Puzzle Solving using Reasoning of Large Language Models: A Survey

Anonymous ACL submission

Abstract

Exploring the capabilities of Large Language Models (LLMs) in puzzle solving unveils critical insights into their potential and challenges in AI, marking a significant step towards understanding their applicability in complex reasoning tasks. This survey leverages a unique taxonomy—dividing puzzles into rule-based and rule-less categories—to critically assess LLMs through various methodologies, including prompting techniques, neuro-symbolic approaches, and fine-tuning. Through a critical review of relevant datasets and benchmarks, we assess LLMs' performance, identifying significant challenges in complex puzzle scenarios. Our findings highlight the disparity between LLM capabilities and human-like reasoning, particularly in those requiring advanced logical inference. The survey underscores the necessity for novel strategies and richer datasets to advance LLMs' puzzle-solving proficiency and contribute to AI's logical reasoning and creative problem-solving advancements.

1 Introduction

Recent developments in LLMs such as GPT-3 (Brown et al., 2020) and GPT-4 (OpenAI et al., 2023) have showcased their logical reasoning abilities across various domains (Liu et al., 2023a,b; Bao et al., 2023; Creswell et al., 2022). Despite these advances and their demonstrated capabilities in deductive reasoning (Saparov et al., 2023), LLMs face limitations in inductive reasoning settings, as analyzed by Xu et al. (2023a); Bang et al. (2023). The specific application of LLMs to puzzle solving, has not been thoroughly summarized.

Our main contributions are as follows: (1) We introduce a distinction between rule-based and rule-less puzzles (§2), highlighting the varied knowledge demands necessary to tackle them. (2) We analyze the methodologies LLMs use to solve puzzles (§3), assessing their impact on each category

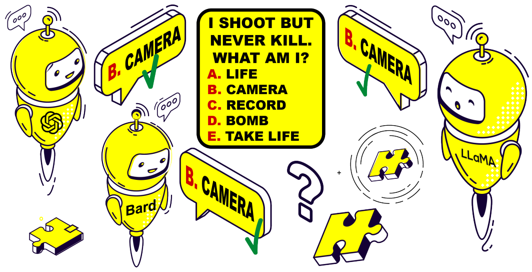

Figure 1: Riddle from RiddleSense (Lin et al., 2021). GPT-4, LLaMA2-70B and Bard chose the right answer.

and comparing them with conventional problem-solving methods. (3) A detailed exploration of existing benchmarks that gauge models' reasoning abilities is conducted (§4). (4) Finally, this paper offers a detailed view of the present obstacles faced in puzzle-solving with LLMs and highlights a wide array of prospects for future research (§5).

Our categorization diverges from existing logical reasoning taxonomies by emphasizing on the underlying cognitive processes and the skills required for puzzle solving, rather than the question format (Luo et al., 2023) or the nature of reasoning (deductive, inductive, abductive) (Luo et al., 2023; Yu et al., 2023a; Yang et al., 2023b; Qiao et al., 2022; Huang and Chang, 2022; Flach and Kakas, 2000). For instance, the existence of rules in puzzles such as Sudoku, Crosswords, or Minesweeper necessitates additional skills (e.g. strategy development) to correctly understand the game's rules or the ability to correctly format the output. In contrast, rule-less puzzles, such as riddles (Figure 1), programming challenges, and commonsense reasoning problems, leverage the model's inherent knowledge for solution derivation.

In our work, we define puzzles as problems that test cognitive abilities including logical reasoning, spatial cognition, and creative thinking by requiring the solver to discern patterns, apply deduction, and combine insights from available information in order to arrive at the correct solution. Notably,

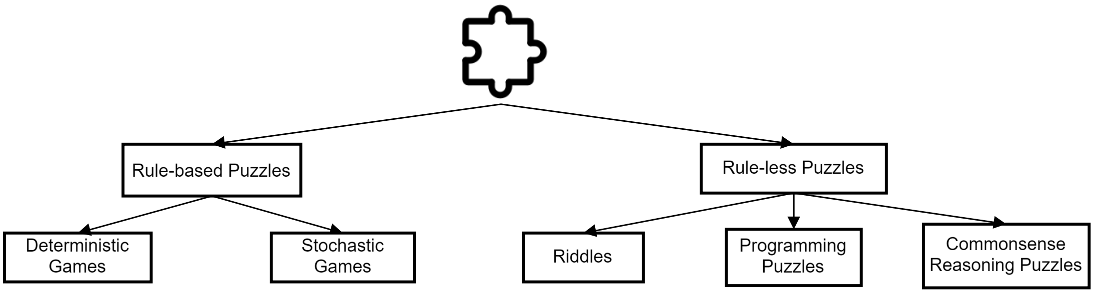

Figure 2: Taxonomy of Puzzles

we exclude puzzles that cannot be expressed in text in any way, such as jigsaw puzzles (Markaki and Panagiotakis, 2022), or problems that require multi-modal understanding abilities of LLMs (Chia et al., 2024; Ghosal et al., 2024). Mathematical puzzles are also excluded, as this area diligently covered by the recent work of Liu et al. (2023c).

2 Categorization of Puzzle Problems

In assessing LLMs' reasoning capabilities, it is essential to categorize puzzles into coherent groups. We distinguish puzzles by their reliance on formal rules or broader world knowledge accompanied by general inferential skills, as illustrated in Figure 2. This categorization not only highlights the cognitive diversity puzzles present, but also aligns with distinct reasoning challenges: rule-based puzzles demand logical deduction and strategic foresight within closed environments with defined parameters, whereas rule-less puzzles require general reasoning abilities, interpreting situations and explaining events by drawing inferences based on practical knowledge about the everyday world.

By separating puzzles into these categories, we aim to provide a nuanced analysis of LLMs' problem-solving abilities, reflecting on both structured challenges and those necessitating broader inferential reasoning.

2.1 Rule-based Puzzles

Rule-based Puzzles provide the model with explicit victory conditions, legal move sets or state transition rules. We further subdivide this category based on whether the state transitions are deterministic or incorporate randomness.

Deterministic games always produce the same successor state given a current game state and action taken according to the rules. For example, in Chess, making a move always yields one unambiguous new board layout. Other examples include Sudoku, maze navigation, or solving a Rubik's cube. The model should learn strategies that operate within the possibility space defined by legal game mechanics.

Stochastic games incorporate randomness or hidden information, i.e. the same player action can lead to different probability distributions over next states. Examples include Minesweeper (hidden bomb locations) or card games e.g. Poker where opponents hold private hands. Mastering these games requires reasoning over uncertain states, planning multiple moves in advance and managing risk.

Thus, while both subgroups require logical reasoning bounded by formal rules, stochastic games pose the additional challenge of decision-making under uncertainty. Excelling in deterministic games enables pure reliance on deduction and forward search, while stochastic environments also require abilities for probabilistic inference, risk analysis, and reasoning with incomplete information.

