GenDec: A Generative Question-decomposition method for Multi-hop Question-answering

Anonymous ACL submission

Abstract

Multi-hop QA involves step-by-step reasoning to answer complex questions and find multiple relevant supporting facts. Previous question-decomposition research on multi-hop QA has shown that performance can be boosted by first decomposing questions into simpler, single-hop sub-questions (QD), and then answering them one by one in a specific order. However, such decomposition often leads to error propagation during QA: 1) incorrect QD leads to wrong QA results; 2) wrong answers to a previous sub-question compromise the next sub-question. In this work, we propose GenDec, a generative QD-based model for multi-hop QA from the perspective of explainable QA by generating independent and complete sub-questions based on incorporating supporting facts. This approach first introduces sub-questions in retrieving relevant passages at each hop and fuses features of sub-questions into QA reasoning, which enables it to provide an explainable reasoning process for its answers. We evaluate GenDec by comparing it with existing QD-based and other strong QA models and the results show GenDec outperforms all QD-based multi-hop QA models for answer spans on the HotpotQA and 2WikihopMultiHopQA datasets. We also conduct experiments with the large language models (LLMs) ChatGPT and LLaMA to illustrate the impact of QD on QA tasks in the LLM era.

1 Introduction

Multi-hop QA (MQA) is a task that requires multiple reasoning steps over multiple information sources (e.g., text paragraphs). While explicit question decomposition (QD), which involves breaking down complex questions into simpler and more straightforward sub-questions, has long been an approach in developing robust and interpretable question-answering (QA) models and systems, most MQA models, e.g., DFGN (Qiu et al., 2019), DecompRC (Min et al., 2019a), CogQA (Ding et al., 2019), HGN (Fang et al., 2019b), C2F Reader (Shao et al., 2020a), and BFR-Graph (Huang and Yang, 2021) illustrate how demonstrating the reasoning ability of a model in multi-hop questions remains a challenge. For example, Tang et al. (2020b) proposes a human-verified sub-question dataset derived from HotpotQA (Yang et al., 2018a) and conducts experiments on sub-question reasoning. The results indicated that DFGN, DecompRC, and CogQA performed badly on answering sub-questions, even when they found the correct answers to multi-hop questions because they usually bypass the correct reasoning process and fail to reason intermediate answers to sub-questions.

Thus, understanding and potentially decomposing multi-hop questions into finer-grained sub-questions is a key desired step in QA. To accurately answer a multi-hop question, traditionally QD + QA methods start by decomposing the given multi-hop question into simpler sub-questions, attempting to answer them in a specific order, and then finally aggregating the information obtained from all sub-questions.

Through a preliminary investigation, we find that QD remains a major bottleneck in MQA. Previous QD methods Min et al. (2019b); Perez et al. (2020a) first decompose multi-hop questions into **dependent** sub-questions, e.g., in figure 1, the original question is decomposed into *"Who is the record holder for Argentine PGA Championship tournaments? "* and *How many tournaments did [Ans of Sub Q1] win?"* and QA models need to correctly answer sub-question 1 and fill it into sub-question 2 and then answer it to get the final answer. Such QD+QA method suffers from error propagation, where incorrectly answering any of the sub-questions may lead to a wrong final answer. GenDec mitigates this error-propagation problem during reasoning since the decomposed sub-questions are independent and complete, thus not requiring answers in a specific order as was the case in previ-

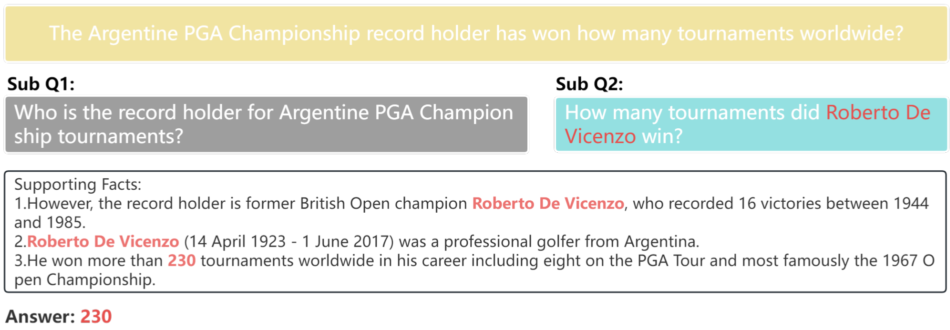

Figure 1: Example of multi-hop and decomposed sub-questions from the HotpotQA dataset. The original question is shown in gold and the decomposed ones in gray and cyan. *"Roberto de Vincenzo"* in supporting facts is the answer to sub-question Q1 and also part of the sub-question Q2. The literal *"230"* is the answer of sub-question Q2.

ous models. We fuse the sub-questions into the QA model to provide the appropriate reasoning chain.

We propose **GenDec**, a generative-based QD method that incorporates supporting facts including evidence for decomposing independent sub-questions that do not require answers in order. After QD, GenDec combines the sub-questions into a paragraph retrieval module by computing attention with each paragraph. These fuses sub-questions are fused into a multi-hop QA module. Figure 1 shows the decomposition results of GenDec over the HotpotQA dataset. The original multi-hop question *"The Argentine PGA Championship record holder has won how many tournaments worldwide?"* is decomposed into independent sub-questions: *"Who is the record holder for Argentine PGA Championship tournaments? "* and *How many tournaments did Roberto De Vicenzo win?"*.

GenDec is thus less vulnerable to different types of question issues than other QA models as it only needs supporting facts as extra decomposing information and does not need to consider hop relations nor answer the order of sub-questions. We further evaluate the effectiveness of our system in multi-hop QA to illustrate that QD still plays a vital role in QA in the large language model (LLM) era. Our contributions are as follows:

- We develop a generative QD-based model that can directly generate natural language sub-questions by incorporating evidence hidden in supporting facts.
- Detailed experimental results show that incorporating the generated sub-questions into paragraph retrieval and QA modules allow GenDec to outperform all QD-based QA models and other strong baselines.
- We explore the potential usage of LLMs (e.g., LLaMA or ChatGPT) and demonstrate QD still plays a vital role in QA in the LLM era.

