QDMR-based Planning-and-Solving Prompting for Complex Reasoning Tasks

Jinfeng Huang[1*], Qiaoqiao She[2*], Wenbin Jiang[3‡], Hua Wu[2], Yang Hao[2], Tong Xu[1‡]
[1] University of Science and Technology of China, Hefei, China
[2] Baidu Inc, Beijing, China
[3] School of Artificial Intelligence, Beijing Normal University, Beijing, China
huangjf11@mail.ustc.edu.cn, tongxu@ustc.edu.cn
{sheqiaoqiao, wu_hua, haoyang03}@baidu.com, jiangwenbin@bnu.edu.cn

Abstract

Chain-of-Thought prompting has improved reasoning capability of large language models (LLM). However, it still is challenging to guarantee the effectiveness and stability for questions requiring complicated reasoning. Recently, Plan-and-Solve prompting enhances the reasoning capability for complex questions by planning the solution steps firstly and then solving them step by step, but it suffers the difficulty to represent and execute the problem-solving logic of complex questions. To deal with these challenges, in this work, we propose a novel Plan-and-Solve prompting method based on Question Decomposition Meaning Representation (QDMR). Specifically, this method first allows the LLM to generate a QDMR graph to represent the problem-solving logic, which is a directed acyclic graph composed of sub-questions. Then, the LLM generates a specific solving process based on the QDMR graph. When solving each sub-question, it can locate the preceding sub-questions and their answers according to the QDMR graph, and then utilize this information for solution. Compared with existing Plan-and-Solve prompting techniques, our method can not only represent the problem-solving logic of complicated questions more accurately with the aid of QDMR graph, but also deliver the dependence information accurately for different solution steps according to the QDMR graph. In addition, with the supervised fine-tuning on the Allen Institute dataset, the decomposing capability of LLM for complicated questions can be considerably enhanced. Extensive experiments show that our method has achieve a great significance in arithmetic reasoning and commonsense reasoning task by comparing the classical Chain-of-Thought prompting and Plan-and-Solve prompting techniques, and the improvements achieved are even greater for problems with more reasoning steps.

Keywords: Question Decomposition Meaning Representation, Planning-and-Solving Prompting, Complex Reasoning

1. Introduction

Prompting techniques (Brown et al., 2020; Min et al., 2022; Dong et al., 2023) enable generative large language models to solve given problems by emulating provided examples. These techniques perform well on many tasks, such as sentiment recognition and topic classification (Wei et al., 2022a). But, they may fall short on tasks requiring reasoning, such as logical reasoning, mathematical calculations, and commonsense reasoning (Rae et al., 2022). Along this line, Chain-of-Thought (CoT) prompting utilizes detailed thought chains that describe the problem-solving process to inspire the large language model to solve questions step by step. However, generative language models have a low sensitivity to historical information generated earlier in the solving process (Touvron et al., 2023; Anil et al., 2023; Bubeck et al., 2023), which hinders accurate execution of long-span reasoning processes for complex reasoning questions with multiple reasoning steps, and the reasoning effect and stability will be dimininshed. (Fu et al., 2023; Yao et al., 2023b; Press et al., 2023)

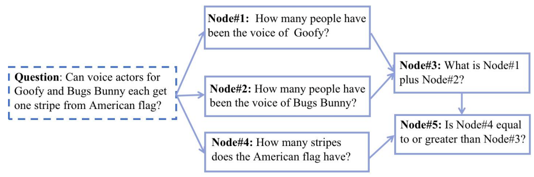

Figure 1: An example of Question Decomposition Meaning Representation(QDMR) Graph. Node#i represents the *i*th sub-question after decomposing the original question.

Recently, Plan-and-Solve prompting has attracted wide attention, which plans the solution steps for a given question at first, and then solves it step-by-step, while the guiding role of solving plans significantly improves the reasoning ability and stability of complex problems. But, it has the defect of not being able to accurately represent and execute the problem-solving logic. Besides, during the problem-solving phase, every step undertaken

* These authors contributed to the work equllly and should be regarded as co-first authors.
‡ Corresponding authors.

LREC-COLING 2024, pages 13395–13406
20-25 May, 2024.

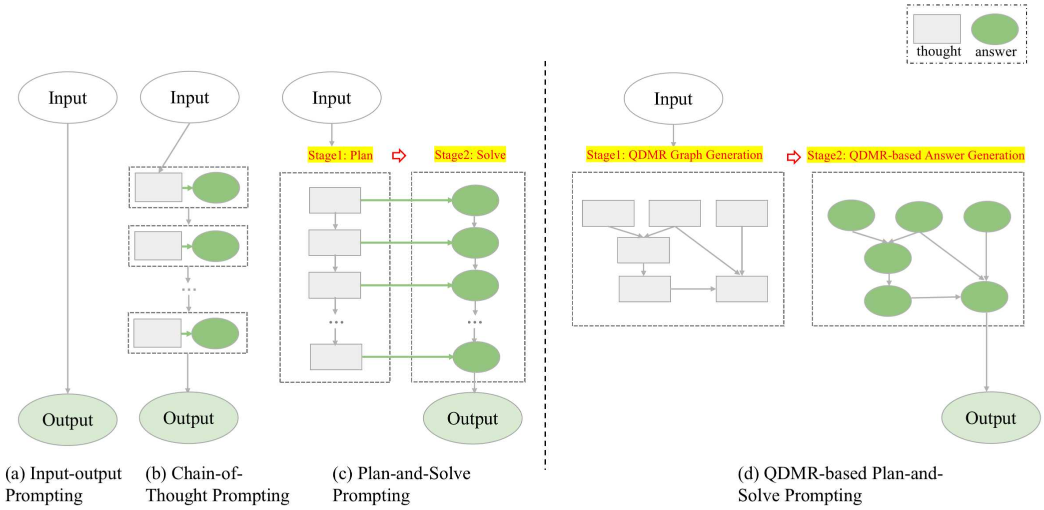

Figure 2: A diagram showcasing different approaches for problem-solving using LLMs. Each rectangular box symbolizes a thought, while the green ellipse represents corresponding problem-solving procedure related to thought. This coherent language sequence acts as an intermediate step in problem-solving process.

fully capitalizes on the information returned from all preceding steps without distinguishing the significance levels of various information. However, statistical analysis results from Figure 3 indicate that a substantial proportion of the problems have a graphical reasoning structure. This underscores the limitation of solely employing a chain-based resolution approach in precisely modeling the intricate logic inherent to numerous problems.

