create multiple follow-ups for a question and even follow-ups to a follow-up question.

For each paragraph, you're expected to ask **at least 1 follow-up question**. We encourage you to try to make as many follow-ups ("why", "how" or "what" questions) as you can without writing factoid questions (answer to which is a number, data, name, etc). See the examples to understand what are some good/bad questions and answers as well as follow-up pairs.

Q3:How should I provide the answers to the questions I ask?

Figure 7: Question-summary hierarchy annotation instructions. (Page 1 / 4)

A3:For each question you make, you should copy-paste one or more **COMPLETE SENTENCE(S)** -- not just words or phrases -- from the given paragraph as the answer span. Please ensure that you **DO NOT** copy a phrase or word as the answer span! The answer span should either partially or completely answer that question. The answer span could be the sentences that you considered to generate the question or the sentences that you think contains all or part of the intended answer.

Follow the same rules for answering the follow-up questions. We would prefer not having two questions that have the same answer. So, while the answers for two questions can have overlapping answer spans, the two answers shouldn't be exactly the same. Please include an answer for every question you write.

Q4:How many question-answer pairs should I make?

A4:You should try to write as many questions or follow-up questions per summary paragraph as you can. In principle, you're expected to construct **at least 4 questions** (**3** if there are only **3** sentences in the paragraph), including **at least 1 follow-up question**.

Q5:How do I format my generated question and answer in my annotation file?

A5:Your responses for each summary paragraph will go in the text box provided below the summary paragraph. You start with the first question Q1 and then continue with a follow-up question Q1.1 or a non-follow-up question Q2 and so on. PLEASE follow the formatting shown in the EXAMPLES!

Examples

Example 1

Summary Paragraph: In September 2014, GAO reported on the Department of Veterans Affairs' (VA) Program of Comprehensive Assistance for Family Caregivers (Family Caregiver Program) and found that the program office had limitations with its information technology (IT) system—the Caregiver Application Tracker (CAT). Specifically, the program did not have ready access to workload data that would allow it to monitor the effects of the program on VA medical centers' resources. VA has initiated various projects since 2015 to implement a new system, but has not yet been successful in its efforts. Specifically, in July 2015 VA initiated a project to improve the reliability of CAT's data, called CAT Rescue. However, the department reported in January 2017 that it had identified numerous defects during system testing. The project ended in April 2018 before any new system capabilities were implemented. A companion project was initiated in September 2015 to develop the Caregivers Tool (CareT), a new system intended to replace CAT. The CareT project was expected to use improved data from CAT Rescue, while also adding new system capabilities. However, the user acceptance testing of CareT identified the need for the department to develop more system capabilities than originally planned. Further, VA reported that implementing a system by October 1, 2018, as specified in the Maintaining Internal Systems and Strengthening Integrated Outside Networks Act of 2018 (MISSION Act), was not feasible. Subsequently, VA terminated CareT in February 2019. The department initiated another project in March 2019 to implement a new system, the Caregiver Record Management Application (CARMA). GAO has ongoing work to evaluate the department's efforts to implement an IT system to support the Family Caregiver Program as required by the MISSION Act.

*For the given summary paragraph, the following are **GOOD** question-answer pairs. Q2.1, Q2.2 are follow-up questions of Q2.*

Figure 8: Question-summary hierarchy annotation instructions. (Page 2 / 4)

Q1:What were the findings of the GAO report?

A1:In September 2014, GAO reported on the Department of Veterans Affairs' (VA) Program of Comprehensive Assistance for Family Caregivers (Family Caregiver Program) and found that the program office had limitations with its information technology (IT) system—the Caregiver Application Tracker (CAT). Specifically, the program did not have ready access to workload data that would allow it to monitor the effects of the program on VA medical centers' resources.

Q2:How has the VA attempted to improve the CAT program?

A2:VA has initiated various projects since 2015 to implement a new system, but has not yet been successful in its efforts. Specifically, in July 2015 VA initiated a project to improve the reliability of CAT's data, called CAT Rescue. A companion project was initiated in September 2015 to develop the Caregivers Tool (CareT), a new system intended to replace CAT. The department initiated another project in March 2019 to implement a new system, the Caregiver Record Management Application (CARMA).

Q2.1:Why did CAT Rescue end in April 2018?

A2.1:However, the department reported in January 2017 that it had identified numerous defects during system testing. The project ended in April 2018 before any new system capabilities were implemented.

Q2.2:Why was the CareT Program unsuccessful?

A2.2:The CareT project was expected to use improved data from CAT Rescue, while also adding new system capabilities. However, the user acceptance testing of CareT identified the need for the department to develop more system capabilities than originally planned. Further, VA reported that implementing a system by October 1, 2018, as specified in the Maintaining Internal Systems and Strengthening Integrated Outside Networks Act of 2018 (MISSION Act), was not feasible. Subsequently, VA terminated CareT in February 2019.

