Exploring and Evaluating the Reasoning Steps in Multi-hop Machine Reading Comprehension

by

HO THI XANH

Dissertation

submitted to the Department of Informatics

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

The Graduate University for Advanced Studies, SOKENDAI

September 2024

Abstract

Machine reading comprehension (MRC) aims to teach computers to read and understand unstructured text to answer a given question. Various directions have emerged from the foundational structure of the MRC task, including multi-hop MRC and conversational MRC. We note that multi-hop MRC requires models to undergo multiple steps to derive the ultimate answer, making it inherently involve reasoning steps within the question-answering (QA) process. This characteristic offers valuable avenues for exploration and model evaluation. Consequently, this thesis endeavors to explore and assess the reasoning steps within multi-hop MRC.

There are two primary issues associated with the multi-hop MRC task. The first pertains to reasoning shortcuts, wherein a machine learning model can answer questions without engaging in multi-hop reasoning. Consequently, the model may lack a genuine understanding of the question's context and underlying concepts, thereby constraining its capacity to offer meaningful responses. This compromise undermines the reliability and effectiveness of the model, particularly in tasks demanding comprehensive understanding and reasoning. The second concern pertains to the absence of explanatory information. Such information is crucial for comprehending the rationale behind models' predictions for specific questions. However, the majority of existing datasets lack comprehensive tasks designed to provide explanations for predicted answers. While there is one dataset available (the R^4C dataset), its limited size precludes its utility as a multi-hop dataset with comprehensive explanations for training end-to-end systems.

Our thesis mainly focuses on exploring and evaluating the reasoning steps in multi-hop MRC; it includes seven chapters. In Chapter 1, we discuss two prevalent issues: reasoning shortcuts and missing explanations, which serve as the primary

motivations for our research. In Chapter 2, we provide an overview of the multi-hop MRC task, encompassing the definition of reasoning steps and various representations thereof. Additionally, we explore the intricacies of reasoning shortcuts within multi-hop MRC, delineating their definition and existing methods for detection. In Chapter 3, we construct a new multi-hop dataset, 2WikiMultiHopQA, which is a large and high-quality dataset providing comprehensive explanations for predictions. Through experiments, we demonstrate that our dataset poses a challenge for multi-hop models, ensuring that multi-hop reasoning is required. In Chapter 4, we analyze the effectiveness of the reasoning steps in the question-answering process in the triple-form. We consider three aspects in our analyses: QA performance, reasoning shortcuts, and robustness. Our analyses show that the reasoning steps can help improve QA performance and reduce reasoning shortcuts, but they do not make the models more robust against adversarial questions. In Chapter 5, we enhance the existing reasoning steps (triple-form) in comparison questions, a sub-type of multi-hop questions by introducing three probing tasks: extraction, reasoning, and robustness. The current reasoning steps in triple-form are limited by the fixed schedule of Wikidata; therefore, we represent the reasoning steps in this chapter in a sub-QA format. We then evaluate top-performing models on our dataset. The results reveal that the models may not possess the ability to subtract two dates even when fine-tuned on our dataset. Additionally, we analyze the effectiveness of the reasoning steps in the sub-QA format and find that they can help to improve QA performance. In Chapter 6, we examine the behaviors of large language models (LLMs) across three stages in the QA process: question decomposition, subproblem solving, and composition. To comprehensively investigate LLMs, we consider three different scenarios for the input context: without context, with unstructured context, and with structured context. Our main finding is that the current LLMs fail to decompose complex questions into sub-questions, which is key in the QA process. In Chapter 7, we conclude our research and discuss about potential future works. The proposed datasets and analyses in this thesis contribute to building a more explainable QA system in the future. They also contribute to a better understanding of the models in the QA process when dealing with multi-hop questions.

Acknowledgments

I would like to express my special appreciation to my advisor, Professor Akiko Aizawa. I remember the time when I was an internship student in her lab. At that time, I was pretty nervous, and my English skills were quite poor. However, she listened to all my presentations patiently and gave me many helpful comments and suggestions. I wouldn't be here without her support. I also learned a lot from her hard-working attitude. It has inspired me greatly to work hard.

I thank Saku Sugawara, Florian Boudin, Katsumi Inoue, Ken Satoh, and Hideaki Takeda for being on the committee of my thesis. They have provided me with questions and feedback that help improve my research. Especially thanks to Saku Sugawara, who has helped and supported me a lot during my PhD journey. And Florian Boudin, who has discussed and motivated me in my research in the final year.

I also want to express my appreciation to all members of Aizawa Lab for their support and feedback. A special thanks to the three secretaries of the lab: Katsu-san, Takenaka-san, and Ito-san.

I also thank my teachers, seniors, and friends at the VNU University of Science for their invaluable support: Hong Nhung, Minh, Hai Van, Kien, Thach, Hong Huy, Minh Duc, and Thy Thy. I want to express special thanks to Khoa-san for your collaboration and support.

I thank my friends for their invaluable support, especially my best friends, Truc and Trang, who has listened to me and supported me. I also want to express thanks to Huan-san, a friend and former roommate, who has shown me many lessons, especially through her actions, which have represented and explained the meaning of the sentence: "When a wise person hears the Tao, this person will practice it diligently".

I would like to thank my family for always being by my side and giving me constant

love. Especially thanks to my mother, who reads books with me when I'm lazy or lost.

I'm thankful to Shen Yun dancers. They are roughly the same age as me, but their hard work inspires me a lot. Especially thanks to Melody Qin and Henry Hung, their videos have inspired me when I lose motivation, get lazy, and get bored. They also show me the meaning of the phrase "Hard work always pays off".

I would like to express my gratitude to Master Li, who has taught me the principles of the universe: Truthfulness - Compassion - Forbearance. I can't imagine how bad I will be in the future if I do not know Falun Dafa. My life has changed. I'm still on the way to assimilate myself with Truthfulness - Compassion - Forbearance. Sometimes I have not behaved well when comparing my actions with these three words, but with Truthfulness - Compassion - Forbearance in my heart, I will try my best for the rest of my life. I also thank other Falun Dafa practitioners who have shared many encouraging stories about their cultivations. Their stories have inspired me a lot in my cultivation and my life.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The long-standing goal of natural language understanding (NLU) is to develop a machine that can understand natural languages. Machine reading comprehension (MRC) is one of the most important tasks that can be used to evaluate NLU. MRC aims to teach computers to read and understand unstructured text to answer a given question. Several directions have developed based on the basic structure of the MRC task, such as multi-hop MRC and conversational MRC. The multi-hop MRC task requires a model to read and perform multi-hop reasoning over multiple paragraphs to answer a given question, while Conversational MRC emphasizes conversation more in chat interactions. We observe that multi-hop MRC task is an important potential direction for the community by the following attributes: (i) Multi-hop MRC dataset is helpful for *testing the reasoning skills* of a model. (ii) Multi-hop MRC can be used to *evaluate the explainability* of a model. (iii) Multi-hop MRC is useful in *applications* (e.g., question answering system). Therefore, in this thesis, we focus on the multi-hop MRC

task.

There are two main issues with the existing multi-hop MRC datasets. The first issue is reasoning shortcuts. We consider shortcuts as statistical correlations in the data that allow a machine learning model to achieve high performance on a task without acquiring all the intended knowledge. Currently, several types of shortcuts and biases have been detected by the community: word overlap shortcut, entity type matching, and position bias. These reasoning shortcuts make the evaluation of the model unclear, and humans still cannot obtain a good understanding of the model's abilities. For example, Figure 1.1 presents an example of reasoning shortcuts in the multi-hop MRC task. We expect the models to perform reasoning step by step to obtain the final answer, which in this case is "January 19, 1985". However, the models can use several types of shortcuts to obtain the answer without performing the expected reasoning. In this case, the models can rely on entity type matching to obtain the final answer since the question asks about the date, and there is only one available date information in the provided context.

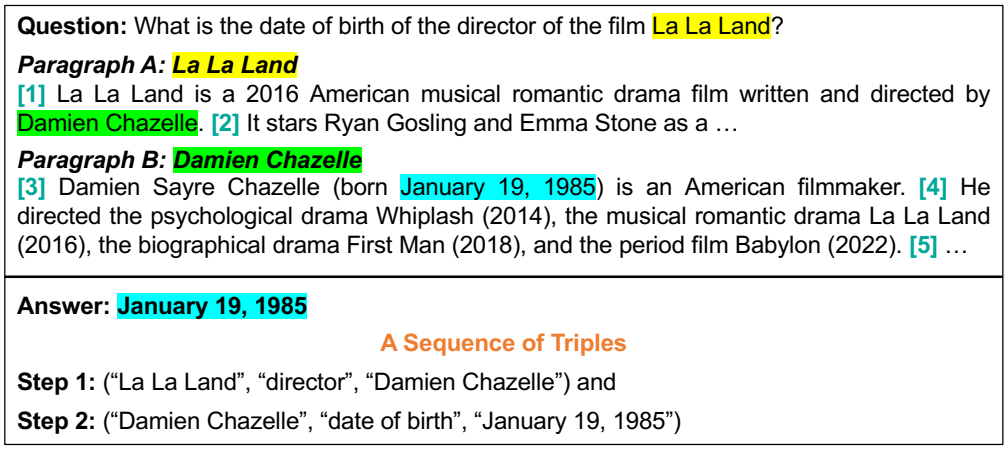

Figure 1.1: An example of reasoning shortcuts in the multi-hop MRC task.

The issue of reasoning shortcuts has been discovered by the community over the past few years. Here are some of them. Through analysis, (Chen et al., 2016) revealed that the current systems excel at matching questions to local contexts when answering the question and the CNN/Daily Mail dataset (Hermann et al., 2015) has weak reasoning and inference skills. Via adversarial, (Jia and Liang, 2017) demonstrated that the current models did not understand natural language precisely. In addition, (Sugawara et al.,

2018) showed that many datasets contain a large number of "easy" questions that can be easily answered based on the first few words of the question. Further, data analyses (Chen and Durrett, 2019; Min et al., 2019a) revealed that many examples in HotpotQA do not require multi-hop reasoning to solve.

The second issue is the missing explanation. Explanation information is important for understanding why models predict certain answers to questions. Currently, there are four multi-hop datasets over textual data: ComplexWebQuestions (Talmor and Berant, 2018), QAngaroo (Welbl et al., 2018), HotpotQA (Yang et al., 2018), and R^4C (Inoue et al., 2020). Table 1.1 summarizes the main information of these datasets. The first two datasets were created by incorporating the documents (from Web or Wikipedia) with a knowledge base (KB). Owing to their building procedures, these datasets have no information to explain the predicted answers. Meanwhile, the other two datasets were created mainly based on crowdsourcing. In HotpotQA, the authors introduced the sentence-level supporting facts (SFs) information that are used to explain the predicted answers. However, as discussed in Inoue et al. (2020), the task of classifying sentence-level SFs is a binary classification task that is incapable of evaluating the reasoning and inference skills of the model. R^4C is created based on HotpotQA and has 4,588 questions. However, the small size of the dataset implies that the dataset cannot be used as a multi-hop dataset with a comprehensive explanation for training end-to-end systems.

Task/Dataset	**Explanations**		**Size**
	Justification	Introspective	
Our work	✓	✓	192,606
ComplexWebQuestions (Talmor and Berant, 2018)	✗	✗	34,689
QAngaroo (Welbl et al., 2018)	✗	✗	53,826
HotpotQA (Yang et al., 2018)	✓	✗	112,779
R^4C (Inoue et al., 2020)	✓	✓	4,588

Table 1.1: Existing multi-hop datasets.

In this thesis, we aim to address these two issues by utilizing the reasoning steps in the QA process. We outline our hypothesis by answering the following questions:

- Research Question 1 (**RQ1**): What are the underlying reasoning steps in multi-

hop MRC? To utilize the information about reasoning steps, we should first understand what the reasoning steps are in the multi-hop MRC task. A clear definition and related information also help us establish a solid background before utilizing them.

- Research Question 2 (**RQ2**): Can we utilize the underlying reasoning steps in multi-hop MRC as evaluation tasks for the QA process? Our second research question is related to the use of reasoning steps. We want to explore whether we can utilize the reasoning steps and formalize them as a new task to help evaluate the reasoning abilities of the models.

- Research Question 3 (**RQ3**): What are the effectiveness of the reasoning steps in the QA Process? We analyze the effectiveness of the reasoning steps in both forms: triple-form and sub-questions-form. During the analyses, we consider three aspects: QA performance, reasoning shortcuts, and robustness.

- Research Question 4 (**RQ4**): What are the gaps between LLMs and humans in performing step-by-step in the QA process? We observe that in order to answer complex multi-hop questions, humans often perform the following three steps: question decomposition, subproblem solving, and composition. We aim to investigate how models behave in answering complex questions compared to the human reasoning process.

1.2 Thesis Outline

We organize our thesis by addressing the above research questions.

- **RQ1**: Regarding the first research question, we provide an overview of the multi-hop MRC task, define the reasoning steps in multi-hop MRC, and discuss issues related to reasoning shortcuts in Chapter 2.

- **RQ2**: For the second research question, we propose a new multi-hop dataset, namely 2WikiMultihopQA, which can be used to evaluate the reasoning steps of the models. We present details about the dataset construction and experiments in Chapter 3.

- **RQ3**: For the third research question, we address it in two chapters. In Chapter 4, we analyze the effectiveness of the reasoning steps in the QA process using the triple form of the reasoning steps. Meanwhile, in Chapter 5, we analyze the effectiveness of the the reasoning steps in the sub-question form.
- **RQ4**: In Chapter 6, we answer the fourth research question by measuring the gap between human reasoning processes and the models when answering multi-hop questions.
- Finally, we conclude our thesis in Chapter 7.

1.3 Contributions

The contributions of this thesis are summarized as follows:

- Chapter 2: We introduce some basic information about the MRC task and the multi-hop MRC task, including the definition of the reasoning steps and different forms used to represent them. We discuss the issues of reasoning shortcuts in the multi-hop MRC task and present existing methods for measuring shortcuts.
- Chapter 3: We present a new multi-hop QA dataset, called 2WikiMultiHopQA, which uses structured and unstructured data. In our dataset, we introduce the evidence information containing a reasoning path for multi-hop questions. The evidence information has two benefits: (i) providing a comprehensive explanation for predictions and (ii) evaluating the reasoning skills of a model. We carefully design a pipeline and a set of templates when generating a question–answer pair that guarantees the multi-hop steps and the quality of the questions. We also exploit the structured format in Wikidata and use logical rules to create questions that are natural but still require multi-hop reasoning. Through experiments, we demonstrate that our dataset is challenging for multi-hop models and it ensures that multi-hop reasoning is required.
- Chapter 4: We analyze the effectiveness of underlying reasoning (UR) tasks (including both sentence-level and entity-level tasks) in three aspects: (1) QA performance, (2) reasoning shortcuts, and (3) robustness. While the previous

models have not been explicitly trained on an entity-level reasoning prediction task, we build a multi-task model that performs three tasks together: sentence-level supporting facts prediction, entity-level reasoning prediction, and answer prediction. Experimental results on 2WikiMultiHopQA and HotpotQA-small datasets reveal that (1) UR tasks can improve QA performance. Using four debiased datasets that are newly created, we demonstrate that (2) UR tasks are helpful in preventing reasoning shortcuts in the multi-hop QA task. However, we find that (3) UR tasks do not contribute to improving the robustness of the model on adversarial questions, such as sub-questions and inverted questions.

- Chapter 5: We first propose a dataset, *HieraDate*, with three probing tasks in addition to the main question: extraction, reasoning, and robustness. Our dataset is created by enhancing two previous multi-hop datasets, HotpotQA and 2WikiMultiHopQA, focusing on multi-hop questions on date information that involve both comparison and numerical reasoning. We then evaluate the ability of existing models to understand date information. Our experimental results reveal that the multi-hop models do not have the ability to subtract two dates even when they perform well in date comparison and number subtraction tasks. Other results reveal that our probing questions can help to improve the performance of the models (e.g., by +10.3 F1) on the main QA task and our dataset can be used for data augmentation to improve the robustness of the models.

- Chapter 6: We examine the models' behaviors across three stages in the question-answering process: question decomposition, subproblem solving, and composition. To comprehensively investigate LLMs, we use three different scenarios for the input context: without context, with unstructured context, and with structured context. Our experiments on two multi-hop datasets and three different sizes of LLMs show that: 1) LLMs fail to decompose complex questions into sub-questions. Further analysis reveals no correlation between the question decomposition stage and QA performance. Additionally, the results indicate that models resort to reasoning shortcuts when addressing complex questions, rather than employing a step-by-step reasoning process. 2) In the subproblem solving stage, LLMs can successfully answer sub-questions when provided with

either unstructured or structured context but fail when no context is provided. This indicates that LLMs, including GPT-3.5, still lack the ability to memorize factual knowledge during pre-training. 3) Both Llama 2 13B and 70B struggle in performing at the composition stage on comparison questions.

2

An Overview of the Multi-hop MRC Task

In this chapter, we first introduce the machine reading comprehension (MRC) task. We then present the multi-hop MRC task, which is a potential direction for developing MRC in the future. After that, we discuss about the reasoning steps in the multi-hop MRC task. Finally, we show the reasoning shortcut issue in the multi-hop MRC task.

2.1 Machine Reading Comprehension Task

Natural Language Processing (NLP) aims to build a machine that can understand natural languages like a human. Reading comprehension is one of the most important skills for the task. For humans, according to Wikipedia[1] "Reading comprehension is the ability to process text, understand its meaning, and to integrate with what the reader already knows". For machines, MRC aims at teaching computers to read and understand unstructured text automatically. The task can be formulated as follows:

[1] https://en.wikipedia.org/wiki/Reading_comprehension

2.1 Machine Reading Comprehension Task

- Input: a question Q and a document or a paragraph P.
- Output: an answer A for the question Q.

Figure 2.1 presents an example of the MRC task from the SQuAD dataset (Rajpurkar et al., 2016). The input contains a paragraph and a question that asks for some information related to the paragraph. The output is an answer to the question.

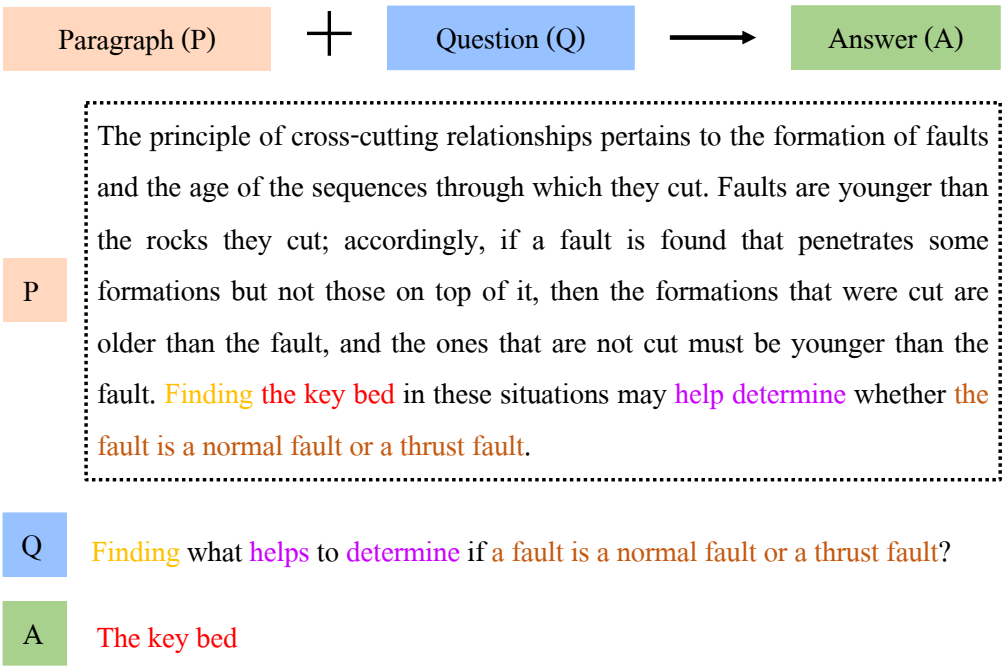

Figure 2.1: An example of the MRC task from the SQuAD dataset.

In recent years, a lot of datasets have been created, such as CNN/Daily Mail (Hermann et al., 2015) and SQuAD (Rajpurkar et al., 2016, 2018). Currently, many current models (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019) have defeated humans on the performance of SQuAD, as shown on its leaderboard.[2] However, such performances do not indicate that these models can completely understand the text.

Specifically, through analysis, Chen et al. (2016) revealed that the current systems excel at matching questions to local contexts when answering the question and the CNN/Daily Mail dataset has weak reasoning and inference skills. Using an adversarial method, Jia and Liang (2017) demonstrated that the current models do not precisely

[2] https://rajpurkar.github.io/SQuAD-explorer/

understand natural language. Moreover, Sugawara et al. (2018) demonstrated that many datasets contain a considerable number of easy instances that can be answered based on the first few words of the questions.

Many attempts have been made to address the issues described above, including unanswerable questions (Rajpurkar et al., 2018), knowledge-based MRC (Lai et al., 2017), conversational MRC (Reddy et al., 2019), and multi-hop MRC (Welbl et al., 2018). In this thesis, we focus on multi-hop MRC, which requires a model to read and perform multi-hop reasoning over multiple paragraphs to answer a given question.

2.2 Multi-hop Machine Reading Comprehension Task

Multi-hop MRC datasets require a model to read and perform multi-hop reasoning over multiple paragraphs to answer a question. Currently, there are four multi-hop datasets over textual data: ComplexWebQuestions (Talmor and Berant, 2018), QAngaroo (Welbl et al., 2018), HotpotQA (Yang et al., 2018), and R^4C (Inoue et al., 2020).

Question: Who is the paternal grandfather of Joan of Valois, Countess of Beaumont?

Paragraph A: Joan of Valois, Countess of Beaumont

[1] Joan of Valois (1304 – 1363) was the daughter of Charles of Valois and his second wife Catherine I of Courtenay, titular empress of Constantinople. ...

Paragraph B: Charles, Count of Valois

[2] Charles of Valois (12 March 1270 – 16 December 1325), the third son of Philip III of France and Isabella of Aragon, was a member of the House of Capet and founder of the House of Valois, whose rule over France would start in 1328. [3] Charles ruled several principalities. [4] He held in appanage the counties of Valois, Alencon and Perche.

Answer: Philip III of France

Sentence-level supporting facts: 1, 2

Evidence: ("Joan of Valois, Countess of Beaumont", "father", "Charles of Valois") and ("Charles of Valois", "father", "Philip III of France")

Figure 2.2: Example of the multi-hop MRC task.

We observe that the multi-hop MRC task is an important potential direction for the community because of the following attributes:

(i) Multi-hop MRC dataset is helpful for *testing the reasoning skills* of a model. To answer a multi-hop question, models must perform multiple reasoning steps (see

Figure 2.2). Each step often corresponds to several reasoning skills, such as comparisons and bridging entities.

(ii) Multi-hop MRC can be used to *evaluate the explainability* of a model. The internal reasoning process from a question to an answer involves multiple steps. Instead of evaluating models based solely on answer prediction task, previous studies (Yang et al., 2018; Ho et al., 2020; Inoue et al., 2020) have utilized internal reasoning information to evaluate the explainability of models.

(iii) Multi-hop MRC is useful in *applications*. Chen et al. (2017) introduced a way to construct a QA system by combining information retrieval (IR) and the MRC model. The MRC model in their system was designed for answering simple questions. However, questions in real-world QA systems can be complex and require many steps to be answered; Multi-hop MRC is an important component for answering those questions. Another application of multi-hop MRC is domain-specific information extraction, such as the discovery of drug-drug interactions by gathering information from different medical documents (Welbl et al., 2018).

2.3 Reasoning Steps in Multi-hop MRC

Owing to the diversity of multi-hop questions and the fact that there are many ways to discover the internal reasoning processes from question to answer, currently, there are many forms to represent a step in the reasoning path from question to answer. We present *three scenarios* with three different definitions of the steps in the path from question to answer below.

Scenario 1 - A Step is a Sub-task: As discussed in previous works (Talmor and Berant, 2018; Min et al., 2019b), multi-hop questions can be decomposed into multiple simple sub-questions. For example, consider the question *Which team does the player named 2015 Diamond Head Classic' s MVP play for?* We can split this question into two sub-questions: (a) *Which player was named 2015 Diamond Head Classic's MVP?* and (b) *Which team does ANS play for?* (*ANS* is the answer to the first sub-question). In this manner, we can consider predicting the answer to a sub-question as a step in the primary answering process.

Scenario 2 - A Step is a Triple: Previous works (Ho et al., 2020; Inoue et al., 2020) introduced a reasoning chain that describes relationships from the entities in the question to answer to explain the answers. Each triple in the reasoning chain can be considered as a step in the reasoning path from question to answer (see Figure 2.2).

Scenario 3 - A Step is a Sequence of Tokens Containing a Single Operator: There are several works (Cao et al., 2022; Wolfson et al., 2020) on both MRC and QA over KB that have introduced an explicit reasoning process from question to answer. Wolfson et al. (2020) introduced a question decomposition meaning representation (QDMR) that contains a set of steps to find an answer. A step in QDMR is a sequence of tokens. Each step corresponds to a single query operator based on a set of predefined operators (e.g., group or sort). It is noteworthy that the QDMR is used as additional supervision data for training and not for evaluating the internal reasoning information.

2.4 Reasoning Shortcuts Issues in Multi-hop MRC

2.4.1 Definition

Reasoning Shortcut We define shortcuts as statistical correlations in the data that allow a machine learning model to achieve high performance on a task without acquiring all the intended knowledge. When these shortcuts happen in a task that was supposed to require a reasoning step, we denominate it reasoning shortcut. The most important side-effect of shortcut learning is under-performance on adversarial or out-of-distribution (OOD) data.

When working with research related to reasoning shortcuts, we consider the definitions of other related terms—such as adversarial example, challenge set, robustness, and generalization—in addition to the definition of reasoning shortcut. We present the definitions of these terms below.

Adversarial Example Following previous studies (Geirhos et al., 2020; Schlegel et al., 2020; Zhang et al., 2020), we define adversarial examples as those that are designed to mislead machine learning models but not humans. Usually, the perceived difficulty for a human remains unchanged, while models fail due to their shortcut behavior.

Challenge Set An evaluation dataset that highlights a particularly difficult aspect of a task, such as overcoming a prominent shortcut. These datasets are important to allow comparison between methods, and assess the progress made in shortcut mitigations.

Robustness and Generalization We define a model as robust if its performance remains relatively unaltered under adversarial attacks. Similarly, a model has the ability to generalize if it can perform well on OOD test data.

2.4.2 Measuring Shortcuts in the MRC Task

Measuring the presence of shortcuts is an important first step necessary to understand the behavior of models and the biases present in the training data. In this thesis, we consider three methods that are popular in the MRC task in general, and one method that is primarily applied to the multi-hop MRC task. It is noted that the three methods used in the MRC task are also applicable to the multi-hop MRC task.

Adversarial Data Evaluation Adversarial samples are a clear and convenient way to highlight shortcut behavior because they are easy to construct and pose no additional difficulty for humans to solve. Adversarial methods either add or edit an original example to create a more challenging example. Based on the change of the gold label, we divide this method into two main groups: (1) label-preserved (the answer is unchanged) and (2) label-changed (the answer is changed).