2 Related Work

2.1 Multi-hop Question-answering

Multi-hop QA requires more than one reasoning step in multiple paragraphs to answer a question. For example, multi-hop QA in DROP (Dua et al., 2019) requires numerical reasoning such as addition and subtraction. Yang et al. (2018b) proposed the HotpotQA dataset that contains 113K multi-hop QA pairs collected from Wikipedia articles by crowd-sourcing. Ho et al. (2020a) presented 2WikiMultiHopQA, which uses structured and unstructured data and introduces the evidence information containing a reasoning path for multi-hop questions.

2.2 Question Decomposition

Several studies conducted QD in complex QA tasks by using different methods. Wolfson et al. (2020a) and Talmor and Berant (2018), inspired by SQL and SPARQL query, proposed rule-based methods. However, they failed to generalize into different types of questions because of the limited rules. Min et al. (2019b) proposed a supervised QD method with human-labeling data to predict the text span of sub-questions. ONUS (Perez et al., 2020a) is a one-to-N unsupervised sequence transduction method that uses supervision information of pseudo-decompositions from Common Crawl to map complex questions into simpler questions and

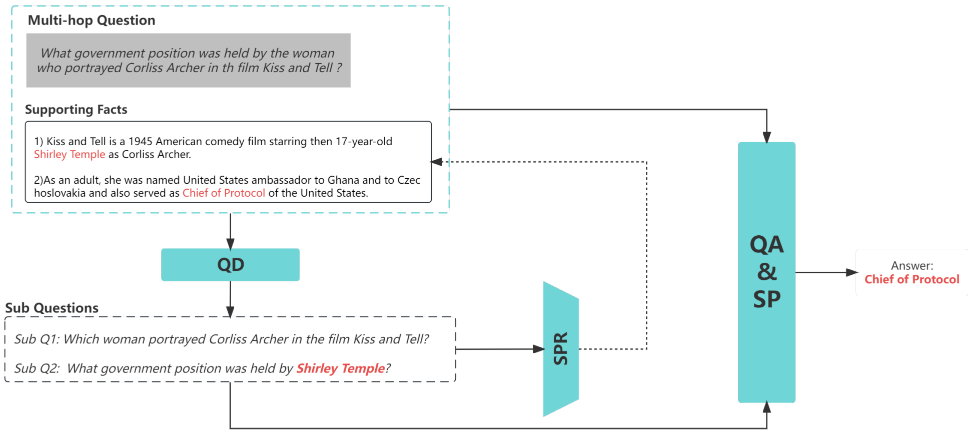

Figure 2: Pipeline of GenDec. From top to bottom. We first carry out Question Decomposition (QD) to decompose a multi-hop question into its sub-questions and then train a sub-question-enhanced paragraph retrieval module (SPR). We then input multi-hop questions, sub-questions, as well as retrieved paragraphs, into the sub-question-enhanced QA module to extract the final answers.

recompose intermediate answers of sub-questions for reasoning final answers. These supervised and unsupervised QD methods decompose complex questions into two sub-questions but are not applicable to real scenarios. Deng et al. (2022b) proposed an Abstract Meaning Representation (AMR)-based QD method that trains an AMR-to-text generation model on the QDMR (Wolfson et al., 2020b) dataset. The entity description graph (EDG)-based QD method (Hu et al., 2021b) represents the structure of complex questions to solve the question-understanding and component-linking problems of knowledge base QA tasks. Zhou et al. (2022) pre-trained Decomp-T5 on human-collected parallel news to improve the ability of semantic understanding for QD. Instead of answering sub-questions one by one, Guo et al. (2022) directly concatenated sub-questions with the original question and context to leverage the reading-comprehension model to predict the answer.

2.3 Large Language Models on Complex Reasoning

LLMs have shown reasoning abilities over several tasks, such as multi-hop QA (Bang et al., 2023), commonsense reasoning (Liu et al., 2022), and table QA (Chen, 2022). Chain-of-thought (CoT) (Wei et al., 2022) leverages a series of intermediate reasoning steps, achieving better reasoning performance on complex tasks. Jin and Lu (2023) proposed a framework called Tabular Chain of Thought (Tab-CoT) that can perform step-by-step reasoning on complex tableQA tasks by creating a table without fine-tuning by combining the table header with related column names as a prompt. Khot et al. (2022) proposed an approach called Decomposed Prompting to solve complex tasks by decomposing them into simple sub-tasks that can be delegated to a shared library of prompting-based LLMs dedicated to these sub-tasks.

However, these studies only decomposed questions into sub-questions and the latter sub-questions always rely on previous sub-questions. When the previous sub-questions are incorrectly answered, the latter sub-questions are also prone to be incorrectly answered.

3 GenDec

As discussed in the preceding section, previous QD-based QA methods fail to solve the error-propagation problem during the answer reasoning process as they decompose questions into sub-questions. GenDec's approach consists of three main components: (1) a generative QD module, to generate independent sub-questions with supporting facts; (2) a sub-question-enhanced paragraph-filtering module, that serves both the QD and QA modules; and (3) a sub-question enhanced QA module, which fuses features of sub-questions for QA and supporting-facts prediction. Figure 2 shows the overall framework of GenDec.