2.2 Rule-less Puzzles

Unlike rule-bounded puzzles, rule-less problems rely more on flexible thinking and real-world knowledge to interpret vague situations and infer unobserved details. Rather than testing systematic search or strategic planning, these puzzles measure cognitive skills for contextual interpretation, conceptual combination, and reasoning from common experiences. The following fall under this category.

Riddles utilize clever wordplay and literary devices to conceal answers. For example, "What gets wetter the more it dries?" obscures the solution of "a towel" through metaphor. Solving riddles requires making abstract connections between concepts hidden in lyrical language. This assesses skills for fluid reasoning, conceptual blending, and lateral thinking to decode linguistic relationships.

Programming Puzzles provide code snippets and require analyzing or modifying the underlying program logic. Schuster et al. (2021) define a programming puzzle as a short Python program f, and the goal is to find an input which makes f return True. Such puzzles assess skills like tracing execution, fixing errors, or anticipating outputs based on coding semantics. For example, the following puzzle tests understanding programming semantics to predict a system's behaviour:

```python
def mystery(x):
    return x // 2
print(mystery(10))
```

Commonsense Reasoning Puzzles depict typical situations omitting key details. Solvers must explain events by inferring plausible implicit assumptions about motivations, causes and effects. For instance, the question "A man who was outside in the rain without an umbrella or hat didn't get a single hair on his head wet. Why?" requires pragmatic analysis of unstated contextual factors.

3 Methods and Strategies

In applying LLMs to puzzle solving, a wide array of methods and strategies enhances complex reasoning and performance. This section outlines the approaches used to address puzzles, aiming to highlight their application within this unique context. Given the extensive literature on prompt engineering and related methods Besta et al. (2024); Chen et al. (2023); Yu et al. (2023b); Chu et al. (2023); Qiao et al. (2022); Liu et al. (2021), we concentrate on the techniques most prevalent for puzzle solving, instead of describing each method separately. We divide existing methods into prompting techniques, neuro-symbolic approaches for puzzle translation and fine-tuning for specific domains. A detailed overview of the methods utilized across different puzzle categories is presented in Table 1. We also discuss how conventional methods have faced these problems before the LLM era (App. A.2).

3.1 Prompting Methods

Prompting strategies that provide intermediate reasoning steps are pivotal in enhancing the puzzle-solving capabilities of language models. The **few-shot in-context learning** paradigm offers one or more demonstrations within prompts, significantly improving performance for both rule-based and rule-less puzzles by showcasing the reasoning process without additional training (Brown et al., 2020; Dong et al., 2023; Zhou et al., 2022).

Recent works focus on how different 'thought structures' can guide LLMs to the final solution.

Chain topologies, which include **Chain-of-Thought (CoT)** (Wei et al., 2022; Kojima et al., 2022) have been applied to all kinds of puzzles, demonstrating their superiority over simple IO prompts. **Self-Refine** (Madaan et al., 2023) is used for the Game of 24 (rule-based/deterministic), outperforming CoT with a 13% higher success rate (Yao et al., 2023). Gu et al. (2023) use **Automatic CoT** (Zhang et al., 2022), **Complexity CoT** (Zhang et al., 2022) and **Plan-and-Solve** (Wang et al., 2023a) in a rule-less detective-style benchmark, with none of the methods clearly outperforming CoT across all tested LLMs. The best results are achieved by **Detective Thinking Prompt**, a CoT-like method introduced in the same study, which does not exceed the 61.6% accuracy score of the best model, GPT-4. Schuster et al. (2021) exclusively utilized the solutions to programming puzzles that the model had already solved as examples, surpassing alternative approaches.

Tree topologies cover a variety of methods. **Self-Consistency (SC)** (Wang et al., 2022) has been tested on rule-based/deterministic puzzles, such as the 8-puzzle, Game of 24 and Pocket Cube, as well as on rule-less commonsense reasoning puzzles, showcasing a small gain in the first category over CoT (Ding et al., 2023; Yao et al., 2023; Mo and Xin, 2023) and no clear benefit in the second one (Gu et al., 2023). **Tree-of-Thought(s) (ToT)** (Yao et al., 2023; Long, 2023) has been exclusively applied to rule-based/deterministic puzzles so far, achieving significantly improved success rates over CoT, with increases ranging from 26% (Mo and Xin, 2023) to 70% (Yao et al., 2023) depending on the puzzle and the depth of the tree, despite the increased LLM invocations (Ding et al., 2023). **Tree-of-Uncertain-Thought (TouT)** (Mo and Xin, 2023) achieved even better results than ToT on the same challenges, with a 9% higher success rate on the Game of 24 and 3% on mini-crosswords. Finally, **Inference-Exclusion-Prompting (IEP)** (Tong et al., 2023) delivered some of the best results on riddles and commonsense puzzles when combined with CoT, scoring 82% on puzzles–up from 81% with zero-shot CoT–and 79% on riddles, compared to 82% with zero-shot CoT.

Graph topologies entail the following: **Graph-of-Thought(s) (GoT)** (Besta et al., 2023; Lei et al., 2023) and **Everything-of-Thought (XoT)** (Ding et al., 2023) have been used to solve rule-based/deterministic puzzles. While GoT has shown poorer results compared to ToT, with a decrease ranging from 2% to 6% (Ding et al., 2023), XoT has been recognized as the most effective method for these puzzles, achieving improvements in results from 53% to 69% compared to ToT, while presenting the fewest LLM invocations among the methods tested, including CoT, SC, ToT, and GoT.

A brief analysis of the aforementioned thought structures is presented in Appendix A.1, while a more detailed one can be found in the work of Besta et al. (2024). Beyond the aforementioned methods, the use of extra information such as **hints** for riddles and commonsense puzzles, or **introductions** and **summarizations** of the puzzles, has also been employed. The inclusion of supplementary details appears to yield positive results, although this is not always the case; for instance, Chinese riddles typically show worse results when hints are used (Zhang and Wan, 2021).

3.2 Puzzle Translation

In this subsection, we summarize the **neuro-symbolic techniques** used by LLMs to translate text puzzles from natural language into forms more amenable to solutions by external tools. Notably, these methods do not test the LLMs' puzzle solving capacity but rather assess their ability to encode puzzles into appropriate representations.

The primary approach involves using LLMs to generate **logic rules** from the puzzle's natural language and subsequently solve it using a symbolic solver. Ishay et al. (2023) employ GPT-3 and GPT-4 to transform logic puzzles, such as chess puzzles, Jobs puzzle and Sudoku (rule-based/deterministic) into Answer Set Programming (ASP) formats by generating predicates and rules. They demonstrate that this method achieved significant results, with GPT-4 scoring 92% accuracy in a logic puzzles dataset Mitra and Baral (2015), compared to 7% in few-shot and 21% in zero-shot settings with the same model. They note that in few-shot settings, LLMs can generate complex programs that humans can easily refine and correct in case of code errors. Additionally, similar frameworks such as Logic-LM (Pan et al., 2023a), LINC (Olausson et al., 2023) and Yang et al. (2023a)'s method show promising results in logical reasoning tasks, although not specifically in puzzle settings.