In this work, we propose a novel Plan-and-Solve prompting method, based on Question Decomposition Meaning Representation (QDMR) (Wolfson et al., 2020). With the help of structured QDMR Graph, our model can represent and execute graphical problem-solving logic, as shown in Figure 1. This method consists of two explicit generative stages illustrated in Figure 2(d). In detail, the first phase allows large language model to generate a QDMR graph for given question, which is a directed acyclic graph describing the precise topological dependencies among different sub-questions. Compared with the linear solution plan of existing Plan-and-Solve prompting techniques, it could accurately describe complex problem-solving logic. In the second phase, the large language model generates solving process based on the QDMR graph. When solving each sub-question, the model can refer QDMR graph to make use of preceding sub-questions and their answers. Compared to the sequential solution plan of existing Plan-and-Solve prompting techniques, our approach can generate the answer to current sub-question based on a more precise context. Intuitively, by first generating a precise QDMR graph and then following its topological order for solving, our method simulates

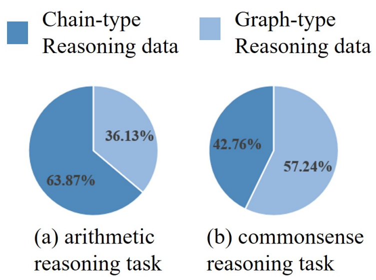

Figure 3: The proportion of chain-type and graph-type reasoning data in arithmetic and commonsense datasets, as illustrated in Section 4.1

an intelligent agent's operation in a prompt-driven manner, enabling the language model to plan and solve complex questions without multiple invocations. This strategy endows our method with the complex reasoning capability of intelligent agents, while also retaining the simplicity and effectiveness of the prompting techniques. Additionally, through instruction tuning, it is feasible to produce more precise QDMR graph, consequently enhancing the final reasoning accuracy.

We compare the performance of our methods with existing ones on complex reasoning tasks, i.e., arithmetic reasoning and commonsense reasoning. Experiments show that under the contextual learning setting, our method achieves a significant performance boost compared with Plan-and-Solve prompting and classic Chain-of-Thought prompting, especially for questions with more reasoning steps. Moreover, under the supervised fine-tuning setup, the quality of the generated QDMR graph greatly

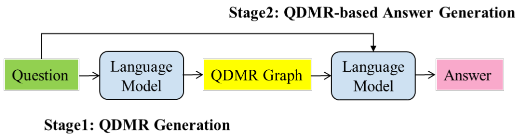

Figure 4: QDMR based Plan-and-Solve prompting solving a complex reasoning problem in two stages: (1) generate the QDMR graph for given question, which decompose the question into sub-question diagram; (2) query the language model to solve the sub-questions in topological order based on the QDMR graph. When solving each sub-question, only information from its preceding sub-questions is used. See detailed process of the following two stages in Figure 5 and Figure 6.

improved, and our method showed even more significant improvements against existing prompting techniques, pointing towards directions for future enhancements.

2. Background

Large language models (Touvron et al., 2023; Anil et al., 2023; Ouyang et al., 2022) have instruction-following capability through prompting learning. Considering that our approach is indeed an improved version of the Plan-and-Solve prompting technique, we introduce the technical background first, including Input-output prompting (Min et al., 2022; Dong et al., 2023; Penedo et al., 2023), Chain-of-Thought prompting (Wei et al., 2022b; Kojima et al., 2022; Zhang et al., 2023c), and Plan-and-Solve prompting (Wang et al., 2023b; Zhou et al., 2023; Jiang et al., 2023) techniques.

Input-output prompting: As shown in Figure 2(a), Input-Output prompting directly turns a problem input x into output y with large language model.

Chain-of-Thought prompting: For tasks that require multi-step reasoning, Input-output prompting techniques are less effective. As shown in Figure 2(b), Chain-of-Thought prompting uses "thought chain" as prompting information. It reflects the step-by-step solution process, thus prompting the language model to gradually generate fine-grained solution and ultimately generate the answer.

Plan-and-Solve prompting: When the complexity of task increases, the enhancement attributed to Chain-of-Thought prompting methodology is progressively attenuated. As shown in Figure 2(c), Plan-and-Solve Prompting first generates a coarse-grained task plan, followed by a fine-grained solving process, which achieves better effect and stability than Chain-of-Thought prompting.

With above prompting strategies, extensive research enhances model's reasoning capability though improving prompting technique, e.g., techniques for selecting or generating better prompting examples (Zhang et al., 2022; Shum et al., 2023; Ho et al., 2023; Kim et al., 2023), voting fusion or selection of reasoning answers (Wang et al., 2023d; Yoran et al., 2023; Wang et al., 2023c; Jain et al., 2023; Huang et al., 2022a), task decomposition based on planning mechanisms (Ahn et al., 2022; Lin et al., 2023; Lynch et al., 2023; Brohan et al., 2023; Liang et al., 2023), introducing dynamic exploratory and trial-and-error mechanisms during reasoning process (Yao et al., 2023b; Ruan et al., 2023; Chen et al., 2023) and so on. Our method can be used in combination with all the above techniques to achieve better results, a detailed discussion of which is beyond the scope of this context.

3. Methodology

3.1. Overall Architecture

In this work, we propose a novel Plan-and-Solve prompting method based on Question Decomposition Meaning Representation (QDMR). Figure 4 illustrates overall architecture which consists of two phases, within this framework, the large language model sequentially generates symbolic sequences that align with the two delineated phases. In Section 3.3, we elaborate initial phase to generate QDMR graph for given question. Subsequently, in Section 3.4, we describe the second phase to solve problems based on the QDMR graph.