3.1 Question Decomposition Module

We explore different model architectures for the QD module, i.e., generative language models (e.g., BART, T5), LLMs, and traditional syntactic-

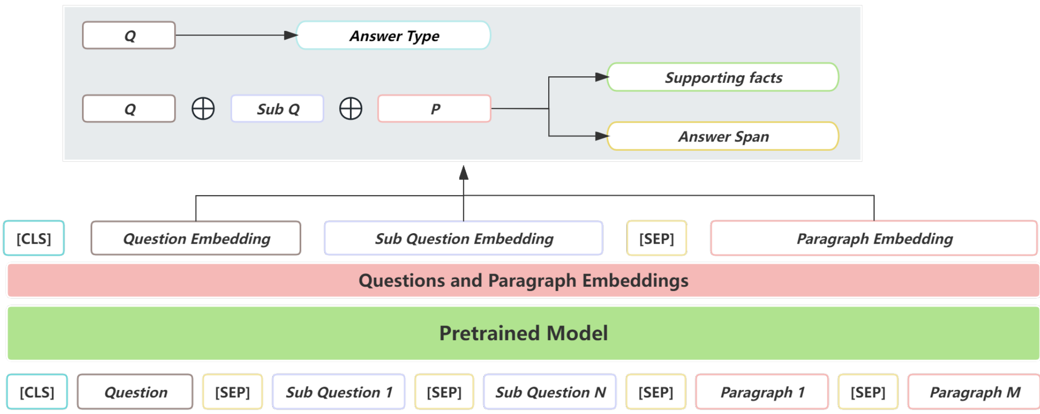

Figure 3: Architecture of QA module.

parsing models. We use BART-large (Lewis et al., 2019) and T5-large (Raffel et al., 2020) as the generative language models in GenDec. Considering the computing resources and model availability, we also use LLaMA-7B (Touvron et al., 2023) with the Low-Rank Adaptation (LoRA) technique (Hu et al., 2021a) for training an LLM-based QD, as a design alternative for evaluation. Finally, we make use of syntactic parsing, including constituency parsing and dependency parsing, to directly break multi-hop questions into sub-questions to compare the impact of not incorporating supporting facts with other generative QD-based QA models.

3.1.1 Generative Question Decomposition

To ensure the sub-questions are answerable by the QA module, we train a text-to-text generation model on the sub-question dataset from HotpotQA Khot et al. (2021).

We use BART-large and T5-large models as backend models and fine-tune them on the sub-question dataset to generate sub-questions. We use the supporting facts p and question q as input to train a question-generator model $G : (p, q) \Rightarrow sub\_qs$, where sub_qs is the generated sub-question set. Such a generator, G, produces the two sub-questions in the example in Figure 1. The details of finetuning T5-large and BART-large are given in Appendix A.

3.1.2 Syntactic Parsing for Question Decomposition

For the QD comparison on not incorporating supporting facts, we use traditional syntactic parsing, including constituency parsing and dependency parsing, which are directly applied to break multi-hop questions based on their sentence structure.

We use syntactic parsing to recognize the specific constituents of multi-hop questions, such as clauses, noun phrases (NPs), or conjunctions, by constructing a constituency parsing tree and dependency-parsing graph and searching for potential sub-questions. Multi-hop questions can generally be divided into two types: bridge and comparison questions. Bridge questions are complex sentences that contain subordinate clauses, while comparison questions are compound sentences that contain coordinate conjunctions such as "and", "or", and "but".

Sub Question Extraction For bridge questions, we use Benepar (Kitaev and Klein, 2018a), a state-of-the-art (SOTA) model for constituency parsing to recognize each constituent in multi-hop questions in a constituency-parsing tree from top to bottom and apply a depth-first search (DFS) algorithm to search for potential sub-question. For comparison-type questions, Gao et al. (2021b) proposed an ABCD model that constructs a graph for decomposing coordinate sentences. We use Benepar and ABCD for decomposing bridge and comparison questions, respectively. Further details are provided in Appendix B.

3.1.3 Large Language Models in Question Decomposition

Differently from typical QD-based QA models, we also explore leveraging powerful LLMs with few-shot prompting as a plugin for GenDec to decompose complex multi-hop questions and reason with the help of supporting facts. Despite the remarkable advancements brought about by LLMs, commercial models come with certain limitations that hinder transparent and open research. Therefore, we fine-tune LLaMA-7B (Touvron et al., 2023) with LoRA (Hu et al., 2021a) under low resource conditions as our LLM of use[1]. The details of finetuning LLaMA are presented in Appendix A.

3.2 Sub-question-enhanced Paragraph Retrieval

Multi-hop question answering takes textual context into account and usually, MQA datasets include multiple paragraphs as question context (e.g., HotpotQA and 2WikiMultiHopQA datasets include 10 paragraphs per question). However, including all such paragraphs is not ideal due to noise and size (length). Therefore, paragraph retrieval plays a vital role in both QA and QD modules, since GenDec utilizes information from sub-questions and can thus focus on the more relevant data.

We propose sub-question-enhanced paragraph retrieval (**SPR**), which utilizes an encoder and a classification head to compute scores for each paragraph. Given a k-hop question Q, generated k sub-questions $q_1, ...q_k$, and a candidate set with n passages as $\mathcal{P} = \{p_1, p_2, ..., p_n\}$, SPR aims to retrieve a relevant paragraph set $(\hat{p}_1, \hat{p}_2, ..., \hat{p}_k)$ that relates to the k sub-questions and the k-hop question Q. Most existing work formulates it as a one-step or two-step sequence labeling task, classifying every passage $p_i \in \mathcal{P}$ as relevant or not.

A passage $p_i \in \mathcal{P}$ corresponds to the question Q and j-th sub-question $q_j \in \mathcal{S}$. Consequently, we also denote the output score of SPR as $S(\hat{p}_i|Q, q_j)$, given the concatenated sequence of question, sub-question, and passages identified so far, $(Q, q_j, \hat{p}_i)$.

We use the DeBERTa model (He et al., 2021) as an encoder to derive embeddings for the concatenated sequence $(Q, q_j, \hat{p}_i)$ and the output $\acute{o}_i \in \mathbb{R}^n$. Subsequently, a fully connected layer is added after DeBERTa to project the final dimension of the "[CLS]" representations of these embeddings into a 2-dimensional space, representing "irrelevant" and "relevant" respectively. The logit in the "relevant" side serves as the score for each paragraph. This scoring process is denoted by a function $S(\hat{p}_i|Q, q_j)$. In SPR, we optimize the classification of each combination of question, sub-question, and paragraph using Cross-Entropy loss.

$$\mathcal{L}_j = -\sum_{q_i \in \mathcal{S}} \sum_{\hat{p}_i \in \mathcal{P}} l_{j,p} log S(\hat{p}_i|Q, q_j) + (1 - l_{j,p}) log(1 - S(\hat{p}_i|Q, q_j)) \tag{1}$$

where $l_{j,p}$ is the label of $\hat{p}_i$ and $S(\hat{p}_i|Q, q_j)$ is the score function predicted by the model.