While neuro-symbolic approaches have been applied to puzzle translation into logic rules, we have found no studies on transforming puzzles from natural language into **code**. However, techniques such as Program of Thoughts (PoT) prompting (Chen et al., 2022) and Program-Aided Language (PAL) (Gao et al., 2022) employ models to convert reasoning into Python programs for logical and mathematical reasoning datasets. Therefore, we encourage the research community to explore these methods for puzzle-solving tasks as well.

Given the structured nature of rule-based puzzles, this approach is inherently suitable for them. Consequently, it is logical that no studies have yet been conducted on rule-less puzzles in this context.

3.3 Fine-Tuning

Fine-tuning LLMs emerges as a potent strategy for enhancing their reasoning capabilities, ranging from general logical reasoning to specific puzzle-solving skills. Models such as LoGiPT (Feng et al., 2023a) and LogiT5 (Luo et al., 2023) demonstrate improved logical reasoning, mimicking human-like problem-solving processes. In the realm of riddles, the study of Lin et al. (2021) illustrates that models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2019) perform better when trained on both RiddleSense Lin et al. (2021) and CommonsenseQA (Talmor et al., 2019) datasets, leveraging commonsense knowledge effectively. Moreover, Zhang and Wan (2021) report that combining fine-tuning on ALBERT-XXL with transfer learning from CommonsenseQA achieved the highest accuracy, noting a 4% improvement over simple fine-tuning. In the domain of rule-based deterministic puzzles, Noever and Burdick (2021) observe suboptimal results when fine-tuning GPT-2 on Sudoku, Rubik's Cube and Mazes, potentially due to a brief fine-tuning period and limited training examples. Regarding crosswords, various studies (Rozner et al., 2021; Efrat et al., 2021) show mixed results, with some fine-tuned LLMs outperforming non-neural baselines and others not, highlighting the inherent challenge of cryptic crosswords for LLMs. Kazemi et al. (2023) demonstrate that fine-tuning LLMs with proofs and CoT under rule-based contexts yields some of the best results. Lastly, the effectiveness of fine-tuning extends to commonsense reasoning (Del and Fishel, 2022) and programming puzzles (Schuster et al., 2021), showcasing its broad applicability across puzzle categories.

4 Datasets, Benchmarks and Tasks

Exploring diverse datasets, benchmarks, and tasks is crucial for evaluating LLMs in puzzle-solving. This section examines datasets within our puzzle taxonomy, encompassing formats, evaluation metrics, and methodologies. Table 2 provides a detailed summary of datasets utilized across the taxonomy's categories, organized according to puzzle type. The analysis demonstrates LLMs' versatility and the impact of techniques discussed in §3.

4.1 Rule-based Puzzles

We explore rule-based puzzles to assess LLMs' understanding within structured, closed-world environments. This includes deterministic puzzles such as Sudoku, Rubik's Cube, Crosswords, and the 8-puzzle, where solutions follow a set of defined rules. In contrast, stochastic games e.g. Minesweeper, card and social deduction games present variable outcomes from the same actions due to hidden factors. Research predominantly focuses on deterministic puzzles, highlighting a gap in addressing stochastic puzzle uncertainties—a promising direction for future research.

4.1.1 Deterministic Puzzles

Sudoku serves as a prime benchmark for LLMs due to its logical complexity. Noever and Burdick (2021) fine-tune GPT-2 (Radford et al., 2019) on 1M Sudoku games, experimenting with compact single-string format, with empty cells represented by "-", and posited that a matrix representation may enhance the model's learning efficacy. Long (2023) uses nested lists for puzzle representation[1], finding the Tree-of-Thought (ToT) method most effective, especially for smaller puzzles. Ishay et al. (2023) explore neuro-symbolic approaches across Sudoku, Jobs puzzles and logic puzzles, demonstrating that well-prompted LLMs can accurately generate answer set programming rules.

For **Rubik's Cube** and **Maze solvers**, Noever and Burdick (2021) assess GPT-2's spatial reasoning using over 2,400 Rubik's Cube samples and 10K mazes. Despite limited fine-tuning and token constrains, GPT-2 successfully solved the Rubik's Cube in 1 out of 7 attempts, showing potential despite a high rate of valid though incorrect solutions. Ding et al. (2023) apply multiple methods such as CoT, Self-Consistency, and various Thoughts (ToT, GoT, XoT) on a 2×2×2 Rubik's Cube using GPT-3.5 and GPT-4. XoT with self-revision emerges as most accurate, significantly outperforming others with a 77.6% success rate.

Exploring LLM versatility, Ding et al. (2023) evaluate the effectiveness of XoT on the spatial **8-Puzzle** and numerical **Game of 24**. The 8-Puzzle's goal configuration challenges are solved with a remarkable 93.2% accuracy across 419 puzzles using XoT with revision, showcasing superior efficiency over few-shot prompting and CoT. This high accuracy, coupled with a reduced number of LLM invocations, underscores the efficiency and potential of XoT in complex puzzle-solving contexts.

As for **Crosswords**, Rozner et al. (2021) and Efrat et al. (2021) fine-tune T5 models (Raffel et al., 2019) on extensive datasets of individual cryptic clues, revealing T5's advantage over traditional methods and highlighting areas for improvement, particularly with quick clues and specified answer lengths. Kulshreshtha et al. (2022)'s comparison of BART (Lewis et al., 2019) and T5 indicate a sub-30% accuracy for clue-answer tasks, with retrieval-augmented generation transformers surpassing fine-tuned LLMs. Additionally, Yao et al. (2023) apply 5-shot prompting and ToT to GPT-4 on Crossword puzzles significantly improving performance by solving 4 out of 20 puzzles and achieving a 60% word-level success rate.

Feng et al. (2023b) fine-tune two models, "ChessGPT" and "ChessCLIP," using a collection of 3.2M **chess puzzles** from the Lichess dataset[2]. Each puzzle in the dataset include annotations for its rating, theme, and solution.

At last, Kazemi et al. (2023) unveil **BoardgameQA**, a dataset featuring multi-choice questions against a backdrop of contradictory facts and rules. Models should navigate through these complexities to provide free-text answers. Their evaluation reveals that fine-tuning BERT-large and T5-XXL with proofs emerges as the most effective method, contrary to few-shot prompting on PaLM with CoT. Moreover, the presence of extra or conflicting information decreases accuracy.

4.1.2 Stochastic Puzzles

The **BoardgameQA** benchmark (Kazemi et al., 2023) also explores scenarios with missing information, which fall under the stochastic puzzle category. It is shown that as missing information increases, the accuracy of fine-tuned models decreases. However, this heightened difficulty does not similarly impact the performance of prompt-tuned and few-shot learning methods, which is likely due to the larger models that were applied.