(1) Label-Preserved We divide this group into the following four types.

Context-modification: Jia and Liang (2017) is the first work that proposed adversarial examples for evaluating reading comprehension systems. They insert distractor sentences into the original context of SQuAD (`AddSent`); their experimental results demonstrate that the current models fail to answer the modified examples. After that, Wang and Bansal (2018) extend the `AddSent` method by adding the distractor sentence into various locations in the context (`AddSentDiverse`). In addition to showing that the models are fragile on `AddSentDiverse`, they also showcase how their method improves robustness. In the same direction, Tran et al. (2023) introduce a negation attack for SQuAD 2.0 to make models produce unanswerable responses

to answerable questions. They found that the performance of models trained on SQuAD 2.0 drops significantly on the negation attack. Wallace et al. (2019) propose a universal adversarial attack by using gradient-guided search for many tasks in NLP, including MRC. Their triggers can attack 72% of 'why' questions in SQuAD, making them produce the same answers. They also reveal that the models are based heavily on the words around the answer in the paragraph and question types when producing an answer.

For the multi-hop MRC task, instead of adding distractor sentences, Jiang and Bansal (2019a) add a distractor paragraph. They demonstrate that many examples in the HotpotQA dataset contain reasoning shortcuts, where the models can answer the question by using word matching.

Question-modification: Ribeiro et al. (2018) introduce a set of rules that modify some characters in the question but still keep the semantics. These are called semantically equivalent adversarial rules (SEARs). Their experimental results show that models are weak to these changes; the model's predictions are changed after applying these rules. After that, Rychalska et al. (2018) modify the questions by changing some important words using the LIME (Locally Interpretable Model Agnostic Explanations) framework (Ribeiro et al., 2016). They show that performance decreases when some words in the questions are replaced with their synonyms. Later in this direction, Gan and Ng (2019) introduce two approaches to paraphrase the questions: (1) make them similar to the original questions to test model over-sensitivity, and (2) use context words near an incorrect answer candidate. They show that their models drop in performance on both types of paraphrased questions.

Option-modification: Lin et al. (2021) modify options in the multiple-choice dataset RACE while keeping the passage and the question. Specifically, they replace one wrong option among the four candidates with an irrelevant option which is chosen from a set of options via experiments. Their results reveal that the models exploited statistical biases in the datasets when answering the questions.

Mix-modification: Si et al. (2021) introduce four types of adversarial attacks and create a new benchmark for evaluating robustness in MRC. Their dataset, AdvRACE, is created by modifying RACE. Their experimental results show that the models are vulnerable under all of these attacks; meanwhile, their dataset, AdvRACE can be served as test data for evaluating robustness. Al-Negheimish et al. (2021) evaluate

top-performing models in the DROP leaderboard on a variety of modified versions of the DROP dataset. They find that the models do not reason about the question and the content of the passage, instead exploiting spurious patterns in the dataset to obtain the answers.

(2) Label-Changed For the case when the dataset modifications include a change in the answer, Ribeiro et al. (2019) automatically generate new question-answering (QA) pairs that represent the same information as the original QA but in a different way. Via evaluation, they find that the models lack real comprehension skills for their correct predicted answers in the original QA samples. Gardner et al. (2020) manually create contrast sets for 10 NLP datasets (including MRC) where the context is slightly modified and the answer is changed. Their experimental results show that model performance dramatically drops on the contrast sets. Schlegel et al. (2021) create a SAM (Semantics Altering Modifications) dataset by modifying the context that makes a change in the answer. They reveal that most of the current models are struggling with their proposed dataset. More recently, Geva et al. (2022) automatically generate adversarial samples by changing the reasoning path through question decomposition. Their results reveal that the performance of the models drops on the adversarial dataset Additionally, several studies (Nakanishi et al., 2018; Rajpurkar et al., 2018; Trivedi et al., 2020) introduce unanswerable questions by modifying the context or adding new questions.

Artifact-Based Models Artifact-based models are trained on insufficient or incomplete data, such as question-only, passage-only, or single-paragraph-only (in multi-hop tasks). If these models perform well, it can be inferred that the missing information was not necessary, and shortcuts were used within the provided data.

Kaushik and Lipton (2018) perform various experiments on 5 MRC datasets by using two artifact-based models: question-only and passage-only. Their results reveal that the models can achieve higher scores when they are trained in this way. For example, in task 18 of the bAbI dataset, the question-only approach obtains 91%, while the best performance of a standard model is 93%. These results indicate that the models are not solving the task in the manner expected, and instead abuse shortcuts. After that, Si et al. (2019) use the same methods as Kaushik and Lipton (2018); they also add

another experiment by shuffling the words in the context. They suggest that there exist artifacts and statistical cues in five multiple-choice datasets.

In multi-hop MRC, where at least two paragraphs are required to answer the question, Min et al. (2019a) and Chen and Durrett (2019) design a sentence-factored model and a single-paragraph BERT-based model respectively. The introduced models are not trained in the full context; therefore, they should not have the ability to answer the questions. However, their results show that these models can answer a large portion of examples; this indicates that these models do not perform multi-hop reasoning in the QA process. With the same idea, Trivedi et al. (2020) introduce the DiRe (Disconnected Reasoning) condition by removing the connection of the two supporting facts to measure reasoning shortcuts. They conclude that there had not been much progress in multi-hop reasoning.

Different from the above works, Sugawara et al. (2018) use the first few words in the question instead of using the full question. It was revealed that the BiDAF model (Seo et al., 2017) can infer the answer by using entity type matching. Sen and Saffari (2020) expand the idea in Sugawara et al. (2018) by using BERT and find that again, the model can answer the questions without using most or all of the words in the question.

There is one special case in this group where a shortcut is detected by using a subset of the dataset. Specifically, Ko et al. (2020) demonstrate the presence of the position bias in the SQuAD dataset by training the models on a subset of SQuAD in which the answer is in the first sentence of the context. Model performance drops significantly when evaluated on the SQuAD development set.

Language Understanding Skills Evaluation For humans to answer the question in the MRC task correctly, it requires several skills such as entity linking or coreference resolution. In this part, we explore approaches that evaluate models on these basic skills. Models that answer the task correctly but fail at the required skills for the task can be said to be abusing shortcuts.

Ribeiro et al. (2020) introduce CheckList, a list of basic linguistic skills to test the models comprehensively. Includes several skills, such as temporal reasoning, negation, coreference resolution and semantic role labeling. Their results show that models do not have the abilities to handle these skills. As an example, given the context "Aaron is an editor. Mark is an actor." and the question "Who is not an actor?", the

model incorrectly predicts "Mark". At the same time, Dunietz et al. (2020) introduce a template of understanding, which is "a set of question templates", to systematically test the comprehension abilities of the models regarding the content. Through a pilot experiment, they show that the XLNet (Yang et al., 2019) model performs worse on their designed questions.

Wu et al. (2021) introduce seven MRC skills that are related to discourse relations, such as negative causality reasoning and explicit conditional reasoning. Their results show that the three datasets (SQuAD, SQuAD 2.0, and SWAG (Zellers et al., 2018)) are insufficient for evaluating the understanding of discourse relations. Prior to this, Sugawara et al. (2020) analyze 10 datasets with 12 requisite skills. They also conclude that most existing MRC datasets might be insufficient for evaluating the discourse relations understanding.

We argue that evaluating the models on more basic NLP skills is an effective way to ensure that the models follow what humans do in the QA process. Future studies for carefully designing a set of language skills corresponding to each evaluation test data would be a need.

2.4.3 Measuring Shortcuts in the Multi-hop MRC Task

The underlying reasoning process from question to answer is important information to verify whether the models know how to answer the question in a step-by-step manner. One special requirement for this method is that it is only applicable to complex questions, such as multi-hop questions. The reasoning steps from question to answer of complex questions can be used to design new sub-tasks or to evaluate the reasoning abilities of the model. We name this method 'intermediate reasoning task evaluation.'

Tang et al. (2021) simply evaluate the underlying reasoning process via a set of sub-questions for bridge questions. It reveals that the existing multi-hop models do not have the ability to answer the sub-questions well, and many of them are answered incorrectly while their corresponding multi-hop questions are correctly predicted. After that, Ho et al. (2022) evaluate the underlying reasoning process for comparison questions by introducing the HieraDate dataset with three probing sub-questions: extraction, reasoning, and robustness. They find that even when the model is fine-tuned on the reasoning sub-questions, it does not have the ability to subtract two dates,

although it can subtract two numbers.

Inoue et al. (2020) and Ho et al. (2020) propose a new task for predicting or generating the reasoning chain; it is called derivation prediction in R^4C and evidence generation in 2WikiMultiHopQA (2Wiki; Ho et al., 2020). Wolfson et al. (2020) propose the 'Break It Down' dataset that contains an ordered list of steps in the process from question to answer. However, these steps are only used for training, not for evaluation. After that, Geva et al. (2021) introduce the StrategyQA dataset with sub-questions to explain the answers. Recently, Trivedi et al. (2022) propose the MuSiQue dataset, which is constructed via single-hop question composition. Most of these existing multi-hop datasets contain only a small number of reasoning steps, which are easy for the models. To solve this issue, Ribeiro et al. (2023) introduce the STREET dataset with more reasoning steps in the QA process. STREET requires a model not only to predict an answer for the question but also to generate a step-by-step structured explanation to explain the answer. Their results reveal that few-shot prompting GPT-3 and fine-tuned T5 do not possess sufficient skills to generate the structured reasoning steps.

3

2WikiMultiHopQA Dataset

A multi-hop question answering (QA) dataset aims to test reasoning and inference skills by requiring a model to read multiple paragraphs to answer a given question. However, current datasets do not provide a complete explanation for the reasoning process from the question to the answer. Further, previous studies revealed that many examples in existing multi-hop datasets do not require multi-hop reasoning to answer a question. In this chapter, we present a new multi-hop QA dataset, called 2WikiMultiHopQA, which uses structured and unstructured data. In our dataset, we introduce the evidence information containing a reasoning path for multi-hop questions. The evidence information has two benefits: (i) providing a comprehensive explanation for predictions and (ii) evaluating the reasoning skills of a model. We carefully design a pipeline and a set of templates when generating a question–answer pair that guarantees the multi-hop steps and the quality of the questions. We also exploit the structured format in Wikidata and use logical rules to create questions that are natural but still require multi-hop reasoning. Through experiments, we demonstrate that our dataset is challenging for multi-hop models and it ensures that

multi-hop reasoning is required.

3.1 Introduction

Machine reading comprehension (MRC) aims at teaching machines to read and understand given text. Many current models (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019) have defeated humans on the performance of SQuAD (Rajpurkar et al., 2016, 2018), as shown on its leaderboard.[1] However, such performances do not indicate that these models can completely understand the text. Specifically, using an adversarial method, Jia and Liang (2017) demonstrated that the current models do not precisely understand natural language. Moreover, Sugawara et al. (2018) demonstrated that many datasets contain a considerable number of easy instances that can be answered based on the first few words of the questions.

Multi-hop MRC datasets require a model to read and perform multi-hop reasoning over multiple paragraphs to answer a question. Currently, there are four multi-hop datasets over textual data: ComplexWebQuestions (Talmor and Berant, 2018), QAngaroo (Welbl et al., 2018), HotpotQA (Yang et al., 2018), and R^4C (Inoue et al., 2020). The first two datasets were created by incorporating the documents (from Web or Wikipedia) with a knowledge base (KB). Owing to their building procedures, these datasets have no information to explain the predicted answers. Meanwhile, the other two datasets were created mainly based on crowdsourcing. In HotpotQA, the authors introduced the sentence-level supporting facts (SFs) information that are used to explain the predicted answers. However, as discussed in Inoue et al. (2020), the task of classifying sentence-level SFs is a binary classification task that is incapable of evaluating the reasoning and inference skills of the model. Further, data analyses (Chen and Durrett, 2019; Min et al., 2019a) revealed that many examples in HotpotQA do not require multi-hop reasoning to solve.

Recently, to evaluate the internal reasoning of the reading comprehension system, Inoue et al. (2020) proposed a new dataset R^4C that requires systems to provide an answer and derivations. A derivation is a semi-structured natural language form that is used to explain the answers. R^4C is created based on HotpotQA and has 4,588

[1] https://rajpurkar.github.io/SQuAD-explorer/

questions. However, the small size of the dataset implies that the dataset cannot be used as a multi-hop dataset with a comprehensive explanation for training end-to-end systems.

In this chapter, we create a large and high quality multi-hop dataset 2WikiMultiHopQA[2] with a comprehensive explanation by combining structured and unstructured data. To enhance the explanation and evaluation process when answering a multi-hop question on Wikipedia articles, we introduce new information in each sample, namely *evidence* that contains comprehensive and concise information to explain the predictions. Evidence information in our dataset is a set of triples, where each triple is a structured data *(subject entity, property, object entity)* obtained from the Wikidata (see Figure 3.1 for an example).

Question: Who is the paternal grandfather of Joan of Valois, Countess of Beaumont?

Paragraph A: Joan of Valois, Countess of Beaumont

[1] Joan of Valois (1304 – 1363) was the daughter of Charles of Valois and his second wife Catherine I of Courtenay, titular empress of Constantinople. ...

Paragraph B: Charles, Count of Valois

[2] Charles of Valois (12 March 1270 – 16 December 1325), the third son of Philip III of France and Isabella of Aragon, was a member of the House of Capet and founder of the House of Valois, whose rule over France would start in 1328. [3] Charles ruled several principalities. [4] He held in appanage the counties of Valois, Alencon and Perche.

Answer: Philip III of France

Sentence-level supporting facts: 1, 2

Evidence: ("Joan of Valois, Countess of Beaumont", "father", "Charles of Valois") and ("Charles of Valois", "father", "Philip III of France")

Figure 3.1: Example of an inference question in our dataset. The difference between our dataset and HotpotQA is the evidence information that explains the reasoning path.

Our dataset has four types of questions: comparison, inference, compositional, and bridge comparison. All questions in our dataset are created by using a set of predefined templates. Min et al. (2019a) classified the comparison questions in HotpotQA in three types: multi-hop, context-dependent multi-hop, and single-hop. Based on this classification, we removed all templates in our list that make questions become single-hop or context-dependent multi-hop to ensure that our comparison questions and

[2] 2Wiki is a combination of Wikipedia and Wikidata.

bridge-comparison questions are multi-hop. We carefully designed a pipeline to utilize the intersection information between the summary[3] of Wikipedia articles and Wikidata and have a special treatment for each type of question that guarantees multi-hop steps and the quality of the questions. Further, by utilizing the logical rule information in the knowledge graph, such as $father(a, b) \wedge father(b, c) \Rightarrow grandfather(a, c)$, we can create more natural questions that still require multi-hop reasoning.

We conducted two different evaluations on our dataset: difficulty and multi-hop reasoning of the dataset. To evaluate the difficulty, we used a multi-hop model to compare the performance of HotpotQA and our dataset. Overall, the results from our dataset are lower than those observed in HotpotQA, while human scores are comparable on both datasets. This suggests that the number of difficult questions in our dataset is greater than that in HotpotQA. Similar to Min et al. (2019a), we used a single-hop BERT model (Devlin et al., 2019) to test the multi-hop reasoning in our dataset. The result of our dataset is lower than the result of HotpotQA by 8.7 F1, indicating that a lot of examples in our dataset require multi-hop reasoning to be solved. Through experiments, we confirmed that although our dataset is generated by hand-crafted templates and the set of predefined logical rules, it is challenging for multi-hop models and requires multi-hop reasoning.

In summary, our main contributions are as follows: (1) We use Wikipedia and Wikidata to create a large and high quality multi-hop dataset that has comprehensive explanations from question to answer. (2) We provide new information in each sample—evidence information useful for interpreting the predictions and testing the reasoning and inference skills of the model. (3) We use logical rules to generate a simple natural question but still require the model to undertake multi-hop reasoning when answering a question. The full dataset, baseline model, and all information that we used when constructing the dataset are available at https://github.com/Alab-NII/2wikimultihop.

3.2 Related Work

Multi-hop Questions in MRC Domain Currently, four multi-hop MRC datasets proposed for textual data: QAngaroo (Welbl et al., 2018), ComplexWebQuestions (Tal-

[3]Another name is "short description"; The short description at the top of an article that summarizes the content. See also https://en.wikipedia.org/wiki/Wikipedia:Short_description

mor and Berant, 2018), HotpotQA (Yang et al., 2018), and R^4C (Inoue et al., 2020). QAngaroo (Welbl et al., 2018) was the first dataset to introduce multi-hop reasoning in MRC. This dataset contains two sub-datasets called WikiHop and MedHop in the open domain and medicine domain, respectively. The dataset was automatically constructed using KB and Wikipedia. Subsequently, Talmor and Berant (2018) introduced ComplexWebQuestions, a dataset created by making the WebQuestionSP dataset (Yih et al., 2016) more complicated. Owing to their building procedures, both datasets do not provide any information to explain the predicted answers. Later, Yang et al. (2018) introduced HotpotQA, a crowdsourced dataset. In HotpotQA, the authors introduced new information called *sentence-level supporting facts*, which are sets of sentences that support answers. They also introduced a new task called sentence-level supporting fact prediction, which is a binary classification task. This type of explanation is called a justification explanation (collection of evidence to support a decision). Subsequently, Inoue et al. (2020) introduced a new dataset called R^4C that provides both justification and introspective explanations (how a decision is made).

Recently, Chen et al. (2020) introduced the HybridQA dataset—a multi-hop question answering over both tabular and textual data. The dataset was created by crowdsourcing based on Wikipedia tables and Wikipedia articles.

Multi-hop Questions in KB Domain Question answering over the knowledge graph has been investigated for decades. However, most current datasets (Berant et al., 2013; Bordes et al., 2015; Yih et al., 2015; Diefenbach et al., 2017) consist of simple questions (single-hop). Zhang et al. (2018b) introduced the METAQA dataset that contains both single-hop and multi-hop questions. Abujabal et al. (2017) introduced the ComplexQuestions dataset comprising 150 compositional questions. All of these datasets are solved by using the KB only. Our dataset is constructed based on the intersection between Wikipedia and Wikidata. Therefore, it can be solved by using structured or unstructured data.

Compositional Knowledge Base Inference Extracting Horn rules from the KB has been studied extensively in the Inductive Logic Programming literature (Quinlan, 1990; Muggleton, 1995). From the KB, there are several approaches that mine association rules (Agrawal et al., 1993) and several mine logical rules (Schoenmackers et al., 2010;

Galárraga et al., 2013). We observed that these rules can be used to test the reasoning skill of the model. Therefore, in this chapter, we utilized the logical rules in the form: $r_1(a, b) \wedge r_2(b, c) \Rightarrow r(a, c)$. ComplexWebQuestions and QAngaroo datasets are also utilized KB when constructing the dataset, but they do not utilize the logical rules as we did.

RC Datasets with Explanations Table 3.1 presents several existing datasets that provide explanations. HotpotQA and R^4C are the most similar works to ours. HotpotQA provides a justification explanation (collections of evidence to support the decision) in the form of a set of sentence-level SFs. R^4C provides both justification and introspective explanations (how a decision is made). Our study also provides both justification and introspective explanations. The difference is that the explanation in our dataset is a set of triples, where each triple is a structured data obtained from Wikidata. Meanwhile, the explanation in R^4C is a set of semi-structured data. R^4C is created based on HotpotQA and has 4,588 questions. The small size of the dataset implies that it cannot be used for training end-to-end neural network models involving the multi-hop reasoning with comprehensive explanation.

Task/Dataset	**Explanations**		**Size**
	Justification	Introspective	
Our work	✓	✓	192,606
R^4C (Inoue et al., 2020)	✓	✓	4,588
CoS-E (Rajani et al., 2019)		✓	19,522
HotpotQA (Yang et al., 2018)	✓		112,779
Science Exam QA (Jansen et al., 2016)		✓	363

Table 3.1: Comparison with other datasets with explanations.

3.3 Task Overview

3.3.1 Task Formalization and Metrics

We formulated (1) answer prediction, (2) sentence-level SFs prediction, and (3) evidence generation tasks as follows:

- Input: a question Q and a set of documents D.
- Output: (1) find an answer A (a textual span in D) for Q, (2) find a set of sentence-level SFs (sentences) in D that a model used to answer Q, and (3) generate a set of evidence E which consists of triples that describes the reasoning path from Q to A.

We evaluate the three tasks by using two evaluation metrics: exact match (EM) and F1 score. Following previous work (Yang et al., 2018), to assess the entire capacity of the model, we introduced joint metrics that combine the evaluation of answer spans, sentence-level SFs, and evidence as follows:

$$Joint\ F1 = \frac{2P^{joint}R^{joint}}{P^{joint} + R^{joint}} \tag{3.1}$$

where $P^{joint} = P^{ans}P^{sup}P^{evi}$ and $R^{joint} = R^{ans}R^{sup}R^{evi}$. (P^{ans}, R^{ans}), (P^{sup}, R^{sup}), and (P^{evi}, R^{evi}) denote the precision and recall of the answer spans, sentence-level SFs, and evidence, respectively. Joint EM is 1 only when all the three tasks obtain an exact match or otherwise 0.

3.3.2 Question Types

In our dataset, we have the following four types of questions: (1) comparison, (2) inference, (3) compositional, and (4) bridge-comparison. The inference and compositional questions are the two subtypes of the bridge question which comprises a bridge entity that connects the two paragraphs (Yang et al., 2018).

1. **Comparison Question** is a type of question that compares two or more entities from the same group in some aspects of the entity (Yang et al., 2018). For instance, a comparison question compares two or more people with the *date of birth* or *date of death* (e.g., *Who was born first, Albert Einstein or Abraham Lincoln?*).

2. **Inference Question** is created from the two triples (e, r_1, e_1) and (e_1, r_2, e_2) in the KB. We utilized the logical rule to acquire the new triple (e, r, e_2), where r is the inference relation obtained from the two relations r_1 and r_2. A question–answer pair is created by using the new triple (e, r, e_2), its question is created from (e, r) and its answer is e_2. For instance, using two triples *(Abraham Lincoln, mother, Nancy Hanks Lincoln)* and *(Nancy Hanks Lincoln, father, James Hanks)*, we obtain a new triple *(Abraham Lincoln, maternal grandfather, James Hanks)*. A question is: *Who is the maternal grandfather of Abraham Lincoln?* An answer is *James Hanks* (Section 3.4.2).

3. **Compositional Question** is created from the two triples (e, r_1, e_1) and (e_1, r_2, e_2) in the KB. Compared with inference question, the difference is that no inference relation r exists from the two relations r_1 and r_2. For instance, there are two triples *(La La Land, distributor, Summit Entertainment)* and *(Summit Entertainment, founded by, Bernd Eichinger)*. There is no inference relation r from the two relations *distributor* and *founded-by*. In this case, a question is created from the entity e and the two relations r_1 and r_2: *Who is the founder of the company that distributed La La Land film?* An answer is the entity e_2 of the second triple: *Bernd Eichinger* (Section 3.4.2).

4. **Bridge-comparison Question** is a type of question that combines the bridge question with the comparison question. It requires both finding the bridge entities and doing comparisons to obtain the answer. For instance, instead of directly compare two films, we compare the information of the directors of the two films, e.g., *Which movie has the director born first, La La Land or Tenet?* To answer this type of question, the model needs to find the bridge entity that connects the two paragraphs, one about the film and one about the director, to get the date of birth information. Then, making a comparison to obtain the final answer.

3.4 Data Collection

The dataset is created by using both Wikipedia and Wikidata. We first briefly introduce Wikipedia and Wikidata. After that, we present the details of the dataset generation

procedure.

3.4.1 Wikipedia and Wikidata

We utilized both text descriptions from Wikipedia[4] and a set of statements from Wikidata to construct our dataset. We used only a summary from each Wikipedia article as a paragraph that describes an entity. Wikidata[5] is a collaborative KB that stores data in a structured format. Wikidata contains a set of statements (each statement includes a property and an object entity) to describe the entity. There is a connection between Wikipedia and Wikidata for each entity. From Wikidata, we can extract a triple (s, r, o), where s is a subject entity, r is a property or relation, and o is an object entity. A statement for the entity s is (r, o). An object entity can be another entity or the date value.

We used both dump[6] and online version of Wikipedia and Wikidata. We downloaded the dump of English Wikipedia on January 1, 2020, and the dump of English Wikidata on December 31, 2019. From Wikidata and Wikipedia, we obtained 5,950,475 entities. Based on the value of the property *instance of* in Wikidata, we categorized all entities into 23,763 groups. In this dataset, we focused on the most popular entities (top-50 for comparison questions). When checking the requirements to ensure the multi-hop reasoning of the dataset, several entities in the multi-path are not present in the dump version; in such situations, we used the online version of Wikipedia and Wikidata.

We observed that the quality of the dataset depends on the quality of the intersection information between Wikipedia and Wikidata. Specifically, for the property related to date information, such as *publication date* and *date of birth*, information between Wikipedia and Wikidata is quite consistent. Meanwhile, for the property *occupation*, information between Wikipedia and Wikidata is inconsistent. For instance, the Wikipedia of the entity *Ebenezer Adam* is as follows: "*Ebenezer Adam was a Ghanaian educationist and politician.*"; meanwhile, the value from Wikidata of the property *occupation* is *politician*. In such situations, we manually check all samples related to the property to ensure dataset quality. For the property related to the country name, we handled many different similar names by using the aliases of the entity and the set of

[4] https://www.wikipedia.org
[5] https://www.wikidata.org
[6] https://dumps.wikimedia.org/

demonyms. Moreover, to guarantee the quality of the dataset, we only focused on the set of properties with high consistency between Wikipedia and Wikidata.

We used both Stanford CoreNLP (Manning et al., 2014) and Spacy to perform sentence segmentation for the context.

3.4.2 Dataset Generation Process

Generating a multi-hop dataset in our framework involves three main steps: (1) create a set of templates, (2) generate data, and (3) post-process generated data. After obtaining the generated data, we used a model to split the data into *train*, *dev*, and *test* sets.

(1) Create a Set of Templates

Comparison Questions For the comparison question, first, we used Spacy[7] to extract named entity recognition (NER) tags and labels for all comparison questions in the training data of HotpotQA (17,456 questions). Then, we obtained a set of templates L by replacing the words in the questions with the labels obtained from the NER tagger. We manually created a set of templates based on L for entities in the top-50 most popular entities in Wikipedia. We also discarded all templates that made questions become single-hop or context-dependent multi-hop as discussed in Min et al. (2019a). Table 3.2 presents all information of our comparison question. We can use more entities and properties from Wikidata to create a dataset. In this version of the dataset, we focused on the top-50 popular entities in Wikipedia and Wikidata. To ensure dataset quality, we used the set of properties as described in the table. For each combination between the entity and the property, we have various templates for asking questions to ensure diversity in the questions.

Bridge-comparison Questions Based on the templates of the comparison question, we manually enhanced it to create the templates for bridge-comparison questions. The top-3 popular entities on Wikipedia and Wikidata are *human*, *taxon*, and *film*. In this type of question, we focused on the combination between *human* and *film*. Table 3.3 presents the combination between the relations from the two entities *human* and *film* in our dataset.

