SEMI-AUTOMATIC PROMPTING APPROACH WITH QUESTION DECOMPOSITION FOR MULTI-HOP QUESTION ANSWERING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

ARIF OZAN KIZILDAĞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

SEPTEMBER 2023

Semi-Automatic Prompting Approach with Question Decomposition for Multi-Hop Question Answering

submitted by **ARIF OZAN KIZILDAĞ** in partial fulfillment of the requirements for the degree of **Master of Science in Information Systems Department, Middle East Technical University** by,

Prof. Dr. Banu Günel Kılıç
Dean, **Graduate School of Informatics** ________________

Prof. Dr. Altan Koçyiğit
Head of Department, **Information Systems** ________________

Prof. Dr. Tuğba Taşkaya Temizel
Supervisor, **Department of Data Informatics, METU** ________________

Examining Committee Members:

Assoc. Prof. Dr. Erhan Eren
Information Systems, METU ________________

Prof. Dr. Tuğba Taşkaya Temizel
Department of Data Informatics, METU ________________

Assist. Prof. Dr. Didem Ölçer
Computer Engineering, Başkent University ________________

Date: 07.09.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Arif Ozan Kızıldağ

Signature :

ABSTRACT

SEMI-AUTOMATIC PROMPTING APPROACH WITH QUESTION DECOMPOSITION FOR MULTI-HOP QUESTION ANSWERING

Kızıldağ, Arif Ozan
M.S., Department of Information Systems
Supervisor: Prof. Dr. Tuğba Taşkaya Temizel

September 2023, 44 pages

With the help of large language models, prompt engineering enables easy access to vast knowledge for various applications. However, limited research has been done on multi-hop question answering using this approach. This thesis introduces a new semi-automatic prompting method for answering two-hop questions. The method involves creating a prompt with automatically selected examples by grouping answer-named entities from the training set and using a chain-of-thought principle. The results demonstrate comparable performance to fine-tuned models on the MuSiQue dataset. Ablation studies further validate the effectiveness of each component in the proposed method. The approach has the potential to be applied to more complex multi-hop question-answering systems while upholding performance on par with other state-of-the-art techniques.

Keywords: Large language models, question decomposition, prompt engineering, multi-hop question answering

ÖZ

ÇOKLU ADIMLI SORU CEVAPLAMA İÇİN SORU PARÇALAMA İLE YARI OTOMATİK İSTEMLEME YAKLAŞIMI

Kızıldağ, Arif Ozan
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Prof. Dr. Tuğba Taşkaya Temizel

Eylül 2023, 44 sayfa

İstem mühendisliği büyük dil modellerinin yardımıyla çeşitli uygulamalar için geniş bir bilgiye kolay erişim sağlar. Ancak, çoklu adımlı soru cevaplama konusunda bu yaklaşımı kullanan sınırlı araştırma yapılmıştır. Bu tez, iki adımlı soruları cevaplamak için yeni bir yarı otomatik istem yöntemi tanıtmaktadır. Yöntem, eğitim kümesinden cevap adlı varlıkları gruplayarak ve düşünce zinciri prensibi kullanarak otomatik olarak seçilen örneklerle bir istem oluşturmayı içerir. Sonuçlar, MuSiQue veri kümesinde ince ayarlı modellere kıyasla benzer performans göstermektedir. Ablasyon çalışmaları, önerilen yöntemdeki her bileşenin etkinliğini daha da doğrulamaktadır. Bu yaklaşım, diğer son teknoloji tekniklerle aynı düzeyde performansı korurken daha karmaşık çoklu adımlı soru-cevaplama sistemlerine uygulanma potansiyeline sahiptir.

Anahtar Kelimeler: Büyük dil modelleri, soru ayrıştırması, istem mühendisliği, çok adımlı soru yanıtlama

To a Peaceful and Happy Future

ACKNOWLEDGMENTS

First and foremost, I extend my profound gratitude to Prof. Dr. Tuğba Taşkaya Temizel for her unwavering support, patience, and invaluable insights throughout the course of this thesis. Her guidance ensured that I remained focused, steering me in the right direction and ultimately making this work possible. Additionally, it was under her mentorship as a teaching assistant in her NLP course that my passion for natural language processing was kindled.

I would like to thank my mother, Gülseven Kızıldağ; father, Salih Kızıldağ; and sister, Fulya Kızıldağ, for their constant support throughout my life. They have always been there for me during challenging times, just as I have always been there for them.

I am grateful to my friend İlhami Okur for his support and his encouragement to pursue academia.

I also thank the Turkish Scientific and Technological Research Council (TUBITAK) for their financial support with the Directorate of Science Fellowships and Grant Programmes (BIDEB) 2210-A program during my study.

I also extend my appreciation to the instructors and research assistants at METU Informatics Institute. There are too many to name, but each has offered their invaluable support and assistance.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

LLM	Large Language Model
LM	Language Model
NLP	Natural Language Processing
QA	Question Answering

CHAPTER 1

INTRODUCTION

The progress in large language models (LLMs) has facilitated rapid advancement in many areas, including natural language processing (NLP). This progress has encouraged the discovery of new methodologies, further accelerating the development of LLMs. Today, language models with 175 billion parameters have become standard, whereas just a few years ago, models with 1 billion parameters were considered a dream to be reached. The race for a larger model continues, and models with trillions of parameters are now on the horizon.

Larger models with higher computational power have led to the development of many sub-tasks in NLP, allowing them to progress further. In the question-answering (QA) area, the previous norm was reading comprehension, which means answering a question from a given paragraph. However, the development of LLMs led to more complex trends like multi-hop reasoning, which involves more than one step to solve a question. For example, if a question asks to return the highest population of two cities, one first needs to find the populations and then compare them. The new datasets [1] [2] accommodating these trends have included distractor paragraphs, open-domain contexts, and multi-hop questions.

The advent of larger models has shifted the training norm from fine-tuning the pre-trained models to optimizing the input given to the models, known as prompts. This new approach, using zero-shot or few-shot techniques, exploits the vast knowledge stored within the billions of parameters of LLMs. This has led to the emergence of a new field known as prompt engineering, which seeks to optimize prompts given to the models to extract patterns required for specific tasks. Today, this novel technique is being applied in many new areas to test the limits of LLMs.

Prompting techniques are being tested on increasingly complex question-answering models. The aim is to create context-aware models that extract knowledge from these sophisticated models. Prompt engineering has changed the approaches to question-answering models; instead of creating fine-tuned models to find solutions in context, newer models take advantage of the vast knowledge of LLMs to answer questions. Researchers are now working on finding optimal ways to create prompts, aiming to create more efficient and adaptable question-answering models. The synergy between prompting and question-answering methods is opening the way to many advances in various domains, showing the unexplored potential of this combination.

One of the most recent topics in the question-answering area is multi-hop question answering. This field aims to find solutions to complex questions that require multiple steps because they either need information from multiple sources or include questions with more than one sub-question, like “How can we solve question-answering problems stated in recent years’ studies?” This question has a sub-question of “What question-answering problems have been reported in recent years’ studies?” Without answering these questions, no one can find the solution to the main question. Although this kind of question is a natural part of day-to-day speech, it has only recently started to be explored in the NLP area.

One of the ways to solve multi-hop questions is through question decomposition. This approach mainly focuses on dividing questions into the smallest meaningful sub-questions. The advantage of finding sub-questions is that these questions can be more easily solved due to their nature. After solutions to these sub-questions are found, they can be exploited to find the actual solution to the main question. Although the decomposition step creates an additional computational cost, it improves multi-hop question answering through its reasoning.

This thesis will focus on the intersection between multi-hop question answering, prompt engineering, and question decomposition. The proposed methodology explores the effectiveness of this convergence by creating a semi-automatic prompting creation approach. The focus of this thesis is to demonstrate the effectiveness of this approach, specifically within the context of two-hop questions. Utilized prompts are given at the end of the thesis for future work.

1.1 Research Questions

This thesis aims to answer the following questions:

RQ1: How can prompt engineering and decomposition techniques be effectively utilized to address the challenge of multi-hop question-answering in the 2-hop questions using a single prompt?

It is hypothesized that solving multi-hop questions can be improved by adding decomposition to find sub-questions, compared to prompting without decomposition. To explore this, a new prompting methodology is proposed to incorporate decomposition techniques into prompt creation.

RQ2: Does incorporating context in the decomposition process lead to improved outcomes compared to decomposing without context when utilizing prompting techniques?

It is hypothesized that some questions require context to be effectively decomposed into sub-questions. To find an answer to this question, the proposed prompting methodology will include the context during decomposition, and it will be compared to decomposition without context.

RQ3: When creating prompts, does clustering sub-question answers using named entity recognition lead to better outcomes compared to selecting the same named entities for examples for few-shot prompting?

It is hypothesized that during the creation of a prompt, including examples with answers that relate to similar named entities can degrade the quality of the prompting solutions. To study this phenomenon, a novel prompt-creation method is developed to select examples with different representative named entities automatically.