Minesweeper, known for its hidden information and unpredictability, exemplifies stochastic

[1] e.g. [[3,*,*,2], [1,*,3,*],[*,1,*,3],[4,*,*,1]]

[2] https://lichess.org/

puzzles, requiring players to deduce mine locations from numerical clues, challenging spatial reasoning. Li et al. (2023) evaluated LLMs on Minesweeper, comparing table and coordinate representations. Even though GPT-3.5 displayed initial understanding, enhancements like few-shot prompting had minimal effects. Conversely, GPT-4 improved mine identification but struggled to complete boards, highlighting Minesweeper's role in evaluating LLMs' strategic thinking. Experiments favored the coordinate representation over the table format for aiding LLM comprehension.

Card games, notably Poker, exemplify stochastic puzzles where strategic skill is crucial. Simplified Poker variants require players to infer opponents' cards and calculate odds amidst hidden intentions. Gupta (2023) found that in Poker's preflop round, ChatGPT and GPT-4 grasp advanced strategies but do not reach Game Theory Optimal (GTO) play. ChatGPT leans towards a conservative approach, while GPT-4 exhibits more aggressive gameplay. Huang et al. (2024) leverage a Reinforcement Learning-trained OPT-1.3B model on all Poker phases revealing superior outcomes in win rates and efficiency, ultimately showcasing LLMs' adeptness at complex strategies in stochastic settings. An agent that leverages GPT-4 (Guo et al., 2023) also achieves significant results in various imperfect information card games.

Social deduction games, including Werewolf and Avalon, blend logical reasoning with complex social dynamics, making them part of the broader stochastic puzzle domain. Such games challenge players to deduce roles involving unpredictable human behavior. Xu et al. (2023b) propose a Werewolf framework using LLMs without tuning, leveraging historical interactions for strategic decisions and showcasing the models' ability in this context. Similarly, frameworks for Avalon (Wang et al., 2023b; Lan et al., 2023) show how LLMs can navigate scenarios demanding social manipulation and deduction, underscoring LLMs' proficiency in managing the complex interplay of logic and social interaction inherent in such games.

4.2 Rule-less Puzzles

This subsection delves into the diverse datasets related to rule-less puzzles, a category that predominantly encompasses riddles, programming puzzles, and commonsense reasoning challenges. Notably, we specifically focus on puzzles in their traditional sense, thereby excluding code generation datasets, which represent a distinct task type. A majority of rule-less puzzles are structured in a multiple-choice question-answering (QA) format, offering a standardized approach for evaluating LLMs' inferential reasoning. Benchmarks deviating from this format are specially mentioned, providing a broader perspective on the variety of rule-less puzzle datasets and their implications for LLM performance.

4.2.1 Riddles

RiddleSense (Lin et al., 2021) offers a collection of 5.7K vertical thinking riddles, testing pre-trained LMs such as BERT, RoBERTa, ALBERT, and text-to-text QA models including UnifiedQA (Khashabi et al., 2020) and T5. Larger LMs generally demonstrate better performance, with UnifiedQA using T5-3B leading, yet struggling with metaphors and counterfactual situations.

Complementing this, **BrainTeaser** (Jiang et al., 2023) introduces 1119 lateral thinking puzzles. It contrasts instruction-based models (ChatGPT, T0, and FlanT5 (Chung et al., 2022)) with commonsense ones (including RoBERTa variants and CAR (Wang et al., 2023c)). ChatGPT excels in both sentence-based and word-based puzzles, indicating its strength in lateral thinking. However, overall, LLMs still face challenges in exhibiting lateral thinking, with common errors in memorization and commonsense association. This dataset highlights the varied dimensions of reasoning that riddles can test, from vertical logic to lateral inference.

BiRdQA (Zhang and Wan, 2021) explores the multilingual aspect of riddles, encompassing English and Chinese puzzles, while evaluating monolingual LMs (BERT, RoBERTa), as well as multilingual ones (mBERT, XLM-R (Conneau et al., 2019)). The use of brief riddle introductions and hints is also tested. Findings reveal a significant performance gap between LMs and human-level understanding, with monolingual models generally outperforming multilingual ones. Interestingly, additional context such as Wikipedia introductions and hints varied in effectiveness, with such aids benefiting English but not Chinese riddles.

CC-Riddle centers on 27K Chinese character riddles, involving multiple-choice, generative, and retrieval-based formats (Xu et al., 2022). Evaluation demonstrates that models encountered difficulties in comprehension and exhibited misunderstandings, revealing the complexities inherent in character-based riddles.

In contrast, **PUZZLEQA** (Zhao and Anderson,

2023) offers 558 word puzzles in multiple choice and free-text formats. Larger models, e.g. GPT-3/3.5 show higher accuracy, especially in multiple-choice settings. However, methods such as CoT combined with summarization do not significantly enhance performance, pointing to the ongoing challenges in free-response puzzle solving.

Finally, **MARB** (Tong et al., 2023) encompasses a variety of riddle tasks. Several methodologies including zero-shot, CoT, IEP, and few-shot prompting are tested on models such as GPT-4 and PaLM2-540B (Anil et al., 2023). The combination of IEP and CoT emerged as the most effective method, highlighting the value of integrating multiple approaches for diverse riddle types. The dataset also includes commonsense puzzles (§4.2.3), showing similar trends with riddles.

4.2.2 Programming Puzzles

P3 (Python Programming Puzzles) (Schuster et al., 2021) offers a range of Python programming challenges, from straightforward string manipulations to complex tasks, such as the Tower of Hanoi and algorithmic puzzles, requiring from the model to find an input that makes the program f return "True". Models applied to these puzzles include enumerative solvers for building Abstract Syntax Trees and autoregressive Language Model Solvers such as GPT-3 and Codex (Chen et al., 2021), employing varied prompting techniques. The evaluation metric pass@k, indicates the models' ability to solve a puzzle within a given number of attempts (Chen et al., 2021). Results show a correlation between puzzle difficulty for both models and humans, with descriptive prompts enhancing model performance. Interestingly, models proficient in code completion solved more puzzles with fewer tries, highlighting the importance of specialized capabilities in programming challenges.

Savelka et al. (2023) introduce a dataset comprised of 530 code snippets from programming courses, presenting puzzles in a multiple-choice format. The distinction between questions with and without code snippets offers a unique perspective on LLMs' problem-solving strategies. The dataset categorizes questions into six types, including true/-false and output prediction. GPT models were evaluated, revealing that code inclusion significantly increases puzzle complexity. Accuracy rates vary, with higher performance on completion-oriented questions, suggesting that LLMs' effectiveness can depend heavily on question format and content.

While both P3 and Programming Snippets Dataset address programming puzzles, they do so in markedly different ways. P3's focus on finding correct Python program inputs contrasts with the multiple-choice format of the Programming Snippets Dataset. However, both datasets reveal key insights: descriptive prompts aid problem-solving, and question format significantly influences LLM performance.

4.2.3 Commonsense Reasoning Puzzles

True Detective (Del and Fishel, 2022) presents detective puzzles in long-form stories, challenging LLMs such as GPT-3.5/4 to draw conclusions. Various methods, including CoT and Golden-CoT are used, revealing difficulties in making final inferences despite all information being available. While Vanilla and CoT approaches perform close to random, Golden CoT shows significantly better accuracy, especially on GPT-4.