[7] https://spacy.io/

Entity Type	Property	#Templates
Human	date of birth	7
	date of death	3
	date of birth and date of death (year old)	2
	occupation	18
	country of citizenship	11
	place of birth	1
Film	publication date	5
	director	2
	producer	2
	country of origin	7
Album	publication date	5
	producer	2
Musical group	inception	4
	country of origin	7
Song	publication date	5
Museum, Airport, Magazine, Railway station, Business, Building, Church building, High school, School, University	inception	1-3
	country	4
Mountain, River, Island, Lake, Village	country	4

Table 3.2: Templates of comparison questions.

Relation 1	Relation 2
director	date of birth
director	date of death
director	country of citizenship
producer	date of birth
producer	date of death
producer	country of citizenship

Table 3.3: Bridge-comparison question's information.

For each row in Table 3.3, we have several ways to ask a question. For instance, in the first row, with the combination of the two relations *director* and *date of birth*, we have various ways to ask a question, as shown in Table 3.4. To avoid ambiguous cases, we used films that have only one director or one producer. A total of 62 templates was obtained for this type of question.

Templates
Which film has the director born first, #name or #name?
Which film whose director was born first, #name or #name?
Which film has the director who was born first, #name or #name?
Which film has the director born earlier, #name or #name?
Which film has the director who was born earlier, #name or #name?
Which film whose director is younger, #name or #name?
Which film has the director born later, #name or #name?
Which film has the director who was born later, #name or #name?
Which film has the director who is older than the other, #name or #name?
Which film has the director who is older, #name or #name?

Table 3.4: Templates of bridge-comparison questions.

Inference Questions For the inference question, we utilized logical rules in the knowledge graph to create a simple question but still require multi-hop reasoning. Extracting logical rules is a task in the knowledge graph wherein the target makes the graph complete. We observe that logical rules, such as $spouse(a, b) \wedge mother(b, c) \Rightarrow mother\_in\_law(a, c)$, can be used to test the reasoning skill of the model. Based on the results of the AMIE model (Galárraga et al., 2013), we manually checked and verified all logical rules to make it suitable for the Wikidata relations.

We argued that logical rules are difficult to apply to multi-hop questions. We obtained a set of 50 inference relations, but we cannot use all of them into the dataset. For instance, the logical rule is $place\_of\_birth(a, b) \wedge country(b, c) \Rightarrow nationality(a, c)$; this rule easily fails after checking the requirements. To guarantee the multi-hop reasoning of the question, the document describing a person a having a place of birth b should not contain the information about the country c. However, most paragraphs describing humans often contain information on their nationality.

The other issue is ensuring that each sample has only one correct answer on the two gold paragraphs. With the logical rule being $child(a, b) \wedge child(b, c) \Rightarrow grandchild(a, c)$, if a has more than one child, for instance a has three children b_1, b_2 and b_3, then each b has their own children. Therefore, for the question "*Who is the grandchild of a?*", there are several possible answers to this question. To address this issue in our dataset, we only utilized the relation that has only one value in the triple on Wikidata. That is the reason why the number of inference questions in our dataset is quite small. Table 3.5 describes all inference relations used in our dataset. We obtained 28 logical rules.

In most cases, this rule will be correct. However, several rules can be false in some cases. In such situations, based on the Wikidata information, we double-checked the new triple before deciding whether to use it. For instance, the rule is $doctoral\_advisor(a, b) \wedge employer(b, c) \Rightarrow educated\_at(a, c)$, a has an advisor is b, b works at c, and we can infer that a studies at c. There can be exceptions that b works at many places, and c is one of them, but a does not study at c. We used Wikidata to check whether a studies at c before deciding to use it.

After obtain the rules, we manually created all templates for inference questions. Table 3.6 presents a set of templates that we used to generate our inference questions.

Compositional Questions For this type of question, we utilized various entities and properties on Wikidata. We used the following properties (13 properties) as the first relation: *composer, creator, director, editor, father, founded by, has part, manufacturer, mother, performer, presenter, producer, and spouse.* Further, we used the following properties (22 properties) as the second relation: *date of birth, date of death, place of birth, country of citizenship, place of death, cause of death, spouse, occupation, educated at, award received, father, place of burial, child, employer, religion, field of work, mother, inception, country, founded by, student of, and place of detention.* A compositional question was created by combining the first relation and the second relation (ignore duplicate cases except for special cases).

We used the following entities (15 entities) to create this type of question: *human, film, animated feature film, album, university, film production company, business, television program, candy, written work, literary work, musical group, song, magazine, newspaper.* We manually created all templates and obtained a total of 799 templates for compositional questions.

Relation 1	Relation 2	Inference Relation
spouse	spouse	co-husband/co-wife
spouse	father	father-in-law
spouse	mother	mother-in-law
spouse	sibling	sibling-in-law
spouse	child	child/stepchild
father	father	paternal grandfather
father	mother	paternal grandmother
father	spouse	mother/stepmother
father	child	sibling
father	sibling	uncle/aunt
mother	mother	maternal grandmother
mother	father	maternal grandfather
mother	spouse	father/stepfather
mother	child	sibling
mother	sibling	uncle/aunt
child	child	grandchild
child	sibling	child
child	mother	wife
child	father	husband
child	spouse	child-in-law
sibling	sibling	sibling
sibling	spouse	sibling-in-law
sibling	mother	mother
sibling	father	father
doctoral student	educated at	employer
doctoral student	field of work	field of work
doctoral advisor	employer	educated at
doctoral advisor	field of work	field of work

Table 3.5: Inference relation information in our dataset.

Relation	Template(s)
aunt, child-in-law, child, co-husband, co-wife, father-in-law, father, grandchild, grandfather, grandmother, husband, mother-in-law, mother, sibling-in-law, sibling, stepchild, stepfather, stepmother, uncle, wife	Who is the #relation of #name? Who is #name's #relation?
educated at	Which #instance_of_answer did #name study at? Which #instance_of_answer did #name graduate from?
employer	Which #instance_of_answer does #name work at? Where does #name work?
field of study	What is the field of study of #name?

Table 3.6: Templates of inference questions.

(2) Generate Data

Comparison Questions From the set of templates and all entities' information, we generated comparison questions as described in Algorithm 1. For each entity group, we randomly selected two entities: e_1 and e_2. Subsequently, we obtained the set of statements of each entity from Wikidata. Then, we processed the two sets of statements to obtain a set of mutual relations (M) between two entities. We then acquired the Wikipedia information for each entity. For each relation in M, for example, a relation r_1, we checked whether we can use this relation. Because our dataset is a span extraction dataset, the answer is extracted from the Wikipedia article of each entity. With relation r_1, we obtained the two values o_1 and o_2 from the two triples (e_1, r_1, o_1) and (e_2, r_1, o_2) of the two entities, respectively. The requirement here is that the value o_1 must appear in the Wikipedia article for the entity e_1, which is the same condition for the second entity e_2.

When all information passed the requirements, we generated a question–answer pair that includes a question Q, a context C, the sentence-level SFs SF, the evidence E,

Algorithm 1: Comparison Question Generation Procedure

Input: Set of all templates, all entities in the same group, Wikipedia and Wikidata information for each entity

Output: A question–answer pair with these information: question Q, answer A, context C, sentence-level SFs SF, and evidence E

```
while not finished do
    Randomly choose two entities e1 and e2;
    Obtain all triples (relations and objects) of each entity from Wikidata;
    Obtain a set of mutual relations (M) between two entities;
    Obtain Wikipedia information of each entity;
    for each relation in M do
        if pass requirements then
            Choose a template randomly;
            Generate a question Q;
            Obtain a context C;
            Obtain an evidence E;
            Compute an answer A;
            Compute sentence-level SFs SF;
        end
    end
end
```

and an answer A. Q is obtained by replacing the two tokens *#name* in the template by the two entity labels. C is a concatenation of the two Wikipedia articles that describe the two entities. E is the two triples (e_1, r_1, o_1) and (e_2, r_1, o_2). SF is a set of sentence indices where the values o_1 and o_2 are extracted. Based on the type of questions, we undertake comparisons and obtain the final answer A.

Bridge Questions We generated bridge questions as described in Algorithm 2. For each entity group, we randomly selected an entity e and then obtained a set of statements of the entity from Wikidata. Subsequently, based on the first relation information in R (the set of predefined relations), we filtered the set of statements to obtain a set of 1-hop H_1. Next, for each element in H_1, we performed the same process to obtain a set of 2-hop H_2, each element in H_2 is a tuple (e, r_1, e_1, r_2, e_2). For each tuple in H_2, we obtained the Wikipedia articles for two entities e and e_1. Then, we checked the requirements to ensure that this sample can become a multi-hop

Algorithm 2: Bridge Question Generation Procedure

Input: Set of relations R, Wikipedia and Wikidata information for each entity
Output: A question–answer pair with these information: question Q, answer A, context C, sentence-level SFs SF, evidence E

```
1  while not finished do
2      Randomly choose an entity e;
3      Obtain a set of statements (relations and objects) of the entity from
         Wikidata;
4      Filter the set of statements based on the first relation information in R to
         obtain a set of 1-hop H_1;
5      For each element in H_1, do the same process (from Line 3) to obtain a set of
         2-hop H_2, each element in H_2 is a tuple (e, r_1, e_1, r_2, e_2);
6      for each tuple in H_2 do
7          Obtain Wikipedia articles for two entities: e and e_1;
8          if pass requirements then
9              Choose a template randomly based on r_1 and r_2;
10             Generate a question Q;
11             Obtain a context C;
12             Obtain an evidence E;
13             Obtain an answer A;
14             Compute sentence-level SFs SF;
15         end
16     end
17 end
```

dataset. For instance, the two paragraphs p and p_1 describe for e and e_1, respectively (see Figure 3.2). The bridge entity requirement is that p must mention e_1. The span extraction answer requirement is that p_1 must mention e_2. The 2-hop requirements are that p must not contain e_2 and p_1 must not contain e. Finally, we obtained Q, C, SF, E, and A similarly to the process in comparison questions.

(3) Post-process Generated Data

Discard Examples We randomly selected two entities to create a question when generating the data; therefore, a large number of *no* questions exist in the *yes/no* questions. We performed post-processing to finalize the dataset that balances the number of *yes* and *no* questions. Questions could have several true answers in the real

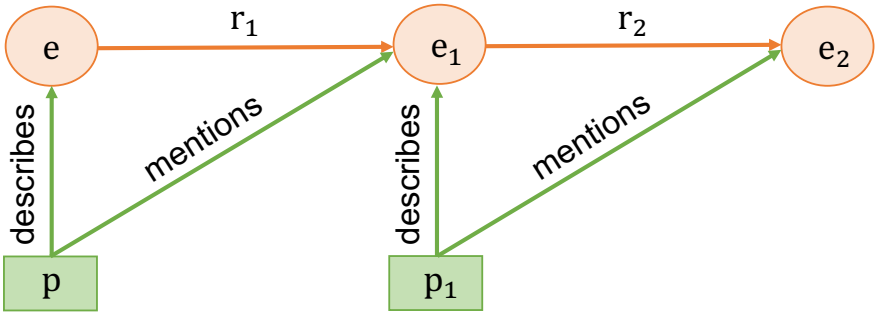

Figure 3.2: The requirements for bridge questions in our dataset.

world. To ensure one sample has only one answer, we discarded all ambiguous cases in the dataset. Specifically, for the bridge questions, we created the data from the two triples (e, r_1, e_1) and (e_1, r_2, e_2). When we have another triple (e, r_1, e_{1*}) that has the same entity and the property with the first triple, it becomes an ambiguous case. Hence, we discarded all such cases in our dataset based on the information from Wikidata. For the comparison questions, when a question is asked for comparing two entities about numerical values, and the values of the two entities are equal, we remove it.

Collect Distractor Paragraphs Following Yang et al. (2018) and Min et al. (2019a), we used bigram tf-idf (Chen et al., 2017) to retrieve the top-50 paragraphs from Wikipedia that are most similar to the question. Then, we used the entity type of the two gold paragraphs (four gold paragraphs for bridge-comparison questions) to select the top-8 paragraphs (top-6 for bridge-comparison questions) and considered it as a set of distractor paragraphs. We shuffled the 10 paragraphs (including gold and distractor paragraphs) and obtained a context.

Dataset Statistics (A Benchmark Setting) We used a single-hop model (Section 3.6) to split the *train*, *dev*, and *test* sets. We conducted five-fold cross-validation on all data. The average F1 score of the model is 86.7%. All questions solved by the single-hop model are considered as a *train-medium* subset. The rest was split into three subsets: *train-hard*, *dev*, and *test* (balancing the number of different types of questions in each subset). Statistics of the data split can be found in Table 3.7. We used *train-medium* and *train-hard* as the training data in our dataset.

Name	Split	#Examples
train-medium	train	154,878
train-hard	train	12,576
dev	dev	12,576
test	test	12,576
Total		192,606

Table 3.7: Data statistics in our dataset.

3.5 Data Analysis

3.5.1 Question and Answer Lengths

We quantitatively analyze the properties of questions and answers for each type of question in our dataset. The statistics of the dataset are presented in Table 3.8. The compositional question has the greatest number of examples, and the inference question has the least number of examples. To ensure one question has only one possible answer, we used the information from Wikidata and removed many inference questions that may have more than one answer. The average question length of the inference questions is the smallest because they are created from one triple. The average question length of the bridge-comparison questions is the largest because it combines both bridge question and comparison question. The average answer lengths of comparison and bridge-comparison questions are smaller than inference and compositional questions. This is because there are many *yes/no* questions in the comparison questions.

Type of Question	#Examples	#Avg. Question	#Avg. Answer
Comparison	57,989	11.97	1.58
Inference	7,478	8.41	3.15
Compositional	86,979	11.43	2.05
Bridge-comparison	40,160	17.01	2.01
Total	192,606	12.64	1.94

Table 3.8: Question and answer lengths across the different type of questions.

3.5.2 Answer Types

We preserved all information when generating the data; hence, we used the answer information (both string and Wikidata id) to classify the types of answers. Based on the value of the property *instance of* in Wikidata, we obtained 708 unique types of answers. The top-5 types of answers in our dataset are: *yes/no* (31.2%), date (16.9%; e.g., July 10, 2010), film (13.5%; e.g., *La La Land*), human (11.7%; e.g., *George Washington*), and big city (4.7%; e.g., *Chicago*). For the remaining types of answers (22.0%), they are various types of entities in Wikidata.

3.5.3 Multi-hop Reasoning Types

Table 3.9 presents different types of multi-hop reasonings in our dataset. Comparison questions require quantitative or logical comparisons between two entities to obtain the answer. The system is required to understand the properties in the question (e.g., *date of birth*). Compositional questions require the system to answer several primitive questions and combine them. For instance, to answer the question *Why did the founder of Versus die?*, the system must answer two sub-questions sequentially: (1) *Who is the founder of Versus?* and (2) *Why did he/she die?*. Inference questions require that the system understands several logical rules. For instance, to find the *grandchild*, first, it should find the *child*. Then, based on the *child*, continue to find the *child*. Bridge-comparison questions require both finding the bridge entity and doing a comparison to obtain the final answer.

3.6 Evaluate the Dataset Quality

We conducted two different evaluations on our dataset: evaluate the difficulty and the multi-hop reasoning.

3.6.1 Evaluate the Difficulty

To evaluate the difficulty, we used the multi-hop model as described in Yang et al. (2018) to obtain the results on HotpotQA (distractor setting) and our dataset. Table 3.10 presents the results. For the SFs prediction task, the scores on our dataset are higher

Reasoning Type	Example
Comparison question: comparing two entities	**Paragraph A:** Theodor Haecker (June 4, 1879 - April 9, 1945) was a ... **Paragraph B:** Harry Vaughan Watkins (10 September 1875 – 16 May 1945) was a Welsh rugby union player ... **Q:** Who lived longer, Theodor Haecker or Harry Vaughan Watkins?
Compositional question: inferring the bridge entity to find the answer	**Paragraph A:** Versus (Versace) is the diffusion line of Italian ..., a gift by the founder Gianni Versace to his sister, Donatella Versace. ... **Paragraph B:** Gianni Versace ... Versace was shot and killed outside ... **Q:** Why did the founder of Versus die?
Inference question: using logical rules and inferring the bridge entity	**Paragraph A:** Dambar Shah (? – 1645) was the king of the Gorkha Kingdom ... He was the father of Krishna Shah. ... **Paragraph B:** Krishna Shah (? – 1661) ... He was the father of Rudra Shah. **Q:** Who is the grandchild of Dambar Shah?
Bridge-comparison question: inferring the bridge entity and doing comparisons	**Paragraph A:** FAQ: Frequently Asked Questions is a feature-length dystopian movie, written and directed by Carlos Atanes and released in 2004. ... **Paragraph B:** The Big Money ... directed by John Paddy Carstairs ... **Paragraph C:** Carlos Atanes is a Spanish film director ... **Paragraph D:** John Paddy Carstairs was a prolific British film director ... **Q:** Are both director of film FAQ: Frequently Asked Questions and director of film The Big Money from the same country?

Table 3.9: Types of multi-hop reasoning in our dataset.

than those on HotpotQA. However, for the answer prediction task, the scores on our dataset are lower than those on HotpotQA. Overall, on the joint metrics, the scores on our dataset are lower than those on HotpotQA. This indicates that given the human performance on both datasets is comparable (see Section 3.7.3), the number of difficult questions in our dataset is greater than that in HotpotQA.

Dataset	Answer		Sp Fact		Joint	
	EM	F1	EM	F1	EM	F1
HotpotQA	44.48	58.54	**20.68**	**65.66**	10.97	40.52
Our Dataset	**34.14**	**40.95**	26.47	66.94	**9.22**	**26.76**

Table 3.10: Results (%) of the multi-hop model on HotpotQA (Yang et al., 2018) and our dataset. "**Sp Fact**" is the abbreviation for the sentence-level supporting facts prediction task.

3.6.2 Evaluate the Multi-hop Reasoning

Similar to Min et al. (2019a), we used a single-hop BERT model (Devlin et al., 2019) to test the multi-hop reasoning in our dataset. The F1 score on HotpotQA is 64.6 (67.0 F1 in Min et al. (2019a)); meanwhile, the F1 score on our dataset is 55.9. The result of our dataset is lower than the result of HotpotQA by 8.7 F1. It indicates that a large number of examples in our dataset require multi-hop reasoning to be solved. Moreover, it is verified that our data generation and our templates guarantee multi-hop reasoning. In summary, these results show that our dataset is challenging for multi-hop models and requires multi-hop reasoning to be solved.

3.7 Experiments

3.7.1 Baseline Models

We modified the baseline model in Yang et al. (2018) and added a new component (the orange block in Figure 3.3) to perform the evidence generation task. We re-used several techniques of the previous baseline, such as bi-attention, to predict the evidence.

Our evidence information is a set of triples, with each triple including *subject entity*, *relation*, and *object entity*. First, we used the question to predict the relations and then used the predicted relations and the context (after predicting sentence-level SFs) to obtain the subject and object entities.

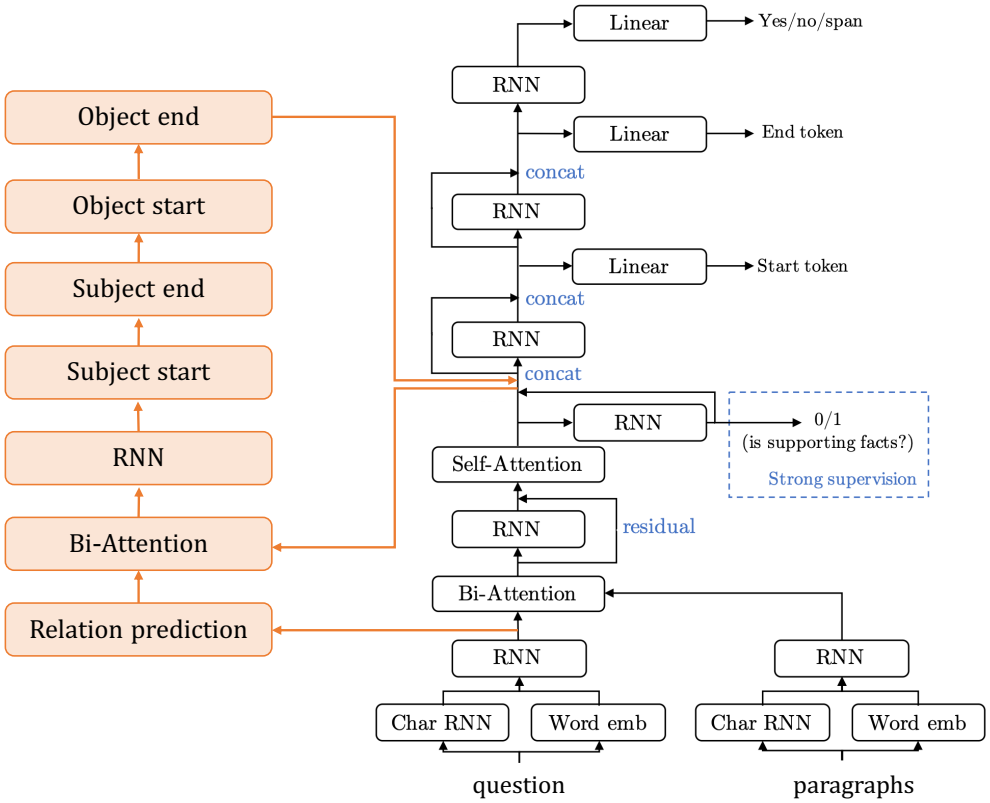

Figure 3.3: Our baseline model. The right part is the baseline model of HotpotQA (Yang et al., 2018).

3.7.2 Baseline Results

Table 3.11 presents the results of our baseline model. We used the evaluation metrics as described in Section 3.3.1. As shown in the table, the scores of the sentence-level SFs prediction task are quite high. This is a binary classification task that classifies whether each sentence is a SF. As discussed, this task is incapable of evaluating the reasoning and inference skills of the model. The scores of the evidence generation task

are quite low which indicates this task is difficult. Our error analysis shows that the model can predict one correct triple in the set of the triples. However, accurately obtaining the set of triples is extremely challenging. This is the reason why the EM score is very low. We believe that adding the evidence generation task is appropriate to test the reasoning and inference skills.

Split/Task	Answer		Sp Fact		Evidence		Joint	
	EM	F1	EM	F1	EM	F1	EM	F1
Dev	38.64	44.74	23.85	64.31	1.42	16.43	0.54	6.04
Test	40.08	46.29	24.99	65.26	1.49	16.63	0.53	6.51

Table 3.11: Results (%) of the baseline model.

To investigate the difficulty of each type of question, we categorized the performance for each type of question (on the test split). Table 3.12 shows the results. For the answer prediction task, the model obtained high scores on inference and compositional questions. Meanwhile, for the sentence-level SFs prediction task, the model obtained high scores on comparison and bridge-comparison questions. Overall, the joint metric score of the inference question is the lowest. This indicates that this type of question is more challenging for the model. The evidence generation task has the lowest score for all types of questions when compared with the other two tasks. This suggests that the evidence generation task is challenging for all types of questions.

Type of Question	Answer		Sp Fact		Evidence		Joint	
	EM	F1	EM	F1	EM	F1	EM	F1
Comparison	26.49	27.86	26.76	65.02	0.00	12.81	0.00	2.47
Inference	44.39	65.28	10.77	49.45	0.00	5.31	0.00	2.78
Compositional	57.92	64.78	18.28	57.44	3.59	20.70	1.27	11.40
Bridge-Comparison	18.47	20.47	43.74	89.16	0.00	19.29	0.00	3.63

Table 3.12: Results (%) of the baseline model on different types of questions.

3.7.3 Human Performance

We obtained a human performance on 100 samples that are randomly chosen from the test split. Each sample was annotated by three workers (graduate students). We provided the question, context, and a set of predefined relations (for the evidence generation task) and asked a worker to provide an answer, a set of sentence-level SFs, and a set of evidence. Similar to the previous work (Yang et al., 2018), we computed the upper bound for human performance by acquiring the maximum EM and F1 for each sample. All the results are presented in Table 3.13.

Setting	Answer		Sp Fact		Evidence		Joint	
	EM	F1	EM	F1	EM	F1	EM	F1
Model	54.00	60.38	29.00	69.90	1.00	18.24	1.00	11.06
Human (average)	80.67	83.21	85.33	92.63	68.00	84.40	63.67	73.18
Human UB	92.00	93.45	89.00	94.75	76.00	88.78	74.00	83.14

Table 3.13: Comparing baseline model performance with human performance (%) on 100 random samples. *UB* represents Upper Bound.

The workers achieved higher performance than that of the model. The human performance for the answer prediction task is 91.0 EM and 91.8 F1. There still seems to be room for improvement, which might be because the mismatch information between Wikipedia and Wikidata makes questions unanswerable (see Section 3.7.4 for an analysis). The human performance of the answer prediction task on our dataset (91.8 F1 UB) shows a relatively small gap against that on HotpotQA (98.8 F1 UB; borrowed from their paper). Although the baseline model is able to predict the answer and sentence-level SFs, it is not very effective at finding the evidence. We also observe that there is a large gap between the performance of human and the model in the evidence generation task (78.8 and 16.7 F1). Therefore, this could be a new challenging task for explaining multi-hop reasoning. We conjecture that the main reason why the score of the evidence generation task was low is the ambiguity in the names of Wikidata. For example, in Wikidata, one person can have multiple names. We use only one name in the ground truth, while the workers can use other names. Future research might explore these issues to ensure the quality of the dataset. Overall, our baseline results

are far behind human performance. This shows that our dataset is challenging and there is ample room for improvement in the future.

3.7.4 Discussion

As mentioned in Section 3.7.3, there are unanswerable questions in our dataset due to the mismatch information between Wikipedia articles and Wikidata knowledge. In the dataset generation process, for a triple (s, r, o), we first checked whether the object entity o appears or not in the Wikipedia article that describes the entity s. Our assumption is that the first sentence in the article in which the object entity o appears is the most important, which we decided to use for the QA pair generation. For instance, we obtained a triple: *(Lord William Beauclerk, mother, Lady Diana de Vere)* from Wikidata, and we obtained a paragraph p from the Wikipedia article that describes *"Lord William Beauclerk"*. We used the object entity *"Lady Diana de Vere"* to obtain the first sentence in p *"Beauclerk was the second son of Charles Beauclerk, 1st Duke of St Albans, and his wife Lady Diana de Vere, ..."* From this sentence, we can infer that the mother of *"Lord William Beauclerk"* is *"Lady Diana de Vere"*. However, because we only checked whether the object entity o appears in the sentence or not, there could be a semantic mismatch between the sentence and the triple. For instance, we obtained a triple: *(Rakel Dink, spouse, Hrant Dink)* from Wikidata, while we obtained the first sentence from Wikipedia article: *"Rakel Dink (born 1959) is a Turkish Armenian human rights activist and head of the Hrant Dink Foundation."* Obviously, from this sentence, we cannot infer that *"Hrant Dink"* is the spouse of *"Rakel Dink"*. Therefore, we defined heuristics to exclude these mismatched cases as much as possible. In particular, we found that some examples have subject entities that are similar/equal to their object entities and are likely to become mismatched cases. For such cases, we manually checked the samples and decided to use or remove them for our final dataset. Nonetheless, there are still cases that our heuristics cannot capture. To estimate how many mismatched cases our heuristics cannot capture in the dataset, we randomly selected 100 samples in the training set and manually checked them. We obtained eight out of 100 samples that have a mismatch between Wikipedia article and Wikidata triple. For the next version of the dataset, we plan to improve our heuristics by building a list of keywords for each relation to check the correspondence between Wikipedia

sentence and Wikidata triple. For instance, we observed that for the relation "*mother*", the sentences often contain phrases: "*son of*", "*daughter of*", "*his mother*", and "*her mother*".