1.2 Contributions of the Study

The main contribution of this thesis is to propose a novel prompting approach that includes decomposition through a chain of thought reasoning. Only a few works combine prompt engineering methodologies with question decomposition using chain-of-thought reasoning. One notable example is Khot et al. [3], which employs a modular approach to create decompositions and then addresses the sub-questions individually. This work integrates a modular approach into a single prompt to address questions using decomposition, excluding document retrieval. This thesis study differs from Khot et al. [3] in the way that (i) decomposition is carried out with the help of the context, (ii) prompt creation is made semi-automatic with the help of answer named entities, and (iii) it uses a single prompt to address both question decomposition, and the question answering.

The proposed method uses context paragraphs for improved decomposition. In literature, questions are often decomposed without considering any context. In this work, the proposed model decomposes questions in the presence of context, using prompt engineering.

The proposed method is a semi-automatic prompt creation method that creates prompts with representative different answer-named entities. There are some works, such as Gao et al.[4] and Jiang et al. [5], that developed methods for automatic prompt creation. In the proposed model, the prompt is generated semi-automatically. While the initial template of the prompt is crafted manually to suit the model's requirements, examples are automatically selected for few-shot prompting. This facilitates the creation of prompts for new models and datasets. To achieve this, the model finds named entities in sub-questions answers to group related questions together.

Based on the work of Zhao et al.[6], a comprehensive ablation study is conducted to evaluate the effects of different choices made when designing the prompt developed in this thesis. This analysis highlights factors such as the role of decomposition and the inclusion of instructions. The study provides a guideline for crafting prompts for multi-hop question answering that involve decompositions.

To sum up, contributions of this study are as follows;

- A semi-automatic prompt creation method that creates prompts with representative different answer-named entities
- The effect of decomposition is shown with the help of context paragraphs.
- A detailed ablation study is carried out to show the impact of different parts of the prompt on the model performance.

1.3 Organization of the Thesis

The organization of this thesis is as follows: Chapter 2 reviews previous literature and outlines the reasoning behind this thesis. Chapter 3 explains the prompting methodology from the ground up. Chapter 4 details the experiments conducted by utilizing the proposed methodology and discusses the results through a detailed ablation study. Chapter 5 discusses the results and talks about possible future works. Chapter 6 concludes the thesis and discusses limitations.

CHAPTER 2

RELATED WORK

In this chapter, related studies are given under four main headings, which are LLMs, multi-hop reasoning, question decomposition, and prompt engineering.

2.1 Large Language Models

LLMs have gained immense popularity in recent years, leading companies to compete for the biggest and most successful models. This trend was made possible by the emergence of the transformer mode [7], which formed the basis of models like BERT (Bidirectional Encoder Representations from Transformers) [8], RoBERTa [9], BART, Longformer [10], and GPT (Generative Pre-trained Transformer) [11]. BERT significantly improved upon the original transformer by introducing pre-training and fine-tuning. Pre-training allows models to learn from vast amounts of unlabeled data, creating representations with neighboring tokens. Fine-tuning then leverages this knowledge for downstream tasks, such as question answering. RoBERTa extends BERT by addressing its limitations through modifications to the pretraining procedure and training on a larger dataset. Similarly, Longformer overcomes BERT's input token limitations by introducing a new attention mechanism that scales linearly with the sequence length. Another model, BART [12], builds upon BERT's bidirectional encoder approach, creating a denoising autoencoder capable of handling a wider range of tasks.

GPT is a series of language models created by OpenAI, utilizing the transformer model and sharing similarities with BERT. These models employ bidirectional representations for context understanding and can be easily fine-tuned for specific tasks by adding an additional linear layer. OpenAI has continued to develop larger models, such as GPT3[13], which has 175 billion parameters and enables few-shot learning. The few-shot learning methodology allows models to reach the underlying knowledge of the LLMs to solve the problems. GPT-3.5, also known as ChatGPT [14], is a version with conversation capabilities, available through an online platform for data collection and fine-tuning, as well as an API for more stable access. OpenAI has announced the development of GPT4[15], the newest model in the GPT family, which possesses multi-modal capabilities, allowing it to process both images and text. Models after that GPT3 are only accessible with OpenAI's API. On the other hand, Meta has created OPT (Open Pre-trained Transformer Language Models) [16] as a rival to GPT3, offering similar performance but being open for use by all researchers. OPT consists of 9 different-sized models, ranging from 125M to 175B parameters.

2.2 Multi-hop Reasoning

Multi-hop reasoning is an area that deals with questions requiring multiple information pieces and the combination of different sources. The name "multi-hop" stems from the need for multiple steps to complete the reasoning. In the NLP domain, it is commonly used in question-answering models. Earlier question-answering datasets, like Squad [2], focused mainly on reading comprehension and did not require multi-hop reasoning. However, advancements in sophisticated models have led to results comparable to human performance on these datasets.

The advanced question-answering datasets, like HotpotQA [17], have introduced 2-hop bridge questions derived from the Wikipedia dump using hyperlinks embedded in the opening sentences of documents. HotpotQA dataset provides two settings: full wiki and distractor. In the full wiki setting, documents need to be extracted from the Wikipedia dump. In the distractor setting, the model is trained using two gold paragraphs accompanied by eight distractor paragraphs.

Another dataset is the 2WikiMultiHopQA dataset, as described by Ho et al. [18], which utilizes both Wikipedia and Wikidata as sources. It incorporates reasoning steps in the dataset and introduces various types of questions, such as comparison, inference, compositional, and bridge questions. Additionally, the dataset includes simple questions generated by rule-based systems, which necessitate multi-hop reasoning for resolution.

The StrategyQA [19] is another multi-hop dataset consisting of yes-or-no questions that are implicitly phrased. It was designed to reduce reasoning shortcuts by framing multi-hop questions without direct references, such as "Did Aristotle use a laptop?". The questions in the dataset were created with a focus on their feasibility and having definite answers. Additionally, StrategyQA is the first dataset to include annotated decompositions that relate to specific paragraphs.

FEVER [20], and FEVEROUS [21] emerged, necessitating data extraction from vast sources like Wikipedia. Some attempts with single-hop models were unsuccessful. To address multi-hop question answering, different approaches were explored. For instance, Team Papelo [22] created a next-hop prediction module in FEVEROUS to retrieve related documents in multiple hops. Similarly, Zang et al. [23] created an iterative document retriever by reranking the documents after each hop. Other approaches involved using graph reasoning [24] or graph networks [25] to map related documents. These approaches leverage the interconnectedness of information in order to answer complex questions. Additionally, some models [26] utilized question decomposition to break down main questions into smaller ones, which can be solved by single-hop question-answering methodologies.

The advancements in language model based approaches have made significant progress in solving multi-hop question-answering tasks. However, some of this progress has been achieved by employing reasoning shortcuts or unintentional data leakage. The MuSiQue dataset [1] is a recent development designed explicitly to prevent these issues. It adopts a bottom-up approach, utilizing various single-hop datasets to construct a directed acyclic graph. Carefully crafted questions are formulated based on this graph. For each question, a total of 20 paragraphs are provided, containing both gold paragraphs and distractor paragraphs. Moreover, to prevent data leakage between the two sets, the train-test split is performed by considering underlying single-hop questions, making this a highly challenging dataset due to its careful construction.

2.3 Question Decompositon

Question decomposition is the act of breaking down questions into smaller pieces, which are referred to as sub-questions. This process is particularly important for solving multi-hop questions. Dividing the main question into multiple sub-questions enables models to comprehend the problems more effectively, discouraging the use of shortcuts and encouraging the application of multi-hop reasoning for problem-solving. Xie et al. [27] found that question decomposition is a useful approach for interpreting question-answering models. However, they also pointed out that the question decomposition model has not matured sufficiently to probe these models effectively.

Patel et al. [28] propose a human-in-the-loop approach to solve questions which are first given to humans to decompose into sub-questions. Then, sub-questions are given to the model, including the final answers. Although this method shows promising results due to human intervention in decomposition, it creates a scaling problem due to this necessity. Hence, the authors only solved 50 randomly selected questions for each dataset. Some of the models utilized unsupervised or distant supervision to solve decomposition problems. Khot et al. [29] proposed the Text Modular Network and Modular QA architecture, in which models create sub-questions with the help of the context and distant supervision hints. After creating this decomposition, a question is divided into five sub-categories: difference, comparison, complementation, composition, and conjunction. For each sub-category, specific rules are set to solve that specific type of question. In this way, their models are able to work in conjunction with small math models like difference. Perez et al. [30], on the other hand, created a list of candidate single-hop questions to match multi-hop questions and then solved the main questions by concatenating sub-questions together. Min et al. [26] proposed the DecompRC model, which utilizes span detection and Bridge and Comparison reasoning types to find sub-questions. This approach is motivated by the fact that decomposition with the help of humans is costly.

Some models like DecomP [3] utilize prompt engineering to decompose the questions. This model utilizes vast knowledge of the LLMs to solve the decomposition problems. This model is explained in detail under the prompt engineering subsection.

2.4 Prompt Engineering

Prompt engineering is a recently emerging area that deals with LLMs and few-shot learning, optimizing prompts for different tasks and providing these prompts to large models to obtain the desired answers. Unlike traditional supervised learning, where the model needs to be fine-tuned to achieve success, in prompting, there is a need to give some questions or sentences with an empty space inside it.[31]

Manual prompts are typically crafted by humans, often requiring multiple attempts and adjustments to get the format right. Creating these prompts demands expertise in the subject and can be time-consuming [31]. In contrast, automatically generated prompts can sometimes use unnatural language, especially when directly optimized. [5].