3.2. Question Decomposition Meaning Representation

The Question Decomposition Meaning Representation graph as introduced by (Wolfson et al., 2020), is meticulously constructed to cater to a specific query, thereby mirroring the intricate logical progression inherent in the problem-solving continuum. This graph, denoted as G, embodies the structure of a directed acyclic graph. Within its framework, nodes epitomize individual sub-questions, whereas the edges elucidate the inter-dependencies and affiliations among these constituent sub-questions. Suppose within the graph G, two directed edges $i \rightarrow k$ and $j \rightarrow k$ are present. This suggests that the resolution of q_k depends on addressing sub-questions q_i and q_j. Specifically, the solution for sub-question q_k mandates the incorporation of content and responses derived from sub-questions q_i and q_j. By formulating the pertinent QDMR graph prior to addressing the question, it facilitates the provision of more accurate guidance cues during the problem-solving phase, thereby enhancing the accuracy and efficacy of the reasoning process.

Figure 1 furnishes a visualization of the QDMR graph. Within this depiction, the intricate question Q: "Can voice actors for Goofy and Bugs Bunny

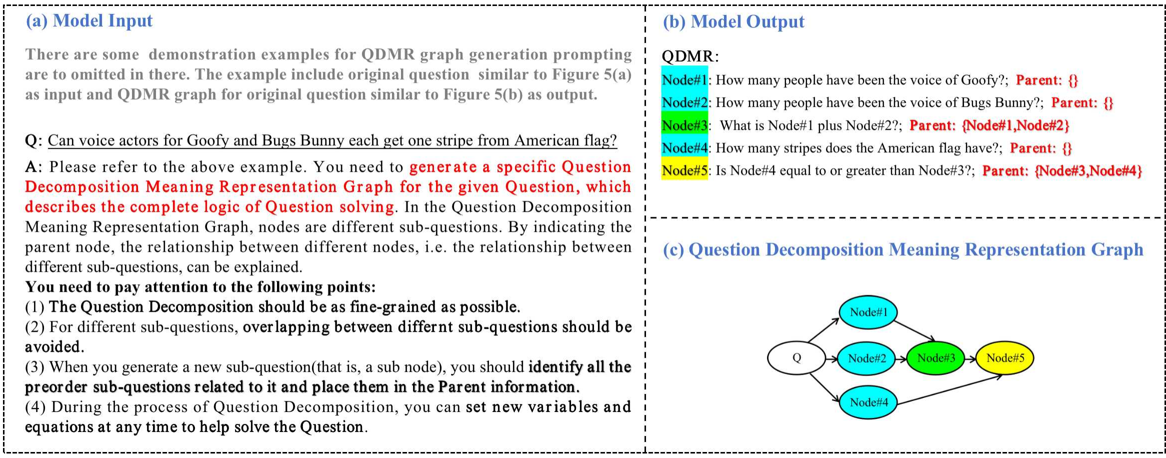

Figure 5: Example of generating QDMR graph through contextual learning with (a) QDMR graph generation Prompting with detailed instructions and (b) model's output, which is a serialized representation of the QDMR graph corresponding to given question and (c) graphical representation of the QDMR graph.

each acquire one stripe from the American flag?" is deconstructed into five intermediary sub-questions. Leveraging broad knowledge, determinations can be drawn for the sub-questions q_1, q_2, q_4 linked to Node#1, Node#2, and Node#4. Upon addressing Node#1 and Node#2, we circumscribe the solution space of Node#3 to the pertinent context, specifically, in response to the query: "How many individuals have served as the voice for both Goofy and Bugs Bunny?". Ultimately, integrating the content information and answers from both Node#3 and Node#4, we can refine the solution space for Node#5, posing sub-question: "Does the count of stripes on the American flag exceed or equate to the total number of individuals who have lent their voice to characters such as Goofy and Bugs Bunny?'". Adhering rigorously to the topological sequencing of the QDMR graph, we execute the designated solution methodology, culminating in the derivation of the answer to the primordial question.

3.3. QDMR Generation

A context-aware learning methodology is employed to guide the language model in the creation of a QDMR graph tailored to intricate question. The QDMR graph serves to illuminate the intricate topological interconnections inherent among sub-questions, a stark contrast to language model's inherent limitation of producing merely linear symbolic sequences. It is imperative to identify a method to encapsulate the QDMR graph using character sequences, derived from language models. To this end, we have engineered a serialized representation technique for the QDMR graph, optimized for active recognition and generation by the language model via contextual learning. Succinctly, each sub-question is allotted a distinct node identifier, with explicit delineation of its dependent parent nodes.

Figure 5 presents the serialized representation schema of the QDMR graph, offering an example on the methodology employed in the generation of QDMR graph via in-context learning. Concretely, all sub-questions is systematically arranged in topological succession. Sub-questions devoid of antecedent nodes present an empty list of parent nodes. In contrast, those associated with parent nodes display a non-vacant list, incorporating the specific node identifiers that correlate with preceding sub-questions. Due to the adoption of a clearly defined symbolic notation, our model is poised to seamlessly execute automatic transitions between the QDMR graph and its serialized embodiment. Explicitly, through the utilization of prompt-based learning methodologies, model is incited to yield a QDMR graph pertinent to the presented question. We curated several pairs encompassing **<Question, QDMR graph>** as prompting examples. These prompting examples, augmented by **essential directive information**, act as foundational prompts. Upon integrating them with the original question, they are input into the language model. This, in turn, produces a serialized delineation of the QDMR graph tailored to the complex query requiring decomposition.