3.8 Conclusion

In this chapter, we presented 2WikiMultiHopQA—a large and high quality multi-hop dataset that provides comprehensive explanations for predictions. We utilized logical rules in the KB to create more natural questions that still require multi-hop reasoning. Through experiments, we demonstrated that our dataset ensures multi-hop reasoning while being challenging for the multi-hop models. We also demonstrated that bootstrapping the multi-hop MRC dataset is beneficial by utilizing large-scale available data on Wikipedia and Wikidata.

4

Analyze Reasoning Steps in the Triple Form

To explain the predicted answers and evaluate the reasoning abilities of models, several studies have utilized underlying reasoning (UR) tasks in multi-hop question answering (QA) datasets. However, it remains an open question as to how effective UR tasks are for the QA task when training models on both tasks in an end-to-end manner. In this chapter, we address this question by analyzing the effectiveness of UR tasks (including both sentence-level and entity-level tasks) in three aspects: (1) QA performance, (2) reasoning shortcuts, and (3) robustness. While the previous models have not been explicitly trained on an entity-level reasoning prediction task, we build a multi-task model that performs three tasks together: sentence-level supporting facts prediction, entity-level reasoning prediction, and answer prediction. Experimental results on 2WikiMultiHopQA and HotpotQA-small datasets reveal that (1) UR tasks can improve QA performance. Using four debiased datasets that are newly created, we demonstrate that (2) UR tasks are helpful in preventing reasoning shortcuts in the multi-hop QA

task. However, we find that (3) UR tasks do not contribute to improving the robustness of the model on adversarial questions, such as sub-questions and inverted questions. We encourage future studies to investigate the effectiveness of entity-level reasoning in the form of natural language questions (e.g., sub-question forms).[1]

4.1 Introduction

The task of multi-hop QA requires a model to read and aggregate information from multiple paragraphs to answer a given question (Figure 4.1a). Several multi-hop QA datasets have been proposed, such as QAngaroo (Welbl et al., 2018), HotpotQA (Yang et al., 2018), and MuSiQue (Trivedi et al., 2022). In HotpotQA, the authors provide sentence-level supporting facts (SFs) to test the reasoning ability and explainability of the models. However, owing to the design of the sentence-level SFs task (binary classification) and the redundant information in the sentences, Inoue et al. (2020) and Ho et al. (2020) show that the sentence-level SFs are insufficient to explain and evaluate multi-hop models in detail. To address this issue, R^4C (Inoue et al., 2020) and 2WikiMultiHopQA (2Wiki; Ho et al., 2020) datasets provide an entity-level reasoning prediction task to explain and evaluate the process of answering questions. Entity-level reasoning information is defined as a set of triples that describes the reasoning path from question to answer (Figure 4.1b).

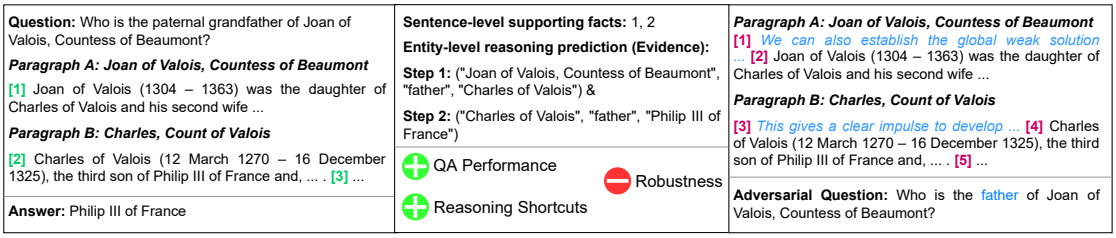

Figure 4.1: Example of (a) a standard multi-hop question, (b) two underlying reasoning tasks in the QA process and three aspects in our analysis, '+' and '-' indicate that the UR tasks have a positive and negative impacts, respectively, and (c) debiased and adversarial examples that are used in our study.

[1]Our data and code are available at https://github.com/Alab-NII/multi-hop-analysis.

Several previous studies (Chen et al., 2019; Fu et al., 2021a) utilize sentence-level SFs and/or entity-level reasoning information to build explainable models by using question decomposition (Min et al., 2019b; Perez et al., 2020) or predicting sentence-level SFs. The advantages of these pipeline models are that they can exploit the underlying reasoning process in QA and their predicted answers are more interpretable. However, the question remains as to how effective training on UR tasks is for the QA task in an end-to-end manner. Although a few end-to-end models have also been introduced (Qiu et al., 2019; Fang et al., 2020), these models are not explicitly trained on entity-level and answer prediction tasks.

In addition to the triple form, the sub-question form is another way to utilize entity-level reasoning information. Specifically, Tang et al. (2021) utilize question decomposition as an additional sub-question evaluation for bridge questions (there are two types of questions: bridge and comparison) in HotpotQA. They only use sub-questions for evaluation and do not fine-tune the models on them. In addition, Ho et al. (2022) use sub-questions for both evaluation and training. However, they only focus on comparison questions for date information. In contrast, we focus on the triple form of the entity-level information and conduct experiments using two datasets, 2Wiki and HotpotQA-small (obtained by combining HotpotQA and R^4C), which include both types of questions.

In this chapter, we analyze the effectiveness of UR tasks (including both sentence-level and entity-level) in three aspects: (1) *QA performance*, (2) *reasoning shortcuts*, and (3) *robustness*. First, QA performance is the final objective of the QA task. We aim to answer the following question: **(RQ1)** *Can the UR tasks improve QA performance?* For the second aspect, previous studies (Chen and Durrett, 2019; Jiang and Bansal, 2019a; Min et al., 2019a; Trivedi et al., 2020) demonstrate that many questions in the multi-hop QA task contain biases and reasoning shortcuts (Geirhos et al., 2020), where the models can answer the questions by using heuristics. Therefore, we aim to ask the following: **(RQ2)** *Can the UR tasks prevent reasoning shortcuts?* For the final aspect, to ensure safe development of NLP models, robustness is one of the important issues and has gained tremendous amount of research (Wang et al., 2022). In this chapter, we aim to test the robustness of the model by asking modified versions of questions, such as sub-questions and inverted questions. Our question is **(RQ3)** *Do the UR tasks make the models more robust?*

There are no existing end-to-end models that can perform three tasks simultaneously (sentence-level SFs prediction, entity-level prediction, and answer prediction); therefore, we first build a multi-hop BigBird-base model (Zaheer et al., 2020) to perform these three tasks simultaneously. We then evaluate our model using two multi-hop datasets: 2Wiki and HotpotQA-small. To investigate the effectiveness of the UR tasks, for each dataset, we conduct three additional experiments in which the model is trained on: (1) answer prediction task, (2) answer prediction and sentence-level prediction tasks, and (3) answer prediction and entity-level prediction tasks. We also create four debiased sets (Figure 4.1c) for 2Wiki and HotpotQA-small for **RQ2**. We create and reuse adversarial questions for 2Wiki and HotpotQA-small for **RQ3**.

The experimental results indicate that the UR tasks can improve QA performance from 77.9 to 79.4 F1 for 2Wiki and from 66.4 to 69.4 F1 for HotpotQA-small **(RQ1)**. The results of the models on the four debiased sets reveal that the UR tasks can be used to reduce reasoning shortcuts **(RQ2)**. Specifically, when the model is trained on both answer prediction and UR tasks, the performance drop of the model on the debiased sets is lower than that when the model is trained only on answer prediction (e.g., 8.9% vs. 13.4% EM). The results also suggest that the UR tasks do not make the model more robust on adversarial questions, such as sub-questions and inverted questions **(RQ3)**. Our analysis shows that correct reconstruction of the entity-level reasoning task contributes to finding the correct answer in only 37.5% of cases. This implies that using entity-level reasoning information in the form of triples does not answer adversarial questions, in this case, the sub-questions. We encourage future work to discover the effectiveness of the entity-level reasoning task in the form of sub-questions that have the same form as multi-hop QA questions.

4.2 Related Work

Multi-hop Datasets and Analyses To test the reasoning abilities of the models, many multi-hop QA datasets (Welbl et al., 2018; Talmor and Berant, 2018; Yang et al., 2018) have been proposed. Recently, Trivedi et al. (2022) introduced MuSiQue, a multi-hop dataset constructed from a composition of single-hop questions. The reason why do we not conduct experiments on MuSiQue is explained in the limitations section.

In addition to Tang et al. (2021) and Ho et al. (2022), the most similar to our research

mentioned in the Introduction, there are some other existing studies (Chen and Durrett, 2019; Jiang and Bansal, 2019a; Min et al., 2019a; Trivedi et al., 2020) on the analysis and investigation of the multi-hop datasets and models. However, most of them do not utilize the internal reasoning information when answering questions.

Multi-hop Models Various directions have been proposed for solving multi-hop datasets, including question decomposition (Talmor and Berant, 2018; Jiang and Bansal, 2019b; Min et al., 2019b; Perez et al., 2020; Wolfson et al., 2020; Fu et al., 2021a), iterative retrieval (Feldman and El-Yaniv, 2019; Asai et al., 2020; Qi et al., 2021), graph neural networks (Song et al., 2018; De Cao et al., 2019; Ding et al., 2019; Qiu et al., 2019; Tu et al., 2019; Fang et al., 2020), and other approaches such as single-hop based models (Yang et al., 2018; Nishida et al., 2019) or transformer-based models (Devlin et al., 2019; Zaheer et al., 2020). Our model is based on the BigBird transformer model.

Other QA Reasoning Datasets In addition to multi-hop reasoning datasets, several other existing datasets also aim to evaluate the reasoning abilities of the models. Some of them are: DROP (Dua et al., 2019) for numerical reasoning; CLUTRR (Sinha et al., 2019), ReClor (Yu et al., 2020), and LogiQA (Liu et al., 2020) for logical reasoning; Quoref (Dasigi et al., 2019) for coreference reasoning; CommonsenseQA (Talmor et al., 2019), MCScript2.0 (Ostermann et al., 2019), and CosmosQA (Huang et al., 2019) for commonsense reasoning. Many of these datasets consist of only a single paragraph in the input or lack explanation information that describes the reasoning process from question to answer. However, our focus is on multi-hop reasoning datasets that contain multiple paragraphs in the input and provide explanatory information for the QA process.

4.3 Background

4.3.1 Reasoning Tasks in Multi-hop QA

We consider UR tasks in multi-hop QA including two levels: *sentence-level* and *entity-level*. The sentence-level SFs prediction task was first introduced by Yang et al. (2018). This task requires a model to predict a set of sentences that is necessary to

answer a question (Figure 4.1).

To evaluate the UR process of the models, derivation and evidence information were introduced in R^4C and 2Wiki, respectively. Both derivation and evidence are sets of triples that represent the reasoning path from question to answer. The difference is the form; derivation in R^4C uses a semi-structured natural language form, whereas evidence in 2Wiki uses a structured form. We conduct experiments with both R^4C (HotpotQA-small) and 2Wiki. For consistency, we use the term *entity-level reasoning prediction task* to denote the derivation task in R^4C and the evidence task in 2Wiki.

4.3.2 Reasoning Shortcuts and Biases

In this chapter, we consider both reasoning shortcuts and biases to be similar. These are spurious correlations in the dataset that allow a model to answer the question correctly without performing the expected reasoning skills, such as comparison and multi-hop reasoning. Following previous studies (Jiang and Bansal, 2019a; Ko et al., 2020), we use the terms *word overlap shortcut* and *position bias*.

To check whether the UR tasks can prevent reasoning shortcuts, we first identify the types of shortcuts that exist in HotpotQA-small and 2Wiki.

Word Overlap Shortcut Using adversarial methods, Jiang and Bansal (2019a) show that examples in HotpotQA often contain word overlap shortcut, where the models can answer the questions by performing word-matching between the question and a sentence in the context. Based on this finding, we automatically calculate the word overlap shortcut for 2Wiki and HotpotQA-small. We observe that the word overlap shortcut is common in bridge questions; therefore, we only calculate the word overlap shortcut for bridge questions in 2Wiki and HotpotQA-small. To check whether a sample contains the word overlap shortcut, we do the following steps:

- Obtain a set of surrounding words S by getting the five words immediately to the left and right of the answer span, then remove stopwords in S.
- Obtain a set of overlapping words (O) between S and a question.
- We consider a sample containing the word overlap shortcut if there are at least two words in O and $\frac{|O|}{|S|} \geq 0.65$. These numbers (threshold) are chosen based on

the evaluation of 40 examples that are manually annotated by the authors.

We find that there are 56 out of 5,791 and 151 out of 715 examples (5,791 and 715 are the numbers of bridge questions in 2Wiki and HotpotQA-small) in the dev. sets of 2Wiki and HotpotQA-small containing the word overlap shortcut.

In summary, we find that the word overlap shortcut is common in HotpotQA-small, but not in 2Wiki. The small sample size of HotpotQA-small (Section 4.5) increases the uncertainty of the obtained results. Therefore, within the scope of this study, we mainly experiment with position bias.

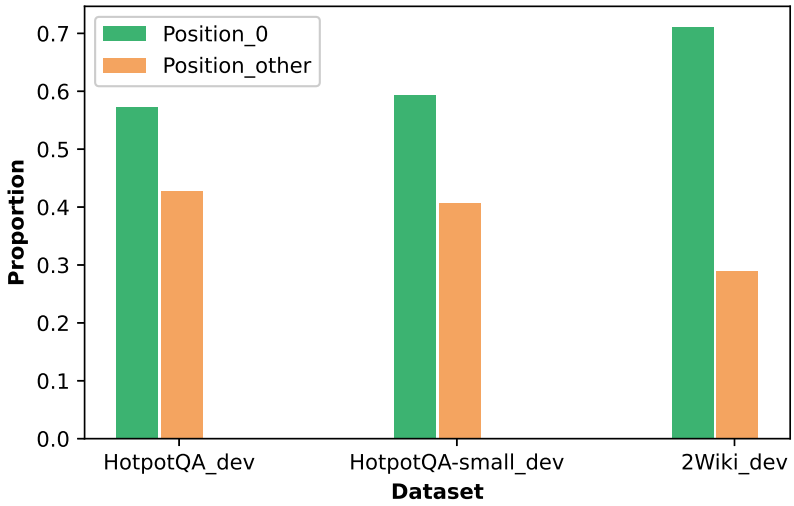

Figure 4.2: Information on the position of sentence-level SFs in the dev. sets of the three datasets.

Position Bias We observe that many examples in 2Wiki contain answers in the first sentence. Therefore, we divide every sentence-level SF in each gold paragraph into two levels: the first sentence (position_0) and the remaining sentences (position_other). Subsequently, we obtain the percentage of each level by dividing the total number of each level (e.g., position_0) by the total number of SFs. Figure 4.2 illustrates the information on the position of sentence-level SFs in dev. sets of three datasets. We find that all three datasets have a bias toward the first sentence. We also find that 2Wiki has more position biases than HotpotQA and HotpotQA-small.

It is noted that there is another type of shortcut, namely, entity-type matching shortcut. Based on the experimental results and human performance, Min et al. (2019a) reveal that examples in HotpotQA contain the entity type matching shortcut, where the models can answer the questions by using the first five tokens in the questions; meanwhile, humans can answer the questions by using the entity type of the paragraphs. Currently, there is no dataset that can prevent the entity-type shortcut; therefore, we do not use this type of shortcut in our experiments.

4.4 Our Multi-task Model

To investigate the usefulness of UR tasks for the QA task, we jointly train the corresponding tasks: sentence-level SFs prediction, entity-level prediction, and answer prediction. Figure 4.3 illustrates our model. To handle long texts, we use the BigBird model (Zaheer et al., 2020), which is available in Hugging Face's transformers repository.[2] Our model comprises three main steps: (1) paragraph selection, (2) context encoding, and (3) multi-task prediction. We use the named entity recognition (NER) models of Spacy[3] and Flair (Akbik et al., 2019) to extract all entities in the context and use them for the entity-level prediction task.

Paragraph Selection Following previous models (Qiu et al., 2019; Fang et al., 2020; Tu et al., 2020), instead of using all the provided paragraphs, we first filter out answer-unrelated paragraphs. We follow the paragraph selection process described in Fang et al. (2020). First, we retrieve first-hop paragraphs by using title matching or entity matching. We then retrieve second-hop paragraphs using the hyperlink information available in Wikipedia. When we retrieve paragraphs, we reuse a paragraph ranker model[4] from the hierarchical graph network (HGN) model (Fang et al., 2020) to rank input paragraphs using the probability of whether they contain sentence-level SFs.

Context Encoding To obtain vector representations for sentences and entities, we first combine all the selected paragraphs into one long paragraph and then concatenate

[2] https://huggingface.co/transformers/model_doc/bigbird.html
[3] https://spacy.io/
[4] https://github.com/yuwfan/HGN

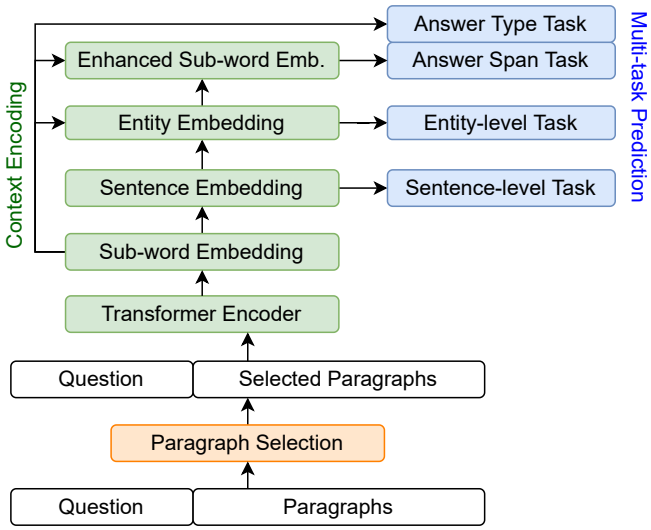

Figure 4.3: Our model has three main steps: paragraph selection, context encoding, and multi-task prediction.

it with the question to form a context C. Specifically,

$C = [[\text{CLS}], q_1, ..., q_m, [\text{SEP}], p_1, ..., p_n, [\text{SEP}]]$, where m and n are the lengths of the question q and the combined paragraph p (all selected paragraphs), respectively. The context C is then tokenized into l sub-words before feeding into BigBird to obtain the contextual representation C' of the sub-words:

$$C' = \text{BigBird}(C) \in \mathbb{R}^{l \times h}, \tag{4.1}$$

where h is the hidden size of the BigBird model. Next, we obtain the representation $s_i \in \mathbb{R}^{2h}$ of the i-th sentence and the representation $e_j \in \mathbb{R}^{4h+d_t}$ of the j-th entity, as follows:

$$\begin{aligned} s_i &= C'_{S^i_{\text{start}}}; C'_{S^i_{\text{end}}} \\ e_j &= C'_{E^j_{\text{start}}}; C'_{E^j_{\text{end}}}; t_j; s_k, \end{aligned} \tag{4.2}$$

where [;] denotes the concatenation of the two vectors, $C'_{S^i_{\text{start}}}$ and $C'_{E^j_{\text{start}}}$ denote the first sub-word representations of the i-th sentence and j-th entity, respectively. $C'_{S^i_{\text{end}}}$ and $C'_{E^j_{\text{end}}}$ denote the last sub-word representations of the i-th sentence and j-th entity, respectively. We enrich the entity embedding e_j by concatenating it with a

d_t-dimensional type embedding t_j and a sentence embedding s_k, where k is the index of the sentence containing the j-th entity.

We also leverage the entity information to improve the contextual representation of sub-words C' as it is mainly used for the answer prediction task, which will be described in the next section. Thus, the enhanced sub-word representation C''_i of the i-th sub-word is calculated as follows:

$$C''_i = C'_i; e_k \in \mathbb{R}^{5h+d_t}, \tag{4.3}$$

where e_k is the embedding of the k-th entity containing the i-th sub-word. Otherwise, e_k is a null vector with the same dimension.

Multi-task Prediction After context encoding, we train our model on three main tasks together: (1) sentence-level prediction, (2) entity-level prediction, (3) and answer prediction. We split the answer prediction task into two sub-tasks, similar to previous studies (Yang et al., 2018; Fang et al., 2020), including answer type prediction and answer span prediction. We train our model by minimizing the joint loss for all tasks, as follows:

$$\begin{aligned} L_{\text{joint}} = \lambda_{\text{sent}} L_{\text{sent}} + \lambda_{\text{ent}} L_{\text{ent}} + \\ \lambda_{\text{ans}} (L_{\text{start}} + L_{\text{end}} + L_{\text{type}}), \end{aligned} \tag{4.4}$$

where λ_{sent}, λ_{ent}, and λ_{ans} are the hyper-parameters for three tasks: sentence-level prediction, entity-level prediction, and answer prediction.

For the sentence-level prediction task, we use a binary classifier to predict whether a sentence is a supporting fact. For the answer type prediction task, we use a 4-way classifier to predict the probabilities of *yes*, *no*, *span*, and *no answer*. Two linear classifiers are used for the answer span prediction task to independently predict the start and end tokens of the answer span.

Different from existing end-to-end models (Qiu et al., 2019; Fang et al., 2020), our model is explicitly trained on the entity-level prediction task. We formalize the entity-level reasoning prediction task as a relation extraction task (Zhang and Wang, 2015). The input is a pair of entities, and the output is the relationship between two entities. From all named entities obtained by using the NER models, we generate a set

Split	2Wiki	HotpotQA-small
Train	167,454	3,671
Dev.	12,576	917
Test	12,576	-
Debiased	12,576 (x4)	917 (x4)
Adversarial	12,576	659 & 134

Table 4.1: Statistics for 2Wiki and HotpotQA-small. There are four debiased sets in 2Wiki and HotpotQA-small. There are one adversarial set in 2Wiki and two adversarial sets in HotpotQA-small.

of entity pairs; for example, given N entities, we obtain $N \times (N-1)$ pairs. For each pair, we predict a relationship in a set of predefined relationships obtained from the training set. We then use cross-entropy as the learning objective.

4.5 Datasets and Evaluation Metrics

We mainly experiment with 2Wiki and HotpotQA-small. We also train and evaluate our model on the full version of HotpotQA. We reuse and create debiased and adversarial sets for the evaluation. Table 4.1 presents the statistics for 2Wiki, HotpotQA-small, and additional evaluation sets. It should be noted that all datasets are in English.

4.5.1 HotpotQA-small

R^4C (Inoue et al., 2020) is created by adding entity-level reasoning information to the samples in HotpotQA. We obtain HotpotQA-small by combining HotpotQA (Yang et al., 2018) with R^4C. HotpotQA-small comprises three tasks as in 2Wiki: (1) sentence-level SFs prediction, (2) entity-level prediction, and (3) answer prediction. First, we re-split the ratio between the training and dev. sets; the new sizes are 3,671 and 917 for the training and dev. sets, respectively (the original sizes are 2,379 and 2,209, respectively). In R^4C, there are three gold annotations for the entity-level prediction task; in 2Wiki, there is only one gold annotation. For consistency in the evaluation and analysis, we randomly choose one annotation from the three annotations for every sample in R^4C.

The entity-level reasoning in R^4C is created by crowdsourcing. We observe that

there are many similar relations in the triples in R^4C, and these relations can be grouped into one. For example, *is in*, *is located in*, *is in the*, and *is located in the* indicate location relation. We also group the relations by removing the context information in the relations; for example, *is a 2015 book by* and *is the second book by* are considered similar to the relation *is a book by*. After grouping, the number of relations in R^4C is 2,526 (it is 4,791 before).

4.5.2 Debiased Dataset

The objective of our debiased dataset is to introduce a small perturbation in each paragraph to mitigate a specific type of bias, in our case, the position bias shown in Figure 4.2. For both 2Wiki and HotpotQA-small, we use the same method to generate four debiased sets: AddUnrelated, AddRelated, Add2, and Add2Swap. The differences between these four sets are whether the sentence is related or unrelated to the paragraph and whether we add one or two sentences into the paragraph. The details of each set are as follows.

AddUnrelated: One sentence unrelated to the paragraph is added. In our experiment, we use a list of sentences in the sentence-level revision dataset (Tan and Lee, 2014). We randomly choose one sentence that has a number of tokens greater than eleven but less than twenty-one.

AddRelated: One sentence that does not have an impact on the meaning or flow of the paragraph is added. In our experiment, we write multiple templates for each entity type (e.g., for a film entity, "#Name is a nice film", where *#Name* is the title of the paragraph), then randomly choose one template, and add it to the paragraph. To detect the type of the paragraph, we use the question type information in 2Wiki and HotpotQA-small, the results of the NER model, and the important keywords in the question (e.g., who, magazine, album, and film).

Add2: AddRelated and AddUnrelated are combined in order.

Add2Swap: The order of AddRelated and AddUnrelated in Add2 is swapped.

4.5.3 Adversarial Dataset

The objective of our adversarial dataset is to check the robustness of the model by asking modified versions of questions. For HotpotQA-small, we reuse two versions of

adversarial examples in Geva et al. (2022). The first one is automatically generated by using the 'Break, Perturb, Build' (BPB) framework in Geva et al. (2022). The BPB framework performs three main steps: (1) breaking a question into multiple reasoning steps, (2) perturbing the reasoning steps by using a list of defined rules, and (3) building new QA samples from the perturbations in step #2. The second version is a subset of the first version and is validated by crowd workers. We only use the examples in these two versions that the original examples appear in HotpotQA-small.

For 2Wiki, no adversarial dataset is available. Based on the idea of the BPB framework in Geva et al. (2022), we apply two main rules from BPB for 2Wiki: (1) replace the comparison operation for comparison questions, and (2) use the prune step for bridge questions. For the first rule, we replace the operation in the comparison questions (e.g., "Who was born first, A or B?" is converted to "Who was born later, A or B?"). For the second rule, we use a sub-question in the QA process as the main question (e.g., for Figure 4.1, we ask, "Who is the father of Joan of Valois?").

4.5.4 Evaluation Metrics

Each task in HotpotQA and 2Wiki is evaluated by using two metrics: exact match (EM) and F1 score. Following the evaluation script in HotpotQA and 2Wiki, we use joint EM and joint F1 to evaluate the entire capacity of the model. For HotpotQA, they are the products of the scores of two tasks: sentence-level prediction and answer prediction. For 2Wiki and HotpotQA-small, they are the products of the scores of three tasks: sentence-level prediction, entity-level prediction, and answer prediction.