Petroni et al.[32] explore using language models as knowledge bases by creating the LAMA dataset, which provides templates for probing them. Brown et al. [13] design manual prefixes to run their GPT-3 model for various NLP tasks. Meanwhile, Schick and Schütze craft prompting templates for text generation and classification in several studies [33] [34]. Schick and Schutze [35] and Schick et al. [36] search predefined prompt temples and find the labels for them.

Gao et al.[4] propose a method to find the optimal prompt format by first selecting label words and then converting them into a prompt format. It utilizes RoBERTa [9] model to fine-tune for prompting.

There are several ways to utilize the prompting methodology with LLMs. One approach is to fine-tune the models using structured prompts. In this methodology, models are trained using structured prompts as inputs from the training set. When validation questions are presented in the same format, answers can be obtained from the model.

A prompt can be run in a few-shot or zero-shot format where there is no fine-tuning required. In zero-shot, only the question to be answered is asked, and the model returns an answer. On the other hand, in few-shot prompting, example questions and answers to solve the main question are carefully selected. These example questions provide the model with guidance on how to solve the problem. These approaches work due to innate connections inside the LLMs. When an instruction is given, the model calls related neurons to return the appropriate responses. This can be achieved with either long instructions or short commands like "Do this". These instructions guide the model's reasoning to achieve results without any fine-tuning. The process is also known as context learning. Dong et al. [37] highlight that this method offers an easy-to-understand interface. It learns in a way similar to humans and does not require any training.

In few-shot prompting, the structure of the prompt is important. There are numerous different elements in the prompts, such as the question, answer, and context. The order of these elements, as well as the order of the question examples, are crucial factors to be considered. Lazaridou et al. [38] demonstrate that placing context in the middle, to bridge the distance between the question and answer decreases the scores of the model. This is attributed to the problem of integrating long context in language models. Similarly, Zhao et al. [6] explore the effects of examples in the few-shot format. They state that language models have a bias towards the most recent example and towards repeated tokens. They also demonstrate that increasing the number of examples does not always yield better results and can sometimes produce worse outcomes.

Another type of prompting is the chain of thought prompting, which generates reasoning steps to find the actual answer instead of creating the answer directly. This way, the model has time to process and reach a reasonable conclusion. It also allows the model to be more interpretable. Wei et al. [39] show that chain of thought prompting outperforms normal prompting methods.

Dua et al. [40] developed an approach to decompose questions using successive prompting. This can be done through either in-context learning or fine-tuning the model. In their method, questions are first split into smaller ones using prompts. After solving these smaller questions, the answers are combined to produce the final solution. Therefore allowing them to separate tasks of decomposition and question answering from each other.

The DecomP model [3], proposed by Tushar Khot et al., is a very recent model that takes the prompting approach to an extreme. It creates prompts as a modular structure by dividing tasks into smaller subtasks and solving them that way. The main controller prompt consists of limited few-shot examples, and at every stage, it refers to other prompts using square brackets. When the main prompt encounters them, it calls the sub-function stated in square brackets and returns the solution from this prompt in the next line, running the prompts again. Prompting ends when the prompt returns 'EOQ' (end of question function). Furthermore, sub-functions can call other sub-functions, creating a hierarchical chain that returns the required solutions. The main advantage of this model is its modularity, allowing it to be adapted to almost any task with little tweaks. However, due to the modular approach, even the simplest tasks will require more than one prompt, increasing the computational cost.

2.5 Conclusion

The General NLP QA area is rapidly developing, with the emergence of larger and better models and approaches leading to improved solutions. This progress has paved the way for exploring new and interesting areas, such as multi-hop reasoning and decomposition. Consequently, more challenging datasets like MuSiQue have started to emerge to accommodate these advancements.

Prompt engineering has provided easy access to LLMs instead of traditional fine-tuning methodologies. This emerging approach has recently attracted the attention of researchers. As far as I know, there is only one research [3] that merges these concepts together, wherein the general focus is not on QA methods but rather on a general framework for mostly logical reasoning data. In their methodology, they introduce a controller prompt and several sub-function prompts. The controller prompt processes the primary question and invokes the sub-function prompts by referring to their tags enclosed in square brackets. Whenever a square bracket is detected in the controller, a new prompt is triggered with the relevant information. The results from this are then relayed back to the controller prompt, and the generation resumes. The process concludes once the end of the question tag is recognized.

This thesis will focus on creating a compact approach to solving multi-hop questions with a semi-automatic prompt-creating approach that utilizes unique characteristics of the MuSiQue dataset. The approach will use a prompting technique to solve questions with a chain of thought reasoning, allowing for the decomposition of complex questions.

CHAPTER 3

METHOD

This thesis proposes a semi-automatic prompt engineering method for solving multi-hop questions using question decomposition. There has been limited research on multi-hop reasoning using a prompt engineering approach. The most similar work [3] uses a controller and sub-functions framework, instead of relying on a single main prompt. Their approach resembles object-oriented programming. The controller prompt initially receives the main question and subsequently invokes single-hop question-answering prompts to address the sub-questions sequentially. These prompts can further call for other prompts to solve the problem. However, there are several differences between their approach and the method proposed in this thesis. Firstly, the proposed method uses context information to decompose the questions more accurately. Some questions are intrinsically created and rely on context for decomposition. However, in their approach, the context was only used during the answering phase, not the decomposition phase. Secondly, the main part of the proposed method consists of only one prompt, unlike the other approach, which uses five different prompts. This approach may reduce system modularity, but it significantly decreases complexity and computational costs. Lastly, the proposed model employs chain-of-thought prompting techniques to achieve superior results, instead of using one-word instructions that would call other prompts.

The proposed method consists of three components; pre-processing, prompt creation, and few-shot prompting. This section discusses each component and the thought processes behind their development.

3.1 The Scope of the Method

The training input data required for the method should include (1) the relevant content for the question e.g. the gold paragraphs with the correct answer to the question, (2) the main question to be answered, (3) exemplar sub-questions as they are important for the semi-automatic prompt generation, and enhancing the model's chain of thought, (4) the correct answer. An example of input data can be seen in Figure 1.

The model is designed to handle bridge questions, which involve selecting embedded questions within each other during the pre-processing step. In multi-hop reasoning, bridge questions are typically connected by a named entity or the answer to a sub-question. These sub-questions are merged with other sub-questions by replacing the named entity with a paraphrased version of the first sub-question. For instance, in Figure 1, named entity "#1" replaced by "the country where the film Duhulu Malak was

Gold paragraph 1: [Title: Duhulu Malak] (Omitted for clarity)
Gold paragraph 2: [Title: Sri Lanka national cricket team] (Omitted for clarity)

Main Question: When did the country where the film Duhulu Malak was produced win the World Cup?

Sub-question 1: Which was the country for Duhulu Malak?
A1: Sri Lanka
Sub-question 2: When did #1 win the world cup

Expected answer: 1996

Figure 1: An example data

Question: When was the institute that owned The Collegian founded?
Sub-question 1: The Collegian → owned by
Sub-question 2: When was #1 birthed?

Figure 2: An example for incomplete sub-question

produced" which is paraphrased version of the sub-question 1. While bridge questions can cover a range of topics, this thesis primarily focuses on general questions rather than logical reasoning.

The scope of the method is defined as follows to reduce the complexity and the computational cost:

- The method works with two-hop questions,
- The method needs relevant content such as gold paragraphs to be able to generate questions.

3.2 Pre-processing

Certain pre-processing methods should be applied to the dataset before applying the model. For instance, the dataset might not include uniformly structured questions, which may impair the performance of the model. Moreover, it might not satisfy the model prerequisites. Therefore, the following steps are conducted:

- Remove the questions if they do not include two hops
- Eliminate the questions containing sub-questions with incomplete sentences. The example in the Figure 2 shows the problem in sub-question 1.
- Select the main questions having a question inside the question. This is achieved by using regex. These questions have sub-questions in which a named entity is replaced by another sub-question. This process and its motivation are explained later in this section.

The first step in the pre-processing involves removing questions with three or more hops. This is because bridge questions with more than two hops are less common compared to those with two

Question: What city is the person who broadened the doctrine of philosophy of language from?
Sub-question 1: Who broadened the doctrine of philosophy of language
Sub-question 2: What city is #1 from?

Figure 3: Example question after pre-processing

Use minimal answers.
What is the named entity type of the following phrase?
Answer

Figure 4: Prompt used for named entity extraction

hops. Additionally, bridge questions with more than two hops can have different structures. For example, there are three sub-questions named as A, B, and C. In a three-hop question, the bridges could be between A and B, and B and C, or A and B directly bridged to C. Similarly, four or more hop questions have more possible bridge connection types. Therefore, only the two-hop questions are selected to simplify the later processes since they have only one possible connection type.

To further reduce the computational cost and create a uniformly distributed dataset, a subset of questions is selected by utilizing named entities inside the sub-questions. This process is done by replacing the named entity of the first sub-question (inside question) with “.*” and doing regex matching with the main question. Only the questions that match regex are kept in the dataset. Figure 3 shows an example question after pre-processing and regex matching between them. As can be seen, sub-question 2 includes the "What city" and "from?" phrases, which is also present in the main question.