3.4. QDMR-based Answer Generation

Utilizing contextual learning, the overarching goal is to guide the language model in solving question based on the QDMR graph. The predominant challenge lies in capacitating the language model to adeptly harness the problem-solving rationale encapsulated by the QDMR graph. Throughout the leftward-to-rightward recursive generation paradigm, the model iteratively produces subsequent symbols, predicated upon both the intro-

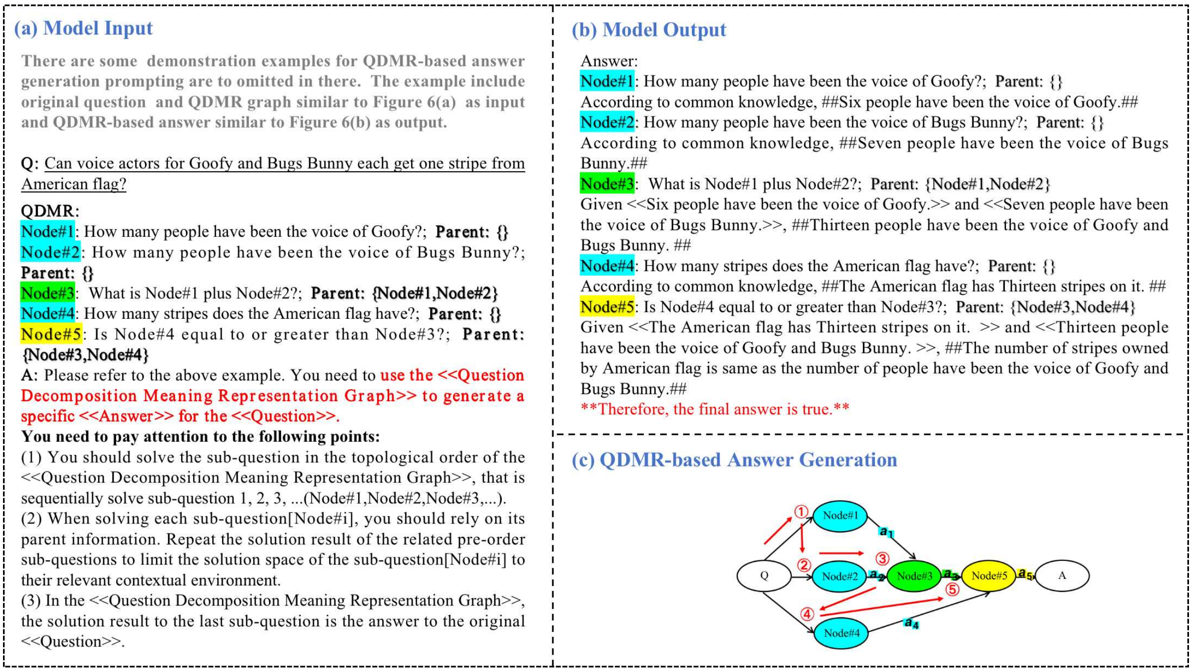

Figure 6: Example of generating answer based on the QDMR graph through contextual learning with (a) QDMR-based answer generation Prompting with detailed instructions and (b) model's output, which includes the topological solution of QDMR graph as well as the answer to the original question and (c) the execution order of sub-question solving based on the QDMR graph.

duced symbols and those previously formulated. As elucidated by prior scholars (Anil et al., 2023; Bubeck et al., 2023), the most contemporaneously produced information exerts paramount influence on the determination to yield the subsequent symbol. Consequently, meticulous orchestration of the content of the solving process is pivotal. This ensures that the content generation sequence impeccably aligns with the topological order delineated within the QDMR graph, thereby aspiring to emulate the structured problem-solving rationale within the serialized output progression.

Figure 6 shows the QDMR-based Answer Generation schema, it presents the process of generating answers based on the QDMR graph. Specifically, the constituent sub-questions undergo resolution in a methodical topological sequence. When addressing each constituent sub-question, one initially revisits both the parent node of the current sub-question and its pertinent answer. Subsequently, the solution for the current sub-question is formulated. This methodology ensures that the resolution process of every sub-question is intricately anchored to its most germane historical data, thereby obviating potential perturbations from extraneous information during the current sub-question's resolution. Specifically, employing the technique of prompt-based learning, inspire model's capability in performing inference and problem-solving. We constructed several pairs encompassing **<QDMR graph, Answer>** as prompting examples. These prompting examples, augmented by **essential directive information**, act as foundational prompts. Upon integrating them with a QDMR graph of original question, they are input into the language model. This, in turn, generates corresponding QDMR-based answer for original question.

3.5. Supervised Finetuning

The approach and data we employ for the QDMR graph and QDMR-based answer representation is highly standardized. Leveraging the existing data (Wolfson et al., 2020; Geva et al., 2021) and high-performance large language model (Touvron et al., 2023; Anil et al., 2023; Bubeck et al., 2023; Du et al., 2022), it is feasible to automatically construct large-scale, high-quality training datasets. Though fine-tuning the large language model in the Supervised Fine-Tuning (SFT) manner, enabling it to generate more precise QDMR graph and corresponding problem-solving workflows.

For the fine-tuning task associated with generation of the QDMR graph, training data have been derived from the Break dataset (Wolfson et al., 2020). For the fine-tuning task of generating QDMR-based answer, we use GPT4 (Bubeck et al., 2023) to generate multiple inference paths and take the inference path with the correct final answer as train sample. Section 4.1 provides specific detail information.

Model	Prompt	Dataset						Average
		AQuA	Gsm8k	MultiArith	SingleEq	Musique	HotpotQA	
text-davinci-002	Cot	0.3743	0.5838	0.9212	0.8571	0.6783	0.8123	0.7045
	PS	**0.4185**	0.6587	**0.9567**	0.8970	0.6854	0.8331	0.7416
	Ours	0.4181	**0.6737**	0.9474	**0.9148**	**0.7005**	**0.8475**	**0.7503**
text-davinci-003	Cot	0.3426	0.7079	0.9434	0.9288	0.6916	0.8519	0.7444
	PS	0.4002	0.7164	0.9506	0.9403	0.7073	**0.8874**	0.7670
	Ours	**0.4194**	**0.7520**	**0.9510**	**0.9591**	**0.7351**	0.8853	**0.7837**
GPT3.5-Turbo	Cot	0.4672	0.8312	0.9677	0.9611	0.6132	0.8586	0.7832
	PS	0.4440	0.8385	**0.9821**	0.9795	0.6467	0.8706	0.7936
	Ours	**0.4688**	**0.8613**	0.9735	**0.9826**	**0.6683**	**0.8997**	**0.8090**

Table 1: Within the experimental setting of in-context learning, accuracy comparison between our method, Chain-of-Thought prompting(Cot) and Plan-and-Solve prompting(PS) on multiple reasoning datasets. The best results are boldfaced.