DetectBench (Gu et al., 2023) containing 1200 questions, also evaluates informal reasoning in real-life contexts. It tests methods such as use of hints, various CoT approaches and detective thinking on models including GPT-4, GPT-3.5, GLM-4 and Llama2. Hints emerges as a powerful aid, with larger models generally outperforming smaller ones. The effectiveness of different approaches vary, with detective thinking effectively assisting most of the models.

Both datasets highlight the complexity of real-life reasoning and detective-style puzzles, demonstrating that hints play a crucial role in aiding both human and model performance.

LatEval (Huang et al., 2023b) introduces a conversational format with English and Chinese stories, requiring players to ask yes/no questions before providing an answer. GPT-3.5, GPT-4, and various other Chat models are evaluated on their ability to ask relevant questions and maintain consistency with the truth. Larger models do not necessarily show advanced performance in question relevance. However, GPT-4 demonstrates the highest answer consistency, though there is still significant room for improvement. The dataset emphasizes the importance of interactive and conversational reasoning in commonsense understanding.

PuzzTe (Szomiu and Groza, 2021), with its array of comparison, knights and knaves, and zebra puzzles, represents a potentially rich resource for LLM testing. Despite not yet being applied to LLMs, its generated puzzle answers by Mace4 model finder and Prover9 theorem prover[3] indicate its potential for future LLM evaluations.

The datasets under investigation demonstrate a variety of methods for evaluating commonsense reasoning in LLMs, ranging from detective-style puzzles to interactive story solving. Although larger models generally exhibit better performance, the complexity of these tasks poses significant challenges. Techniques such as sharing additional information through hints show effectiveness in improving outcomes, yet there remains a considerable gap between the performance of models and humans. It is important to note that in this work, we specifically focus on puzzle-oriented benchmarks, excluding general commonsense reasoning datasets e.g. CommonsenseQA, PIQA (Bisk et al., 2019) or StrategyQA (Geva et al., 2021).

5 Discussion and Future Directions

Applied Methods and Dataset Gaps: Across our puzzle taxonomy (Figure 2), the selection of methods such as few-shot prompting, CoT, introductions and fine-tuning is common across most categories. Rule-based deterministic and rule-less commonsense puzzles show the greatest methodological variety, while riddles are also see diverse approaches. In contrast, rule-based stochastic and rule-less programming puzzles exhibit less variety, likely due to fewer studies in these areas. The lack of benchmarks for stochastic puzzles prompted us to include tasks like card and social deduction games, which share core characteristics with traditional puzzles. This highlights the need for more specialized datasets that adhere closely to defined puzzle structures with missing information elements. Additionally, neuro-symbolic techniques that translate natural language into code remain notably underutilized in puzzle benchmarks, suggesting a potential area for future exploration.

Performance Analysis:

Rule-based / Deterministic: Methods such as ToT and XoT (§ 3), typically enhance model reasoning abilities as the complexity of the structure increases (Ding et al., 2023). Yet, studies in BoardgameQA and crossword puzzles show generally poor model performance.

Rule-based/Stochastic: Fine-tuning is prevalent here, enabling LLMs to grasp basic rules and simpler scenarios. However, they falter in complex settings that require extensive multi-step reasoning (Li et al., 2023).

Rule-less/Riddles & Commonsense: There is a notable performance gap between LLMs and human levels, with methods like CoT improving accuracy but still not matching human evaluation outcomes.

Rule-less/Programming: LLMs find programming puzzles challenging, paralleling human difficulties (Schuster et al., 2021). Tasks involving code analysis and reasoning in multiple-choice formats prove particularly tough (Savelka et al., 2023).

Furthermore, the format of questions significantly affects puzzle-solving effectiveness. Multiple-choice setups simplify tasks for LLMs by narrowing the solution search space, while free-text formats increase the difficulty level.

Puzzle Generation research is currently limited, likely because the ability to understand and solve puzzles is a prerequisite for generating them. In our survey, we primarily focused on puzzle-solving. The few works we found in puzzle generation reveal mixed results. For instance, GPT-3.5's attempts to generate puzzles with answers showed poor outcomes (Zhao and Anderson, 2023). Conversely, the introduction of ACES, an autotelic generation method for diverse programming puzzles, demonstrates how semantic descriptors produced by LLMs can be leveraged for creative puzzle creation (Pourcel et al., 2023). Lastly, there are recent works that have studied the generation of crossword puzzles of different languages, utilizing LLMs (Zugarini et al., 2024; Zeinalipour et al., 2023b,a).

6 Conclusion

In this survey, we propose a taxonomy of puzzles for evaluating LLMs, categorizing them into rule-based (deterministic and stochastic) and rule-less puzzles (riddles, programming, and commonsense reasoning puzzles). We explore a spectrum of methods for LLM-based puzzle solving, ranging from prompting techniques to neuro-symbolic strategies and fine-tuning. By collating existing datasets in this domain, we provide a comprehensive overview of the resources available for such evaluations. Our analysis identifies current challenges, revealing a difficulty of most methods to successfully solve puzzles, while we outline future directions, emphasizing the need for advanced methodologies and diverse datasets to enhance LLMs' proficiency in puzzle solving.

[3] https://www.cs.unm.edu/ mccune/prover9/

7 Limitations

In this study, we provide a survey of puzzle solving using reasoning of Large Language Models. Despite our best efforts, there may be still some limitations that remain in this paper. Firstly, due to the rapidly evolving nature of this field, we continuously add related approaches and analyses, but it is possible that some recent developments may not be included. Also, due to page constraints, we cannot extensively present all the methods nor provide all the technical details. This might limit the depth of understanding for some readers. Our review only includes methods within 4 years, primarily from sources such as ACL, EMNLP, NAACL, NeurIPS, ICLR, and arXiv. We plan to continue following these sources and adding new methods and datasets. Additionally, all our conclusions §6 are based on empirical analysis. While this provides robust evidence, it may not capture all aspects of the problem. Lastly, as with any survey, our interpretations and conclusions §5 are influenced by our own perspectives and understanding of the field. Other researchers might interpret the same studies differently. Despite these limitations, we believe this study provides a valuable overview of the current state of puzzle-solving using reasoning of Large Language Models.

References

Forest Agostinelli, Stephen Marcus McAleer, Alexander Shmakov, and Pierre Baldi. 2019. Solving the rubik's cube with deep reinforcement learning and search. *Nature Machine Intelligence*, 1:356 – 363.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Tachard Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Z. Chen, Eric Chu, J. Clark, Laurent El Shafey, Yanping Huang, Kathleen S. Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Michael Brooks, Michele Catasta, Yongzhou Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, C Crépy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, M. C. D'iaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fan Feng, Vlad Fienber, Markus Freitag, Xavier García, Sebastian Gehrmann, Lucas González, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, An Ren Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wen Hao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Mu-Li Li, Wei Li, Yaguang Li, Jun Yu Li, Hyeontaek Lim, Han Lin, Zhong-Zhong Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Oleksandr Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alexandra Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Marie Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniela Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Ke Xu, Yunhan Xu, Lin Wu Xue, Pengcheng Yin, Jiahui Yu, Qiaoling Zhang, Steven Zheng, Ce Zheng, Wei Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report. *ArXiv*, abs/2305.10403.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *ArXiv*, abs/2302.04023.