Thus, we train a paragraph retrieval model based on DeBERTa (He et al., 2021) to execute binary classification and rank the scores of paragraphs containing the gold supporting facts.

3.3 Sub-question-enhanced QA module

In the QA module, we use multi-task learning to simultaneously predict supporting facts, and extract answer spans by incorporating sub-questions. In order to better evaluate the role of sub-question incorporation, we do not include other additional modules in our model. Instead, we focus on the effects of sub-question incorporation on the performance of the QA module. Additionally, as both HotpotQA and 2WikiMultiHopQA datasets also contain questions with yes/no answers, a common scenario, we include an answer type task. The architecture of our QA module is illustrated in figure 3.

The QA module obtains an initial representation by first combining all retrieved paragraphs into context C, which is concatenated with question Q and sub-questions $\{Sub\_Qs\}$ and fed into DeBERTa. We denote the encoded question and sub-question representations as $\mathbf{Q} = \{\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_{Q-1}\} \in \mathbf{R}^{m \times d}$ and the encoded context representation as $\mathbf{C} = \{\mathbf{c}_0, \mathbf{c}_1, ..., \mathbf{c}_{C-1}\} \in \mathbf{R}^{C \times d}$, where Q is the length of the question. Each $\mathbf{q}_i$ and $\mathbf{c}_j \in \mathbf{R}^d$.

$$\begin{aligned}
\mathbf{P}^i &= \text{DeBERTa}\left(S^{(i)}[d:]\right) \\
\mathbf{sub_q}^i &= \text{DeBERTa}\left(Sub_Q^{(i)}[d:]\right) \\
\mathbf{q} &= \text{DeBERTa}(\mathbf{Q}),
\end{aligned} \tag{2}$$

where $P^{(i)} \in \mathbf{R}^d$, $Sub\_Q^{(i)} \in \mathbf{R}^d$, $\mathbf{Q} \in \mathbf{R}^d$ respectively denote the i-th paragraph, sub-question, and question representations.

To extract answer spans, we use a linear prediction layer on the contextual representation to identify the start and end positions of answers and employ cross-entropy as the loss function. The corresponding loss terms are denoted as $\mathcal{L}_{start}$ and $\mathcal{L}_{end}$, respectively.

[1]https://huggingface.co/decapoda-research/llama-7b-hf

The classification loss for the supporting facts is denoted as $\mathcal{L}_{sup}$, and we jointly optimize all of these objectives in our model.

We also introduce an answer-type classification module trained with cross-entropy loss function.

$$\mathcal{L}_{type} = \mathbb{E}[-\sum_{i=1}^{3} y_i^{type} log(\hat{y}_i^{type})] \tag{3}$$

where $\hat{y}i^{fine}$ denotes the predicted probability of question types classified by our model, and yi^{fine} represents the corresponding one-hot encoded ground-truth distribution. y_i^{type} has three values: 0 denotes a negative answer, 1 denotes a positive answer, and 2 denotes the answer is a span.

The multi-task prediction model's total loss is:

$$\mathcal{L}_{reading} = \lambda_1 \mathcal{L}_{type} + \lambda_2(\mathcal{L}_{start} + \mathcal{L}_{end}) + \lambda_3 \mathcal{L}_{sup} \tag{4}$$

Similarly, we set λ_1, λ_2, and λ_3 all to 1, giving equal importance to each module for multitask learning. The implementation details of the Sub-question-enhanced QA module are described in Appendix A.

4 Experiments and Analysis

This section describes the different utilized datasets to analyse the different characteristics of the problem and our experimental setup.

4.1 Datasets

Question Answering (QA) We evaluate GenDec on the 2WikiMultiHopQA (Ho et al., 2020b) and HotpotQA (Yang et al., 2018a) datasets, which contain 160K and 90K training instances. These two multi-hop QA datasets consist of questions, answers, supporting facts, and a collection of 10 paragraphs as context per question.

Question Decomposition (QD) To train and evaluate GenDec's QD module, we use the sub-questions and answers data processed from the multi-hop HotpotQA dataset Khot et al. (2021) - here named SQA for clarity. These sub-questions are relatively high quality, in that we are able to use them to train a sub-question generator that achieves high task performance on multi-hop QD.

Sub-question Reasoning To evaluate the reasoning ability of GenDec, we also utilize a human-verified sub-question test dataset derived from HotpotQA Tang et al. (2020a) - here named HVSQA for clarity; which provides a strong benchmark to evaluate QA models in answering complex questions via sub-question reasoning.

4.2 Experiment Results

4.3 Quantitative Analysis

We use Exact Match (EM) and F1 scores as evaluation metrics for answer span prediction and supporting facts prediction on the HotpotQA and 2WikiMultiHopQA datasets to compare the performance of GenDec with that of QD-based, GNN-based, and other SOTA QA models. As shown in Table 1, GenDec outperforms all models in both metrics, including the strong baseline consisting of our Question Decomposition method combined with HGN-large (Fang et al., 2019b) (itself a strong GNN-based QA model), on the HotpotQA dataset. The bottom section of the table also shows GenDec also outperforms previous work on the 2WikiMultiHopQA dataset. Table 2 shows the SOTA paragraph retrieval performance of our SPR method against previous strong paragraph retrieval model baselines. Table 3 shows the performance of GenDec and baselines models on the HVSQA dataset (human-verified sub-questions). GenDec achieves SOTA performance compared with the other QA models. Moreover, it is important to note that GenDec also outperforms all other models on sub-question reasoning (1 and 2), which highlights the benefits of our approach in reasoning chains. Lastly, with the help of our QD module, relative F1 scores are boosted by $+6.82\%$ and EM by $+5.45\%$ compared with ONUS (Perez et al., 2020b), which is also a QD-based model. We further verify the effectiveness of GenDec's QD module in an ablation study discussed in the next section.