4. Experiments

4.1. Datasets and Baselines

Datasets of multi-step reasoning tasks i.e., Arithmetic Reasoning as well as Commonsense Reasoning, are included in our experiments. To verify the effectiveness of our method on various generative language models, We experiment on six datasets spanning arithmetic reasoning and commonsense reasoning with five LLMs: **text-davinci-002, text-davinci-003, Gpt3.5-Turbo, Llama2-7B, and Chatglm2-6B** through in-context learning and instruction tuning. **Arithmetic Reasoning:** (1) the AQuA (Ling et al., 2017) dataset of algebraic problems with answers comprised of multiple natural language rationales. (2) the Gsm8k (Cobbe et al., 2021) dataset of high quality grade school math problems with question that can be solved in 2-8 steps. (3) the MultiArith (Roy and Roth, 2015) dataset of mathematical problems that necessitates numerous inference steps for resolution. (4) the SingleEq (Koncel-Kedziorski et al., 2015) dataset of algebraic problems calls for the solution of equations. **Commonsense Reasoning:** (1) the musique dataset (Trivedi et al., 2022) of commonsense problems with 2-4 hops, constructed by composing existing single-hop Wikipedia seed problems. (2) the HotpotQA (Wolfson et al., 2020) dataset of commonsense questions based on Wikipedia that requires reasoning using multiple supporting problems.

Additionally, in order to enhance the capability to generate question decomposition meaning representation graph for given question, we developed a **<Question,QDMR graph>** dataset based on existing BREAK dataset (Wolfson et al., 2020). BREAK dataset is released by the Allen Institute, includes natural language questions and human-annotated question decomposition meaning representations, with a total of 83,978 examples. All questions with QDMR that can be graphed as well as 10% of randomly selected questions with chain QDMR are kept, resulting in a total of 30,000 training samples. Additionally, to improve the ability of answers generated based on the QDMR graph, training data int the format of **<QDMR graph, Answer>** is automatically constructed using the dataset StrategyQA (Geva et al., 2021), which is a commonsense Q&A dataset that includes relevant background knowledge required for problem-solving. Employing the capabilities of GPT-4 (Bubeck et al., 2023), we generate a myriad of potential QDMR-based responses for the posed inquiry. Subsequent evaluations, predicated on criteria such as the veracity, fluidity, and brevity of the generated narratives, guide our selection, resulting in the endorsement of the QDMR-based answer with the highest merit. In the stage of QDMR-based answer generation, there are 2290 training samples produced, with data evaluation accuracy reaching an impressive 98%. For the training data produced above, we carried out instruction tuning on several open-source models, including Llama2-7B, Chatglm2-6B.

4.2. Main Results

Our approach is an improvement on the Plan-and-Solve prompting (Wang et al., 2023b) and Chain-of-Thought prompting (Wei et al., 2022b). Therefore, we first conducted a comprehensive comparison with these two prompting techniques. **Table 1 reports comparison of accuracy in our method with the two prompting techniques on the arithmetic reasoning and commonsense reasoning task in the paradigm of context learning, a significant improvement was observed on most datasets across all baseline models**. Relative to the Chain-of-Thought prompting technique, an average performance increase between 2.59% and 4.58% is documented across all datasets. Conversely, when benchmarked against the Plan-and-Solve prompting technique, the performance escalates on average from 0.87% to 1.66% over several datasets except MultiArith and HotpotQA dataset. By meticulously analyzing a subset of MultiArith dataset, the exception could be due to most reasoning trajectories within MultiArith dataset ex-

Model	Prompt	Dataset						Average
		AQuA	Gsm8k	MultiArith	SingleEq	Musique	HotpotQA	
Llama2-7B	ICL-Cot	0.2638	0.2669	0.6876	0.6561	0.3598	0.4111	0.4409
	ICL-PS	0.2874	0.2775	0.7109	0.6501	0.3950	0.3921	0.4522
	ICL-Ours	0.2969 (38%)	0.2861 (44%)	0.7325 (86%)	0.6782 (74%)	0.3988 (64%)	0.4370 (58%)	0.4716
	SFT-Ours	**0.3110** (42%)	**0.3035** (50%)	**0.7378** (90%)	**0.7003** (90%)	**0.4146** (84%)	**0.4554** (76%)	**0.4871**
Chatglm2-6B	ICL-Cot	0.2717	0.2835	0.7002	0.6765	0.3789	0.3699	0.4468
	ICL-PS	0.3150	0.3078	0.6942	0.6802	0.3907	0.3780	0.4610
	ICL-Ours	0.3254 (36%)	0.3269 (40%)	0.7230 (74%)	0.7037 (66%)	0.4017 (60%)	0.4140 (56%)	0.4825
	SFT-Ours	**0.3465** (48%)	**0.3362** (56%)	**0.7391** (82%)	**0.7285** (78%)	**0.4337** (72%)	**0.4378** (74%)	**0.5036**

Table 2: Comparison of surpervised fine-tuning and in-context learning in producing QDMR graph and related effects on the final reasoning results. The best results are boldfaced. Tip: The first line in ICT-Ours and SFT-Ours represents the Accuracy, the second line in ICT-Ours and SFT-Ours represents quality evaluation of the QDMR graph generated under the current settings.

hibit a linear fashion. Concerning text-davinci-003 model and HotpotQA dataset, the effectiveness of our approach commensurates with Plan-and-Solve prompting technique. This could plausibly be attributed to limitations in the model arising from the quality of the QDMR graph. In the concluding segment of this subsection, we delve into whether the quality of QDMR graph exerts a tangible impact on the final reasoning results.

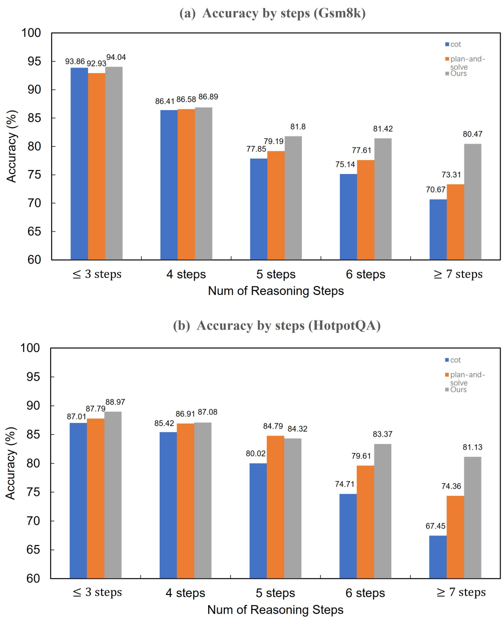

Figure 7: Accuracy (%) of our method compared with chain of thought and plan-and solve prompting , broken down by the number of reasoning steps required in the expected solution.