Qiming Bao, Gaël Gendron, Alex Yuxuan Peng, Wanjun Zhong, Neşet Özkan Tan, Yang Chen, Michael Witbrock, and Jiamou Liu. 2023. A systematic evaluation of large language models on out-of-distribution logical reasoning tasks. *ArXiv*, abs/2310.09430.

Houda Nait El Barj and Theophile Sautory. 2024. Reinforcement learning from llm feedback to counteract goal misgeneralization.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2023. Graph of thoughts: Solving elaborate problems with large language models.

Maciej Besta, Florim Memedi, Zhenyu Zhang, Robert Gerstenberger, Guangyuan Piao, Nils Blach, Piotr Nyczyk, Marcin Copik, Grzegorz Kwaśniewski, Jürgen Müller, Lukas Gianinazzi, Ales Kubicek, Hubert Niewiadomski, Aidan O'Mahony, Onur Mutlu, and Torsten Hoefler. 2024. Demystifying chains, trees, and graphs of thoughts.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *ArXiv*, abs/1911.11641.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack

Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Murray Campbell, A.Joseph Hoane, and Feng hsiung Hsu. 2002. Deep blue. *Artificial Intelligence*, 134(1):57–83.

Banghao Chen, Zhaofeng Zhang, Nicolas Langren'e, and Shengxin Zhu. 2023. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *ArXiv*, abs/2310.14735.

Juntao Chen. 2022. Different algorithms to solve a rubik's cube. *Journal of Physics: Conference Series*, 2386(1):012018.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, Suchir Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *ArXiv*, abs/2211.12588.

Eric C. Chi and Kenneth Lange. 2013. Techniques for solving sudoku puzzles.

Yew Ken Chia, Vernon Toh Yan Han, Deepanway Ghosal, Lidong Bing, and Soujanya Poria. 2024. Puzzlevqa: Diagnosing multimodal reasoning challenges of language models with abstract visual patterns.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2023. A survey of chain of thought reasoning: Advances, frontiers and future. *ArXiv*, abs/2309.15402.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. In *Annual Meeting of the Association for Computational Linguistics*.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *ArXiv*, abs/2205.09712.

Maksym Del and Mark Fishel. 2022. True detective: A deep abductive reasoning benchmark undoable for gpt-3 and challenging for gpt-4. In *STARSEM*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ruomeng Ding, Chaoyun Zhang, Lu Wang, Yong Xu, Ming-Jie Ma, Wei Zhang, Si Qin, S. Rajmohan, Qingwei Lin, and Dongmei Zhang. 2023. Everything of thoughts: Defying the law of penrose triangle for thought generation. *ArXiv*, abs/2311.04254.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Avia Efrat, Uri Shaham, Dan Kilman, and Omer Levy. 2021. Cryptonite: A cryptic crossword benchmark for extreme ambiguity in language. *ArXiv*, abs/2103.01242.

Jiazhan Feng, Ruochen Xu, Junheng Hao, Hiteshi Sharma, Yelong Shen, Dongyan Zhao, and Weizhu Chen. 2023a. Language models can be logical solvers. *ArXiv*, abs/2311.06158.

Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Henry Mguni, Yali Du, and Jun Wang. 2023b. Chessgpt: Bridging policy learning and language modeling. *ArXiv*, abs/2306.09200.

Peter A. Flach and Antonis C. Kakas. 2000. Abductive and inductive reasoning: background and issues.

Yao Fu, Hao-Chun Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. *ArXiv*, abs/2210.00720.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *ArXiv*, abs/2211.10435.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Deepanway Ghosal, Vernon Toh Yan Han, Chia Yew Ken, and Soujanya Poria. 2024. Are language models puzzle prodigies? algorithmic puzzles unveil serious challenges in multimodal reasoning.

Zhouhong Gu, Zihan Li, Lin Zhang, Zhuozhi Xiong, Sihang Jiang, Xiaoxuan Zhu, Shusen Wang, Zili Wang, Jianchen Wang, Haoning Ye, Wenhao Huang, Yikai Zhang, Hongwei Feng, and Yanghua Xiao. 2023. Go beyond the obvious: Probing the gap of informal reasoning ability between humanity and llms by detective reasoning puzzle benchmark.

Jiaxian Guo, Bo Yang, Paul Yoo, Bill Yuchen Lin, Yusuke Iwasawa, and Yutaka Matsuo. 2023. Suspicion-agent: Playing imperfect information games with theory of mind aware gpt-4. *ArXiv*, abs/2309.17277.

Akshat Gupta. 2023. Are chatgpt and gpt-4 good poker players? - a pre-flop analysis. *ArXiv*, abs/2308.12466.

Chenghao Huang, Yanbo Cao, Yinlong Wen, Tao Zhou, and Yanru Zhang. 2024. Pokergpt: An end-to-end lightweight solver for multi-player texas hold'em via large language model. *ArXiv*, abs/2401.06781.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *ArXiv*, abs/2212.10403.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023a. Large language models cannot self-correct reasoning yet. *ArXiv*, abs/2310.01798.

Shulin Huang, Shirong Ma, Yinghui Li, Mengzuo Huang, Wuhe Zou, Weidong Zhang, and Haitao Zheng. 2023b. Lateval: An interactive llms evaluation benchmark with incomplete information from lateral thinking puzzles. *ArXiv*, abs/2308.10855.

Adam Ishay, Zhun Yang, and Joohyung Lee. 2023. Leveraging large language models to generate answer set programs. *ArXiv*, abs/2307.07699.

Yifan Jiang, Filip Ilievski, and Kaixin Ma. 2023. Brainteaser: Lateral thinking puzzles for large language models. In *Conference on Empirical Methods in Natural Language Processing*.

Mehran Kazemi, Quan Yuan, Deepti Bhatia, Najoung Kim, Xin Xu, Vaiva Imbrasaite, and Deepak Ramachandran. 2023. Boardgameqa: A dataset for natural language reasoning with contradictory information. *ArXiv*, abs/2306.07934.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. In *Findings*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *ArXiv*, abs/2205.11916.

Richard E. Korf. 1997. Finding optimal solutions to rubik's cube using pattern databases. In *AAAI/IAAI*.

Saurabh Kulshreshtha, Olga Kovaleva, Namrata Shivagunde, and Anna Rumshisky. 2022. Down and across: Introducing crossword-solving as a new nlp benchmark. *ArXiv*, abs/2205.10442.

Yihuai Lan, Zhiqiang Hu, Lei Wang, Yang Wang, De-Yong Ye, Peilin Zhao, Ee-Peng Lim, Hui Xiong, and Hao Wang. 2023. Llm-based agent society investigation: Collaboration and confrontation in avalon gameplay. *ArXiv*, abs/2310.14985.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.