4.4 Ablation Studies

To evaluate the impact of GenDec's QD module, we conduct an ablation study testing the performance of answering all sub-questions and original questions, with and without the QD module. The results, shown in Table 3, indicate that the QD module shows consistent and significant improved results; improving the F1 score and EM by 3.36 and 2.16, respectively, in the original QA. In answering intermediate answers to sub-questions GenDec w/ QD also improves over w/o QD (improving the F1 score and EM by 2.07 and 3.78, and 4.49 and 4.45 on sub-questions 1 and 2 respectively). The results indicate that the QD module plays an important role

Model	Ans		Sup		Joint	
	EM	F1	EM	F1	EM	F1
HotpotQA test set						
QD-based QA Models						
DecompRC (Min et al., 2019b)	55.20	69.63	-	-	-	-
ONUS (Perez et al., 2020a)	66.33	79.34	-	-	-	-
GNN-based Models						
DFGN (Xiao et al., 2019)	56.31	69.69	51.50	81.62	33.62	59.82
SAE-large (Tu et al., 2020)	66.92	79.62	61.53	86.86	45.36	71.45
C2F Reader(Shao et al., 2020b)	67.98	81.24	60.81	87.63	44.67	72.73
HGN-large (Fang et al., 2019a)	69.22	82.19	62.76	88.47	47.11	74.21
BRF-graph (Huang and Yang, 2021)	70.06	82.20	61.33	88.41	45.92	74.13
AMGN+ (Li et al., 2021)	70.53	83.37	63.57	88.83	47.77	75.24
Other SOTA Models						
FE2H on ALBERT (Li et al., 2022b)	71.89	84.44	64.98	89.14	50.04	76.54
PCL (Deng et al., 2022a)	71.76	84.39	64.61	89.20	49.27	76.56
Smoothing R3 (Yin et al., 2023)	72.07	84.34	65.44	89.55	49.73	76.69
QD + HGN-large	71.73	84.23	64.32	89.46	49.22	75.63
GenDec (DeBERTa-large)	**72.39**	**84.69**	**65.88**	**90.31**	**50.34**	**77.48**
2WikiMultiHotpotQA test set						
CRERC (Fu et al., 2021)	69.58	72.33	82.86	90.68	49.80	58.99
NA-Reviewer (Fu et al., 2022)	76.73	81.91	89.61	94.31	52.75	65.23
BigBird-base model (Ho et al., 2023)	74.05	79.68	77.14	92.13	39.30	63.24
GenDec (DeBERTa-large)	**86.47**	**88.15**	**93.28**	**96.45**	**56.87**	**68.38**

Table 1: Performance of different QA models on test distractor settings of HotpotQA and 2WikiMultihopQA datasets. GenDec outperforms all QD-based and other GNN-based QA models.

Model	**EM**	**F1**
SAE_{large} (Tu et al., 2020)	91.98	95.76
S2G_{large} (Wu et al., 2021)	95.77	97.82
FE2H_{large} (Li et al., 2022a)	96.32	98.02
C2FM_{large} (Yin et al., 2023)	96.85	98.32
SPR (ours)	**97.13**	**98.78**

Table 2: Comparison of our sub-question enhanced paragraph retriever with previous baselines on HotpotQA dev set.

in GenDec in not only its QA ability, but also in intermediate answer reasoning to support answering the final question. We also evaluate the impact of different backend models in our QD module and compare the performances of T5-large, BART-large, SynDec, and LLaMA-7B on the dev distractor setting of HotpotQA. LLaMA-7B achieves the best overall performance on both answer span prediction and supporting facts prediction since it had the best QD performance, with BART-large (even being a much smaller model) presenting a very competitive performance, as shown in Table 6.

4.5 Qualitative Analysis

We compare the QD performance of different LMs (Table 6 in the Appendix) and their impact on QA performance (Table 4). LLaMA-7B achieves SOTA performance in F1 score and EM (6.81 and 9.47 higher, respectively). We also compare F measure, ROUGE-1, ROUGE-L, and BLEU scores of generated sub-questions and LLaMA-7B significantly improves the quality of sub-questions reaching 80.57, 69.48, and 31.32, respectively. Likely due to the different max input lengths of T5-large (512) and BART-large (1024), BART outperforms it since some inputs contain many sentences (including both the multi-hop questions themselves and their supporting facts). GenDec with LLaMA-7B also improves QA performance on the distractor setting dev set, as shown in Table 4, but not substantially. We also evaluate the impact of sub-question

Model	Q_ori		Q_sub1		Q_sub2	
	F1	EM	F1	EM	F1	EM
CogQA	67.82	53.2	69.65	58.6	68.49	54
DFGN	71.96	58.1	68.54	54.6	60.83	49.3
DecompRC	77.61	63.1	75.21	61	70.77	56.8
ONUS	79.25	67.43	77.56	63.89	72.21	57.62
GenDec w/o QD	82.81	70.72	87.45	72.65	80.12	70.38
GenDec w QD	**86.17**	**72.88**	**90.52**	**76.43**	**84.61**	**74.83**

Table 3: Performance comparison between GenDec (with and without the QD module) and other QA models on HVSQA (Tang et al., 2020a), a human-verified sub-question test dataset from HotpotQA.

Model	Ans		Sup		Joint	
	EM	F1	EM	F1	EM	F1
GenDec (BART-large)	70.13	84.47	63.51	89.47	46.12	75.52
GenDec (T5-large)	69.94	84.11	63.32	89.35	46.02	75.69
GenDec (SynDec)	69.34	83.92	61.35	88.21	45.26	74.89
GenDec (LLaMA-7B)	**70.23**	**84.76**	**63.41**	**89.78**	**46.28**	**76.05**

Table 4: Performance of QD module with different generative LMs on SQA Khot et al. (2021), distractor dev set of sub-questions processed from HotpotQA.

Model	Ans		Sup		Joint	
	EM	F1	EM	F1	EM	F1
ChatGPT w/o QD	51.08	74.53	60.61	**87.96**	30.95	65.55
ChatGPT w QD	**56.24**	**76.28**	**60.74**	87.85	**34.16**	**67.01**

Table 5: Performance of ChatGPT (with and without QD) on 1000 samples from HotpotQA's dev set distractor setting data.

on LLM reasoning in table 5. Further analysis of ChatGPT is discussed in Appendix D.