4.6 Results

Currently, there are no existing end-to-end models that explicitly train all three tasks together; therefore, in this chapter, we use our proposed model for analysis. We also compare our model with other previous models on the HotpotQA and 2Wiki datasets. In general, the experimental results indicate that our model is comparable to previous models and can be used for further analyses.

4.6 Results

Dataset	Model	Answer		Sentence-level		Entity-level		Joint	
		EM	F1	EM	F1	EM	F1	EM	F1
HotpotQA	HGN-BERT‡ (Fang et al., 2020)	*N/A*	74.76	*N/A*	86.61	×	×	*N/A*	66.90
	HGN-RoBERTa (Fang et al., 2020)	**68.93**	**82.18**	63.09	**88.59**	×	×	46.46	**74.34**
	SAE-BERT (Tu et al., 2020)	61.32	74.81	58.06	85.27	×	×	39.89	66.45
	SAE-RoBERTa (Tu et al., 2020)	67.70	80.75	**63.30**	87.38	×	×	**46.81**	72.75
	Our BigBird-base	61.90	76.09	58.54	86.93	×	×	39.39	67.81
HotpotQA-small	Our BigBird-base	54.74	69.44	75.14	90.88	6.43	31.05	4.25	21.69
2Wiki	BiDAF (Ho et al., 2020)	36.53	43.93	24.99	65.26	1.07	14.94	0.35	5.41
	CRERC (Fu et al., 2021a)	69.58	72.33	82.86	90.68	**54.86**	68.83	49.80	58.99
	NA-Reviewer (Fu et al., 2021b)	**76.73**	**81.91**	**89.61**	**94.31**	53.66	70.83	**52.75**	**65.23**
	Our BigBird-base	74.05	79.68	77.14	92.13	45.75	**76.64**	39.30	63.24
	Human UB (Ho et al., 2020)	91.00	91.79	88.00	93.75	64.00	78.81	62.00	75.25

Table 4.2: Results (%) of our model and previous models in the dev. set of HotpotQA and in the test set of 2Wiki. We also show the performance of our model in the dev. set of HotpotQA-small. *Answer*, *Sentence-level*, and *Entity-level* represent the answer prediction task, sentence-level prediction task, and entity-level prediction task, respectively. For HGN-BERT, the scores that we obtained (from left to right: 58.93 73.18 54.64 85.34 35.11 64.24) are lower than the reported scores in HGN (Fang et al., 2020); therefore, we show the reported F1 scores in HGN.

4.6.1 Results Comparison

We compare our results with three previous models: BiDAF, CRERC, and NA-Reviewer. BiDAF is a baseline model in Ho et al. (2020). CRERC (Fu et al., 2021a) is a pipeline model that includes three modules: relation extractor, reader, and comparator. NA-Reviewer (Fu et al., 2021b) is an improved version of CRERC, as it addresses the error accumulation issue. It is noted that both CRERC and NA-Reviewer models are evaluated on only 2Wiki.

Table 4.2 presents the results of our model and previous models in the dev. set of HotpotQA and in the test set of 2Wiki. It also shows the performance of our model in the dev. set of HotpotQA-small and human performance in Ho et al. (2020).

Results on HotpotQA Our score is comparable to the BERT-base version of two strong models, SAE (Tu et al., 2020) and HGN (Fang et al., 2020) in the dev. set of the distractor setting in HotpotQA. Specifically, our joint F1 is 67.8, while for SAE-BERT, it is 66.5, and for HGN-BERT, it is 66.9. However, our score is smaller than

the RoBERTa-base of SAE and HGN. They are 72.8 and 74.4 F1 for SAE-RoBERTa and HGN-RoBERTa, respectively. It is noted that we use the BigBird-ITC version in our model. Although the BigBird-ETC version performs better than the BigBird-ITC version, it is not available in Hugging Face. We do not use SAE and HGN for our analyses because these models are not designed to train on the entity-level reasoning prediction task.

Results on HotpotQA-small The scores on HotpotQA-small are lower than those on HotpotQA in the answer prediction task. This result may be explained by the fact that the training size of HotpotQA-small is smaller than HotpotQA (3,671 vs. 90,564). Due to the small size, we only use the gold paragraphs for experiments. That is why the scores on HotpotQA-small are higher than those on HotpotQA in the sentence-level task. For the entity-level task, the EM score is quite low (6.4 EM). A possible reason for this is that there are many relations in HotpotQA-small (2,526 relations); meanwhile, there are only 33 relations in 2Wiki. We observe that the F1 score (31.1 F1) is much better than the EM score. Therefore, we keep using HotpotQA-small for analyses.

Results on 2Wiki Our model significantly outperforms BiDAF in all tasks. Our results are comparable to CRERC. The EM score of our model in the entity-level task is lower than that of CRERC. A possible explanation for this might be that the relation extractor module in CRERC is fine-tuned on 2Wiki; therefore, it can extract entities better than the NER models from Spacy and Flair that are used in our model. However, the F1 score of our model in the entity-level task is higher than that of CRERC. This indicates that our model can correctly obtain a few triples in a set of gold triples for many samples. All our scores (except the F1 score of the entity-level task) are lower than those on NA-Reviewer. Our target is to analyze the UR tasks in an end-to-end model. Although the pipeline models (CRERC and NA-Reviewer) are easy to interpret, we cannot determine how the UR tasks affect answer prediction in an end-to-end model. Therefore, we use the design of our model to perform the analyses in this study.

Dataset	Task Setting	Answer		Sentence-level		Entity-level	
		EM	F1	EM	F1	EM	F1
2Wiki	(1) Ans	72.03	77.87	-	-	-	-
	(2) Ans + Sent	72.82	78.65	78.06	92.38	-	-
	(3) Ans + Ent	72.33	78.21	-	-	46.11	76.65
	(4) Ans + Sent + Ent	**73.60**	**79.37**	78.46	92.68	45.97	76.69
HotpotQA-small	(1) Ans	52.89	66.43	-	-	-	-
	(2) Ans + Sent	54.42	69.03	75.35	91.00	-	-
	(3) Ans + Ent	54.74	69.08	-	-	6.54	31.31
	(4) Ans + Sent + Ent	**54.74**	**69.44**	75.14	90.88	6.43	31.05

Table 4.3: Ablation study results (%) of our model in the dev. sets of 2Wiki and HotpotQA-small. *Ans*, *Sent*, and *Ent* represent the answer prediction task, sentence-level SFs prediction task, and entity-level prediction task, respectively. 'Task Setting' represents the tasks that the model is trained on. '-' indicates the tasks the model is not trained on.

4.6.2 Effectiveness of the UR Tasks

To investigate the effectiveness of the UR tasks, we train the model in four settings: (1) answer prediction only, (2) answer prediction and sentence-level SFs prediction, (3) answer prediction and entity-level prediction, and (4) all three tasks together.

QA Performance (RQ1) Our first research question is whether the UR tasks can improve QA performance. To answer this question, we compare the results of different task settings described above. The results are presented in Table 4.3. For 2Wiki, using sentence-level and entity-level separately (settings #2 and #3), the QA performance does not change significantly. The improvement is significant when we combine both the sentence-level and entity-level (setting #4). Specifically, the scores when the model is trained on the answer prediction task only (setting #1) and on both the answer prediction task and UR tasks (setting #4) are 77.9 and 79.4 F1, respectively. In contrast to 2Wiki, using sentence-level and entity-level separately, there is a larger QA performance improvement in HotpotQA-small. Specifically, the F1 scores of settings #2 and #3 are 69.0 and 69.1, respectively, whereas, the F1 score of the first setting is 66.4. Similar to 2Wiki, there is a large gap between the two settings, #1 and #4 (66.4 F1 and

Dataset	Task Setting	Reduction (%) on Four Debiased Sets							
		AddUnrelated		AddRelated		Add2		Add2Swap	
		EM	F1	EM	F1	EM	F1	EM	F1
2Wiki	(1) Ans	13.40	12.13	3.55	3.46	12.32	11.72	18.99	17.51
	(2) Ans + Sent	11.00	9.71	4.16	4.22	11.22	10.69	17.62	16.24
	(3) Ans + Ent	**7.73**	**6.94**	**2.80**	**2.77**	**8.38**	**7.76**	**13.12**	**12.21**
	(4) Ans + Sent + Ent	8.86	8.11	3.16	3.13	9.09	8.58	14.53	13.77
HotpotQA-small	(1) Ans	3.01	1.53	4.04	1.50	1.65	1.01	3.96	2.47
	(2) Ans + Sent	**1.13**	**1.35**	-0.51	0.19	**0.08**	**0.85**	**1.77**	**1.96**
	(3) Ans + Ent	6.73	5.60	**-0.92**	**0.03**	4.02	3.54	6.89	5.46
	(4) Ans + Sent + Ent	5.05	4.65	1.26	1.25	1.83	2.46	3.58	3.64

Table 4.4: Average performance drop from five times running (smaller is better) of the four settings on the four debiased sets of 2Wiki and HotpotQA-small. The best and worst scores are boldfaced and underlined, respectively.

69.4 F1, respectively).

In summary, these results indicate that both sentence-level and entity-level prediction tasks contribute to improving QA performance. These results align with the findings in Yang et al. (2018), which shows that incorporating the sentence-level SFs prediction task can improve QA performance. We also find that when combining both sentence-level and entity-level prediction tasks, the scores of the answer prediction task are the highest.

Reasoning Shortcuts (RQ2) To investigate whether explicitly optimizing the model on the UR tasks can prevent reasoning shortcuts, we evaluate the four settings of the model on the four debiased sets of 2Wiki and HotpotQA-small. The generation of the debiased sets includes stochastic steps. To minimize the impact of randomness on our reported results, we generate the debiased sets five times and report the average evaluation scores. The average performance drops are presented in Table 4.4.

Overall, for 2Wiki, when the model is trained on only one task (#1), the drop is the largest (except for AddRelated, which is the second largest). When the model is trained only on the answer prediction task, the drops are always higher than those when the model is trained on three tasks. Specifically, the gaps between the two settings, #1 (only answer task) and #4 (all three tasks), are 4.5%, 0.4%, 3.2%, 4.5% (EM score) for AddUnrelated, AddRelated, Add2, and Add2Swap, respectively. These

Task Setting	Dev-adver		Reduction %	
	EM	F1	EM	F1
Ans	37.09	46.07	**48.51**	**40.84**
Ans + Sent	34.26	43.64	52.95	44.51
Ans + Ent	32.67	39.43	54.83	49.58
Ans + Sent + Ent	34.19	42.74	53.55	46.15

Table 4.5: Results of our model in the dev-adversarial set of 2Wiki and the performance drop.

scores indicate that the two tasks, sentence-level and entity-level, positively affect the answer prediction task when the model is trained on three tasks simultaneously.

For HotpotQA-small, we observe that the effectiveness of the UR tasks is inconsistent. For example, for AddUnrelated, when training the model on the three tasks (setting #4), the reduction is larger than that when training on answer task only (setting #1) (5.1 vs. 3.0 EM). However, for AddRelated, the reduction on setting #4 is smaller than that on setting #1 (1.3 vs. 4.0 EM). One possible reason is that the performance of the entity-level task is not good (6.4 EM), which affects the answer prediction task when the model is trained on the three tasks together. Another possible reason is that the position bias in HotpotQA-small is not sufficiently large. We present a detailed analysis in Section 4.6.3 to explain this case.

Robustness (RQ3) To test whether the UR tasks can help to improve the robustness of the model, we evaluate the four settings of the model on the adversarial sets. For 2Wiki, the results are presented in Table 4.5. The scores for all four settings decrease significantly on the adversarial set. The reduction is the smallest when the model is trained on the answer task only. The UR tasks do not make the model more robust on this adversarial set. For HotpotQA-small, we observe the same behavior, that is, when the model is trained on the answer task only, the reduction is the smallest. All results are presented in Table 4.6. These results indicate that both sentence-level and entity-level prediction tasks do not contribute to improving the robustness of the models on adversarial questions, such as sub-questions and inverted questions. We analyze the results in Section 4.6.3.

Task Setting	Dev		Dev-Adver		Adver↓ (%)		Dev-Adver-val		Adver-val↓ (%)	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
(1) Ans	52.89	66.43	40.36	51.23	23.69	**22.88**	37.31	46.69	**29.46**	**29.72**
(2) Ans + Sent	54.42	69.03	41.73	52.50	23.32	23.95	34.33	43.86	36.92	36.46
(3) Ans + Ent	54.74	69.08	42.79	52.16	**21.83**	24.49	27.61	36.86	49.56	46.64
(4) Ans + Sent + Ent	54.74	69.44	40.52	51.14	25.98	26.35	31.34	38.22	42.75	44.96

Table 4.6: Results of our model in the dev. and two dev-adversarial sets of HotpotQA-small. 'Adver' denotes adversarial and 'Adver-val' denotes the adversarial set that was validated by crowd workers.

4.6.3 Analyses

Details of RQ2 To investigate the results concerning RQ2 in more depth, we first analyze the position biases of different types of questions in 2Wiki and HotpotQA-small. Figure 4.4 illustrates the information on the position of sentence-level SFs of comparison and bridge questions in the dev. sets of the two datasets: 2Wiki and HotpotQA-small. As shown in the Figure, the comparison questions have more position biases than the bridge questions in both 2Wiki and HotpotQA-small. Furthermore, we observe that the position bias in the comparison questions in HotpotQA-small is smaller than that in 2Wiki. To evaluate the effectiveness of the position bias for each type of question, we evaluate the four settings of the model on the four debiased sets for each type of question in both datasets. Table 4.8 and Table 4.9 present the performance drop for two types of questions, comparison and bridge questions, in 2Wiki and HotpotQA-small, respectively.

For 2Wiki, we find that most of the answers are in the first sentences in the comparison questions. This large bias is the main reason for the significant reduction in the scores in the comparison questions. 2Wiki has 46.0% of comparison questions. The reduction in comparison questions contributes to the reduction in the entire dataset. In other words, the results of 2Wiki are affected by those of the comparison questions. HotpotQA-small has only 22.0% of comparison questions, and the position bias in the comparison questions was not sufficiently large. Therefore, the position bias does not have a significant impact on the main QA task. In other words, the UR tasks do not have a significant effect.

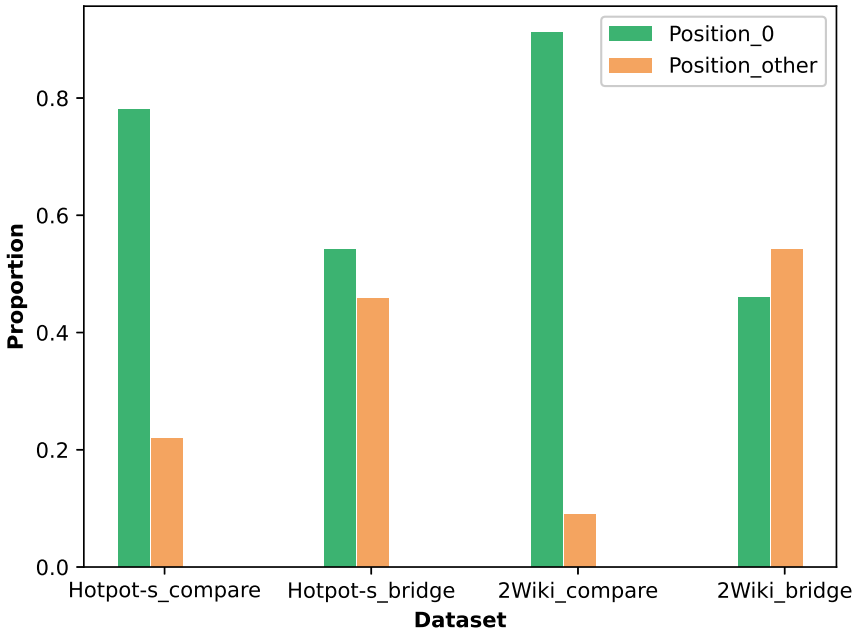

Figure 4.4: Information on the position of sentence-level SFs of comparison and bridge questions in the dev. sets of the two datasets: 2Wiki and HotpotQA-small.

Details of RQ3 The adversarial questions used in RQ3 are the sub-questions in the QA process for bridge questions and the inverted questions for comparison questions. We observe that the triple in the entity-level task is helpful in answering the sub-questions. For example, the triple is: (*Charles of Valois, father, Philip III of France*) and the sub-question is "*Who is the father of Charles of Valois?*". To understand more on the behaviors of the model, we analyze the results from 2Wiki in two settings: (3) Ans + Ent and (4) Ans + Sent + Ent. Table 4.7 presents the detailed results for these two settings. We find that correct reconstruction of the entity-level reasoning task contributes to finding the correct answer only in 32.8% of cases in setting #3 and only in 37.5% of cases in setting #4. Entity-level reasoning in the form of triples has no significant effect on the main QA process. Several examples are presented in Table 4.10.

We conjecture that there are three possible reasons why the UR tasks cannot contribute to the adversarial dataset. The first one is the difference in the form and design of the tasks. Specifically, the entity-level reasoning task is formulated as a

Task Setting	Correct Ans	Correct Ent	Correct Both Ans & Ent
(3) Ans + Ent	4,109	6,851	2,249 (32.8%)
(4) Ans + Sent + Ent	4,300	6,450	2,420 (37.5%)

Table 4.7: Number of correct predicted answers, number of correct predicted entity-level reasoning, and number of examples that have both correct predicted answers and correct predicted entity-level reasoning.

relation extraction task; the input is a pair of entities, and the output is a relation label. Meanwhile, the adversarial dataset is formulated as a QA task; the input is a natural language question, and the output is an answer. The second reason is the incompetence of the entity-level reasoning information. As discussed in Ho et al. (2022), the entity-level reasoning in the comparison questions does not describe the full path from question to answer, and other reasoning operations are required to obtain the answer. The final reason is the manner in which we utilize the entity-level reasoning information. Our model does not consider the order of the triples in the reasoning chain. For example, we do not consider the order of the two steps in Figure 4.1b. We hope that our research will inspire future studies to investigate the effectiveness of the UR tasks in the form of a natural language question, which has the same form as a multi-hop QA question.

4.7 Conclusion

We analyze the effectiveness of the underlying reasoning tasks using two multi-hop datasets: 2Wiki and HotpotQA-small. The results reveal that the underlying reasoning tasks can improve QA performance. Using four debiased sets, we demonstrate that the underlying reasoning tasks can reduce the reasoning shortcuts of the QA task. The results also reveal that the underlying reasoning tasks do not make the models more robust on adversarial examples, such as sub-questions and inverted questions. We encourage future studies to investigate the effectiveness of the entity-level reasoning task in the form of sub-questions.

Dataset	Task Setting	Comparison				Bridge			
		Answer		Answer↓ (%)		Answer		Answer↓ (%)	
		EM	F1	EM	F1	EM	F1	EM	F1
2Wiki									
Dev	Ans	78.98	83.74			66.10	72.85		
	Ans + Sent	79.45	84.21			67.16	73.90		
	Ans + Ent	78.86	83.60			66.75	73.61		
	Ans + Sent + Ent	80.35	85.08			67.84	74.49		
AddUnrelated	Ans	59.51	64.49	24.65	22.99	65.01	71.81	**1.65**	**1.43**
	Ans + Sent	65.55	71.11	17.50	15.56	64.42	71.14	4.08	3.73
	Ans + Ent	67.67	72.84	14.19	**12.87**	65.47	72.44	1.92	1.59
	Ans + Sent + Ent	69.38	74.01	**13.65**	13.01	65.72	72.48	3.13	2.70
AddRelated	Ans	73.60	78.22	6.81	6.59	65.73	72.36	**0.56**	**0.67**
	Ans + Sent	74.87	79.43	5.76	5.68	65.38	71.82	2.65	2.81
	Ans + Ent	75.57	80.17	**4.17**	**4.10**	66.06	72.75	1.03	1.17
	Ans + Sent + Ent	76.69	81.28	4.56	4.47	66.63	73.14	1.78	1.81
Add2	Ans	61.61	65.54	21.99	21.73	64.55	71.60	2.34	1.72
	Ans + Sent	64.93	69.13	18.28	17.91	64.58	71.50	3.84	3.25
	Ans + Ent	67.16	71.43	**14.84**	**14.56**	65.51	72.52	**1.86**	**1.48**
	Ans + Sent + Ent	67.85	72.19	15.56	15.15	66.06	72.94	2.62	2.08
Add2Swap	Ans	51.13	55.50	35.26	33.72	64.42	71.55	**2.54**	**1.78**
	Ans + Sent	55.19	60.21	30.53	28.50	63.96	70.83	4.76	4.15
	Ans + Ent	60.42	64.80	**23.38**	**22.49**	65.04	71.99	2.56	2.20
	Ans + Sent + Ent	60.25	64.37	25.02	24.34	65.51	72.23	3.43	3.03

Table 4.8: Performance drop (smaller is better) for two types of questions (comparison and bridge questions) of the four settings of the model on the four debiased sets of **2Wiki**. The best and worst scores are boldfaced and underlined, respectively.

Dataset	Task Setting	Comparison				Bridge			
		Answer		Answer↓ (%)		Answer		Answer↓ (%)	
		EM	F1	EM	F1	EM	F1	EM	F1
Dev	Ans	56.44	61.86			51.89	67.72		
	Ans + Sent	57.92	63.44			53.43	70.61		
	Ans + Ent	57.92	63.14			53.85	70.75		
	Ans + Sent + Ent	57.43	64.44			53.99	70.86		
AddUnrelated	Ans	50.00	56.24	11.41	9.09	50.77	66.85	**2.16**	**1.28**
	Ans + Sent	52.97	60.64	**8.55**	**4.41**	52.03	68.17	2.62	3.46
	Ans + Ent	51.49	57.43	11.10	9.04	51.33	67.97	4.68	3.93
	Ans + Sent + Ent	47.03	55.59	18.11	13.73	52.17	68.16	3.37	3.81
AddRelated	Ans	53.96	60.48	4.39	2.23	50.07	67.14	3.51	0.86
	Ans + Sent	57.43	63.37	0.85	0.11	54.13	70.54	**-1.31**	0.10
	Ans + Ent	58.91	64.11	**-1.71**	**-1.54**	54.13	70.27	-0.52	0.68
	Ans + Sent + Ent	53.96	61.23	6.04	4.98	54.69	71.24	-1.30	**-0.54**
Add2	Ans	54.46	59.52	3.51	3.78	51.75	67.53	0.27	0.28
	Ans + Sent	58.91	64.31	**-1.71**	**-1.37**	52.45	69.03	1.83	2.24
	Ans + Ent	56.93	62.33	1.71	1.28	50.91	67.81	5.46	4.16
	Ans + Sent + Ent	55.94	62.58	2.59	2.89	54.41	70.82	**-0.78**	**0.06**
Add2Swap	Ans	48.51	53.94	14.05	12.80	50.63	66.94	2.43	**1.15**
	Ans + Sent	53.47	60.30	7.68	4.95	52.03	68.16	2.62	3.47
	Ans + Ent	53.96	60.51	**6.84**	**4.17**	51.05	67.78	5.20	4.20
	Ans + Sent + Ent	50.99	58.64	11.21	9.00	53.29	69.33	**1.30**	2.16

Table 4.9: Performance drop (smaller is better) for two types of questions (comparison and bridge questions) of the four settings of the model on the four debiased sets of **HotpotQA-small**. The best and worst scores are boldfaced and underlined, respectively.

Type	Example
Bridge - Prune	**Paragraph A:** Polish-Russian War (Wojna polsko-ruska) is a 2009 Polish film directed by Xawery Żuławski based on … **Paragraph B:** Xawery Żuławski (born 22 December 1971 in Warsaw) is a Polish film director. …He is the son of actress Małgorzata Braunek and director Andrzej Żuławski. … **Q:** Who is the director of Polish-Russian War? **Predicted answer:** Andrzej Żuławski ✗ **Predicted entity-level:** (“Polish-Russian War”, “director”, “Xawery Żuławski”) ✓
Bridge - Prune	**Paragraph A:** Francesca von Habsburg (born 7 June 1958) is an art collector and the estranged wife of Karl von Habsburg, current head of the House of Habsburg-Lorraine. **Paragraph B:** Michaela von Habsburg was born …She is the twin sister of Monika von Habsburg, and daughter of Otto von Habsburg and Princess Regina of Saxe - Meiningen. **Q:** Who is the spouse of Francesca von Habsburg? **Predicted answer:** Princess Regina of Saxe - Meiningen ✗ **Predicted entity-level:** (“Francesca von Habsburg”, “spouse”, “Karl von Habsburg”) ✓
Comparison - Inverted	**Paragraph A:** Montréal/Les Cèdres Airport is a general aviation aerodrome located approximately west of Montreal, Quebec, Canada near Autoroute 20 west of … **Paragraph B:** Flying J Ranch Airport is a privately owned, public use …The airport is located southwest of the central business district of Pima, a city in Graham County, Arizona, United States and northeast of Tucson International Airport. … **Q:** Are Montréal/Les Cèdres Airport and Flying J Ranch Airport located in different countries? **Predicted answer:** no ✗ **Predicted entity-level:** (“Flying J Ranch Airport”, “country”, “United States”) & (“Montréal/Les Cèdres Airport”, “country”, “Canada”) ✓
Comparison - Inverted	**Paragraph A:** A Romance of the Air is a 1918 American silent drama film based …Directed by Harry Revier, the film was … **Paragraph B:** Harry Revier (16 March 1890 – 13 August 1957) was ... American director … **Paragraph C:** How Moscha Came Back is a 1914 silent film comedy short directed by Phillips Smalley. … **Paragraph D:** Phillips Smalley (August 7, 1865 – May 2, 1939) was an American silent film director and actor. **Q:** Which film has the director who was born later, A Romance of the Air or How Moscha Came Back? **Predicted answer:** How Moscha Came Back ✗ **Predicted entity-level:** (“A Romance of the Air”, “director”, “Harry Revier”), (“How Moscha Came Back”, “director”, “Phillips Smalley”), (“Harry Revier”, “date of birth”, “16 March 1890”), & (“Phillips Smalley”, “date of birth”, “August 7, 1865”) ✓

Table 4.10: Examples of the outputs predicted by our model, which is trained on three tasks simultaneously.

5

Enhance and Analyze Reasoning Steps in the Sub-question Form

Several multi-hop reading comprehension datasets have been proposed to resolve the issue of reasoning shortcuts by which questions can be answered without performing multi-hop reasoning. However, the ability of multi-hop models to perform step-by-step reasoning when finding an answer to a comparison question remains unclear. It is also unclear how questions about the internal reasoning process are useful for training and evaluating question-answering (QA) systems. To evaluate the model precisely in a hierarchical manner, we first propose a dataset, *HieraDate*, with three probing tasks in addition to the main question: extraction, reasoning, and robustness. Our dataset is created by enhancing two previous multi-hop datasets, HotpotQA and 2WikiMultiHopQA, focusing on multi-hop questions on date information that involve both comparison and numerical reasoning. We then evaluate the ability of existing models to understand date information. Our experimental results reveal that the multi-hop models do not have the ability to subtract two dates even when they perform

well in date comparison and number subtraction tasks. Other results reveal that our probing questions can help to improve the performance of the models (e.g., by +10.3 F1) on the main QA task and our dataset can be used for data augmentation to improve the robustness of the models.