To further elucidate the advantages of our approach vs Plan-and-Solve prompting (Wang et al., 2023b) and Chain-of-Thought prompting (Wei et al., 2022b), we evaluated the comparative effects on questions varying reasoning complexities. Given the quintessential nature of GPT3.5-Turbo model, we have elected to employ it as our benchmark model. To gauge the difficulty of the reasoning, simply counted the number of inference steps for resolution. Figure 7 illustrates the comparative performance of our method with these two prompting techniques on problems varying numbers of reasoning steps. For questions necessitating more than seven reasoning steps, our methodology registered a performance augmentation of 9.8% compared to Chain-of-Thought prompting and advanced by 7.16% relative to Plan-and-Solve prompting on the arithmetic reasoning dataset Gsm8k. In the context of the commonsense reasoning dataset HotpotQA, our technique surpassed Chain-of-Thought prompting and Plan-and-Solve prompting by 13.68% and 6.77%, respectively. Such findings accentuate the validity of our method on intricate reasoning problems. Moreover, **as the number of reasoning steps increases, the lead of our method over previous methods gradually expands.**

Finally, to ascertain whether the quality of the question decomposition meaning representation graph exerts an influence on the ultimate reasoning outcomes, we undertook experiments on two models: Llama2-7B and Chatglm2-6B, employing instruction tuning. During the fine-tuning process, a multi-task learning approach was adopted. Assisting the model in learning question decomposition, the initial task was to fine-tune the QDMR graph generation. The second task was to optimize the answer generation, helping the model to learn question solving. In the process of problem-solving, pertinent background knowledge has been integrated to mitigate the impact of factual mistake, facilitating a more precise comparative analysis. Table 2 lists the experimental results. Fine-tuning the model can significantly improve the quality of the generated QDMR graph(**SFT-Ours**) compared to generating QDMR graph through in-context learning(**ICL-**

Dataset	Question	QDMR Graph Generated Through In-context Learning	QDMR Graph Generated Through Supervised Fine-tuning
Musique	What is the occupation of the father of the actor that plays Will Graham on teh series 'Hannibal'?	**Node#1: Who is the father of the actor that plays #4?;Parent:{#4}** Node#2: #1 is the occupation of whom?;Parent:{#1} Node#3: Who plays #4?;Parent:{#4}	Node#1: who plays will graham on hannibal?; Parent:{} Node#2: #1 >>father?; Parent:{#1} Node#3: What job did #2 have?; Parent:{#2}
Gsm8k	In a dance class of 20 students, 20% enrolled in contemporary dance, 25% of the remaining enrolled in jazz dance, and the rest enrolled in hip-hop dance. What percentage of the entire students enrolled in hip-hop dance?	Node#1: How many students enrolled in hip-hop dance?;Parent:{} Node#2: How many students enrolled in contemporary dance?;Parent:{} Node#3: How many students enrolled in jazz dance?;Parent:{} **Node#4: What percentage of the entire students enrolled in hip-hop dance?; Parent:{#1,#2,#3}**	Node#1: How many students are enrolled in contemporary dance?;Parent:{} Node#2: How many students are left after contemporary dance?;Parent:{#1} Node#3: How many students are enrolled in jazz dance?;Parent:{#2} Node#4: How many students enrolled in hip-hop dance?;Parent: {#2,#3} Node#5: What percentage of the entire students are enrolled in hip-hop dance?;Parent:{#4}

Table 3: Examples of QDMR Graph produced by Llama2-7B Model in In-context Learning and Supervised Fine-tuning Settings.

Ours), with an increasing from 4% to 20% across all datasets applied to different models, ultimately leading to an average performance augmentation of 5.15% compared to in-context Learning mode of Cot prompting(**ICL-Cot**) and 3.88% in contrast to in-context Learning mode of Plan-and-Solve prompting(**ICL-PS**), while in-context Learning mode of Our method(**ICL-Ours**) only surpassed **ICL-Cot** and **ICL-PS** by 3.32% and 2.04% on average, respectively. These experimental results conducting across a multitude of models and datasets consistently elucidate that **the quality of the generated QDMR graph profoundly influences the generation of the ultimate answers**. Through case analysis, it has been observed that incremental enhancements in the quality of the graph consistently lead to rectificating solving of samples with complex reasoning structures. As illustrated in Figure 3, a substantial proportion of the problems exhibit a non-linear reasoning structure. Consequently, **by continually enhancing the quality of the QDMR graph, one can more accurately represent the underlying problem-solving logic, leading to improvements in the overall reasoning performance**. Such findings illuminate a potential direction for future research.

4.3. Quality Analysis of QDMR Graph

To assess the quality of the generated QDMR graph, we have implemented a structured approach. Our analysis begins with applying prompt learning methodology, as illustrated in Figure 8. This procedure incorporates GPT4 (Bubeck et al., 2023) to thoroughly evaluate the graph, focusing on attributes such as its completeness, effectiveness, clarity, and redundancy. Subsequently, the graph is assigned to one of several predefined labels: "Excellent", "Good", "Fair", "Poor", and "Very Poor". The subsequent phase of our evaluation strategy entails the selection of random samples from each category of the initially labeled data. A secondary round of manual quality assessment is then conducted, which includes the re-labeling process.

The experimental analysis presented in Section 4.2 illustrates a positive correlation between the progressive improvement of the QDMR graph's

Instruction

You are responsible for developing an intelligent question answering system that will respond to user queries according to the logical topology of the question decomposition meaning representation graph. We aim for the graph to accurately reflect the question-solving logic to ensure the system can correctly comprehend and answer various types of questions.
(Nodes in the graph represent decomposed sub-questions, and edges represent relationships between different sub-questions.)