4.6 Error Analysis

We conduct an error analysis of GenDec's performance by selecting 20 samples from the dev set for evaluation, with 10 correct and 10 incorrect answers to analyze the impact of supporting facts prediction on QD and QA. We find a total of 12 correct supporting facts predictions and 8 incorrect supporting facts predictions among these 20 samples. For the 12 correct Supporting fact Predictions (SPs), we obtain 10 correct and 2 incorrect QD results. For the 10 correct QD results, we finally obtain 8 correct answers and 2 incorrect answers. And for the 8 incorrect SPs, we obtain 7 incorrect QD results and 6 incorrect answers. We then also select 20 samples from the dev set, with 10 correct and 10 incorrect QD results. For the 10 correct QD results, we obtain 8 correct answers, while for the 10 incorrect QD results, we obtain 5 correct answers. We list 6 samples of our selection in Table 8 in the Appendix, showing that questions be well answered based on high-quality QD.

5 Conclusion

We proposed GenDec, a generative-based QD method that generates independent sub-questions based on incorporating supporting facts. Intuitively, the supporting facts inform the reasoning chain of multi-hop questions. To explore this intuition, we train a sub-question-enhanced paragraph retrieval and QA module that incorporates sub-questions and shows that it significantly improves QA. We also explore the possible role of LLMs in QD and QA tasks. Lastly, while GenDec reaches new SOTA results in multi-hop QA, it can still face errors due to incorrect supporting fact predictions influencing the model to incorrectly predict both sub-questions and final answers.

6 Limitations

In this paper, we focus on the impact of QD in multi-hop QA, where the answers to most questions can be decomposed into several independent sub-questions via the fusion of supporting facts. Although GenDec performs very well on QD and QA, one of its limitations is that it is still sensitive to errors in paragraph filtering. The QD results would be affected when given incorrect paragraphs are selected. For future work, we plan to focus on tackling this problem.

References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Wenhu Chen. 2022. Large language models are few(1)-shot table reasoners. In *Findings*.

Zhenyun Deng, Yonghua Zhu, Yang Chen, Qianqian Qi, Michael Witbrock, and Patricia Riddle. 2022a. Prompt-based conservation learning for multi-hop question answering. *arXiv preprint arXiv:2209.06923*.

Zhenyun Deng, Yonghua Zhu, Yang Chen, M. Witbrock, and Patricia J. Riddle. 2022b. Interpretable amr-based question decomposition for multi-hop question answering. In *International Joint Conference on Artificial Intelligence*.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. *arXiv preprint arXiv:1905.05460*.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *North American Chapter of the Association for Computational Linguistics*.

Yuwei Fang, S. Sun, Zhe Gan, Rohit Radhakrishna Pillai, Shuohang Wang, and Jingjing Liu. 2019a. Hierarchical graph network for multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2019b. Hierarchical graph network for multi-hop question answering. *arXiv preprint arXiv:1911.03631*.

Ruiliu Fu, Han Wang, Xuejun Zhang, Jun Zhou, and Yonghong Yan. 2021. Decomposing complex questions makes multi-hop QA easier and more interpretable. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 169–180, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ruiliu Fu, Han Wang, Jun Zhou, and Xuejun Zhang. 2022. Na-reviewer: Reviewing the context to improve the error accumulation issue for multi-hop qa. *Electronics Letters*, 58(6):237–239.

Yanjun Gao, Ting hao Huang, and Rebecca J. Passonneau. 2021a. Abcd: A graph framework to convert complex sentences to a covering set of simple sentences.

Yanjun Gao, Ting-Hao Huang, and Rebecca J. Passonneau. 2021b. ABCD: A graph framework to convert complex sentences to a covering set of simple sentences. 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 3919–3931, Online. Association for Computational Linguistics.

Xiao-Yu Guo, Yuan-Fang Li, and Gholamreza Haffari. 2022. Complex reading comprehension through question decomposition. *ArXiv*, abs/2211.03277.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *ICLR 2021*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020a. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020b. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2023. Analyzing the effectiveness of the underlying reasoning tasks in multi-hop question answering. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1163–1180, Dubrovnik, Croatia. Association for Computational Linguistics.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021a. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Xixin Hu, Yiheng Shu, Xiang Huang, and Yuzhong Qu. 2021b. Edg-based question decomposition for complex question answering over knowledge bases. In *International Workshop on the Semantic Web*.

Yongjie Huang and Meng Yang. 2021. Breadth first reasoning graph for multi-hop question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5810–5821.

Ziqi Jin and Wei Lu. 2023. Tab-cot: Zero-shot tabular chain of thought. *arXiv preprint arXiv:2305.17812*.

Tushar Khot, Daniel Khashabi, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2021. Text modular networks: Learning to decompose tasks in the language of existing models. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, page 1264–1279.

Tushar Khot, H. Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *ArXiv*, abs/2210.02406.

Nikita Kitaev and Dan Klein. 2018a. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018b. Constituency parsing with a self-attentive encoder. *arXiv preprint arXiv:1805.01052*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Ronghan Li, Lifang Wang, Shengli Wang, and Zejun Jiang. 2021. Asynchronous multi-grained graph network for interpretable multi-hop reading comprehension. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3857–3863. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Xin-Yi Li, Wei-Jun Lei, and Yu-Bin Yang. 2022a. From easy to hard: Two-stage selector and reader for multi-hop question answering. *ArXiv preprint*, abs/2205.11729.

Xin-Yi Li, Weixian Lei, and Yubin Yang. 2022b. From easy to hard: Two-stage selector and reader for multi-hop question answering. *ArXiv*, abs/2205.11729.

Jiacheng Liu, Skyler Hallinan, Ximing Lu, Pengfei He, Sean Welleck, Hannaneh Hajishirzi, and Yejin Choi. 2022. Rainier: Reinforced knowledge introspector for commonsense question answering. *ArXiv*, abs/2210.03078.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019a. Multi-hop reading comprehension through question decomposition and rescoring. In *ACL*.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019b. Multi-hop reading comprehension through question decomposition and rescoring. *ArXiv*, abs/1906.02916.