5.1 Introduction

Multi-hop reading comprehension (RC) requires a model to read and aggregate information from multiple paragraphs to answer a given question (Welbl et al., 2018). Several datasets have been proposed for this task, such as HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (2Wiki; Ho et al., 2020). Although the proposed models show promising performances, previous studies (Jiang and Bansal, 2019a; Chen and Durrett, 2019; Min et al., 2019a; Tang et al., 2021) have demonstrated that existing multi-hop datasets contain reasoning shortcuts, in which the model can answer the question without performing multi-hop reasoning.

There are two main types of questions in the previous multi-hop datasets: bridge and comparison. Tang et al. (2021) explored sub-questions in the QA process for model evaluation. However, they only used the bridge questions in HotpotQA and did not fine-tune the previous multi-hop models on their dataset when performing the evaluation. Therefore, it is still unclear about the ability of multi-hop models to perform step-by-step reasoning when finding an answer to a comparison question.

HotpotQA provides sentence-level supporting facts (SFs) to explain the answer. However, as discussed in Inoue et al. (2020) and Ho et al. (2020), the sentence-level SFs cannot fully evaluate the reasoning ability of the models; to solve this issue, in addition to sentence-level SFs, these studies provide a set of triples as the evidence information. For example, for the question in Figure 5.1, the evidence regards the dates of birth and death of two people, e.g., *(Maceo, date of death, July 4, 2001)*. We argue that simply requiring the models to detect a set of triples, in this case, cannot explain the answer to the question and cannot describe the full path from the question to the answer; additional operations, including calculations and comparisons, need to be performed to obtain the final answer.

Question: Who lived longer, Maceo Anderson or Jacek Karpiński?

Paragraph A:* *Maceo Anderson

[1] Maceo Anderson (September 3, 1910 – July 4, 2001 in Los Angeles, California) expressed an interest in dancing at … . [2] ....

Paragraph B:* *Jacek Karpiński

[3] Jacek Karpiński (9 April 1927 – 21 February 2010) was a Polish pioneer in computer … . [4] ....

Answer: Maceo Anderson

What is the date of birth of Maceo Anderson?
What is the date of death of Maceo Anderson?
What is the date of birth of Jacek Karpiński?
What is the date of death of Jacek Karpiński?
} Extraction Task

Reasoning Task:

How old was Maceo Anderson when they died?

How old was Jacek Karpiński when they died?

Full version: Is a 90-year-10-month-1-day-old person older than a 82-year-10-month-12-day-old person?

Year-only version: Is a 90-year-old person older than a 82-year-old person?

Robustness Task: Who lived shorter, Maceo Anderson or Jacek Karpiński?

Figure 5.1: Example of a question in our dataset.

To deal with this issue, we introduce a dataset, *HieraDate*,[1] consisting of the three probing tasks. (1) The extraction task poses sub-questions that are created by converting evidence triples into natural language questions. (2) The reasoning task is pertinent to the combination of triples, involving comparison and numerical reasoning that precisely evaluate the reasoning path of the main questions. (3) The robustness task consists of examples generated by slightly changing the semantics (e.g., *born first* to *born later*) of the original main questions. The purpose of the robustness task is to ensure that the models do not exploit superficial features in answering questions.

Our dataset is created by extending two existing multi-hop datasets, HotpotQA and 2Wiki. As the first step of the proof of concept, we start with the date information through comparison questions because this information is available and straightforward to handle. Moreover, based on the classification of comparison questions in Min et al. (2019a), all comparison questions on date information require multi-hop reasoning for answering. We then use our dataset to evaluate two leading models, HGN (Fang et al., 2020) and NumNet+ (Ran et al., 2019) on two settings: with and without fine-tuning

[1]Our data and code are available at https://github.com/Alab-NII/HieraDate.

on our dataset. We also conduct experiments to investigate whether our probing questions are useful for improving QA performance and whether our dataset can be used for data augmentation.

Our experimental results reveal that existing multi-hop models perform well in the extraction and robustness tasks but fail in the reasoning task when the models are not fine-tuned on our dataset. We observe that with fine-tuning, HGN can perform well in the comparison reasoning task; meanwhile, NumNet+ struggles with subtracting two dates, although it can subtract two numbers. Our analysis shows that questions that require both numerical and comparison reasoning are more difficult than questions that require only comparison reasoning. We also find that training with our probing questions boosts QA performance in our dataset, showing improvement from 77.1 to 82.7 F1 in HGN and from 84.6 to 94.9 F1 in NumNet+. Moreover, our dataset can be used as augmentation data for HotpotQA, 2Wiki, and DROP (Dua et al., 2019), which contributes to improving the robustness of the models trained on these datasets. Our results suggest that a more complete evaluation of the reasoning path may be necessary for better understanding of multi-hop models' behavior. We encourage future research to integrate our probing questions when training and evaluating the models.

5.2 Related Work

In addition to Tang et al. (2021), Al-Negheimish et al. (2021) and Geva et al. (2022) are similar to our study. Al-Negheimish et al. (2021) evaluated the previous models on the DROP dataset to test their numerical reasoning ability. However, they did not investigate the internal reasoning processes of those models. Geva et al. (2022) proposed a framework for creating new examples using the perturbation of the reasoning path. Our work is different in that their focus was on creating a framework, and it does not necessarily ensure the quality of all generated perturbation samples. Moreover, we investigate the QA process in-depth, while Geva et al. (2022) do not include all detailed questions (e.g., they do not include extraction task and comparison reasoning questions in Figure 5.1).

5.3 Dataset Construction

Our dataset is generated by using the two existing multi-hop datasets, HotpotQA and 2Wiki. HotpotQA (Yang et al., 2018), created through crowdsourcing, includes two main types of questions: bridge and comparison. Unlike previous datasets, a set of sentence-level SFs information is introduced, which facilitates explainable reasoning by the system. Because of the dataset construction procedure, there is no available information that can be used to generate sub-questions. 2Wiki (Ho et al., 2020) was created using Wikipedia articles and Wikidata triples. Similar to HotpotQA, it includes two main types of questions: bridge and comparison. In 2Wiki, the authors introduced evidence information that can be used to explain the reasoning chain from question to answer. We used this information for generating sub-questions in our dataset. We present our dataset construction process in the next following sections. In summary, it includes three main steps: obtaining date questions, generating sub-questions and sub-answers, and constructing HieraDate.

5.3.1 Obtain Date Questions

We first sampled the comparison questions in HotpotQA and 2Wiki. We then used a set of predefined keywords, such as *born first* and *lived longer*, to obtain questions regarding the date information. From the train and dev. split, respectively, we obtained 119 (after annotating, only use 114 samples) and 878 samples in HotpotQA, and 984 and 8,745 samples in 2Wiki.

5.3.2 Generate Sub-questions and Sub-answers

In 2Wiki, we used the evidence in the form of triples (e.g., (Maceo, date of death, July 4, 2001)) to automatically generate sub-questions and sub-answers for the extraction task. We used Wikidata IDs (available in 2Wiki) to obtain structured date information to compare and/or subtract two dates when generating questions for the reasoning task. To obtain natural language questions, we wrote ten and five templates for the extraction and reasoning tasks, respectively. Similar to Min et al. (2019b), to evaluate the robustness of the models, we created the adversarial questions by changing the main multi-hop questions such that the new answers are opposite (e.g., we changed the

Task	Templates/Phrases
Extract	What is the birth date of #name? What's the birth date of #name? What is the date of birth of #name? What's the date of birth of #name? When was #name born? What is the death date of #name? What's the death date of #name? What is the date of death of #name? What's the date of death of #name? When did #name die?
Reason	Does #date1 come before #date2? Does #date1 come after #date2? How old was #name when they died? Is a #age1 person younger than a #age2 person? Is a #age1 person older than a #age2 person?
Robust	Born first/earlier ⇔ Born later Born later ⇔ Born first Died first/earlier ⇔ Died later Died later/second/last ⇔ Died first Died more recently ⇔ Died first Lived longer ⇔ Lived shorter

Table 5.1: List of templates and phrases that we used in the dataset creation process. *Extract*, *Reason*, and *Robust* represent the three tasks: extraction, reasoning, and robustness, respectively.

question: "Who lived longer, A or B?" to "Who lived shorter, A or B?"). We observed that the ten phrases (e.g., *born first*) could cover all questions, and used these phrases to generate robustness questions. Table 5.1 presents a set of templates and phrases that we used in the dataset creation process.

In HotpotQA, unlike 2Wiki, no triples are available; therefore, we first prepared triples for the sampled questions, and then performed the same procedure as in 2Wiki to generate all probing questions. To obtain the triples, we first filtered the distractor paragraphs and retained only gold paragraphs. We then used Spacy[2] to extract the

[2]https://spacy.io/

entities in the questions. Further, we manually annotated the date with two formats: unstructured (e.g., 'May 1992') and structured (e.g., month=5). It is noted that we used only the dev. set in HotpotQA.

5.3.3 Construct HieraDate

We created our dev. and test sets from the dev. sets of HotpotQA and 2Wiki, and our training set from the 2Wiki training set. Table 5.2 lists the number of samples for each task and each split in our dataset. Our dataset includes two main types of questions: questions that ask about both date-of-birth and date-of-death information (e.g., "who lived longer"), and those that ask about only the date-of-birth or date-of-death information (e.g., "who was born later"). We call the first type *combined reasoning* because it requires both comparison and numerical reasoning (Figure 5.1). The second type is called *comparison reasoning* (Figure 5.2) because it requires only comparison reasoning. *One combined reasoning* sample has one main multi-hop question, four extraction questions, two numerical reasoning questions, one comparison question, and one robustness question. Meanwhile, *one comparison reasoning* sample has one main multi-hop question, two extraction questions, two comparison questions, and one robustness question.

Split	Main	Extract	Reason	Robust
Train	8745	21340	19415	8745
Dev.	549	1346	1222	549
Test	549	1346	1222	549

Table 5.2: Our dataset statistics. Each main question has the extraction, reasoning, and robustness tasks.

Date Format Wikidata uses a zero value for the dates that miss the month value or day value. In reality, we have no date with month-0 and day-0; therefore, we use a default value "1" for the dates that miss the month value or day value.

Numerical Reasoning Issue In reality, in some cases, the paragraph can contain age information, e.g., "*He died in 1981 at the age of 90*". In this case, the model does not

Question: Who was born first, George Washington or Lawrence Washington?

Paragraph A: George Washington

[1] George Washington (February 22, 1732 – December 14, 1799) was an American political leader, … who served as the first president … [2] ....

Paragraph B: Lawrence Washington

[3] Lawrence Washington (1718–1752) was an American soldier, planter, politician, and prominent landowner in … . [4] ....

Answer: Lawrence Washington

What is the birth date of George Washington?
When was Lawrence Washington born?
} Extraction Task

Reasoning Task:

Full version:	Year-only version:
Does February 22, 1732 come before 1718?	Does 1732 come before 1718?
Does February 22, 1732 come after 1718?	Does 1732 come after 1718?

Robustness Task:

Who was born later, George Washington or Lawrence Washington?

Figure 5.2: Example of a comparison reasoning question in our dataset. There are two main types of questions in our dataset: *combined reasoning* and *comparison reasoning*. Combined reasoning requires comparison and numerical reasoning; meanwhile, comparison reasoning requires only comparison reasoning. This is an example of a comparison reasoning question.

need to perform numerical reasoning. We used rules (e.g., filter whether the context contains the word "age" or not), then manually checked, and found that there are 13 paragraphs in a total of 248 paragraphs (124 examples) in the test set that the age information is available.

Dataset Versions Our dataset has two versions: "normal setting" and "distractor setting". The "normal setting" includes only two gold paragraphs; meanwhile, the "distractor setting" contains ten paragraphs, including two gold paragraphs and eight distractor paragraphs. In this study, we evaluated the previous models on the "normal setting".

5.4 Experiments

To comprehensively evaluate the top-performing multi-hop models, we conducted various experiments, including both with and without fine-tuning on our dataset. In addition, to discover the effectiveness of our dataset, we examine the usefulness of our probing tasks and investigate whether our dataset can be used for data augmentation.

5.4.1 Models

As existing models cannot perform all three tasks, we evaluate these models under two groups: one focused on comparison reasoning (e.g., HGN) and the other focused on numerical reasoning (e.g., NumNet+). HGN (Fang et al., 2020) was designed to deal with HotpotQA, whereas NumNet+ (Ran et al., 2019) was designed to deal with DROP (Dua et al., 2019). Both models can perform on the extraction and robustness tasks. By design, HGN can answer yes/no questions in the comparison reasoning task. Meanwhile, NumNet+ cannot answer yes/no questions. However, it can deal with numerical questions. There are some versions of the NumNet model; in our experiment, we use the NumNet+ version.[3] There are two ways to convert the questions of the extraction task in our dataset to the format of the DROP dataset. One is to use the span format, and the other is to use the date format. In our experiment, we use the span format because it produces better results than the date format.

5.4.2 Main Results

To study the abilities of the models in detail, we evaluate both models on two versions of our dataset: the full-date version (with year-month-date) and the year-only version. We also evaluate the models on two settings: with and without fine-tuning on our dataset. We use all main and probing questions together for fine-tuning the models. It is noted that we only use HieraDate when fine-tuning.

Date Understanding Evaluation Table 5.3 presents all the results of the existing models on our dataset. Regarding the full-date version of the dataset, when the models are not fine-tuned on our dataset, both HGN and NumNet+ fail in the reasoning task.

[3] https://github.com/llamazing/numnet_plus

This can be because the forms of reasoning questions are new to these models or the models do not possess the reasoning abilities as humans do. For the extraction task, HGN performs quite well; meanwhile, NumNet+ performs worse. In the robustness task, the results are comparable with those of the main multi-hop questions. This can be explained by the fact that the patterns of the main multi-hop and robustness questions are similar.

Regarding the year-only version of the dataset, when the models are not fine-tuned on our dataset, the score of the HGN model in the comparison reasoning task does not change much when compared with the full-date version (55.0 vs. 53.1 EM); this indicates that there is not much difference between the full-date and year-only versions when using HGN. For NumNet+, the score of the numerical reasoning task significantly improves when compared with the full-date version (83.1[4] vs. 22.8 F1); this indicates that NumNet+ can perform numerical reasoning in the form of numbers (as years) but cannot in the form of dates.

Fine-tuning	Model	Main		Extraction		Reasoning		Robustness	
		EM	F1	EM	F1	EM (num)	EM (comp)	EM	F1
×	HGN	66.85	76.15	94.58	96.14	*N/A*	53.08	71.95	81.64
	HGN (year-only)	-	-	-	-	*N/A*	55.03	-	-
	NumNet+	67.94	71.57	1.26	47.93	22.79 (F1)	*N/A*	69.58	71.91
	NumNet+ (year-only)	-	-	-	-	83.07	*N/A*	-	-
✓	HGN	78.87	82.69	96.06	97.14	*N/A*	100	76.68	78.58
	HGN (year-only)	77.23	79.24	95.84	96.93	*N/A*	99.90	76.68	78.61
	NumNet+	95.08	95.20	96.36	97.73	35.96 (F1)	*N/A*	94.90	94.93
	NumNet+ (year-only)	94.90	94.93	96.29	97.69	94.36	*N/A*	93.99	94.01
×	SAE	69.76	77.78	82.99	84.73	*N/A*	59.14	69.22	77.82
	SAE (year-only)	-	-	-	-	*N/A*	55.75	-	-
	Human (avg.)	94.00	94.90	99.16	99.53	100	98.06	95.5	95.9
	Human UB	100	100	100	100	100	100	100	100

Table 5.3: Results (%) of the previous models on the test set of our dataset. *Num* denotes numerical reasoning and *comp* denotes comparison reasoning. It is noted that combined reasoning questions require both numerical and comparison reasoning. *N/A* denotes not applicable. "-" indicates that the score is similar to the score of the full-date version in the same setting. *Human UB* represents the human upper bound.

[4]In the year-only version, the EM and F1-score are equal.

When the models are fine-tuned on our dataset, we find that all scores of HGN improve; especially, HGN reaches the highest score in the comparison reasoning task. We conjecture that the low scores when HGN is not fine-tuned on our dataset are because the forms of the comparison reasoning questions are new to this model. Similar to HGN, the scores of the NumNet+ model also improve when it is fine-tuned on our dataset. However, the score in the numerical reasoning task on the full-date version remains low. We observed that when we evaluate NumNet+ on the year-only version, the EM scores are 83.1 and 94.4 in the numerical reasoning task for two cases: without and with fine-tuning on our dataset, respectively. This indicates that NumNet+ could perform subtraction in the form of numbers (as years) but could not in the form of dates.

Evaluation on SAE We also evaluate SAE (Tu et al., 2019) on our dataset. Similar to HGN, SAE was designed to deal with HotpotQA. Similar to HGN, the model cannot perform well on the comparison reasoning questions when it is not fine-tuned on our dataset. As all questions in the comparison reasoning task are yes/no questions, the random score is 50%. The scores of both HGN and SAE are close to the chance score.

Dataset Quality Check To verify the quality of our dataset, we randomly selected 100 samples from the test set and instructed graduate students to conduct the annotation. Each sample was annotated by two annotators. We provided the context and a list of questions to the annotators; the results are reported in Table 5.3. It can be observed that the human upper bound is 100% for all tasks. However, the human average is slightly low. On manually investigating the reason for this low human average, we found that the annotators made careless mistakes in several examples; however, we confirmed that these examples are answerable and reasonable.

5.4.3 Analyses

Difficulty of Reasoning over Dates To discover whether the number of required reasoning skills in each question affects question difficulty, we compared the results of the two main types of questions in our dataset (combined vs. comparison reasoning). Table 5.4 shows the results of the previous models on the test set of our dataset for

different types of questions. As shown in the table, the scores of comparison reasoning questions were always higher than those of combined reasoning questions (85.7 vs. 72.3 F1 in HGN; 98.8 vs. 81.6 F1 in NumNet+). In the current version of the dataset, there are only 22.1% combined reasoning questions. To avoid the data-size bias, we created a HieraDate-small version by randomly choosing the comparison reasoning questions such that the number of combined reasoning questions is equal to the number of comparison reasoning questions. We then conducted experiments on HieraDate-small, and the results are the highlighted rows in Table 5.4. We found similar results as on HieraDate. These results indicate that questions requiring both numerical and comparison reasoning are more difficult than questions that require only comparison reasoning.

Model-type	**Main**		**Extraction**		**Reasoning**		**Robustness**	
	EM	F1	EM	F1	EM (num)	EM (comp)	EM	F1
HGN-all	78.87	82.69	96.06	97.14	*N/A*	100	76.68	78.58
HGN-combined	70.97	72.34	93.95	95.67	*N/A*	100	69.35	71.18
HGN-comparison	81.18	85.71	97.29	98.00	*NO*	100	78.82	80.74
HGN-all	75.40	76.67	95.30	96.71	*N/A*	99.19	76.21	77.26
HGN-combined	66.13	67.50	93.75	95.60	*N/A*	99.19	71.77	72.82
HGN-comparison	84.68	85.85	98.39	98.92	*NO*	99.19	80.65	81.69
NumNet-all	94.90	94.93	96.29	97.69	94.36	*N/A*	93.99	94.01
NumNet-combined	81.45	81.58	94.76	96.81	94.36	*N/A*	79.84	79.95
NumNet-comparison	98.82	98.82	97.18	98.20	*NO*	*N/A*	98.12	98.12
NumNet-all	85.08	85.43	95.97	97.69	94.00	*N/A*	85.48	85.50
NumNet-combined	72.58	73.27	94.76	96.84	94.00	*N/A*	73.39	73.42
NumNet-comparison	97.58	97.58	98.39	99.40	*NO*	*N/A*	97.58	97.58

Table 5.4: Results (%) of the previous models on the test set of our dataset for different types of questions. *Model-type* denotes the model name and the type of question that the model is evaluated on (e.g., HGN-combined: the results of HGN on combined reasoning questions). *Num* denotes numerical reasoning and *comp* denotes comparison reasoning. *N/A* denotes not applicable; meanwhile, *NO* indicates that there are no questions for evaluation. For HGN, we fine-tuned it on the full-date version of our dataset; meanwhile, NumNet+ is fine-tuned on the year-only version of our dataset. In the row with highlight color, the model is trained on HieraDate-small where the number of combined reasoning and comparison reasoning questions are equal.

Are our Probing-questions Useful for Improving the QA Performance? To investigate the effectiveness of our probing questions for improving the QA performance, we trained HGN and NumNet+ on six different combinations of the main and probing tasks. Table 5.5 presents the results of the HGN and NumNet+ models on the test set of our dataset when they are trained on different subsets of our dataset. The results show that each task in our dataset helps to improve the performance of the main QA question. Especially when training the models on all tasks, the results improve significantly in both HGN and NumNet+ compared with the models trained on the main questions only (82.7 vs. 77.1 F1 in HGN; 94.9 vs. 84.6 F1 in NumNet+). This demonstrates that our probing questions not only help to explain the internal reasoning process but also help to improve the score of the main multi-hop questions.

Model	**Training Data**	**#Questions**	**Testing Data**			
			Main	Extract	Comp/Num	Robust
HGN	Main	8,745	77.11	0.0	0.0	75.45
	Main & Extract	30,085	78.37	**97.14**	0.0	78.18
	Main & Reason	24,310	79.06	0.0	99.79	76.62
	Main & Robust	17,490	80.96	0.0	0.0	78.04
	Main & Extract & Reason	45,650	79.97	97.10	99.59	78.40
	All	54,395	**82.69**	**97.14**	**100**	**78.58**
NumNet+	Main	8,745	84.57	0.02	0.0	82.87
	Main & Extract	30,085	92.03	97.75	0.0	89.28
	Main & Reason	12,595	88.92	0.19	94.36	89.83
	Main & Robust #1	17,490	49.86	0.23	0.0	44.84
	Main & Robust #2	17,490	48.54	0.08	0.0	50.42
	Main & Robust #3	17,490	52.95	0.02	0.0	45.24
	Main & Extract & Reason	33,935	92.01	**97.89**	**95.16**	88.91
	All	42,680	**94.93**	97.69	94.36	**94.01**

Table 5.5: F1-score of the HGN and NumNet+ models on the test set of our dataset when they are trained on different subsets of our dataset. *#Questions* represents the number of questions in the training data. *Comp/Num* denotes comparison reasoning or numerical reasoning; for the HGN model, it is comparison reasoning; for the NumNet+ model, it is numerical reasoning. We run three times for the "Main & Robust" setting in the NumNet+ model because the results of this setting are quite different with others.

Data Augmentation We also check whether our dataset can be used for data augmentation. We trained HGN and NumNet+ on two settings, on the original dataset (e.g., HotpotQA) and on the union of the original dataset and our dataset. We use HGN for HotpotQA and 2Wiki; meanwhile, NumNet+ is used for DROP. All results are reported in Table 5.6. There is no significant change on the original datasets (e.g., from 81.1 to 81.4 F1 for HotpotQA); meanwhile, the improvement in our dataset is significant (e.g., from 76.3 to 84.9 F1 on the main QA task). Notably, all models that are trained on the union of the original dataset and our dataset are better in our robustness task. This indicates that our dataset can be used as augmentation data for improving the robustness of the models trained on HotpotQA, 2Wiki, and DROP.

Model	Training Data	#Questions	Evaluation Data					
			Original		Main	Extract	Reason	Robust
			EM	F1	F1	F1	F1	F1
HGN	Hotpot	90,447	67.56	81.13	76.25	94.64	26.03	79.74
	Hotpot & Ours	144,842	**67.99**	**81.44**	**84.93**	**97.09**	**99.95**	**81.18**
	2Wiki	167,454	69.42	74.21	76.69	64.62	0.0	77.35
	2Wiki & Ours	221,849	**69.66**	**75.26**	**85.27**	**97.03**	**99.74**	**82.23**
NumNet+	DROP	77,409	**78.58**	**82.14**	69.06	48.10	79.24	71.37
	DROP & Ours	120,089	78.45	82.06	**95.39**	**97.80**	**94.76**	**94.54**

Table 5.6: The results of the HGN and NumNet+ models on HotpotQA, 2Wiki, DROP, and our dataset. For the *Original* column, the evaluation data is HotpotQA, 2Wiki, and DROP when the model used HotpotQA, 2Wiki, and DROP for training, respectively. All reported scores in this table are average scores from two runs.

5.5 Conclusion

We proposed a new multi-hop RC dataset for comprehensively evaluating the ability of existing models to understand date information. We evaluated the top-performing models on our dataset. The results revealed that the models may not possess the ability to subtract two dates even when fine-tuned on our dataset. We also found that our probing questions could help to improve QA performance, and can be used for data augmentation.

6

Gaps between LLMs and Human Reasoning

With well-designed prompts (e.g., chain-of-thought), previous studies have shown that large language models (LLMs) can perform various reasoning tasks. However, their performance often drops when dealing with more complex reasoning tasks. Currently, most studies have not fully investigated why LLMs fail on complex reasoning tasks. We design experiments to explore how LLMs behave when tackling complex reasoning tasks. Drawing from previous research and the ways often employed by humans to address complex questions, we examine the models' behaviors across three stages in the question-answering (QA) process: question decomposition, subproblem solving, and composition. To comprehensively investigate LLMs, we use three different scenarios for the input context: without context, with unstructured context, and with structured context. Our experiments on two multi-hop datasets and three different sizes of LLMs show that: 1) LLMs fail to decompose complex questions into sub-questions. Further analysis reveals no correlation between the question decomposition stage and

QA performance. Additionally, the results indicate that models resort to reasoning shortcuts when addressing complex questions, rather than employing a step-by-step reasoning process. 2) In the subproblem solving stage, LLMs can successfully answer sub-questions when provided with either unstructured or structured context but fail when no context is provided. This indicates that LLMs, including GPT-3.5, still lack the ability to memorize factual knowledge during pre-training. 3) Both Llama 2 13B and 70B struggle in performing at the composition stage on comparison questions.

6.1 Introduction

With the release of language models (LMs) (Devlin et al., 2019; Yang et al., 2019; Raffel et al., 2020, *inter alia*), especially large language models (LLMs) (Brown et al., 2020; Zhao et al., 2023, *inter alia*), there has been a significant change in the research community. Using well-designed prompts (e.g., chain-of-thought (CoT; Wei et al., 2022)), previous studies (Dua et al., 2022; Kojima et al., 2022) have demonstrated the reasoning abilities of LLMs in performing various tasks, such as arithmetic reasoning (Cobbe et al., 2021), multi-hop (compositional) reasoning (Welbl et al., 2018), and commonsense reasoning (Talmor et al., 2019). However, the performance of LLMs often drops when dealing with more complex reasoning tasks (Khot et al., 2023; Press et al., 2023; Zhou et al., 2023).

To obtain better performance on complex reasoning tasks, most previous studies have focused on improving the prompts, such as introducing new types of prompts that involve question decomposition (Dua et al., 2022; Khot et al., 2023; Press et al., 2023; Zhou et al., 2023) or enhancing the CoT prompting (Wang et al., 2023). For example, Zhou et al. (2023) propose a least-to-most prompt with two different stages (decomposition and subproblem solving) to enable LLMs solving more complex tasks. Although these studies show the performance improvement, it still remains unclear why LLMs fail in complex reasoning questions on multi-hop datasets. Futhermore, the two stages in the previous prompts (e.g., least-to-most) do not apply to all types of questions. For example, in comparison questions, besides decomposing and solving the subproblems, we also need a composition stage to combine or compare multiple sub-answers and obtain the final answer (see Figure 6.1).