You need to evaluate the quality of the provided question decomposition meaning representation graph. The **evaluation criteria** are as follows:
1. Completeness: Does the graph capture the main aspects and relevant information of the question?
2. Effectiveness: Does the graph accurately present the structure and semantic relationships of the question?
3. Clarity: Are there any ambiguities or vagueness in the nodes and edges of the graph that could lead to the system misunderstanding the question?
4. Redundancy: Does the graph contain unnecessary or redundant nodes?

Based on the above evaluation criteria, you can comprehensively assess the provided graph and *categorize it into the following five labels:* {Excellent, Good, Fair, Poor, Very Poor}. Record the final classification result under Classification:

Example

> ####Example of QDMR Graph Quality Analysis
Q + QDMR graph -> graph quality analysis, Classification: label

Input

> ####QDMR Graph to be Analyzed
Q + QDMR graph ->

Figure 8: Prompt Learning Method for Quality Assessment of QDMR Graph.

quality and the consequent enhancement of the solution's efficacy. This observation emphatically underscores the critical importance of augmenting graph quality to bolster question-solving capabilities. We aim to enhance the quality of generated graph by employing various methods such as improving the expression of graph logical structure and supervised fine-tuning adjustments.

Table 3 presents the results of decomposing specific queries via Llama2-7B model. In the in-context learning setting, this model shows limited capability in breaking down the sub-problems contained within the MuSique dataset. In contrast, when applied to the Gsm8k dataset, the model adeptly identifies sub-problems but struggles to arrange them in a coherent sequence that aligns with the logical progression needed for effective resolution. Ideally, the sequence for tackling these problems should be Node#2 -> Node#3 -> Node#1 -> Node#4. However, our observations indicate that supervised fine-tuning significantly improves the quality of the QDMR graph, ensuring an accurate depiction of the sub-problems' interconnections and reflecting the necessary logical flow for problem resolution.

5. Related work

The approach is affiliated with Planning-based reasoning techniques. Planning techniques are employed to generate a decision sequence from an initial state to a target state, allowing for the resolution of complex tasks (Ahn et al., 2022; Singh et al., 2023; Huang et al., 2022b; Lin et al., 2023; Lynch et al., 2023). Adopting planning techniques based on language model to decompose complex tasks into specific instructions, and directing robots or other intelligent agents to perform navigation or other intricate operations (Wang et al., 2023b; Zhou et al., 2023; Jiang et al., 2023; Radhakrishnan et al., 2023; Zhang et al., 2023a). Apply planning techniques to facilitate reasoning, they initially outline a solving strategy for a complex problem and subsequently guide the ensuing reasoning computational process based on this. Our work innovates substantially upon the foundational works mentioned by introducing a QDMR graph that reflects problem-solving logic, precisely characterizing the topological relationship between distinct sub-questions. This graph unveils the comprehensive solution approach. Following the order depicted by this graph, specific reasoning processes are accurately executed.

The operational process of our method is similar to agent-based reasoning techniques. Agent-based techniques realize complex task solving by constantly generating and executing new actions through dynamic interactions between the agent and its environment (Yao et al., 2023b; Wang et al., 2023e; Zhang et al., 2023b; Chen et al., 2023). At the same time, it driven by large language model to resolve complex reasoning question, enhancing reasoning capability through decomposition of sub-questions and multiple iterative interactions. The intrinsic distinction between our method and agent-based approach lies in our employment of context learning strategy to simulate the dynamic reasoning process of the agent. On the one hand, we retain the advantages of agent techniques, decomposing complex questions into precise topological structures, and then progressively solving problems based on decomposition. On the other hand, we circumvent the drawbacks of agent techniques by simulating the agent's reasoning process through a left-to-right recursive generation process, as opposed to invoking language model multiple times.

Beyond the two works mentioned above, there exists some improvement work that have introduced non-linear strategies based on classic Chain-of-Thought prompting, but these cannot be strictly categorized as planning or agent-based works. For instance, (Yao et al., 2023a) proposed Tree-of-Thought, which considers different reasoning paths and evaluates the effect of each step, with the highest-scoring path serving as the reasoning answer. (Wang et al., 2023a) introduced Algorithm of Thoughts Prompting, explores the question solving space and devises strategies from an algorithmic perspective, which can alleviate the shortcomings of high query and computational costs in the Tree-of-Thought prompting. (Besta et al., 2023) proposed Graph-of-Thought Prompting, treating the information generated by the large language model as thought units and continuously generating, combining, and eliminating these units to form a complex thought network with a graph structure. However,even if these methods have indeed transcended the linear pattern of classic chain-of-thought prompting, exhibiting mechanisms like branching, backtracking, and error correction during the solution generation process, they have not effectively modeled and leveraged the intricate dependencies between the sub-questions, and they consume significant computational resources.

6. Conclusion

Building on the Plan-and-Solve prompting technique (Wei et al., 2022a), we propose an improved method based on the Question Decomposition Meaning Representation(QDMR) (Wolfson et al., 2020). This method uses the QDMR graph as a plan, reflecting the complex dependencies between sub-questions, and then guides the answer generation process, ensuring that solving each sub-question can be located to its preceding sub-questions and corresponding answers based on the QDMR graph. As the experiments demonstrate that our approach can achieve significant improvements over existing Plan-and-Solve prompting technique on multiple baseline models and datasets. It suggests that our method can leverage the QDMR graph to more accurately express the logic of complex question solving, and can provide dependency information accurately for district solution steps based on the QDMR graph. In the future, we will continue to explore how to automatically generate QDMR graph for new question and generate answer based on the graph, as well as further improve the quality of QDMR graph generation and following answer generation. We also plan to extend this method to a wider range of complex tasks.

7. Acknowledgement

This work was supported in part by the grants from National Natural Science Foundation of China (No.62222213, U22B2059, 62072423), and the USTC Research Funds of the Double First-Class Initiative (No.YD2150002009), and the Fundamental Research Funds for the Central Universities.

8. Bibliographical References

Michael Ahn, Anthony Brohan, and Noah Brown et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances.