Ethan Perez, Patrick Lewis, Wen tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020a. Unsupervised question decomposition for question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Ethan Perez, Patrick Lewis, Wen tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020b. Unsupervised question decomposition for question answering. In *EMNLP*.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150, Florence, Italy. 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.

Nan Shao, Yiming Cui, Ting Liu, Shijin Wang, and Guoping Hu. 2020a. Is graph structure necessary for multi-hop question answering? *arXiv preprint arXiv:2004.03096*.

Nan Shao, Yiming Cui, Ting Liu, Shijin Wang, and Guoping Hu. 2020b. Is graph structure necessary for multi-hop reasoning? *ArXiv*, abs/2004.03096.

A. Talmor and J. Berant. 2018. The web as a knowledge-base for answering complex questions. In *North American Association for Computational Linguistics (NAACL)*.

Yixuan Tang, Hwee Tou Ng, and Anthony K. H. Tung. 2020a. Do multi-hop question answering systems know how to answer the single-hop sub-questions? In *Conference of the European Chapter of the Association for Computational Linguistics*.

Yixuan Tang, Hwee Tou Ng, and Anthony KH Tung. 2020b. Do multi-hop question answering systems know how to answer the single-hop sub-questions? *arXiv preprint arXiv:2002.09919*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9073–9080.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020a. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020b. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Bohong Wu, Zhuosheng Zhang, and Hai Zhao. 2021. Graph-free multi-hop reading comprehension: A select-to-guide strategy. *ArXiv preprint*, abs/2107.11823.

Yunxuan Xiao, Yanru Qu, Lin Qiu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. *ArXiv*, abs/1905.06933.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018a. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhangyue Yin, Yuxin Wang, Xiannian Hu, Yiguang Wu, Hang Yan, Xinyu Zhang, Zhao Cao, Xuanjing Huang, and Xipeng Qiu. 2023. Rethinking label smoothing on multi-hop question answering. In *China National Conference on Chinese Computational Linguistics*, pages 72–87. Springer.

Ben Zhou, Kyle Richardson, Xiaodong Yu, and Dan Roth. 2022. Learning to decompose: Hypothetical question decomposition based on comparable texts. *ArXiv*, abs/2210.16865.

A Implementation Details

Question Decomposition We use the pre-trained T5-large and BART-large models with max_input_length L = 512, and max_output_length O = 64. During training, we use the Adam optimizer in the QD modules and set batch size to 32 and learning rate to 5e-5. All experiments utilized two TITAN RTX GPUs.

Question Answering We choose DeBERTa-v2-large as backend model and set number of epochs to 12 and batch size to 4. We use BERTAdam with learning rate of 5e-6 for the optimization and set max position embeddings to 1024.

Fine-tuning LLaMA To fine-tune LLaMA, considering computing resources, we select LLaMA-7B as backbone, batch size of 4, number of epochs is 3, learning rate is 3e-4, LoRA alpha of 16, and LoRA dropout of 0.05.

B SynDec

Implementation Details We leverage the well-trained models from Benepar[2] and ABCD[3] to build the constituency-parsing tree and dependency-parsing graph. For bridge-type QD, the threshold t is the key hyper-parameter, which is set to 5.

Bridge Question Decomposition Bridge questions are typically compound sentences that contain a clause that is modified by a relative clause. This enables the extraction of a clause or NP from the original question and treating it as a sub-question.

Constituency parsing We use Benepar (Kitaev and Klein, 2018b), a SOTA model for constituency parsing, to recognize each constituency in multi-hop questions in the constituency-parsing tree from top to bottom and output the sub-tree, the label of which belongs to the label set L = {NML, S, SBAR, SQ, SINV, NP}, where and other labels represent different types of clauses, e.g., subordinate clause (SBAR) and declarative clause (SINV).

Sub-question Extraction We used a search algorithm to find the sub-question in the constituency-parsing tree, where each node represents a text span

[2]https://spacy.io/universe/project/self-attentive-parser
[3]https://github.com/serenayj/ABCD-ACL2021

Models	Metric			
	F Measure	Rouge1	Rouge-L	BLEU
BART-LARGE	74.41	73.85	62.68	26.94
T5-LARGE	72.85	71.27	60.12	24.37
LLAMA-7B	**81.32**	**80.57**	**69.48**	**31.22**

Table 6: Generative QD performance of different generative LMs on test instances of HOTPOTQA sub-questions. Results are averaged on 1549 test instances.

GENDEC (OURS)
Sub-question 1: Which South Korean boy group had their debut album in 2014? **Sub-question 2:** WINNER was formed by who?
SYNDEC (SYNTACTIC PARSING)
Sub-question 1: a South Korean boy group that was formed by who? **Sub-question 2:** 2014 S/S is the debut album of ?
MODULARQA (Khot et al., 2021)
Sub-question 1: What is the name of the South Korean group that had their debut album in 2014? **Sub-question 2:** What was WINNER formed by?
DECOMPRC (Min et al., 2019b)
Sub-question 1: 2014 S/S is the debut album of which South Korean boy group? **Sub-question 2:** which formed by who ?

Table 7: QD examples produced by {GENDEC, SYNDEC, MODULARQA, DECOMPRC} for question "2014 S/S is the debut album of a South Korean boy group that was formed by who?".

of the original sentence with a specific tag. Basically, it searches a node with any of the labels in L from a root of the tree in a depth-first manner.

DFS algorithm for Bridge QD We introduce the DFS algorithm to find the sub-sentence from the root node to leaves in the constituency-parsing tree, where each node represents a text span of the original sentence with a specific tag. To prevent a multi-hop question from being decomposed into too many incomplete text segments (some clauses may contain a shorter clause) and output the right constituency, we used the following searching rules in the DFS algorithm:
1) The DFS algorithm starts from the root node of the constituency-parsing tree and visits all the children of the current node.
2) If the label of Node$_i$ is not in L, continue searching its children nodes.
3) If the label Node$_i$ is in L, and the length of the text span of Node$_i$ is larger than threshold t, the algorithm outputs this node as a potential sub-question and stops the loop.
4) If the label of Node$_i$ is an NP, but none of the children is in L, and the length of the text span of Node$_i$ is larger than t, then the algorithm output this node as a potential sub-question and stops the loop.