Current LLMs still function as black boxes to humans. Therefore, conducting

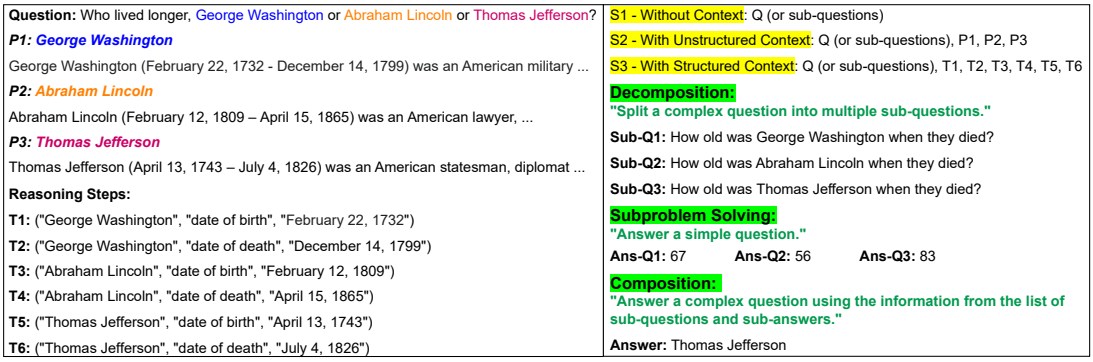

Figure 6.1: An example in our dataset. We conduct experiments with three scenarios: (S1) without context, (S2) with unstructured context, and (S3) with structured context. P, S, T represent paragraph, scenario, and triple, respectively. The information on the left side represents the sample input, while the right side shows our designed experiments.

a careful investigation as to why LLMs fail and at which stage they fail offers numerous benefits when working with these models. Our investigation is based on two assumptions: (1) humans can perform generalization tasks, and (2) humans often undergo three main stages in answering complex questions: 1) decomposition, 2) subproblem solving, and 3) composition. The second assumption is derived from previous systems (Perez et al., 2020; Khot et al., 2023) and from our observation of how humans answer complex questions. To assess the LLM performance in detail, we analyze the behaviors of LLMs in the question-answering process across these three stages. Additionally, to comprehensively investigate LLMs, we design three different scenarios for the input text: (S1) without context, (S2) with unstructured context (natural language text), and (S3) with structured context (the information in a structured format). Figure 6.1 shows the three different scenarios for the input context (yellow part) in our experiments.

By conducting experiments on three models (Llama 2 13B and 70B, and GPT-3.5-turbo-1106) and two multi-hop datasets, we find that 1) compared to the human QA process, LLMs fail to decompose complex questions into sub-questions. Upon further analysis, we find no correlation between the decomposition stage and the final QA performance when the model is asked to directly answer the complex question. Instead, the model performs reasoning shortcuts to answer the complex

questions in an end-to-end manner. 2) In the subproblem solving stage, our results reveal that models can effectively answer sub-questions when presented with either unstructured or structured context. However, they encounter difficulties in answering sub-questions when no context is provided. This underscores the ongoing limitation of LLMs, including GPT-3.5, in retaining factual knowledge during pre-training. 3) In the composition stage, we find that both Llama 2 13B and 70B struggle to perform comparisons involving multiple dates and numbers when the number of hops increases. In contrast, GPT-3.5 does not face as much difficulty in achieving high scores at this stage.

6.2 Related Work

The idea of decomposing a complex question into sub-questions in the multi-hop MRC task was first proposed by Talmor and Berant (2018). The authors introduced a framework that answers a complex question by decomposing it into a sequence of simple questions, which are then answered using a search engine. The final answer is obtained by applying operations such as union and intersection. In general, the three steps in their framework are very similar to ours. The difference lies in the purpose: they aim to propose a framework, while we aim to use these three steps to investigate the abilities of LLMs. There are some upcoming works (Min et al., 2019b; Perez et al., 2020; Fu et al., 2021a) that attempt to propose different ways to decompose complex questions, such as using unsupervised methods (Perez et al., 2020) or recasting question decomposition as a span prediction task (Min et al., 2019b). Recently, to improve task performance, some works (Dua et al., 2022; Khot et al., 2023; Press et al., 2023; Zhou et al., 2023) have incorporated question decomposition into their prompts.

6.3 Datasets

We focus on the multi-hop reasoning task (Welbl et al., 2018; Yang et al., 2018), which requires models to gather information from multiple paragraphs, and then perform compositional reasoning to obtain the final answer. For our designed experiments, datasets should meet the following three requirements: 1) include both unstructured

and structured context, 2) include sub-questions and sub-answers, and 3) contain various hops for complex questions. Due to these requirements, we mainly experiment with two multi-hop datasets: Date-complex and 2Wiki-complex. These datasets for solving complex tasks were derived by expanding upon existing datasets: HieraDate (Ho et al., 2022) and 2WikiMultiHopQA (2Wiki; Ho et al., 2020). We present the number of samples per hop and the total number in these two datasets in Table 6.1.

	Date-complex	**2Wiki-complex**
2-hop	300	191
3-hop	300	413
4-hop	300	270
5-hop	300	116
6-hop	300	-
7-hop	300	-
Total	1,800	990

Table 6.1: The number of samples per hop and the total number of samples for two datasets, Date-complex and 2Wiki-complex.

6.3.1 Date-complex

Date-complex is an extension of the HieraDate dataset (Ho et al., 2022). There are three main steps in the dataset construction process: 1) obtain a list of people with specific attribute information (e.g., date of birth), 2) automatically generate samples, and 3) manually verify random samples. Specifically, from the development set in HieraDate, we obtain a list of people with dates of birth and/or death, along with a descriptive paragraph. We then define six main types of question templates: *born first*, *born later*, *died first*, *died later*, *lived longer*, and *lived shorter*. We use these templates to automatically generate the samples. In our dataset, we generate 300 samples per hop ranging from 2-hop to 7-hop. To ensure the quality of the dataset, we randomly choose 6 samples for each hop and manually verify that they are reasonable and answerable.

6.3.2 2Wiki-complex

2Wiki-complex is an extension of the 2Wiki dataset (Ho et al., 2020). There are four main steps in the dataset construction process: 1) obtain basic 2-hop samples, 2) manually verify 2-hop samples and related Wikidata triples, 3) prepare templates, and 4) automatically generate the samples.

Specifically, from the development set of 2Wiki, we only retain the questions related to father or mother relations (1,025 samples). The primary reason for this is that the combination of the two parent relations can form a new relational word, such as 'mother' and 'mother' combining to create 'maternal grandmother'. Since 2Wiki is constructed automatically, it contains unanswerable questions. Therefore, we randomly selected 200 samples and manually verified them to ensure that all samples we use for generating more hops are answerable, resulting in 191 2-hop samples.

Each 2-hop sample corresponds to two triples, (e_1, r_1, e_2) and (e_2, r_2, e_3), and two paragraphs (p_1 and p_2) that describe the two entities, e_1 and e_2. To obtain a new 3-hop sample from a 2-hop sample, we start with the entity e_3 and obtain a list of new triples, such as (e_3, r_{31}, e_4) and (e_3, r_{32}, e_4). We manually verify this list of new triples. When a triple (e.g., (e_3, r_{31}, e_4)) is confirmed to be correct, we use the three triples (e_1, r_1, e_2), (e_2, r_2, e_3), and (e_3, r_{31}, e_4) to generate a 3-hop sample. To ensure the automation of the generation process, we prepare a list of templates for questions when extending from 2-hop to 3-hop in Table 6.2. For example, if the 2-hop question is 'Who is the maternal grandmother of person A?' and the new triple is (person A, date of birth, May 13, 1899), then the corresponding 3-hop question is 'What is the date of birth of the maternal grandmother of person A?'. We obtain 413 3-hop samples. Using the same process, we obtain 270 4-hop samples and 116 5-hop samples.

Relation_1	Relation_2	Relation_3	Question_template
father	father	father	Who is the paternal grandfather' s father of #name?
father	father	mother	Who is the paternal grandfather' s mother of #name?
father	father	spouse	Who is the paternal grandfather's wife of #name?
father	father	date of birth	What is the date of birth of paternal grandfather of #name?
father	father	date of death	What is the date of death of paternal grandfather of #name?
father	father	place of birth	What is the place of birth of paternal grandfather of #name?
father	father	place of death	What is the place of death of paternal grandfather of #name?
father	father	country of citizenship	What is the nationality of paternal grandfather of #name?
father	father	cause of death	What is the cause of death of paternal grandfather of #name?
father	mother	father	Who is the paternal grandmother' s father of #name?
father	mother	mother	Who is the paternal grandmother' s mother of #name?
father	mother	spouse	Who is the paternal grandmother's husband of #name?
father	mother	date of birth	What is the date of birth of paternal grandmother of #name?
father	mother	date of death	What is the date of death of paternal grandmother of #name?
father	mother	place of birth	What is the place of birth of paternal grandmother of #name?
father	mother	place of death	What is the place of death of paternal grandmother of #name?
father	mother	country of citizenship	What is the nationality of paternal grandmother of #name?
father	mother	cause of death	What is the cause of death of paternal grandmother of #name?
mother	mother	father	Who is the maternal grandmother' s father of #name?
mother	mother	mother	Who is the maternal grandmother' s mother of #name?
mother	mother	spouse	Who is the maternal grandmother's husband of #name?
mother	mother	date of birth	What is the date of birth of maternal grandmother of #name?
mother	mother	date of death	What is the date of death of maternal grandmother of #name?
mother	mother	place of birth	What is the place of birth of maternal grandmother of #name?
mother	mother	place of death	What is the place of death of maternal grandmother of #name?
mother	mother	country of citizenship	What is the nationality of maternal grandmother of #name?
mother	mother	cause of death	What is the cause of death of maternal grandmother of #name?
mother	father	father	Who is the maternal grandfather' s father of #name?
mother	father	mother	Who is the maternal grandfather' s mother of #name?
mother	father	spouse	Who is the maternal grandfather's wife of #name?
mother	father	date of birth	What is the date of birth of maternal grandfather of #name?
mother	father	date of death	What is the date of death of maternal grandfather of #name?
mother	father	place of birth	What is the place of birth of maternal grandfather of #name?
mother	father	place of death	What is the place of death of maternal grandfather of #name?
mother	father	country of citizenship	What is the nationality of maternal grandfather of #name?
mother	father	cause of death	What is the cause of death of maternal grandfather of #name?

Table 6.2: List of question templates that are used to extend a question from 2-hop to 3-hop.

6.4 Experiments

6.4.1 Experimental Settings

We conduct our experiments using three different sizes of models, Llama 2 (13B and 70B) (Touvron et al., 2023) and GPT-3.5-turbo-1106 (referred to as GPT-3.5), and three types of prompts, zero-shot, standard few-shot and CoT.. Table 6.3, Table 6.4, Table 6.5,

and Table 6.6 present all prompts used for running GPT-3.5, covering the full QA process, the decomposition stage, the subproblem-solving stage, and the composition stage, respectively.

Zero-shot Prompting We ask the models to perform the task without providing any exemplars in the prompt.

Standard Few-shot Prompting In addition to asking the models to do the task, we also provide one or a few examples in the prompt.

CoT Prompting We also use CoT prompting for the full QA process to check the ability of the model on solving complex tasks.

Setting	Example
	2Wiki-complex
Zero-shot	Answer a complex question below with the provided context, response only the answer in the 'Answer: ' part and the explanation in the 'Explanation: ' part. Context: str_context\n\n —\n\nQuestion: str_question \nAnswer: \nExplanation:
Few-shot (unstructured context)	Answer a complex question below with the provided context, response only the answer in the 'Answer: ' part, don' t repeat any words in the question. Here is an example: Context: Prince Dmitri Alexandrovich of Russia. Prince Dmitri Alexandrovich of Russia (15 August 1901 – 7 July 1980) was the fourth son and fifth child of Grand Duke Alexander Mikhailovich of Russia and Grand Duchess Xenia Alexandrovna of Russia. He was a nephew of Tsar Nicholas II of Russia. Grand Duchess Xenia Alexandrovna of Russia. Grand Duchess Xenia Alexandrovna of Russia (– 20 April 1960) was the elder daughter and fourth child . . . \n\n—\n\n Question: Who is the maternal grandmother of Prince Dmitri Alexandrovich Of Russia? \n Answer: Maria Feodorovna \n Context: str_context\n\n —\n\nQuestion: str_question \nAnswer:
CoT (unstructured context)	Answer a complex question below with the provided context, response only the answer in the 'Answer: ' part, don' t repeat any words in the question. Here is an example: Context: Prince Dmitri Alexandrovich of Russia. Prince Dmitri Alexandrovich of Russia (15 August 1901 – 7 July 1980) was the fourth son and fifth child of Grand Duke Alexander Mikhailovich of Russia and Grand Duchess Xenia Alexandrovna of Russia. He was a nephew of Tsar Nicholas II of Russia. Grand Duchess Xenia Alexandrovna of Russia. Grand Duchess Xenia Alexandrovna of Russia (– 20 April 1960) was the elder daughter and fourth child . . . \n\n—\n\n Question: Who is the maternal grandmother of Prince Dmitri Alexandrovich Of Russia? \n Guideline: The first step is to find the mother of Prince Dmitri Alexandrovich Of Russia. Based on the context, we discover that Grand Duchess Xenia Alexandrovna of Russia is the mother of Prince Dmitri Alexandrovich Of Russia. Next, we continue to find the mother of Grand Duchess Xenia Alexandrovna of Russia. Based on the context, we learn that Maria Feodorovna is the mother of Grand Duchess Xenia Alexandrovna of Russia. Therefore, we can infer that the answer is Maria Feodorovna. Answer: Maria Feodorovna \n Context: str_context\n\n —\n\nQuestion: str_question \nAnswer:

Table 6.3: Examples of the prompts that we use for running GPT-3.5 in the full QA process. str_question is the input question for the required task and str_context is the provided context..

Setting	Example
	2Wiki-complex
Zero-shot	Split a complex question into multiple sub-questions. Fill in the sub-questions under the 'Sub-question-' section, starting with Sub-question-1, then Sub-question-2, and so on. Complex question: str_question
Few-shot	Split a complex question ... (*similar to Zero-shot*) Here are two complex questions, each with its sub-questions. Complex question: Who is the maternal grandmother of George Washington? Sub-question-1: Who is the mother of George Washington? Sub-question-2: Who is the mother of the mother of George Washington? Complex question: What is the date of birth of paternal grandfather of Abraham Lincoln? Sub-question-1: Who is the father of Abraham Lincoln? Sub-question-2: Who is the father of the father of Abraham Lincoln? Sub-question-3: What is the date of birth of the father of the father of Abraham Lincoln? Complex question: str_question
Few-shot Dependent	Complex question: Who is the maternal grandmother of George Washington? Sub-question-1: Who is the mother of George Washington? Sub-question-2: Who is the mother of Sub-Answer-1? Complex question: What is the date of birth of paternal grandfather of Abraham Lincoln? Sub-question-1: Who is the father of Abraham Lincoln? Sub-question-2: Who is the father of Sub-Answer-1? Sub-question-3: What is the date of birth of Sub-Answer-2?
	Date-complex
Few-shot	Split a complex question ... (*similar to Zero-shot*) Here are three complex questions, each with its sub-questions. Complex question: Who was born first, Subhash Bapurao Wankhede or Frederick Howard Bryant? Sub-question-1: When was Subhash Bapurao Wankhede born? Sub-question-2: When was Frederick Howard Bryant born? Complex question: Who died later, Martial Singher or David Feintuch? Sub-question-1: When did Martial Singher die? Sub-question-2: When did David Feintuch die? Complex question: Who lived longer, Olaf Andreas Gulbransson or Pepe Hern? Sub-question-1: When was Olaf Andreas Gulbransson born? Sub-question-2: When did Olaf Andreas Gulbransson die? Sub-question-3: How old was Olaf Andreas Gulbransson when they died? Sub-question-4: When was Pepe Hern born? Sub-question-5: When did Pepe Hern die? Sub-question-6: How old was Pepe Hern when they died? Complex question: str_question

Table 6.4: Examples of the prompts that we use for running GPT-3.5 in the **decomposition stage**.

Setting	Example
	2Wiki-complex
Zero-shot	Answer a simple question below using the information in the provided context. Response only the answer in the 'Answer: ' part and the explanation in the 'Explanation: ' part. Context: str_context\n\n —\n\nQuestion: str_question \nAnswer: \nExplanation:
Few-shot (unstructured context)	Answer a simple question below ... (*similar to Zero-shot*) Here is an example: Context: George Washington was born on February 22, 1732, at Popes Creek in Westmoreland County, Virginia. He was the first of six children of Augustine and Mary Ball Washington. His father was a justice of the peace and a prominent public figure who had four additional children from his first marriage to Jane Butler. \n\n—\n\n Question: Who is the mother of George Washington? \n Answer: Mary Ball Washington \n Explanation: Based on the provided context, we can infer that the answer is Mary Ball Washington. Context: str_context\n\n—\n\nQuestion: str_question \nAnswer: \nExplanation:
Few-shot (structured context)	Answer a simple question below ... (*similar to Zero-shot*) Here is an example: Context: Prince Dmitri Alexandrovich of Russia, mother, Grand Duchess Xenia Alexandrovna of Russia \n Grand Duchess Xenia Alexandrovna of Russia, mother, Maria Feodorovna \n\n—\n\n Question: Who is the mother of George Washington? \n Answer: Mary Ball Washington \n Explanation: Based on the provided context, we can infer that the answer is Mary Ball Washington. Context: str_context\n\n—\n\nQuestion: str_question \nAnswer: \nExplanation:
	Date-complex
Few-shot (structured context)	Answer a simple question below ... (*similar to Zero-shot*) Here is an example: Context: George Washington, date of birth, February 22, 1732 \n George Washington, date of death, December 14, 1799 \n George Washington, age, 67 \n\n—\n\n Question: How old was George Washington when they died? \n Answer: 67 \n Explanation: Based on the context, we can infer that the answer is 67 because George Washington died on December 14, 1799, and was born on February 22, 1732. Context: str_context\n\n—\n\nQuestion: str_question \nAnswer: \nExplanation:

Table 6.5: Examples of the prompts that we use for running GPT-3.5 in the **subproblem solving stage**. str_question is the input question for the required task and str_context is the provided context. Both datasets use the same prompt for the zero-shot setting. The example for few-shot with unstructured context in Date-complex has a similar format as in 2Wiki-complex.

6.4 Experiments

Setting	Example
	2Wiki-complex
Zero-shot	Answer a complex question using the information from the list of sub-questions and sub-answers. Response only the answer in the 'Answer: ' part and the explanation in the 'Explanation: ' part. List of sub-questions and sub-answers: subqa_list \n\n —\n\n Complex question: str_question \nAnswer: \nExplanation:
Few-shot	Answer a complex question using the information from the list of sub-questions and sub-answers. Response only the answer in the 'Answer: ' part and the explanation in the 'Explanation: ' part. Here is an example: List of sub-questions and sub-answers: Sub-question-1: Who is the father of Henry Iii, Prince Of Anhalt-Aschersleben? Sub-answer-1: Henry II, Prince of Anhalt-Aschersleben Sub-question-2: Who is the father of Henry Ii, Prince Of Anhalt-Aschersleben? Sub-answer-2: Henry I, Count of Anhalt \n\n —\n\n Complex question: Who is Henry Iii, Prince Of Anhalt-Aschersleben's paternal grandfather? \nAnswer: Henry I, Count of Anhalt \nExplanation: Based on the information, we can infer that the answer is Henry I, Count of Anhalt. List of sub-questions and sub-answers: subqa_list \n\n —\n\n Complex question: str_question \nAnswer: \nExplanation:
	Date-complex
Few-shot	Answer a complex question using the information from the list of sub-questions and sub-answers. Response only the answer in the 'Answer: ' part and the explanation in the 'Explanation: ' part. Here is an example: List of sub-questions and sub-answers: Sub-question-1: What is the birth date of Joy? Sub-answer-1: 11 September 1950 Sub-question-2: What is the birth date of Emma? Sub-answer-2: 2 August 1981 \n\n —\n\n Complex question:Who was born later, Joy or Emma? \nAnswer: Emma \nExplanation: Based on the birth date of Joy and Emma, we can infer that the answer is Emma because 1981 is larger than 1950. List of sub-questions and sub-answers: subqa_list \n\n —\n\n Complex question: str_question \nAnswer: \nExplanation:

Table 6.6: Examples of the prompts that we use for running GPT-3.5 in the **composition stage**. str_question is the input question for the required task and subqa_list represents a list of sub-questions and sub-answers. Both datasets use the same prompt for the zero-shot setting.

6.4.2 Results

Decomposition Stage

In this stage, the input is a complex question and the output is a list of sub-questions. Automatic evaluation is difficult since there are many valid ways to split a complex question into sub-questions. We observe that under the few-shot setting, models can follow the exemplars in the prompt to perform the split. Therefore, we have designed an automatic evaluation for Date-complex in the few-shot setting (scores marked with a star in Table 6.7). Meanwhile, for the remaining settings, we conduct manual evaluations, randomly assessing 300 samples (50 per hop) for Date-complex and 200 samples (50 per hop) for 2Wiki-complex.

Dataset	Llama 2 13B		Llama 2 70B		GPT-3.5	
	Zero	Few	Zero	Few	Zero	Few
Date	19.0	89.0*	58.0	97.3*	87.0	100*
2Wiki	20.5	26.0	19.0	57.0	28.0	62.5

Table 6.7: Average accuracies in the decomposition stage across hops. Zero and few represent zero-shot and few-shot settings, respectively.

Table 6.7 presents average accuracies of models from different numbers of hops. We observe that models perform well at breaking down comparison questions in the Date-complex dataset under the few-shot setting. However, in the zero-shot setting, a smaller model, such as Llama 2 13B, struggles to correctly split the complex questions. Regarding the 2Wiki-complex dataset, we find that even a large model like GPT-3.5, under the few-shot setting, cannot accurately decompose complex questions. After manually examining GPT-3.5's predicted sub-questions, we identified three main error types: exceeding one step, missing one step, and incorrectly splitting relations, with the most common being missing one step. Table 6.8 presents several error cases of GPT-3.5 in the decomposition stage.

Number of hop	Example & Error Type
2-hop	Question: Who is Catherine Fitzcharles's paternal grandmother? (Acc-strict: 0; Acc-soft: 0; %-Sub-questions: 100% (2/2) Sub-question-1: Who is the father of Catherine Fitzcharles? Sub-question-2: Who is the father of the father of Catherine Fitzcharles? Sub-question-3: Who is the mother of the father of Catherine Fitzcharles? Sub-question-4: Who is the mother of Catherine Fitzcharles's paternal grandmother?
3-hop	Question: Who is the maternal grandfather' s father of Guiscarda, Viscountess of Béarn? (Acc-strict: 0; Acc-soft: 1; %-Sub-questions: 66.7% (2/3) Sub-question-1: Who is the maternal grandfather of Guiscarda, Viscountess of Béarn? Sub-question-2: Who is the father of the maternal grandfather of Guiscarda, Viscountess of Béarn?
3-hop	Question: What is the date of birth of maternal grandfather of Hans Albrecht? (Acc-strict: 0; Acc-soft: 0; %-Sub-questions: 0% (0/3) Sub-question-1: Who is the father of Hans Albrecht? Sub-question-2: Who is the father of the father of Hans Albrecht? Sub-question-3: What is the date of birth of the father of the father of Hans Albrecht?
3-hop	Question: What is the date of death of paternal grandmother of Ari'i-Otare Terii-maeva-rua III? (Acc-strict: 0; Acc-soft: 0; %-Sub-questions: 33.3% (1/3) Sub-question-1: Who is the father of Ari'i-Otare Terii-maeva-rua III? Sub-question-2: Who is the father of Sub-Answer-1? Sub-question-3: What is the date of death of Sub-Answer-2?
4-hop	Question: Who is the father' s father' s father' s father of Richard of Normandy? (Acc-strict: 0; Acc-soft: 0; %-Sub-questions: 75.0% (3/4) Sub-question-1: Who is the father of Richard of Normandy? Sub-question-2: Who is the father of the father of Richard of Normandy? Sub-question-3: Who is the father of the father of the father of Richard of Normandy?
5-hop	Question: What is the place of death of father' s father' s father' s father of James FitzGerald, 8th Earl of Desmond? (Acc-strict: 0; Acc-soft: 0; %-Sub-questions: 60.0% (3/5) Sub-question-1: Who is the father of James FitzGerald, 8th Earl of Desmond? Sub-question-2: Who is the father of the father of James FitzGerald, 8th Earl of Desmond? Sub-question-3: Who is the father of the father of the father of James FitzGerald, 8th Earl of Desmond? Sub-question-4: What is the place of death of the father of the father of the father of James FitzGerald, 8th Earl of Desmond?

Table 6.8: Examples of error cases of GPT-3.5 in the decomposition stage.

While the reasoning steps of these sub-questions may lack one step, they can still lead to the correct answer. To gain deeper insights into such cases, we further evaluate 2Wiki-complex by introducing two new evaluation scores: soft accuracy and the percentage of correct sub-questions. The models achieve a score of one for

Acc-strict when they include the complete list of subquestions in the reasoning path. The models achieve a score of one for Acc-soft when they can reach the correct path to the final answer, even if they miss some steps or if some steps are not in the correct order initially. By default, if the models achieve a score of one for Acc-strict, they also achieve a score of one for Acc-soft and 100% for the percentage of correct sub-questions. For each sample that does not achieve a score of one for Acc-strict and Acc-soft, we record the number of correct sub-questions. The percentage of correct sub-questions is calculated by dividing the number of correct sub-questions by the number of gold sub-questions. The models can obtain one for Acc-soft but it is not necessary that they obtain all correct sub-questions. Table 6.9 presents all scores for 2Wiki-complex. For GPT-3.5, the soft accuracy is 79.0% which is higher than the current strict accuracy, 62.5%. However, this score is still low for a large model like GPT-3.5 on performing decomposing complex questions.

Model	**Zero-shot**			**Few-shot**		
	Acc-strict	Acc-soft	%-Sub-questions	Acc-strict	Acc-soft	%-Sub-questions
Llama 2 13B	20.5	42.0	67.1	26.0	37.0	62.9
Llama 2 70B	19.0	65.5	75.0	57.0	69.5	86.0
GPT-3.5	28.0	74.5	70.4	62.5	79.0	86.2

Table 6.9: All average scores in the decomposition stage across hops for the 2Wiki-complex dataset.

Subproblem Solving Stage

In this stage, the input is a simple question with or without context, and the output is an answer to the simple question. We experiment with three different scenarios: (S1) without context, (S2) with unstructured context, and (S3) with structured context. It is noted that in the case of the Date-complex dataset, the sub-questions are related to the date of birth, date of death, or the age of a person. In the scenario without context, the models refuse to answer these sub-questions.