Rohan Anil, Andrew M. Dai, and Orhan Firat et al. 2023. Palm 2 technical report.

Maciej Besta, Nils Blach, and Ales Kubicek et al. 2023. Graph of thoughts: Solving elaborate problems with large language models.

Anthony Brohan, Noah Brown, and Justice Carbajal et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control.

Tom Brown, Benjamin Mann, and Ryder et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Sébastien Bubeck, Varun Chandrasekaran, and Ronen Eldan et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4.

Liting Chen, Lu Wang, and Hang Dong et al. 2023. Introspective tips: Large language model for in-context decision making.

Qingxiu Dong, Lei Li, and Damai Dai et al. 2023. A survey on in-context learning.

Zhengxiao Du, Yujie Qian, and Xiao et al. Liu. 2022. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. Association for Computational Linguistics.

Yao Fu, Hao Peng, and Ashish Sabharwal et al. 2023. Complexity-based prompting for multi-step reasoning.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada. Association for Computational Linguistics.

Jiaxin Huang, Shixiang Shane Gu, and Le Hou et al. 2022a. Large language models can self-improve.

Wenlong Huang, Fei Xia, and Ted Xiao et al. 2022b. Inner monologue: Embodied reasoning through planning with language models.

Siddhartha Jain, Xiaofei Ma, Anoop Deoras, and Bing Xiang. 2023. Self-consistency for open-ended generations.

Xue Jiang, Yihong Dong, and Lecheng Wang et al. 2023. Self-planning code generation with large language models.

Seungone Kim, Se June Joo, and Doyoung Kim et al. 2023. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning.

Takeshi Kojima, Shixiang (Shane) Gu, and Reid et al. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Jacky Liang, Wenlong Huang, and Xia et al. 2023. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500.

Bill Yuchen Lin, Chengsong Huang, and Qian et al. Liu. 2023. On grounded planning for embodied tasks with language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):13192–13200.

Corey Lynch, Ayzaan Wahid, and Tompson et al. 2023. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, pages 1–8.

Sewon Min, Xinxi Lyu, and Ari et al. Holtzman. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, and Xu et al. Jiang. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Guilherme Penedo, Quentin Malartic, and Daniel Hesslow et al. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only.

Ofir Press, Muru Zhang, and Sewon Min et al. 2023. Measuring and narrowing the compositionality gap in language models.

Ansh Radhakrishnan, Karina Nguyen, and Anna Chen et al. 2023. Question decomposition improves the faithfulness of model-generated reasoning.

Jack W. Rae, Sebastian Borgeaud, and Trevor Cai et al. 2022. Scaling language models: Methods, analysis & insights from training gopher.

Jingqing Ruan, Yihong Chen, and Bin Zhang et al. 2023. Tptu: Task planning and tool usage of large language model-based ai agents.

KaShun Shum, Shizhe Diao, and Tong Zhang. 2023. Automatic prompt augmentation and selection with chain-of-thought from labeled data.

Ishika Singh, Valts Blukis, and Mousavian et al. 2023. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530.

Hugo Touvron, Louis Martin, and Kevin Stone et al. 2023. Llama 2: Open foundation and fine-tuned chat models.

Lei Wang, Chen Ma, and Xueyang Feng et al. 2023a. A survey on large language model based autonomous agents.

Lei Wang, Wanyu Xu, and Yihuai et al. Lan. 2023b. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

Peifeng Wang, Zhengyang Wang, and Zheng et al. Li. 2023c. SCOTT: Self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5546–5558, Toronto, Canada. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, and Dale Schuurmans et al. 2023d. Self-consistency improves chain of thought reasoning in language models.

Zhenhailong Wang, Shaoguang Mao, and Wenshan Wu et al. 2023e. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration.

Jason Wei, Maarten Bosma, and Vincent Y. Zhao et al. 2022a. Finetuned language models are zero-shot learners.

Jason Wei, Xuezhi Wang, and Schuurmans et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Tomer Wolfson, Mor Geva, and Gupta et al. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Shunyu Yao, Dian Yu, and Jeffrey Zhao et al. 2023a. Tree of thoughts: Deliberate problem solving with large language models.

Shunyu Yao, Jeffrey Zhao, and Dian Yu et al. 2023b. React: Synergizing reasoning and acting in language models.

Ori Yoran, Tomer Wolfson, and Ben Bogin et al. 2023. Answering questions by meta-reasoning over multiple chains of thought.

Jiajie Zhang, Shulin Cao, and Tingjian et al. Zhang. 2023a. Reasoning over hierarchical question decomposition tree for explainable question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14556–14570, Toronto, Canada. Association for Computational Linguistics.

Yifan Zhang, Jingqin Yang, and Yang Yuan et al. 2023b. Cumulative reasoning with large language models.

Zhuosheng Zhang, Aston Zhang, and Mu Li et al. 2023c. Multimodal chain-of-thought reasoning in language models.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models.

Denny Zhou, Nathanael Schärli, and Le Hou et al. 2023. Least-to-most prompting enables complex reasoning in large language models.

9. Language Resource References

Karl Cobbe and Vineet Kosaraju and Mohammad Bavarian et al. 2021. *Training Verifiers to Solve Math Word Problems*. Computing Research Repository.

Geva, Mor and Khashabi, Daniel and Segal et al. 2021. *Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies*. MIT Press.

Koncel-Kedziorski, Rik and Hajishirzi, Hannaneh and Sabharwal et al. 2015. *Parsing Algebraic Word Problems into Equations*. MIT Press.

Ling, Wang and Yogatama, Dani and Dyer, Chris and Blunsom, Phil. 2017. *Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems*. Association for Computational Linguistics.

Roy, Subhro and Roth, Dan. 2015. *Solving General Arithmetic Word Problems*. Association for Computational Linguistics.

Trivedi, Harsh and Balasubramanian, Niranjan and Khot, Tushar and Sabharwal, Ashish. 2022. *MuSiQue: Multihop Questions via Single-hop Question Composition*. MIT Press.

Wolfson, Tomer and Geva, Mor and Gupta et al. 2020. *Break It Down: A Question Understanding Benchmark*. MIT Press.