3.3 Prompt Creation

Prompt creation is a two-step process. The first step is to cluster sub-question answers based on their named entities in the training dataset. In this way, different prompts can be created according to the named entity types. The problem of repetitive use of similar answers in the prompt makes the model biased towards certain answers and therefore degrades performance. This issue will be discussed in Chapter 4.

The answers are clustered with the help of ChatGPT. The prompt in Figure 4 is used to retrieve the named entity types of all the questions in the training dataset.

The first line of the prompt helps reduce the size of the output, and prevents ChatGPT from printing a long string of text. The second line specifies the rules of the selection of the named entity. The third line represents the answer to be extracted from the dataset. Once they are extracted, all the results are converted to lowercase, and punctuation is removed, such as transforming answers like “Location”, “location”, and “location” into one. Initially, there were 56 named entities for the inside question answers, and 151 for the main answers. However, the output included similar named entity categories,

Context1: [Title: (title1)] (context1)
Context2: [Title: (title2)] (context2)
Q: (main question)
Decompose Q to the maximum number of meaningful sub-questions. Use minimal answers and keep named entities, adjectives, adverbs when possible in your answers.
Q1: (sub-question1)
A1: (sub-question1 answer)
Q2: (sub-question2)
A2: (sub-question2 answer)

Figure 5: Main prompt

such as “brand” and “brand company”. To further reduce the number of named entities, ChatGPT is used to reduce the group by giving the list of named entities. The utilized prompt consists of a list containing all the named entities and their number of occurrences in the dataset, along with a line stating, ’merge similar named entities.’ The response returned a list of named entities grouped by their similarities. Finally, 25 named entities for the inside question answers and 67 for the main answers are obtained. Figure 8 in Appendix B shows the distribution of the named entities before and after merging.

The second step is the creation of the actual prompt itself. The main bulk of the prompt is selected by the questions from the training set utilizing the method above. The prompt utilized is shown in Figure 5.

Other prompting methods are also tested, showing that the best results were obtained using this particular format, which was documented as an ablation study in Section 4.5. During decomposition, certain questions require context to be present due to implicit wording in the main questions. Similarly, titles are necessary because they do not appear in the context. The following instruction is added to the prompt: "According to context above, answer the following question with the shortest answer".

Each prompt consists of six questions from the training set and one from the validation set. The question from the validation set is always positioned as the last question in the prompt. For validation, sub-questions and sub-question answers from the prompt are removed because these lines are expected to be generated by the model. Examples are selected from the training set, where questions are grouped by their answer named entities. After the prompt format is manually crafted, the model automatically chooses six examples to finalize the prompt.

3.4 Few-shot Prompting

After prompt generation, ChatGPT or GPT 3.5 is used for the prompts and to obtain the predictions. Open AI’s API provides significant computing power with the cost of 0.0015$ per 1000 input tokens and 0.002$ per 1000 output tokens (June 2023). Although it still has a considerable cost (using this model, running 200 prompts costs around 1$), compared to the other options, it is favorable considering the complexity of creating and debugging an LLM.

Another model, OPT (Open Pretrained Transformer), is also tested on local devices. A commercial computer with an Nvidia 3060 mobile with 6GB of memory could barely run a 1.3B parameter model. Another device with Nvidia TitanX can run a 2.7B parameter version. The problem is that smaller models generally have worse scores than the larger versions on prompting [13]. Another option was using TRUBA (Turkish National Science e-Infrastructure). The problem is that the large models require a lot of GPUs. Clusters with 16 GB V100 GPUs, OPT-175B requires 22 (https://alpa.ai/opt). It means almost reserving all the infrastructure, meaning waiting nearly a month in the queue to obtain the results (not including unexpected errors, which resets waiting time).

Comparing these two factors, ChatGPT selected for modeling purposes, which tends to give long and drawn-out answers. To solve it, the prompt contains a set of rules for the answers, shown in the last part. After obtaining the prompt results, the last line generated by the model is retrieved, which includes “A2” and extracts the results. The extracted result was compared to the actual results with Exact Match (EM) and F1 score metrics. These results are discussed in Chapter 4.

CHAPTER 4

EXPERIMENTS

This chapter presents the experiments to test the proposed model's performance, including sub-sections on Dataset, Experimental Setup, Compared Models, Results, and Ablation Study.

4.1 Dataset

In this thesis, the MuSiQue, a multi-hop question-answering dataset, is used [1]. It consists of 24,814 two to four-hop questions. Most of the questions fall into the 2-hop category, whereas the 3-hop and 4-hop questions constitute a much smaller portion of the data set due to their distinct structures.

The dataset is specifically designed in a bottom-up approach to minimize data leakage and disconnected reasoning. To achieve this, the dataset was thoroughly constructed with certain rules in mind. Each question in the dataset is accompanied by its decompositions. These sub-questions were obtained from five different single-hop datasets: SQuAD [2], T-REx [41], Natural Questions [42], MLQA [43], and Zero Shot RE [44]. The questions were then organized into a directed acyclic graph, and new multi-hop questions were generated by establishing links between questions using named entities.

The dataset was split into train, validation, and test sets in such a way that no single-hop question was shared between different sets. This approach prevents data leakage between the training and validation sets. Table 1 presents the breakdown of the examples in each set, categorized by the number of hops.

Table 1: Dataset statistics of MuSiQue-Ans

Hop Count	Train	Dev	Test	Total
2-hop	14376	1252	1271	16899
3-hop	4387	760	763	5910
4-hop	1175	405	425	2005
Total	19938	2417	2459	24814

The MuSiQue dataset offers two options: answerable and full. The answerable option includes only the questions that can be answered, along with their corresponding context. The full option includes answerable questions and the same questions with one of the required contexts removed. For example, the full dataset contains both the example shown in Figure 1 and another version with gold paragraph 2 removed.

For this thesis, the MuSiQue-answerable option is utilized as the gold paragraphs are needed as relevant context for the method. Using the full dataset requires a proper method to find relevant documents, but the proposed model lacks this capability. Unanswerable questions in the MuSiQue full rely on this document finding process.

4.2 Experimental Setup

The experiments conducted in July 2023 used the OpenAI API with the model "gpt-3.5-turbo" or ChatGPT. Few-shot prompting examples were pre-processed using the steps from Chapter 3. A total of 10 example questions were selected to test the proposed model. Six of these examples were incorporated into the proposed model, while the remaining examples were employed for ablation studies that require more than six examples only. The examples for few-shot prompting and the main prompt format can be found in Appendix A.2.

4.3 Compared Models

To ensure a fair comparison of the proposed model, two baseline models with different prompting approaches and incorporated four additional baseline models [1] specifically designed for the MuSiQue dataset are utilized. One baseline approach utilizes the few-shot method, where only questions and answers are provided. Another approach employs a zero-shot methodology. On the other hand, baselines from the MuSiQue dataset use fine-tuning techniques with varying strategies. The following subsections provide a detailed description of these models.

4.3.1 Baseline 1

The first baseline model's prompts consist of questions, sub-questions, and answers without context or chain-of-thought reasoning. Figure 6 shows the prompting format of the first baseline model. The questions used as examples for n-shot prompting are identical to those tested with the proposed model, ensuring a fair comparison. The order of the example questions was kept the same, with only the context, and the instruction stating the generation rules. The full prompt can be seen in the Appendix A.1.

4.3.2 Baseline 2

In the second baseline approach, ChatGPT is used to answer questions directly in a zero-shot format without providing any examples or decomposition. A chain-of-thought reasoning step with additional instructions is added to get more concise answers. Figure 7 shows the prompt format of the second baseline model.

Q: (Example question)
Q1: (sub-question1)
A1: (sub-question1 answer)
Q2: (sub-question2)
A2: (sub-question2 answer)

.
.
.

Q: (Example question)
Q1: (sub-question1)
A1: (sub-question1 answer)
Q2: (sub-question2)
A2: (sub-question2 answer)

Q: (main question)

Figure 6: Baseline1 prompt format

Context1:[Title: (title1)](context1)
Context2:[Title: (title2)] (context2)
According to context above, answer the following question with the shortest answer:
Q: (main question)

Figure 7: Baseline2 prompt format

4.3.3 State of the Art Models

The MuSiQue dataset also incorporates four baseline models for comparison purposes, as described in Chapter 2. The first model is an end-to-end approach, which takes context and questions as input and directly provides answers as output. The second model, known as the "select and answer" model, operates in two steps: it first selects relevant context paragraphs and then answers the question based on the selected paragraphs. Both of these models are fine-tuned using Longformer [10] and the selection mechanism of the second model utilizes RoBerta model [9].

The other two models are the step execution models. They start by creating a directed acyclic graph composed of decompositions. This is achieved by training with BART-Large [12] on gold decompositions. Subsequently, the first two models utilize these decompositions to generate answers.

Please note that these models utilize paragraph retrieval, either adding extra steps to extract gold paragraphs or using all paragraphs from a dataset where each question has a total of 20 paragraphs. In this thesis, gold paragraphs are used directly due to the token limit of the GPT model. Moreover, using gold paragraphs simplifies the problem by focusing on decomposition as the main objective.