Figure 4 shows an output of Benepar and the search process of the DFS algorithm. The blue arrow shows the search direction of the DFS algorithm. The algorithm finds the NP *"the woman who portrayed Corliss Archer in the film kiss and tell"* with the tag `'NP'`, where the label of its child is SBAR. Therefore, we go to the node SBAR and find that all the labels of its children are in L. It then finishes searching the parsing tree and returns to the parent node and outputs it as the sub-question. The pseudo-code of the DFS algorithm is shown in Algorithm 1.

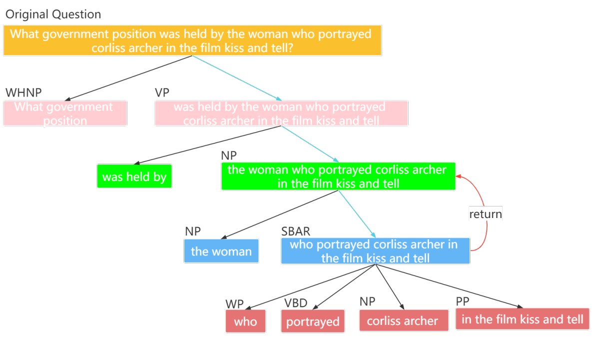

Figure 4: Constituency-parsing tree output from Benepar and search process of DFS algorithm

Initialization: *threshold* $t \leftarrow 5$
Initialization: $Clause\_labels \leftarrow$

$['NML','S','SBAR','SQ','SINV']$

Initialization: $NP\_label \leftarrow' NP'$
$StartfromRoot$
repeat
  $Subtree \leftarrow Root.child$
  if $Subtree.label$ in Clause_labels and $Subtree.length >= t$ **then**
    Output $Subree$ and Stop loop
  else if $Subtree.label$ == NP_label and $Subtree.length >= t$ **then**
    $Continue$
  else
    $Return$
  end if
  $ROOT \leftarrow Subtree$
until $Subtree.length$ <= 1 or $Subtree.node$ == Leaf

C DFS algorithm for Bridge QD

Comparison QD A comparison question is a coordinate sentence with conjoined verb phrases. To decompose the question, certain words from the original sentence need to be dropped or retained and rewritten into two sub-sentences that do not overlap.

For example, if we decompose the question *"Were Pavel Urysohn and Leonid Levin known for the same type of work?"*, we need to recognize the two subjects *"Pavel Urysohn"* and *"Leonid Levin"*. Subjects should be retained and the coordinate conjunction ('cc') *"and"* should be dropped.

We used the ABCD model (Gao et al., 2021a), which accepts, breaks, copies, and drops words from the complex coordinate sentences and produces sub-sentences by constructing a dependency-parsing graph and using the DFS algorithm to search and segment graph. We applied the well-trained ABCD model to decompose comparison-type questions.

D ChatGPT on Multi-hop QA

We also evaluated the performance of ChatGPT with and without QD on 1000 samples of dev distractor settings. Figure 5 shows the used with QD and without QD prompt settings. We selected the 1-shot setting in which ChatGPT is given one example from the training set with two prompts, one is reasoning over sub-questions and the other is directly reasoning answers. As shown in Table 5, ChatGPT with additional sub-question information performs better than without sub-questions. ChatGPT with QD prompting achieves higher answer span extraction on the F1 score (76.28) and EM (56.24). However, both ChatGPT with QD prompting and ChatGPT without QD prompting are still lower than current QA models.

Original Question	Sub-questions	Intermediate Answers	Answer
Were Scott Derrickson and Ed Wood of the same nationality?	What was Scott Derrickson's nationality? What was Ed Wood's nationality? ✓	American ✓	Yes ✓
What government position was held by the woman who portrayed Corliss Archer in the film Kiss and Tell?	Who portrayed Corliss Archer in Kiss and Tell? What position was held by Shirley Temple? ✓	Shirley Temple ✓	Chief of Protocol ✓
The director of the romantic comedy B̈ig Stone Gapïs based in what New York City neighborhood?	Who is the director of the romantic comedy Big Stone Gap? In what New York City neighborhood is Adriana Trigiani based? ✓	Adriana Trigiani ✓	Greenwich Village ✓
Are Random House Tower and 888 7th Avenue both used for real estate?	The Random House Tower used as real estate? What is 888 7th Avenue used also for? ✗	Used ✗	No ✗
What is the name of the executive producer of the film that has a score composed by Jerry Goldsmith?	What is the name of the film of which Jerry Goldsmith composed the score? Which co-writer of Alien was also an executive producer? ✓	Alien ✓	Francis Ford Coppola ✗
Alvaro Mexia had a diplomatic mission with which tribe of indigenous people?	Who was given a diplomatic mission to the native populations living south of St. Augustine and in the Cape Canaveral area? What is the name of the indigenous tribe of Florida? ✗	Alvaro Mexia ✗	Indigenous peoples of Florida ✗

Table 8: Examples of 3 correct samples and 3 incorrect samples from dev set of HotpotQA

1-shot prompting with QD

Input: Here is an example:

Q1: [Question], Supporting Facts: [SPs], Sub-Questions:[Sub_Qs], The Answer is [Answer].

Now given a new example, Q: [Question], Supporting Facts: [SPs]. let's think step by step, please decompose the question into sub-questions based on supporting facts and give the answer:

Output: The answer is :[Answer]

1-shot prompting

Input: Here is an example:

Q1: [Question], Supporting Facts: [SPs], The Answer is [Answer].

Now given a new example, Q: [Question], Supporting Facts: [SPs]. let's think step by step, please use them to give the answer:

Output: The answer is :[Answer]

Figure 5: Prompting examples of different settings.