Bin Lei, Pei-Hung Lin, Chunhua Liao, and Caiwen Ding. 2023. Boosting logical reasoning in large language models through a new framework: The graph of thought. *ArXiv*, abs/2308.08614.

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

Yinghao Li, Haorui Wang, and Chao Zhang. 2023. Assessing logical puzzle solving in large language models: Insights from a minesweeper case study. *ArXiv*, abs/2311.07387.

Bill Yuchen Lin, Ziyi Wu, Yichi Yang, Dong-Ho Lee, and Xiang Ren. 2021. Riddlesense: Reasoning about riddle questions featuring linguistic creativity and commonsense knowledge. In *Findings*.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yuexin Zhang. 2023a. Evaluating the logical reasoning ability of chatgpt and gpt-4. *ArXiv*, abs/2304.03439.

Hanmeng Liu, Zhiyang Teng, Ruoxi Ning, Jian Liu, Qiji Zhou, and Yuexin Zhang. 2023b. Glore: Evaluating logical reasoning of large language models. *ArXiv*, abs/2310.09107.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55:1 – 35.

Wentao Liu, Hanglei Hu, Jie Zhou, Yuyang Ding, Junsong Li, Jiayi Zeng, Mengliang He, Qin Chen, Bo Jiang, Aimin Zhou, and Liang He. 2023c. Mathematical language models: A survey.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Jieyi Long. 2023. Large language model guided tree-of-thought. *ArXiv*, abs/2305.08291.

Man Luo, Shrinidhi Kumbhar, Ming shen, Mihir Parmar, Neeraj Varshney, Pratyay Banerjee, Somak Aditya, and Chitta Baral. 2023. Towards logiglue: A brief survey and a benchmark for analyzing logical reasoning capabilities of language models. *ArXiv*, abs/2310.00836.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *ArXiv*, abs/2303.17651.

Smaragda Markaki and Costas Panagiotakis. 2022. Jigsaw puzzle solving techniques and applications: a survey. *The Visual Computer*, 39:4405 – 4421.

Stephen McAleer, Forest Agostinelli, Alexander Shmakov, and Pierre Baldi. 2018. Solving the rubik's cube without human knowledge.

Arindam Mitra and Chitta Baral. 2015. Learning to automatically solve logic grid puzzles. In *Conference on Empirical Methods in Natural Language Processing*.

Shentong Mo and Miao Xin. 2023. Tree of uncertain thoughts reasoning for large language models. *ArXiv*, abs/2309.07694.

David A. Noever and Ryerson Burdick. 2021. Puzzle solving without search or human knowledge: An unnatural language approach. *ArXiv*, abs/2109.02797.

Theo X. Olausson, Alex Gu, Benjamin Lipkin, Cedegao Zhang, Armando Solar-Lezama, Josh Tenenbaum, and Roger Levy. 2023. Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Conference on Empirical Methods in Natural Language Processing*.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023a. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *ArXiv*, abs/2305.12295.

Liangming Pan, Michael Stephen Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023b. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *ArXiv*, abs/2308.03188.

Julien Pourcel, Cédric Colas, Pierre-Yves Oudeyer, and Laetitia Teodorescu. 2023. Aces: Generating diverse programming puzzles with autotelic language models and semantic descriptors. *ArXiv*, abs/2310.10692.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Reasoning with language model prompting: A survey. *ArXiv*, abs/2212.09597.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Josh Rozner, Christopher Potts, and Kyle Mahowald. 2021. Decrypting cryptic crosswords: Semantically complex wordplay puzzles as a target for nlp. *ArXiv*, abs/2104.08620.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Seyed Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the general deductive reasoning capacity of large language models using ood examples. *ArXiv*, abs/2305.15269.

Jaromir Savelka, Arav Agarwal, Christopher Bogart, and Majd Sakr. 2023. Large language models (gpt) struggle to answer multiple-choice questions about code.

Tal Schuster, A. Kalyan, Oleksandr Polozov, and Adam Tauman Kalai. 2021. Programming puzzles. *ArXiv*, abs/2106.05784.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm.

Helmut Simonis. 2005. Sudoku as a constraint problem.

Chris Studholme. 2001. Minesweeper as a constraint satisfaction problem.

Roxana Szomiu and Adrian Groza. 2021. A puzzle-based dataset for natural language inference. *ArXiv*, abs/2112.05742.

Kyo Takano. 2023. Self-supervision is all you need for solving rubik's cube.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *ArXiv*, abs/1811.00937.

Yongqi Tong, Yifan Wang, Dawei Li, Sizhe Wang, Zi Lin, Simeng Han, and Jingbo Shang. 2023. Eliminating reasoning via inferring with planning: A new framework to guide llms' non-linear thinking. *ArXiv*, abs/2310.12342.

Gladys Tyen, Hassan Mansoor, Peter Chen, Tony Mak, and Victor Carbune. 2023. Llms cannot find reasoning errors, but can correct them! *ArXiv*, abs/2311.08516.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Annual Meeting of the Association for Computational Linguistics*.

Shenzhi Wang, Chang Liu, Zilong Zheng, Siyuan Qi, Shuo Chen, Qisen Yang, Andrew Zhao, Chaofei Wang, Shiji Song, and Gao Huang. 2023b. Avalon's game of thoughts: Battle against deception through recursive contemplation. *ArXiv*, abs/2310.01320.

Weiqi Wang, Tianqing Fang, Wenxuan Ding, Baixuan Xu, Xin Liu, Yangqiu Song, and Antoine Bosselut. 2023c. Car: Conceptualization-augmented reasoner for zero-shot commonsense question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Huai hsin Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171.

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

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Kang Liu, and Jun Zhao. 2022. Large language models are better reasoners with self-verification. In *Conference on Empirical Methods in Natural Language Processing*.

Fan Xu, Yunxiang Zhang, and Xiao-Yi Wan. 2022. Cc-riddle: A question answering dataset of chinese character riddles. *ArXiv*, abs/2206.13778.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2023a. Are large language models really good logical reasoners? a comprehensive evaluation and beyond.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. 2023b. Exploring large language models for communication games: An empirical study on werewolf. *ArXiv*, abs/2309.04658.

Sen Yang, Xin Li, Leyang Cui, Li Bing, and Wai Lam. 2023a. Neuro-symbolic integration brings causal and reliable reasoning proofs. *ArXiv*, abs/2311.09802.

Zonglin Yang, Xinya Du, Rui Mao, Jinjie Ni, and E. Cambria. 2023b. Logical reasoning over natural language as knowledge representation: A survey. *ArXiv*, abs/2303.12023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *ArXiv*, abs/2305.10601.

Fei Yu, Hongbo Zhang, Prayag Tiwari, and Benyou Wang. 2023a. Natural language reasoning, a survey.

Zihan Yu, Liang He, Zhen Wu, Xinyu Dai, and Jiajun Chen. 2023b. Towards better chain-of-thought prompting strategies: A survey. *ArXiv*, abs/2310.04959.