Table 6.10 presents the average exact match (EM) of GPT-3.5 on Date-complex and 2Wiki-complex when solving sub-questions. Results of Llama 2 13B and 70B are presented in Table 6.11. In summary, the models are able to successfully solve the

Dataset	Without		Unstructured		Structured	
	Zero	Few	Zero	Few	Zero	Few
Date	-	-	82.9	83.4	98.0	91.9
2Wiki	17.8	24.2	69.8	72.1	80.7	84.9

Table 6.10: Average EM of GPT-3.5 on sub-questions.

sub-question tasks when provided with context. When comparing scenarios with and without context, we observe higher scores in the context scenario (including both S2 and S3) than in the scenario without context. This indicates that current LLMs are still not good at memorizing factual knowledge during pre-training when solving simple factual questions available on Wikipedia pages. When comparing scenarios with unstructured and structured contexts, we see that working with structured context yields higher scores. This raises a new question: Is it because the model works well with structured context or performs well with short context? We leave this for future research.

Dataset	Without Context				Unstructured				Structured Context			
	Zero		Few		Zero		Few		Zero		Few	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Llama 2 13B												
Date-complex	-	-	-	-	81.4	82.9	71.1	71.8	82.1	82.2	82.7	82.7
2Wiki-complex	6.2	33.6	3.4	28.8	37.9	61.5	29.8	52.0	67.2	79.0	58.4	71.4
Llama 2 70B												
Date-complex	-	-	-	-	83.8	84.3	87.5	87.7	82.8	82.8	82.9	82.9
2Wiki-complex	9.2	36.2	9.0	38.0	43.7	63.7	40.6	59.6	63.2	74.6	76.6	84.1

Table 6.11: Average EM and F1 scores of Llama 2 13B and 70B on Date-complex and 2Wiki-complex when solving sub-questions. Zero and few represent zero-shot and few-shot settings, respectively.

Composition Stage

In this stage, the input is a list of sub-questions with sub-answers and a complex question, and the output is the answer to the complex question. It aims to assess the

Dataset	Llama 2 13B		Llama 2 70B		GPT-3.5	
	Zero	Few	Zero	Few	Zero	Few
Date	56.3	64.0	78.3	74.9	91.3	94.3
2Wiki	93.9	96.0	97.6	99.5	95.0	90.5

Table 6.12: Average EM scores in the composition stage.

models' abilities to compose and compare multiple values for the final answer.

Table 6.12 presents average EM scores from different numbers of hops of the models in the composition stage. We observe that both Llama 2 (13B and 70B) struggle to perform comparisons between multiple dates or numbers to obtain the final answer for comparison questions in Date-complex. In contrast, GPT-3.5 does not face challenges when comparing multiple dates or numbers. For the 2Wiki-complex, we find that all models can easily obtain the final answer because the answer to the complex question is the final sub-answer in the list of provided sub-questions & sub-answers.

6.4.3 Decomposition vs. Final Performance

To explore the connection between the decomposition stage and GPT-3.5's final QA performance, we measure their correlation using the 200 samples from Section 6.4.2. The Pearson correlation coefficient scores are presented in Table 6.13, revealing no correlation between the decomposition stage and the final QA performance. Our hypothesis is that, contrary to our expectation of a step-by-step reasoning process, models may employ shortcuts (Chen and Durrett, 2019) to answer questions, impacting their final QA performance.

	Zero-shot	Few-shot	CoT
Strict Accuracy	0.092	0.080	0.088
Soft Accuracy	0.034	0.022	-0.011

Table 6.13: Pearson correlation coefficient scores between the decomposition stage and the final QA performance.

To verify our hypothesis that the model can perform reasoning shortcuts in answering multi-hop questions directly, we further run experiments on GPT-3.5 with

various settings of the context. All scores are presented in Table 6.14. It is noted that the gold paragraphs are the list of paragraphs that contain the information to answer the question. This list of gold paragraphs is ordered as the reasoning steps; it means that the last gold paragraph is the paragraph that contains the final answer for the complex question.

As shown in the Table, the model achieves the highest EM score when using all gold paragraphs. The model obtains the lowest scores when no gold paragraphs are input. In settings (3) and (4), the model's performance dramatically decreases when one paragraph is removed, particularly the last paragraph. However, removing the first paragraph has a less significant impact on the model's performance. In both cases, the model theoretically shouldn't be able to answer the question due to the lack of sufficient gold paragraphs. This validates our hypothesis: in the full QA performance setting, the model can employ reasoning shortcuts instead of engaging in step-by-step reasoning like humans.

Setting	**EM**	**F1**
(1) Without context	2.0	8.5
(2) All gold paragraphs	**43.0**	54.3
(3) Remove first gold paragraph	40.5	**57.5**
(4) Remove last gold paragraph	3.5	19.5
(5) Only last gold paragraph	37.5	56.2

Table 6.14: EM and F1 scores of the QA performance with various settings of the context input.

6.5 Conclusion

We introduce two new complex multi-hop datasets that range from 2-hop to 5-hop for 2Wiki-complex and from 2-hop to 7-hop for Date-complex. We then use these datasets to investigate why LLMs fail in solving complex multi-hop questions. To comprehensively investigate the behavior of LLMs, we split the QA process into three main stages and experiment with three different scenarios of the context. Our experiments show that the main reason is their inability to decompose complex questions into sub-questions. We find that LLMs, even GPT-3.5, still lack the ability

to answer simple sub-questions when no context is provided. These sub-questions involve simple factual knowledge available in Wikipedia pages, indicating that LLMs still face challenges in memorizing factual knowledge during pre-training. In the composition stage, we find that both Llama 2 13B and 70B struggle in performing comparisons of multiple dates/numbers to obtain the final answer.

7 Conclusion and Discussion

7.1 Conclusion

In this dissertation, we explored and evaluated the reasoning steps in the multi-hop MRC task.

In Chapter 2, we provided an overview of the multi-hop MRC task, including the definition of the reasoning steps and different forms used to represent them. We discussed the issues of reasoning shortcuts in the MRC task. Additionally, we summarized the available techniques for measuring shortcuts.

In Chapter 3, we presented 2WikiMultiHopQA—a large and high quality multi-hop dataset that provides comprehensive explanations for predictions. We utilized logical rules in the KB to create more natural questions that still require multi-hop reasoning. Through experiments, we demonstrated that our dataset ensures multi-hop reasoning while being challenging for the multi-hop models. We also demonstrated that bootstrapping the multi-hop MRC dataset is beneficial by utilizing large-scale available data on Wikipedia and Wikidata.

In Chapter 4, we analyzed the effectiveness of the underlying reasoning tasks using two multi-hop datasets: 2Wiki and HotpotQA-small. The results revealed that the underlying reasoning tasks can improve QA performance. Using four debiased sets, we demonstrated that the underlying reasoning tasks can reduce the reasoning shortcuts of the QA task. The results also revealed that the underlying reasoning tasks do not make the models more robust on adversarial examples, such as sub-questions and inverted questions.

In Chapter 5, we proposed a new multi-hop RC dataset for comprehensively evaluating the ability of existing models to understand date information. We evaluated the top-performing models on our dataset. The results revealed that the models may not possess the ability to subtract two dates even when fine-tuned on our dataset. We also found that our probing questions could help to improve QA performance, and can be used for data augmentation.

In Chapter 6, we introduced two new complex multi-hop datasets that range from 2-hop to 5-hop for 2Wiki-complex and from 2-hop to 7-hop for Date-complex. We then used these datasets to investigate why LLMs fail in solving complex multi-hop questions. To comprehensively investigate the behavior of LLMs, we split the QA process into three main stages and experiment with three different scenarios of the context. Our experiments showed that the main reason is their inability to decompose complex questions into sub-questions. We found that LLMs, even GPT-3.5, still lack the ability to answer simple sub-questions when no context is provided. In the composition stage, we found that both Llama 2 13B and 70B struggle in performing comparisons of multiple dates/numbers to obtain the final answer.

Altogether, we utilized the reasoning steps to form a new task and explored their effectiveness. However, defining the reasoning steps can be quite difficult for different types of questions, such as commonsense questions. Our understanding of the models is still limited; we still don't fully comprehend how the models obtain the answers. Our findings in Chapter 6 revealed that the models do not follow the human reasoning process, indicating that they use their own methods to answer questions.

7.2 Discussion and Future Works

Discussion Most of our datasets are generated semi-automatically, incorporating human involvement in certain stages, such as template creation, while using code to automatically generate samples in the final steps. As a result, some errors may remain undetected during the creation process. For example, as discussed in Section 3.7.4, we found that in the 2Wiki dataset, 8 out of 100 samples are unanswerable due to mismatches between Wikipedia articles and Wikidata triples. Another disadvantage of synthetic datasets is their lack of diversity, as many samples use the same template. Despite these disadvantages, synthetic datasets offer several significant advantages. Firstly, cost efficiency: creating synthetic datasets is less expensive than using human crowdsourcing for the entire dataset. Secondly, control: with synthetic datasets, we can manage all templates and precisely determine the number of samples for each template as needed. Thirdly, data size: creating synthetic datasets typically has no limitations on size; we can generate as many samples as needed. In summary, each method of dataset creation has its own advantages and disadvantages. Before our dataset creation (2WikiMultiHopQA), QAngaroo (Welbl et al., 2018) was the first multi-hop reasoning dataset. It was automatically constructed using knowledge bases and Wikipedia. QAngaroo contains two sub-datasets: WikiHop, in the open domain, and MedHop, in the medical domain.

Today, with the advent of LLMs, the need for extensive examples in model training has diminished. However, the benefits of cost efficiency and control remain valuable to the community when using synthetic datasets. We find that combining human and machine efforts in dataset creation is currently the most effective approach. Initially, code or machines can generate samples, followed by human verification in the final step to ensure quality.

Future Works For future work, we have two possible directions. The first one is to improve the existing multi-hop datasets. We observe that most existing multi-hop datasets have extractive answer types, which can lead models to take shortcuts in answering questions. We suggest that, instead of using extractive answer types, we can introduce another type of reasoning on top of existing multi-hop questions, which convert the answer type from extractive to generative. For example, given the question

"What is the date of birth of the director of the film La La Land?", we can pose a new question: "What is the next day after the date of birth of the director of the film La La Land?" The second one is to construct a multi-hop dataset in another domain, such as scientific text. Currently, there are no multi-hop MRC datasets that utilize full-text of multiple scientific articles. If we can construct a new multi-hop dataset on scientific domain, it would be useful to help the process of reading scientific papers.

Some Recent and Emerging Studies Recently, with the success of LLMs, many new research studies related to LLMs have been released. We observe that there are two research directions related to our thesis.

The first area of focus is the interpretability of LLMs. Since LLMs remain a black box to humans, understanding their operation is crucial. The multi-hop reasoning task is suitable for conducting this kind of internal investigation for two reasons: (1) we can control the number of reasoning steps in the QA process, and (2) the information to reason from is not explicitly provided in the input. Recently, several research studies (Sakarvadia et al., 2023; Yang et al., 2024; Biran et al., 2024) have conducted experiments and analyses to investigate the internal mechanisms of LLMs in solving two-hop queries.

The second research direction is about discovering the reasoning abilities of LLMs. Our thesis focuses on multi-hop reasoning, but there are many other types of reasoning that are also important to investigate before using LLMs in real-world applications. These include deductive reasoning, temporal reasoning, logical reasoning, arithmetic reasoning, common sense reasoning, multimodal reasoning, and symbolic reasoning. Recently, some works have focused on these types of reasoning, such as Mondorf and Plank (2024) for deductive reasoning, Xiong et al. (2024) for temporal reasoning, and Wang et al. (2024) for logical reasoning.

Bibliography

[1] Abdalghani Abujabal, Mohamed Yahya, Mirek Riedewald, and Gerhard Weikum. Automated template generation for question answering over knowledge graphs. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1191–1200, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee. ISBN 9781450349130. doi: 10.1145/3038912.3052583. URL https://doi.org/10.1145/3038912.3052583.

[2] Rakesh Agrawal, Tomasz Imieliundefinedski, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, page 207–216, New York, NY, USA, 1993. Association for Computing Machinery. ISBN 0897915925. doi: 10.1145/170035.170072. URL https://doi.org/10.1145/170035.170072.

[3] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In Waleed Ammar, Annie Louis, and Nasrin Mostafazadeh, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4010. URL https://aclanthology.org/N19-4010.

[4] Hadeel Al-Negheimish, Pranava Madhyastha, and Alessandra Russo. Numerical reasoning in machine reading comprehension tasks: are we there yet? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9643–9649, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-main.759.

[5] Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations*, 2020.

[6] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1160.

[7] Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. Hopping too late: Exploring the limitations of large language models on multi-hop queries. *arXiv:2406.12775*, 2024. URL https://arxiv.org/abs/2406.12775.

[8] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. volume abs/1506.02075, 2015. URL http://arxiv.org/abs/1506.02075.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[10] Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. KQA pro: A dataset with explicit compositional programs for complex question answering over knowledge base. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

6101–6119, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.422. URL https://aclanthology.org/2022.acl-long.422.

[11] Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1223. URL https://www.aclweb.org/anthology/P16-1223.

[12] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1171. URL https://www.aclweb.org/anthology/P17-1171.

[13] Jifan Chen and Greg Durrett. Understanding dataset design choices for Multi-hop reasoning. 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 4026–4032, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1405. URL https://www.aclweb.org/anthology/N19-1405.

[14] Jifan Chen, Shih ting Lin, and Greg Durrett. Multi-hop question answering via reasoning chains. *arXiv*, 2019.

[15] Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.91. URL https://www.aclweb.org/anthology/2020.findings-emnlp.91.

[16] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021.

[17] Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 5925–5932, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1606. URL https://aclanthology.org/D19-1606.

[18] Nicola De Cao, Wilker Aziz, and Ivan Titov. Question answering by reasoning across documents with graph convolutional networks. 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 2306–2317, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1240. URL https://www.aclweb.org/anthology/N19-1240.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

[20] Dennis Diefenbach, Thomas Tanon, Kamal Singh, and Pierre Maret. Question answering benchmarks for Wikidata. 10 2017.

[21] Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2694–2703, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1259. URL https://www.aclweb.org/anthology/P19-1259.

[22] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Jill Burstein, Christy Doran, and Thamar Solorio,

editors, *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 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246.

[23] Dheeru Dua, Shivanshu Gupta, Sameer Singh, and Matt Gardner. Successive prompting for decomposing complex questions. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1251–1265, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.81. URL https://aclanthology.org/2022.emnlp-main.81.

[24] Jesse Dunietz, Greg Burnham, Akash Bharadwaj, Owen Rambow, Jennifer Chu-Carroll, and Dave Ferrucci. To test machine comprehension, start by defining comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7839–7859, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.701. URL https://aclanthology.org/2020.acl-main.701.

[25] Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823–8838, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.710. URL https://www.aclweb.org/anthology/2020.emnlp-main.710.

[26] Yair Feldman and Ran El-Yaniv. Multi-hop paragraph retrieval for open-domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2309, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1222. URL https://www.aclweb.org/anthology/P19-1222.

[27] Ruiliu Fu, Han Wang, Xuejun Zhang, Jun Zhou, and Yonghong Yan. Decomposing complex questions makes multi-hop QA easier and more interpretable. In Marie-

Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 169–180, Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.17. URL https://aclanthology.org/2021.findings-emnlp.17.

[28] Ruiliu Fu, Han Wang, Jun Zhou, and Xuejun Zhang. Na-reviewer: Reviewing the context to improve the error accumulation issue for multi-hop qa. *Electronics Letters*, 12 2021b. doi: 10.1049/ell2.12411.

[29] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 413–422, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320351. doi: 10.1145/2488388.2488425. URL https://doi.org/10.1145/2488388.2488425.

[30] Wee Chung Gan and Hwee Tou Ng. Improving the robustness of question answering systems to question paraphrasing. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1610. URL https://aclanthology.org/P19-1610.

[31] Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.117. URL https://aclanthology.org/2020.findings-emnlp.117.

[32] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, Nov 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL http://dx.doi.org/10.1038/s42256-020-00257-z.

[33] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 04 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00370. URL https://doi.org/10.1162/tacl_a_00370.

[34] Mor Geva, Tomer Wolfson, and Jonathan Berant. Break, perturb, build: Automatic perturbation of reasoning paths through question decomposition. *Transactions of the Association for Computational Linguistics*, 10:111–126, 2022. doi: 10.1162/tacl_a_00450. URL https://aclanthology.org/2022.tacl-1.7.

[35] Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA, 2015. MIT Press.

[36] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 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), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://www.aclweb.org/anthology/2020.coling-main.580.

[37] Xanh Ho, Saku Sugawara, and Akiko Aizawa. How well do multi-hop reading comprehension models understand date information? In Yulan He, Heng Ji, Sujian Li, Yang Liu, and Chua-Hui Chang, editors, *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 470–479, Online only, November 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.aacl-short.58.

[38] Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 2391–2401, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1243. URL https://aclanthology.org/D19-1243.

[39] Naoya Inoue, Pontus Stenetorp, and Kentaro Inui. R4C: A benchmark for evaluating RC systems to get the right answer for the right reason. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6740–6750, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.602. URL https://www.aclweb.org/anthology/2020.acl-main.602.

[40] Peter Jansen, Niranjan Balasubramanian, Mihai Surdeanu, and Peter Clark. What's in an explanation? Characterizing knowledge and inference requirements for elementary science exams. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2956–2965, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1278.

[41] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL https://www.aclweb.org/anthology/D17-1215.

[42] Yichen Jiang and Mohit Bansal. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2726–2736, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1262. URL https://www.aclweb.org/anthology/P19-1262.

[43] Yichen Jiang and Mohit Bansal. Self-assembling modular networks for interpretable multi-hop reasoning. 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 4474–4484, Hong Kong, China, November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1455. URL https://www.aclweb.org/anthology/D19-1455.

[44] Divyansh Kaushik and Zachary C. Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1546. URL https://aclanthology.org/D18-1546.

[45] Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=_nGgzQjzaRy.

[46] Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. Look at the first sentence: Position bias in question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1109–1121, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.84. URL https://aclanthology.org/2020.emnlp-main.84.

[47] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

[48] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794,

Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://www.aclweb.org/anthology/D17-1082.

[49] Jieyu Lin, Jiajie Zou, and Nai Ding. Using adversarial attacks to reveal the statistical bias in machine reading comprehension models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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 333–342, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.43. URL https://aclanthology.org/2021.acl-short.43.

[50] Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. LogiQA: A challenge dataset for machine reading comprehension with logical reasoning. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/501. URL https://doi.org/10.24963/ijcai.2020/501. Main track.

[51] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv*, 2019.

[52] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-5010. URL https://www.aclweb.org/anthology/P14-5010.

[53] Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. Compositional questions do not necessitate multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4249–4257, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1416. URL https://www.aclweb.org/anthology/P19-1416.

[54] Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1613. URL https://www.aclweb.org/anthology/P19-1613.

[55] Philipp Mondorf and Barbara Plank. Comparing inferential strategies of humans and large language models in deductive reasoning. *arXiv:2402.14856*, 2024. URL https://arxiv.org/abs/2402.14856.

[56] Stephen Muggleton. Inverse entailment and Progol, 1995.

[57] Mao Nakanishi, Tetsunori Kobayashi, and Yoshihiko Hayashi. Answerable or not: Devising a dataset for extending machine reading comprehension. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 973–983, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1083.

[58] Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2335–2345, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1225. URL https://aclanthology.org/P19-1225.

[59] Simon Ostermann, Michael Roth, and Manfred Pinkal. MCScript2.0: A machine comprehension corpus focused on script events and participants. In Rada Mihalcea, Ekaterina Shutova, Lun-Wei Ku, Kilian Evang, and Soujanya Poria, editors, *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 103–117, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/S19-1012. URL https://aclanthology.org/S19-1012.

[60] Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. Unsupervised question decomposition for question answering. In *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8864–8880, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.713. URL https://www.aclweb.org/anthology/2020.emnlp-main.713.

[61] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378.

[62] Peng Qi, Haejun Lee, Tg Sido, and Christopher Manning. Answering open-domain questions of varying reasoning steps from text. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3599–3614, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.292. URL https://aclanthology.org/2021.emnlp-main.292.

[63] Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 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, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1617. URL https://www.aclweb.org/anthology/P19-1617.

[64] J. R. Quinlan. Learning logical definitions from relations. volume 5, page 239–266, USA, September 1990. Kluwer Academic Publishers. doi: 10.1023/A:1022699322624. URL https://doi.org/10.1023/A:1022699322624.

[65] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020. ISSN 1532-4435.

[66] Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! Leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1487. URL https://www.aclweb.org/anthology/P19-1487.

[67] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://www.aclweb.org/anthology/D16-1264.

[68] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://www.aclweb.org/anthology/P18-2124.

[69] Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. NumNet: Machine reading comprehension with numerical reasoning. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 2474–2484, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1251. URL https://aclanthology.org/D19-1251.

[70] Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, March 2019. doi: 10.1162/tacl_a_00266. URL https://www.aclweb.org/anthology/Q19-1016.

[71] Danilo Neves Ribeiro, Shen Wang, Xiaofei Ma, Henghui Zhu, Rui Dong, Deguang Kong, Juliette Burger, Anjelica Ramos, zhiheng huang, William Yang Wang, George Karypis, Bing Xiang, and Dan Roth. STREET: A MULTI-TASK STRUCTURED REASONING

AND EXPLANATION BENCHMARK. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1C_kSW1-k0.

[72] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

[73] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging NLP models. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1079. URL https://aclanthology.org/P18-1079.

[74] Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. Are red roses red? evaluating consistency of question-answering models. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6174–6184, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1621. URL https://aclanthology.org/P19-1621.

[75] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442.

[76] Barbara Rychalska, Dominika Basaj, and Przemyslaw Biecek. Are you tough enough? framework for robustness validation of machine comprehension systems. In *NeurIPS IRASL Workshop*, 2018.

[77] Mansi Sakarvadia, Aswathy Ajith, Arham Khan, Daniel Grzenda, Nathaniel Hudson, André Bauer, Kyle Chard, and Ian Foster. Memory injections: Correcting multi-hop reasoning failures during inference in transformer-based language models. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi, editors, *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 342–356, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.26. URL https://aclanthology.org/2023.blackboxnlp-1.26.

[78] Viktor Schlegel, Goran Nenadic, and Riza Batista-Navarro. Beyond leaderboards: A survey of methods for revealing weaknesses in natural language inference data and models. *arXiv:2005.14709*, 2020.

[79] Viktor Schlegel, Goran Nenadic, and Riza Batista-Navarro. Semantics altering modifications for evaluating comprehension in machine reading. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13762–13770, May 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17622.

[80] Stefan Schoenmackers, Jesse Davis, Oren Etzioni, and Daniel Weld. Learning first-order Horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1088–1098, Cambridge, MA, October 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D10-1106.

[81] Priyanka Sen and Amir Saffari. What do models learn from question answering datasets? In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2429–2438, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.190. URL https://aclanthology.org/2020.emnlp-main.190.

[82] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HJ0UKP9ge.

[83] Chenglei Si, Shuohang Wang, Min-Yen Kan, and Jing Jiang. What does bert learn from multiple-choice reading comprehension datasets? *arXiv:1910.12391*, 2019.

[84] Chenglei Si, Ziqing Yang, Yiming Cui, Wentao Ma, Ting Liu, and Shijin Wang. Benchmarking robustness of machine reading comprehension models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 634–644, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.56. URL https://aclanthology.org/2021.findings-acl.56.

[85] Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 4506–4515, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1458. URL https://aclanthology.org/D19-1458.

[86] Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks. *arXiv:1809.02040*, 2018.

[87] Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1453. URL https://www.aclweb.org/anthology/D18-1453.

[88] Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and Akiko Aizawa. Assessing the benchmarking capacity of machine reading comprehension datasets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8918–8927, Apr. 2020. doi: 10.1609/aaai.v34i05.6422. URL https://ojs.aaai.org/index.php/AAAI/article/view/6422.

[89] Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex 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 641–651, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1059. URL https://www.aclweb.org/anthology/N18-1059.

[90] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421.

[91] Chenhao Tan and Lillian Lee. A corpus of sentence-level revisions in academic writing: A step towards understanding statement strength in communication. In Kristina Toutanova and Hua Wu, editors, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 403–408, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2066. URL https://aclanthology.org/P14-2066.

[92] Yixuan Tang, Hwee Tou Ng, and Anthony Tung. Do multi-hop question answering systems know how to answer the single-hop sub-questions? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3244–3249, Online, April 2021. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.eacl-main.283.

[93] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian,

Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.

[94] Son Quoc Tran, Phong Nguyen-Thuan Do, Uyen Le, and Matt Kretchmar. The impacts of unanswerable questions on the robustness of machine reading comprehension models. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1543–1557, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.113. URL https://aclanthology.org/2023.eacl-main.113.

[95] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Is multihop QA in DiRe condition? measuring and reducing disconnected reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8846–8863, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.712. URL https://www.aclweb.org/anthology/2020.emnlp-main.712.

[96] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. ♫ MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31.

[97] Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2704–2713, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1260. URL https://www.aclweb.org/anthology/P19-1260.

[98] Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *AAAI*, 2020.

[99] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 2153–2162, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1221. URL https://aclanthology.org/D19-1221.

[100] Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. Can llms reason with rules? logic scaffolding for stress-testing and improving llms. *arXiv:2402.11442*, 2024. URL https://arxiv.org/abs/2402.11442.

[101] Xuezhi Wang, Haohan Wang, and Diyi Yang. Measure and improve robustness in NLP models: A survey. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.339. URL https://aclanthology.org/2022.naacl-main.339.

[102] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

[103] Yicheng Wang and Mohit Bansal. Robust machine comprehension models via adversarial training. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 575–581, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2091. URL https://aclanthology.org/N18-2091.

[104] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed,

A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[105] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302, 2018. doi: 10.1162/tacl_a_00021. URL https://www.aclweb.org/anthology/Q18-1021.

[106] Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198, 2020. doi: 10.1162/tacl_a_00309. URL https://www.aclweb.org/anthology/2020.tacl-1.13.

[107] Yulong Wu, Viktor Schlegel, and Riza Batista-Navarro. Is the understanding of explicit discourse relations required in machine reading comprehension? In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3565–3579, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.311. URL https://aclanthology.org/2021.eacl-main.311.

[108] Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. Large language models can learn temporal reasoning. *arXiv:2401.06853*, 2024. URL https://arxiv.org/abs/2401.06853.

[109] Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? *arXiv:2402.16837*, 2024. URL https://arxiv.org/abs/2402.16837.

[110] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 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, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://www.aclweb.org/anthology/D18-1259.

[111] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

[112] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1128. URL https://www.aclweb.org/anthology/P15-1128.

[113] Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2033. URL https://www.aclweb.org/anthology/P16-2033.

[114] Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. Reclor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgJtT4tvB.

[115] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf.

[116] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-

scale adversarial dataset for grounded commonsense inference. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1009. URL https://aclanthology.org/D18-1009.

[117] Dongxu Zhang and Dong Wang. Relation classification via recurrent neural network. *arXiv:1508.01006*, 2015.

[118] Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(3), apr 2020. ISSN 2157-6904. doi: 10.1145/3374217. URL https://doi.org/10.1145/3374217.

[119] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018b.

[120] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *arXiv:2303.18223*, 2023.

[121] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.