4.4 Results

As explained above, six different baseline models are selected for comparison with the proposed model. The first comparison was made with the models published by the authors of the dataset. On their GitHub page, validation predictions for each model are available. To obtain predictions for the subset used in this thesis, the predictions are extracted by matching question IDs between the subset and the predictions. The results obtained for each of the four models are presented in Table 2. It is worth mentioning that the F1 scores were 0.02 higher than the results on the whole dataset.

Table 2: Evaluation results

Model	**EM**	**F1**
dev_end2end_model	0.3385	0.4355
select_answer_model	0.3906	0.4853
step_execution_by_end2end_model	0.4167	0.4652
step_execution_by_select_answer_model	**0.4375**	**0.5158**

The proposed method explained in Chapter 3 is used for testing its performance. After preprocessing the dataset and obtaining the subset, six example questions are extracted for n-shot prompting from the training set. In this process, after identifying named entities from answers to the first and second sub-questions for the whole training subset, a random question is selected. Then, any other questions containing the same named entities are removed from the selection pool. This procedure is repeated until six examples are extracted for prompting. Questions are then put into the proposed prompt format. The prompt can be found in Appendix A.2. After creating the n-shot part of the prompt, a question from the validation set is merged by utilizing the same format except for sub-questions and answers, which are expected to be generated by ChatGPT.

The comparisons are carried out using the exact match and F1 scores. Additionally, the answers that ChatGPT did not provide are collected by filtering the outputs for negative words such as 'no', 'not', and 'unknown'. This was necessary because ChatGPT tends to provide "do not know" answers.

Table 3: Evaluation results: baseline1, baseline2, and proposed model

Model	**Exact Match (EM)**	**F1 Score**	**"Do not know" Predictions**
Baseline1 (Run1)	0.146	0.238	40
Baseline1 (Run2)	0.130	0.233	37
Baseline1 (Run3)	0.161	0.270	34
Baseline2 (Run1)	0.448	0.581	5
Baseline2 (Run2)	0.448	0.588	5
Baseline2 (Run3)	0.471	0.584	5
Proposed Model(Run1)	**0.620**	**0.732**	5
Proposed Model(Run2)	0.619	0.725	**2**
Proposed Model(Run3)	**0.620**	0.728	3

The results are presented in Table 3. Due to inconsistencies in the performance of ChatGPT, three different results are collected for each model with the same selected examples. Among the three models, the "Baseline 1" performed the poorest. This approach relies heavily on the internalized knowledge of the language model to solve the problem. It achieved an average exact match score of 0.15 and an F1 score of 0.25. The "Baseline 1" model provided more "do not know" answers, indicating that the model generated answers without any knowledge of the contexts.

The second baseline model demonstrated comparable performance to the proposed model, although with a slight 0.17 margin lower in both the exact match and F1 scores. During the experiments, it is observed that this approach can generate correct responses if allowed to provide long answers. However, when employing the chain of thought reasoning to limit the output length, it tended to select incorrect outputs.

Lastly, the proposed model outperformed the other baseline models, achieving an average exact match score of 0.62 and an F1 score of 0.73. Although the presence of "do not know" answers decreased overall, this could be attributed to the context and example questions. The proposed model demonstrated superior performance compared to the baselines.

4.5 Ablation Study

During the development of the proposed model, several key decisions were made. To demonstrate the effectiveness of these decisions, six different ablation studies were conducted to analyze the effects of each decision. These decisions have a direct impact on the model's prompt, which in turn influences the results generated by the model. The prompts for the ablation study were designed by selectively removing or altering specific sections of the original prompt. Each subsection of the study provides an explanation of a decision step and its respective effects on the actual model.

4.5.1 N-shot Number of Example Questions

Zao et al. [6] state the importance of the structure of the prompt in a few-shot format, which has a tendency to high variance due to selected examples. This variance is caused by models copying hidden correlations between example questions, even down to small details such as the second character of the answers or the number of words in the sub-questions. To mitigate these hidden correlations, increasing the number of examples can be helpful, but it introduces challenges, such as lengthy prompt size. This increase in prompt size not only escalates computational costs, but also poses issues with the input size limitations of the model. The default model of ChatGPT supports up to 4k input tokens, and accommodating varying context sizes requires a flexible input size.

Table 4 displays various example sizes and their corresponding results on the utilized subset. Additional examples were chosen from the dataset to increase the example size, following the same method of selecting them based on different answer-named entities. The model is tested with up to 10 examples; however, the input size limitation of 4k tokens prevented me from running more than seven examples. The results show that an increase in the example size generally leads to improved performance, except for the case of 4 example prompts. The drop in performance could be attributed to two factors: random chance or a hidden correlation between the fourth example and the first three. The models with 6 and 7 few-shot examples demonstrated the best performance in terms of exact match, while the 7-example solution yielded the highest score in the F1 category.

The proposed model uses six examples for two reasons: First, no notable improvements are observed when the example size is increased to seven. Second, selecting six example models provides the prompt with a margin for error, allowing it to accommodate examples with larger contexts.

Table 4: The first ablation study shows the performance of the model with respect to the different number of examples in the prompt

Model	Exact Match (EM)	F1 Score	"Do not know" Predictions
Model 1 Example	0.526	0.632	6
Model 2 Example	0.554	0.689	7
Model 3 Example	0.583	0.707	8
Model 4 Example	0.563	0.683	3
Model 5 Example	0.604	0.720	3
Model 6 Example (Proposed, Average)	**0.620**	0.728	3
Model 7 Example	**0.620**	**0.752**	**2**

4.5.2 The Impact of Including the Context

Another thing to consider is the impact of adding context to the prompt. LLMs are generally believed to contain vast knowledge and can easily answer common questions without any context. This means that adding context could have minimal effects due to question decomposition. To test this, a new prompt is created by removing context from the main prompt. This new prompt consists of six examples, each containing a main question, instructions, sub-questions, and sub-question answers. This prompt is a similar model except for the instructions, which provide a chain of thought reasoning similar to the proposed model.

As shown in Table 6, the result demonstrates a significant performance loss in terms of exact match and F1 scores, as well as more negative predictions. Compared to the first baseline, it has fewer negative predictions but yields similar results. This means that adding the instructions alone allows the model to answer more questions, although incorrectly.

4.5.2.1 The Impact of Inaccurate Selection of the Context

After examining the system's performance without any context, the impact of introducing random context is explored further. To do this, two random contexts are selected from the dataset for each question in the validation set while keeping the example contexts the same as in the proposed model.

The results in Table 6 indicate that its effects are the worst among all the ablations conducted, even compared to not using any context. This suggests that using incorrect context impacts the performance significantly.

4.5.2.2 The Impact of the Context on Decomposition

The impact of using context during decomposition is assessed by calculating F1 scores for each sub-question. Table 5 presents results for inner (sub-question1) and outer questions(sub-question2). Some question decompositions include only one sub-question. This is due to missing context or because the question is directly answered by the first sub-question. The Table provides F1 scores both with and without considering these questions.

The results show that outer questions have a higher overall F1 score than inner questions. This is expected because, during the preprocessing step, the structure of the sub-question is retained in the main question. Whether answers from one sub-question are included or excluded, the F1 scores for inner questions remain close to each other. However, the score for outer questions increases when the exclusion of the F1 score calculation occurs. Furthermore, adding context during the decomposition process slightly improves the performance of the decomposed question. At the same time, it significantly increases the answer performance.

Table 5: Results sub-question F1 scores according to context

Model	**Sub-Question1 F1**	**Sub-Question2 F1**	**One Sub-Question Questions Removed**
with incorrect context	0.644	0.727	-
without context	0.690	**0.858**	-
Proposed (average)	**0.692**	0.840	-
with incorrect context	0.646	0.801	18
without context	0.690	0.858	**0**
Proposed (average)	**0.693**	**0.880**	9

Table 6: Results of ablation studies with modified prompts

Model	Exact Match (EM)	F1 Score	"Do not know" Predictions
with incorrect context	0.099	0.130	55
without context	0.135	0.234	14
without chain of thought	0.500	0.662	10
with additional instruction	0.505	0.629	10
without decomposition	0.542	0.695	**2**
same named entity	0.542	0.689	6
changed order of examples	0.600	0.714	7
Proposed	**0.620**	**0.728**	3

4.5.3 The Impact of Chain of Thought

The model presented in this thesis utilizes chain-of-thought reasoning by incorporating instructions, as demonstrated in Chapter 3. It is proposed that adding this reasoning can improve the model's overall reasoning capabilities. However, by increasing the prompt size with unnecessary tokens, the model's performance may also be negatively affected. To assess the impact of this reasoning approach, two tests are conducted. The first test involved removing the instructions entirely, while the second test increased the word count by adding more words.

4.5.3.1 Removal of the Chain of Thought

In the first test, the instructions from the prompt are removed. Hence, the new prompt consists of contexts, main questions, sub-questions, and sub-question answers. The same six examples are used for a fair comparison.

As shown in Table 6, without the context, the results yielded an exact match (EM) score of 0.5 and an F1 score of 0.662. Additionally, there were 10 question responses containing unknown results. This represents a decrease of 0.12 points in the EM score and 0.5 points in the F1 score compared to the proposed model. The significant decrease in the EM score compared to the F1 score is related to the instructions, which states the rules for selecting the answers.