Q3:What has been the GAO's response to the VA's efforts?

A3:GAO has ongoing work to evaluate the department's efforts to implement an IT system to support the Family Caregiver Program as required by the MISSION Act.

*For the given summary paragraph, the following are **BAD** question-answer pairs.*

Q1:When did GAO report on the Department of Veterans Affairs' (VA) Program of Comprehensive Assistance for Family Caregivers? (The question is a "when" question asking for the simple fact of date)

A1:In September 2014 (The answer is not a full sentence.)

Q2:Who initiated the CAT Rescue project to improve the reliability of CAT 's data? (The question is a "who" question asking for the simple fact of the initiator's name)

A2:Specifically, in July 2015 VA initiated a project to improve the reliability of CAT's data, called CAT Rescue.

Example 2

Summary Paragraph: The marine transportation system is a critical part of the nation's infrastructure. To facilitate the safety and efficiency of this system, the Coast Guard maintains aids-to-navigation (ATON), such as buoys and beacons, and conducts domestic icebreaking in the Great Lakes, St. Lawrence Seaway, and northeast coast. To conduct these missions, the Coast Guard has a fleet of more than 200 vessels, ranging from 225-foot seagoing

Figure 9: Question-summary hierarchy annotation instructions. (Page 3 / 4)

buoy tenders and 140-foot domestic icebreakers to 21-foot boats. After the terrorist attacks of September 11, 2001, many of these assets took on additional responsibilities for security patrols and other homeland security duties. Although some assets have been recently acquired, many others are reaching or have exceeded their design service lives, raising concerns about how well and for how much longer these older assets may be able to carry out their missions. In response, GAO examined (1) recent trends in the amount of time these assets have spent performing missions; (2) asset condition and its effect on mission performance; and (3) the actions taken by the Coast Guard to continue to achieve the missions of these assets. To conduct this work, GAO reviewed Coast Guard documents, interviewed Coast Guard officials, and made site visits to various locations around the country. In commenting on a draft of this report, the Coast Guard provided technical comments, which were incorporated as appropriate.

*For the given summary paragraph, the following are **GOOD** question-answer pairs. **(Notice the follow-up questions)**Q1.1, Q1.2 are follow-up questions of Q1 and Q1.2.1 is a follow-up question of Q1.2.*

Q1:How does the Coast Guard maintain the safety and efficacy of the country's marine transportation system?

A1:The marine transportation system is a critical part of the nation's infrastructure. To facilitate the safety and efficiency of this system, the Coast Guard maintains aids-to-navigation (ATON), such as buoys and beacons, and conducts domestic icebreaking in the Great Lakes, St. Lawrence Seaway, and northeast coast. To conduct these missions, the Coast Guard has a fleet of more than 200 vessels, ranging from 225-foot seagoing buoy tenders and 140-foot domestic icebreakers to 21-foot boats.

Q1.1:How did the terrorist attacks of September 11 affect the Coast Guard's work in maintaining the marine transport system?

A1.1:After the terrorist attacks of September 11, 2001, many of these assets took on additional responsibilities for security patrols and other homeland security duties.

Q1.2:What are the concerns regarding Coast Guard assets for maintaining the marine transport system?

A1.2:Although some assets have been recently acquired, many others are reaching or have exceeded their design service lives, raising concerns about how well and for how much longer these older assets may be able to carry out their missions.

Q1.2.1:How has GAO responded to these concerns?

A1.2.1:In response, GAO examined (1) recent trends in the amount of time these assets have spent performing missions; (2) asset condition and its effect on mission performance; and (3) the actions taken by the Coast Guard to continue to achieve the missions of these assets. To conduct this work, GAO reviewed Coast Guard documents, interviewed Coast Guard officials, and made site visits to various locations around the country.

Figure 10: Question-summary hierarchy annotation instructions. (Page 4 / 4)

In this study, you will evaluate 50 sets of **question-summary (QS) hierarchies** produced by five systems. The hierarchy is presented by the IDs of questions and summaries (e.g., Q1 is the parent of Q1.1 and Q1.2). We also consider there is a dummy root to be the parent of the top-level questions (e.g., Q1, Q2).
Please go through the hierarchy generated by each system in order. For each QS pair in the hierarchy, you need to **adjust it step by step** such that it has the most appropriate QS pair as its parent. Meanwhile, please also check if the question can be **answered** by its corresponding summary. The descriptions of how to make the adjustment and determine answerability are detailed as follows with an example.

Example

(DUMMY ROOT)
Q1: What did state officials report about the effectiveness of identification verification procedures?
A1: State officials interviewed by GAO report that identity verification procedures have been effective at combating certain kinds of fraud, but vulnerabilities remain. Officials in most of the 11 states GAO contacted reported a decline in the use of counterfeit identity documents, and officials in states using facial recognition said they detected a number of identity theft attempts.
Q1.1: How can criminals use someone else's identity to get a license in another state?
A1.1: However, criminals can still steal the identity of someone in one state and use it to get a license in another because states lack the capacity to consistently detect such cross-state fraud.
Q1.1.1: What is one solution to this existing issue?
A1.1.1: For example, one state officials told GAO a check against the problem driver database (Problem Driver Pointer System) will not detect a license in another state if it is not associated with any driving violation.
Q2: ...
A2: ...