4.5.3.2 Effect of Length of the Chain of Thought

The other side of the scale is increasing the instructions. For this purpose, ChatGPT is asked to bloat the instructions. The following new instructions are obtained;

> In order to achieve comprehensive decomposition, ensure that sentence Q is fragmented into an extensive assortment of profoundly meaningful sub-questions. Employ succinct answers that retain named entities, adjectives, and adverbs whenever feasible, thereby optimizing the process.

This new line replaced the original instructions in the prompt and obtained a new prompt consisting of contexts, main questions, sub-questions, and sub-questions answers.

As shown in Table 6, without the context, the results yielded an exact match (EM) score of 0.505 and an F1 score of 0.669. Additionally, there were 10 question responses containing unknown results. The results are very similar to the prompt without context. From the results, it can be deduced that bloating the prompt with an increased number of unnecessary tokens creates complexity for the model. Moreover, increased tokens create distance between the main question and the answers, causing the model to miss strong reasoning relations between them.

Based on both results, it can be deduced that when adding a chain of thought reasoning to a prompt model, careful consideration must be given. These lines should be concise to avoid creating distance between the related elements.

4.5.4 Effectiveness of Decomposition

To investigate the impact of decomposition on multi-hop questions, the main focus of this thesis revolves around the question of decomposing them. In order to observe the effects of decomposition, a new prompt was created by removing the sub-questions and their corresponding answers. Instead, the final solution was substituted in their place. As a result, the prompt now comprises contexts, main questions, instructions, and the answer.

As shown in Table 6, the model without the decomposition component exhibits a significant drop in the EM (exact match) score, while experiencing only a slight decrease in the F1 score. This drop can be attributed to the model producing additional artifacts instead of solely returning the answers. When compared to the second baseline model without examples, this model performs better. However, with only the answers provided, the model fails to identify the required solution structure correctly.

4.5.5 Effectiveness of Named Entity Grouping

The last ablation study examines the effects of selecting different named entities during the prompt creation process to demonstrate their effectiveness. For this purpose, six examples are chosen with the same named entities, where the answers to the first sub-questions were selected as locations, and the answers to the second sub-questions were chosen as numbers. Apart from that, the same prompt structure is maintained.

In Table 6, it can be observed that utilizing the same named entities resulted in worse performance compared to the proposed model, with an EM score of 0.521 and an F1 score of 0.665. The hidden correlations between examples can arise from selecting the same structure, evidently contributing to the decrease in performance. This correlation, in turn, leads to a loss in performance.

4.5.6 Order of the Examples

The problem of recency bias arises when creating prompts, as models tend to establish stronger connections between related texts that are close to each other. This applies to various elements of questions, such as context, sub-questions, main question, and answer, as well as between examples used for few-shot prompting [38]. To demonstrate this effect, the order of the examples in the model is rearranged.

To achieve this, a new prompt is created by swapping the places of questions in the first and fourth places with the last and third places. The new prompt was then put to the test again utilizing ChatGPT, and the results can be seen in Table 6. The obtained results are very close to the proposed model, with a slight reduction in EM and F1 scores.

CHAPTER 5

DISCUSSION AND FUTURE WORK

In this chapter, the results related to the research questions, followed by an exploration of potential future work, will be discussed.

5.1 Discussions

The results from the previous chapter show that the proposed model has an advantage over the other prompting techniques that do not include parts of the proposed model.

RQ1: How can prompt engineering and decomposition techniques be effectively utilized to address the challenge of multi-hop question-answering in the 2-hop questions using a single prompt?

The proposed model demonstrates how to integrate prompt engineering and decomposition using the MuSiQue dataset and GPT-3.5. The results indicate that using decomposition in a prompt combined with a chain of thought reasoning yields better performance than using without it. This chain of thought approach enhances the model's scores by allowing it to think and reason through multiple steps, ultimately leading to a more informed conclusion.

The proposed model is designed and tested for 2-hop questions. This can be extended by either adding multi-hop questions with more than two hops or using a broader subset of questions. In the broader subset approach, it is important to select questions carefully. This means having an equal number of examples from different question types while still grouping them by named entities as in the original model. For questions with more than two hops, new examples should be added. The main challenge is that questions with three or more hops can have varied connections between their sub-questions. It is essential to provide examples from all these connections in the few-shot samples for the model to handle such questions correctly.

The proposed model can be extended by retrieving documents between gold and distractor paragraphs. This can involve adding a new prompt before the current model, which will determine the appropriate context for the questions. This selection can be done using a zero-shot approach or by including one or two examples in a few-shot approach. The method can also benefit from the conventional capabilities of ChatGPT. Instead of sending just one prompt to the model and using the given answer, one could extend the conversation with the OpenAI API by adding more instructions before or after the initial method. While this can offer more flexibility, it could also raise the computational costs significantly due to repeated prompt submissions.

In the ablation study, six different versions of the proposed model are tested by removing parts of the prompt. The study showed that removing the context and giving incorrect context led to the most significant performance drop. On the other hand, changing the order of the questions resulted in the least performance drop. These can be attributed to the method's reliance on the context and the named entity grouping of different questions respectively. Using models other than ChatGPT might yield different results, especially if the new language model is trained on the MuSiQue dataset or datasets from which it sources sub-questions. The results of the ablation study could also vary depending on the internal reasoning of other models.

RQ2: Does incorporating context in the decomposition process lead to improved outcomes compared to decomposing without context when utilizing prompting techniques?

In Chapter 4's ablation study, the context is removed from the prompt to highlight the difference between its inclusion and exclusion. the effects of using gold paragraphs are tested against random contexts selected for each question. Finally, the F1 score for sub-questions is calculated to gauge the influence of context on the generated sub-questions.

The results clearly show that excluding the context leads to a significant drop in performance when answering questions. However, sub-question F1 scores indicate a slight increase in performance. When context is provided during decomposition in the proposed method, the model can directly answer the questions instead of creating meaningful questions. This happens either because the model has knowledge without needing the context or the context helps the model find a reasoning shortcut. On the other hand, using the wrong context results in lower F1 scores compared to both the no-context and proposed methods. This suggests that selecting the wrong context hinders the decomposition of questions.

RQ3: When creating prompts, does clustering sub-question answers using named entity recognition lead to better outcomes compared to selecting the same named entities for examples for few-shot prompting?

The proposed model processes the answers to sub-questions provided in the dataset with respect to the named entity and selects a set of questions where the sub-questions differ from each other. On the other hand, the ablation study demonstrates that the worst possible outcome is observed when examples are randomly sampled for few-shot prompting by selecting the same named entities for sub-questions. The results show reduced performance compared to the proposed model. From this, it can be deduced that clustering the answers to sub-questions and using them in the few shot prompting yields better outcomes.

5.2 Future Work

Several potential future works can stem from this research. First and foremost, the proposed prompting model can be modularized to enhance its effectiveness. Instead of using a single large prompt to process the data, the prompt could be divided into several smaller prompts, one for decomposition and another for answering sub-questions. The first prompt breaks down questions into sub-questions using context. When a sub-question is identified, a new prompt is called to address it. The answer from this new prompt is then fed back to the original model to continue the decomposition. Once the final answer is reached, the prompting stops. This strategy would enable the improvement of the method by adding additional modules for processing sub-questions.

Another way to improve the proposed model is through the selection of 'gold' paragraphs. Instead of solely relying on these gold paragraphs, a document retrieval approach can be employed to retrieve the most relevant context automatically. This adaptation will make the model more versatile, enabling its application across various datasets and a wider range of cases. This approach can also be combined with modularity to enhance the solution.

The proposed model can also be extended to handle a diverse set of question types. The current method filters out similar two-hop questions by focusing on sub-questions within the main question. It does this by replacing a named entity with a sub-question. The method only selects questions where the surrounding context remains identical on a token basis. This approach can be improved to consider questions where the surrounding context is paraphrased. Additionally, the selection could be extended to include questions where sub-questions are intertwined. When creating the prompt, it is important to include all question types in several step-by-step examples while maintaining a balanced representation of each type. Alternatively, a universal prompt can be developed to address various question types, which include more than two-hop questions. Similarly, examples for each type of question need to be present to ensure the model's success.

The model could be improved to accommodate questions with more than two hops. These types of questions contain different connections between sub-questions within the same hop. Taking this into consideration, a new prompting methodology could be developed to extend the range of the prompts.

One of the ways to improve the approach is by using advanced LLMs for prompting. In prompt engineering, larger models typically perform better than smaller ones because of their high reasoning capabilities. However, smaller models can excel when they are carefully fine-tuned. One option is to fine-tune a model of similar size, or another is to use a newer model like GPT-4 or GPT-3.5 with a 16k token input. Whichever is chosen, these models will need more computational power due to their bigger size or the additional fine-tuning process.

CHAPTER 6

CONCLUSION AND LIMITATIONS

This thesis offers a guideline for using large language models to solve two-hop questions through prompting methodology and decomposition. The research evaluates the effectiveness of prompting models in decomposing and answering multi-hop questions. It also explores the impact of example selection in few-shot prompting techniques. Moreover, the research analyzes the consequences of the design choices in few-shot prompting. This study utilizes the MuSiQue dataset, which includes question decompositions due to its bottom-up creation method. A subset of this dataset is used, excluding questions with more than two hops or decompositions with two words.