Step-by-step Adjustment: In QS hierarchies, the children of a QS pair ask about follow-up information that could be specific descriptions or elaborations of the content in the QS pair. For each QS pair, you need to first determine another QS pair (or the dummy root) as its parent such that they form the best follow-up relation. After identifying the most appropriate parent, adjustment of the QS pair is conducted step by step. In each step, you can attach the QS pair to its grandparent or sibling (i.e., the parent or child of its current parent).
Please report the **number of steps** required to complete the adjustment. If no adjustment is needed, please report 0.
For example, the most appropriate parent for Q1.1 is the DUMMY ROOT because it asks about a concrete flaw of the identification verification procedure while Q1 and A1 talk about the effectiveness of the procedure. These two questions are regarding the current status of the identification verification procedure and they should be at the same level. As there is an edge between Q1 and DUMMY ROOT, you only need **one** step to finish attaching Q1.1 to DUMMY ROOT. (Q1 $\rightarrow$ DUMMY ROOT).
Note that the parent-child relation remains unchanged for the children and descendants of an adjusted QS pair. For example, after attaching Q1.1 to DUMMY ROOT, attaching Q1.1.1 to Q2 only needs two steps as Q1.1 is already attached to DUMMY ROOT (Q1.1 $\rightarrow$ DUMMY ROOT $\rightarrow$ Q2).

Answerability: Whether the question can be answered by the associated summary.
Please select "True" or "False" for each QS pair.
For example, Q1.1.1 is not answerable because A1.1.1 does not mention any solution. Both Q1 and Q1.1 are answerable.

Figure 11: Human evaluation guidelines.

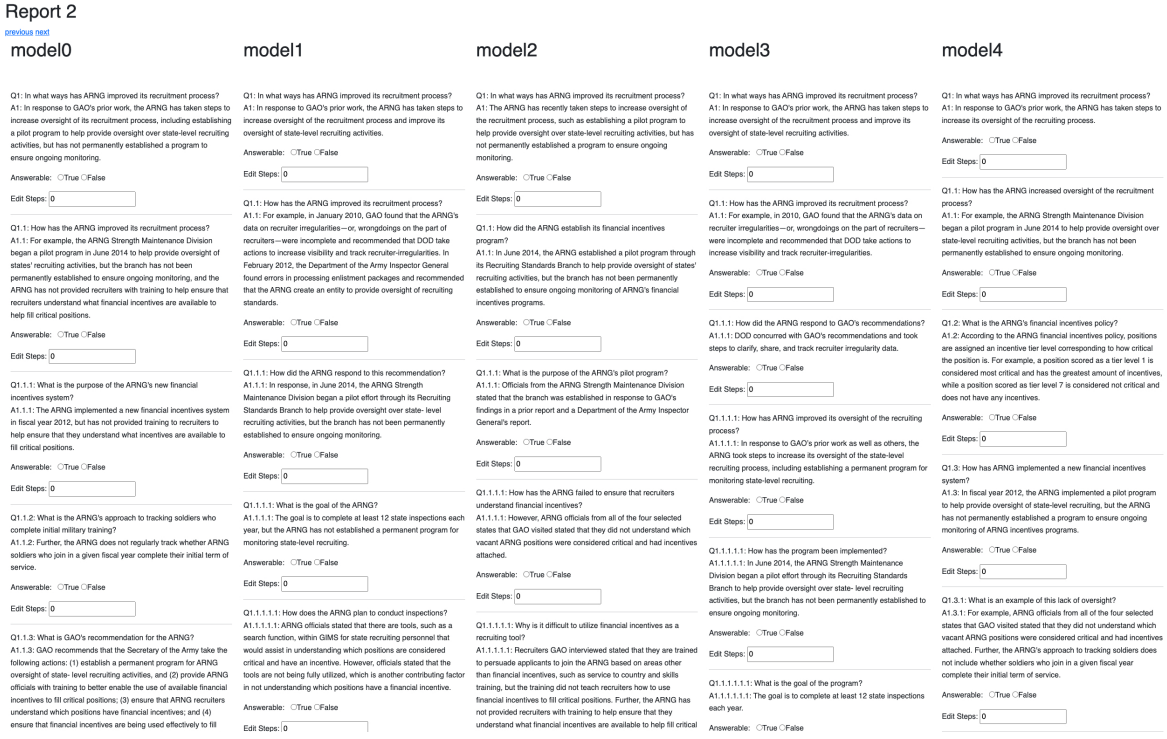

Figure 12: Screenshot of the human evaluation interface.