The study examines the use of the ChatGPT model for few-shot prompting. Initially, the model clusters questions based on named entities found in their sub-question answers using a zero-shot prompting approach. After this clustering, the clusters are combined into more generalized versions. Example questions are then selected, ensuring no overlapping named entities. The prompts are structured as follows: Contexts -> main question -> sub-questions with answers. To reduce the variability in results due to the prompting approach, the model is run three times. For comparison, two baseline prompting models are also run three times each to minimize the effect of randomness. One of these baseline models operates in a zero-shot format, while the other runs without any context or instructions. Four additional baseline models, which are fine-tuned rather than prompted, are sourced from the dataset paper. The proposed model outperforms all baseline models.

Six ablation studies were conducted to evaluate the impact of various decisions made during prompt creation: the number of example questions, context, chain of thought, decomposition, named entity grouping, and order of examples. Results indicate that removing the context or using the wrong context results in the most significant performance drop, as the model becomes unaware of how to answer the questions. Conversely, altering the order of the examples has a minimal effect on performance. Though changing the order can initially decrease performance due to named entity selection during prompt creation, this effect diminishes over time. As indicated in the Zhao et al.[6], the number of examples has varying effects on few-shot prompts. In this study, the ablation shows the effects of example prompts, with ChatGPT able to accommodate up to seven examples. There is a noted performance increase with additional examples, but the marginal gain diminishes as the number of examples increases.

The primary constraint of this thesis is the computational cost associated with LLMs. To mitigate this, most aspects of the proposed model have been refined. This necessity primarily stems from OpenAI's API policy that measures cost based on the count of input and output tokens. Although the cost per token is relatively small, the prompting method requires a high number of input tokens per prompt, substantially increasing the overall cost.

In order to reduce the overall computational cost, several key decisions were made during the creation process of the proposed model. The first of these decisions was to utilize only gold paragraphs. The inclusion of misleading paragraphs would have increased the context size tenfold. Therefore, the proposed model only employs gold paragraphs. Similarly, the proposed model uses only 2-hop questions to reduce the complexity of the prompts as well as the reduction in the context size.

This thesis's final strategy for reducing computational cost is to work with a subset of the dataset. This subset consists of approximately 2000 training examples and around 200 validation examples. The proposed model is only run on the validation set. The overall cost is around 1 $ per run of the validation set.

A limitation of the model in use is LLM's 4000-token limit, which restricts the space available for examples in few-shot prompting. As a result, there is a trade-off between the number of contexts for an example and the number of examples themselves. To accommodate this limitation, the proposed model selects two contexts (gold paragraphs) and includes six examples.

The proposed models reliance on ChatGPT is a limitation. As mentioned in the discussions and future work section, using different LLMs can present various challenges. If a smaller LLM like BERT is used, it might save on computational costs but might sacrifice performance and input token size. Conversely, a newer or larger model might deliver better results and handle more input tokens, but at a much higher computational cost. When testing different models, it is essential to balance computational cost against performance. The ideal model should be selected based on both performance requirements and computational constraints. While using more examples or a larger subset can enhance the model's performance, it may also significantly increase computing costs.

Another limitation of the proposed model is that it requires pre-decomposed questions for semi-automatic prompt creation. Since the decomposition of questions using a language model is not yet a mature topic, one either needs datasets with available decompositions or the decompositions must be created by humans.

REFERENCES

[1] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "♪ MuSiQue: Multihop questions via single-hop question composition," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 539–554, 2022.

[2] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 2383–2392, Association for Computational Linguistics, Nov. 2016.

[3] T. Khot, H. Trivedi, M. Finlayson, Y. Fu, K. Richardson, P. Clark, and A. Sabharwal, "Decomposed prompting: A modular approach for solving complex tasks," *arXiv preprint arXiv:2210.02406*, 2022.

[4] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Online), pp. 3816–3830, Association for Computational Linguistics, Aug. 2021.

[5] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, "How Can We Know What Language Models Know?," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 423–438, 07 2020.

[6] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," in *International Conference on Machine Learning*, pp. 12697–12706, PMLR, 2021.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. 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)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ArXiv*, vol. abs/1907.11692, 2019.

[10] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv:2004.05150*, 2020.

[11] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018. Accessed: 2023-08-17.

[12] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 7871–7880, Association for Computational Linguistics, July 2020.

[13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 1877–1901, Curran Associates, Inc., 2020.

[14] OpenAI, "OpenAI: Introducing chatgpt." `https://openai.com/blog/chatgpt`, 2022. Accessed: 2023-08-17.

[15] OpenAI, "Gpt-4 technical report," *ArXiv*, vol. abs/2303.08774, 2023.

[16] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, *et al.*, "Opt: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022.

[17] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. 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*, (Brussels, Belgium), pp. 2369–2380, Association for Computational Linguistics, Oct.-Nov. 2018.

[18] X. Ho, A.-K. Duong Nguyen, S. Sugawara, and A. Aizawa, "Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps," in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)), pp. 6609–6625, International Committee on Computational Linguistics, Dec. 2020.

[19] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant, "Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 346–361, 04 2021.

[20] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: a large-scale dataset for fact extraction and VERification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 809–819, Association for Computational Linguistics, June 2018.

[21] R. Aly, Z. Guo, M. S. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, and A. Mittal, "The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task," in *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, (Dominican Republic), pp. 1–13, Association for Computational Linguistics, Nov. 2021.

[22] C. Malon, “Team papelo at FEVEROUS: Multi-hop evidence pursuit,” in *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, (Dominican Republic), pp. 40–49, Association for Computational Linguistics, Nov. 2021.

[23] Y. Zhang, P. Nie, A. Ramamurthy, and L. Song, “Answering any-hop open-domain questions with iterative document reranking,” in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’21, (New York, NY, USA), p. 481–490, Association for Computing Machinery, 2021.

[24] Y. Li, W. Li, and L. Nie, “Dynamic graph reasoning for conversational open-domain question answering,” *ACM Trans. Inf. Syst.*, vol. 40, jan 2022.

[25] N. Kotonya, T. Spooner, D. Magazzeni, and F. Toni, “Graph reasoning with context-aware linearization for interpretable fact extraction and verification,” in *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, (Dominican Republic), pp. 21–30, Association for Computational Linguistics, Nov. 2021.

[26] S. Min, V. Zhong, L. Zettlemoyer, and H. Hajishirzi, “Multi-hop reading comprehension through question decomposition and rescoring,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 6097–6109, Association for Computational Linguistics, July 2019.

[27] K. Xie, S. Wiegreffe, and M. Riedl, “Calibrating trust of multi-hop question answering systems with decompositional probes,” in *Findings of the Association for Computational Linguistics: EMNLP 2022*, (Abu Dhabi, United Arab Emirates), pp. 2888–2902, Association for Computational Linguistics, Dec. 2022.

[28] P. Patel, S. Mishra, M. Parmar, and C. Baral, “Is a question decomposition unit all we need?,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, (Abu Dhabi, United Arab Emirates), pp. 4553–4569, Association for Computational Linguistics, Dec. 2022.

[29] T. Khot, D. Khashabi, K. Richardson, P. Clark, and A. Sabharwal, “Text modular networks: Learning to decompose tasks in the language of existing models,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Online), pp. 1264–1279, Association for Computational Linguistics, June 2021.

[30] E. Perez, P. Lewis, W.-t. Yih, K. Cho, and D. Kiela, “Unsupervised question decomposition for question answering,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 8864–8880, Association for Computational Linguistics, Nov. 2020.

[31] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing,” *ACM Comput. Surv.*, vol. 55, jan 2023.

[32] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, “Language models as knowledge bases?,” 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), (Hong Kong, China), pp. 2463–2473, Association for Computational Linguistics, Nov. 2019.

[33] T. Schick and H. Schütze, "Few-shot text generation with natural language instructions," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, (Online and Punta Cana, Dominican Republic), pp. 390–402, Association for Computational Linguistics, Nov. 2021.

[34] T. Schick and H. Schütze, "It's not just size that matters: Small language models are also few-shot learners," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Online), pp. 2339–2352, Association for Computational Linguistics, June 2021.

[35] T. Schick and H. Schütze, "Exploiting cloze-questions for few-shot text classification and natural language inference," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, (Online), pp. 255–269, Association for Computational Linguistics, Apr. 2021.

[36] T. Schick, H. Schmid, and H. Schütze, "Automatically identifying words that can serve as labels for few-shot text classification," in *Proceedings of the 28th International Conference on Computational Linguistics*, (Barcelona, Spain (Online)), pp. 5569–5578, International Committee on Computational Linguistics, Dec. 2020.

[37] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui, "A survey on in-context learning," *arXiv preprint arXiv:2301.00234*, 2023.

[38] A. Lazaridou, E. Gribovskaya, W. Stokowiec, and N. Grigorev, "Internet-augmented language models through few-shot prompting for open-domain question answering," *arXiv preprint arXiv:2203.05115*, 2022.

[39] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems* (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds.), vol. 35, pp. 24824–24837, Curran Associates, Inc., 2022.

[40] D. Dua, S. Gupta, S. Singh, and M. Gardner, "Successive prompting for decomposing complex questions," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, (Abu Dhabi, United Arab Emirates), pp. 1251–1265, Association for Computational Linguistics, Dec. 2022.

[41] H. Elsahar, P. Vougiouklis, A. Remaci, C. Gravier, J. Hare, F. Laforest, and E. Simperl, "T-REx: A large scale alignment of natural language with knowledge base triples," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, (Miyazaki, Japan), European Language Resources Association (ELRA), May 2018.

[42] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural questions: A benchmark for question answering research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 452–466, 2019.

[43] P. Lewis, B. Oguz, R. Rinott, S. Riedel, and H. Schwenk, “MLQA: Evaluating cross-lingual extractive question answering,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 7315–7330, Association for Computational Linguistics, July 2020.

[44] O. Levy, M. Seo, E. Choi, and L. Zettlemoyer, “Zero-shot relation extraction via reading comprehension,” in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, (Vancouver, Canada), pp. 333–342, Association for Computational Linguistics, Aug. 2017.

APPENDIX A

FULL PROMPTS USED IN THIS STUDY.

A.1 Baseline Prompt

Q: How high is the highest point in the place where Tadeusz Peiper died?
Q1: Where did Tadeusz Peiper live when he died?
A1: Warsaw
Q2: How high is the highest point in #1 ?
A2: 115.7 metres

Q: In what year did the author of A Child's Garden of Verses die?
Q1: who penned a child's garden of verses
A1: Robert Louis Stevenson
Q2: In what year did #1 die?
A2: 1894

Q: How did the group that Tuvalu signed an agreement to ally with, rank Switzerland's economy?
Q1: With what group does the agreement form an alliance?
A1: European Union
Q2: How did #1 rank Switzerland's economy?
A2: Europe's most innovative country

Q: What manager of the performer of Whole Lotta Love tried to sign Queen?
Q1: Who performed Whole Lotta Love?
A1: Led Zeppelin
Q2: What manager of #1 tried to sign Queen?
A2: Peter Grant

Q: What was depicted on the banners of the religious group strongly opposing the idea of Neoplatonism in the First crusade?
Q1: Which religious group strongly opposed the idea of Neoplatonism?
A1: Christians
Q2: What was depicted on the banners of #1 in the First crusade?
A2: a red cross on a white field

Q: What did the war the AMX-30 was in inadvertently do in the early 1990s?
Q1: Which war was AMX-30 in?
A1: Gulf War
Q2: What did #1 inadvertently do in the early 1990s?
A2: radicalize the Islamist movement

A.2 Main Prompt

Context1: [Title: Tadeusz Peiper] (omitted for clarity)
Context2: [Title: Warsaw] (omitted for clarity)
Decompose Q to the maximum number of meaningful sub-questions. Use minimal answers and keep named entities, adjectives, adverbs when possible in your answers.
Q1: Where did Tadeusz Peiper live when he died?
A1: Warsaw
Q2: How high is the highest point in #1 ?
A2: 115.7 metres

Context1: [Title: A Child's Garden of Verses] (omitted for clarity)
Context2: [Title: Samoa] (omitted for clarity)
Q: In what year did the author of A Child's Garden of Verses die?
Decompose Q to the maximum number of meaningful sub-questions. Use minimal answers and keep named entities, adjectives, adverbs when possible in your answers.
Q1: who penned a child's garden of verses
A1: Robert Louis Stevenson
Q2: In what year did #1 die?
A2: 1894

Context1: [Title: Tuvalu] (omitted for clarity)
Context2: [Title: Switzerland](omitted for clarity)
Q: How did the group that Tuvalu signed an agreement to ally with, rank Switzerland's economy?
Decompose Q to the maximum number of meaningful sub-questions. Use minimal answers and keep named entities, adjectives, adverbs when possible in your answers.
Q1: With what group does the agreement form an alliance?
A1: European Union
Q2: How did #1 rank Switzerland's economy?
A2: Europe's most innovative country

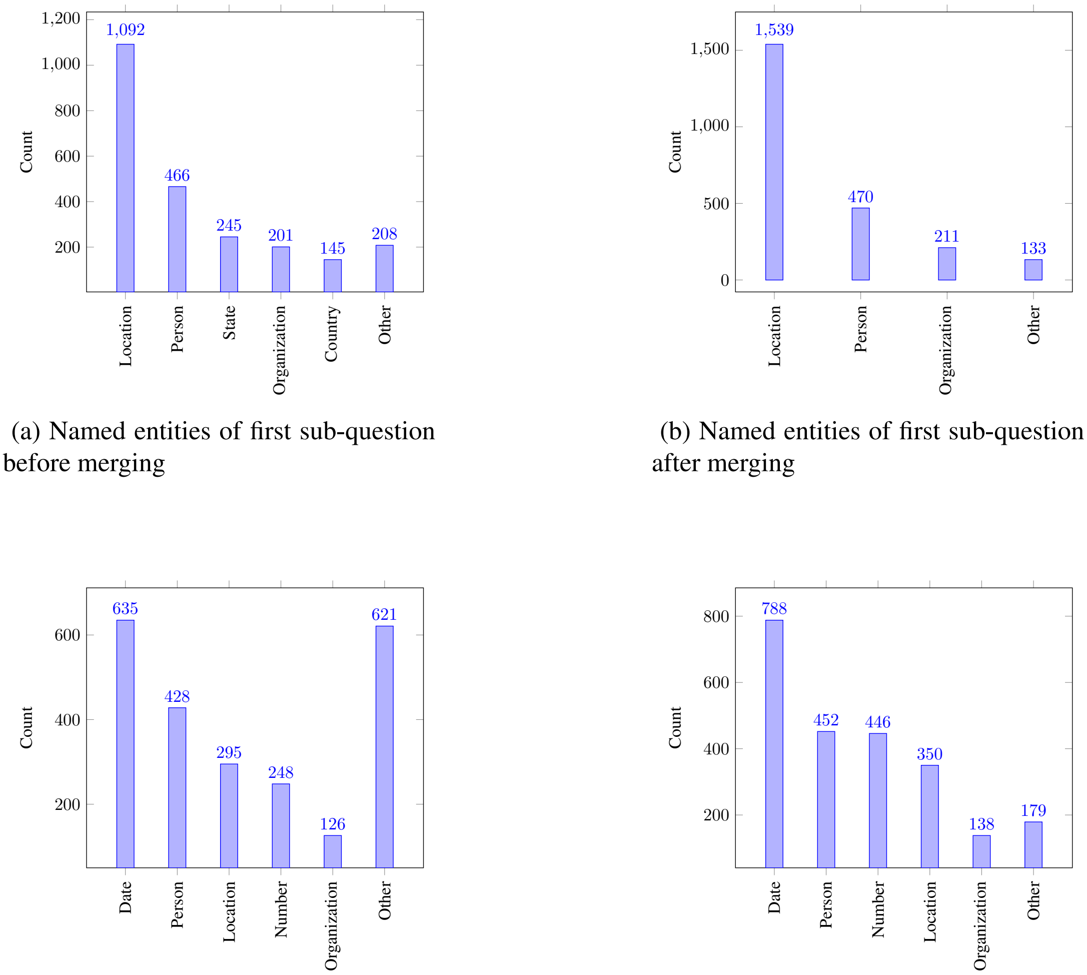
Context1: [Title: Whole Lotta Love](omitted for clarity)
Context2: [Title: Queen (band)] (omitted for clarity)
Decompose Q to the maximum number of meaningful sub-questions. Use minimal answers and keep named entities, adjectives, adverbs when possible in your answers.
Q1: Who performed Whole Lotta Love?
A1: Led Zeppelin
Q2: What manager of #1 tried to sign Queen?
A2: Peter Grant

Context1: [Title: Materialism] (omitted for clarity)
Context2: [Title: Red](omitted for clarity)
Q: What was depicted on the banners of the religious group strongly opposing the idea of Neoplatonism in the First crusade?
Decompose Q to the maximum number of meaningful sub-questions. Use minimal answers and keep named entities, adjectives, adverbs when possible in your answers.
Q1: Which religious group strongly opposed the idea of Neoplatonism?
A1: Christians
Q2: What was depicted on the banners of #1 in the First crusade?
A2: a red cross on a white field

Context1: [Title: AMX-30] (omitted for clarity)
Context2: [Title: Islamism] (omitted for clarity)
Q: What did the war the AMX-30 was in inadvertently do in the early 1990s?
Decompose Q to the maximum number of meaningful sub-questions. Use minimal answers and keep named entities, adjectives, adverbs when possible in your answers.
Q1: Which war was AMX-30 in?
A1: Gulf War
Q2: What did #1 inadvertently do in the early 1990s?
A2: radicalize the Islamist movement

APPENDIX B

NAMED ENTITY DISTRIBUTION OF SUB-QUESTION ANSWERS



(a) Named entities of first sub-question before merging

(b) Named entities of first sub-question after merging

(c) Named entities of second sub-question before merging

(d) Named entities of second sub-question after merging

Figure 8: Named entities before and after processing