Kamyar Zeinalipour, Tommaso laquinta, Asya Zanollo, Giovanni Angelini, Leonardo Rigutini, Marco Maggini, and Marco Gori. 2023a. Italian crossword generator: Enhancing education through interactive word puzzles.

Kamyar Zeinalipour, Mohamed Saad, Marco Maggini, and Marco Gori. 2023b. Arabicros: Ai-powered arabic crossword puzzle generation for educational applications. In *Proceedings of ArabicNLP 2023*. Association for Computational Linguistics.

Yunxiang Zhang and Xiaojun Wan. 2021. Birdqa: A bilingual dataset for question answering on tricky riddles. *ArXiv*, abs/2109.11087.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alexander J. Smola. 2022. Automatic chain of thought prompting in large language models. *ArXiv*, abs/2210.03493.

Jingmiao Zhao and Carolyn Jane Anderson. 2023. Solving and generating npr sunday puzzles with large language models.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *ArXiv*, abs/2211.01910.

Andrea Zugarini, Kamyar Zeinalipour, Surya Sai Kadali, Marco Maggini, Marco Gori, and Leonardo Rigutini. 2024. Clue-instruct: Text-based clue generation for educational crossword puzzles.

A Appendix

A.1 Prompting Topologies

The **chain-of-thought (CoT)** paradigm involves step-wise explanatory reasoning chains, bolstering capabilities even in zero-shot settings with instructions such as "Let's think step-by-step" (Wei et al., 2022; Kojima et al., 2022). Complementing this, **self-consistency** generates multiple solution paths, selecting the most coherent one (Wang et al., 2022).

Automatic CoT (auto-CoT) autonomously generates diverse reasoning chains for various questions (Zhang et al., 2022), while the **complexity of prompted chains** influences accuracy, as more intricate reasoning steps often enhance performance in complex inference tasks (Fu et al., 2022). This entails generating diverse reasoning chains and selecting outcomes that showcase deeper reasoning capabilities.

Golden CoT offers ground-truth reasoning chains to address limitations of basic prompting, reducing model hallucination risks (Del and Fishel, 2022). The **Plan-and-Solve (PS)** method breaks down tasks into subtasks for more structured solving (Wang et al., 2023a), while **Self-Question** guides models through a four-step process to enhance informal reasoning (Gu et al., 2023).

Exploring automated feedback, Pan et al. (2023b) examined **self-correction** within LLMs, noting its varied impact on logical reasoning. While instances of performance enhancement exist (Weng et al., 2022; Madaan et al., 2023), broader gains are often elusive, with some strategies even detracting from overall reasoning accuracy (Huang et al., 2023a). However, Tyen et al. (2023) highlight the potential of backtracking methods, which, when informed about the specific location of errors, significantly boost the model's correction abilities.

Detective Thinking Prompt (Gu et al., 2023) encourages the model to consider and analyze multiple clues or pieces of evidence within a given scenario, sequentially building towards a conclusion, much like solving a mystery. This type of prompting can help the model to handle complex scenarios where synthesizing disparate information correctly is crucial to generating accurate and logical outcomes.

The **Tree-of-Uncertain-Thought (TouT)** prompting method structures problem-solving into a tree where each branch explores different uncertain reasoning pathways, allowing for multiple potential solutions (Mo and Xin, 2023). In contrast, the **Tree-of-Thought(s)(ToT)** method (Yao et al., 2023; Long, 2023) focuses on a more linear and deterministic approach, systematically breaking down problems into a single coherent pathway towards a solution. The **Graph-of-Thought(s) (GoT)** method (Besta et al., 2023; Lei et al., 2023) structures problem-solving by mapping out various interconnected reasoning pathways, allowing language models to explore and evaluate multiple solutions simultaneously within a flexible, network-like framework.

The **Everything of Thoughts (XoT)** framework integrates Monte Carlo Tree Search (MCTS) with LLMs for enhanced thought generation, showing remarkable performance in complex puzzles (Ding et al., 2023). Additionally, **Inference-Exclusion Prompting (IEP)** employs a combination of forward and backward reasoning to approximate human logic more closely (Tong et al., 2023).

A.2 Conventional Methods

AI and Machine Learning methods have long been applied to puzzles and games, with algorithms like Deep Blue (Campbell et al., 2002) and AlphaZero (Silver et al., 2017) for Chess and Go, renowned for their exceptional results. This section contrasts "traditional" methods used to solve various puzzles with those derived from large language models (LLMs). Note that the aim of this paper isn't to determine the superior method for each puzzle, but to highlight the distinctive reasoning abilities of LLMs within diverse puzzle contexts. We particularly focus on rule-based puzzles, extensively addressed using conventional methods due to their structured, well-defined environments which require systematic strategies to achieve a solution. Conversely, rule-less puzzles such as riddles primarily test the logical, commonsense reasoning and creativity of models, without a clear path of steps to follow in order to find the solution, so we do not analyze this category.

Chi and Lange (2013) utilized three techniques to solve **Sudoku**: backtracking, simulated annealing, and alternating projections. The backtracking method, a brute-force depth-first search, consistently resolves puzzles across all difficulty levels, albeit slowly. Constraint programming transforms Sudoku into a constraint satisfaction problem, swiftly enforcing constraints to deduce solutions, often within milliseconds (Simonis, 2005). These methods always find a solution for Sudoku puzzle, in contrast with LLMs that have not achieved results better than 80% for 5x5 puzzles (Long, 2023).

In their study on **Rubik's Cube**, Chen (2022) employed several traditional methods including Korf's algorithm (Korf, 1997), which combines Iterative-Deepening Depth-First Search (IDDFS) with the A* algorithm and a heuristic search database. Both Thistlethwaite's [4] and Kociemba's [5] algorithms utilize group theory and similar search techniques to streamline the solving process, with Kociemba's version enhancing efficiency by simplifying the group structure. While all these algorithms effectively solve the Rubik's Cube—a task challenging for LLMs—Korf's method is particularly noted for its efficiency. Additionally, the study explored a machine learning strategy that integrates Monte-Carlo Tree Search (MCTS) with breadth-first search, yielding more optimized solutions, albeit at a lower efficiency. There have also been various attemts to solve Rubik's Cube using Reinforcement Learning (RL) like DeepCubeA (McAleer et al., 2018; Agostinelli et al., 2019) and others (Takano, 2023), which although find a solution in relatively few steps are time-consuming, with duration varying from 38.7 to 75.6 seconds (Takano, 2023).

Mazes are puzzles that can be solved by applying simple algorithms like depth-first search, A* or Trémaux's algorithm. However these problems are good for testing the spatial reasoning of LLMs. RL has also been utilized to solve mazes with (Barj and

[4] https://www.jaapsch.net/puzzles/thistle.htm
[5] https://kociemba.org/

Sautory, 2024) leveraging LLM feedback during training.

In Ding et al. (2023) MCTS has been used to solve **Game of 24**, **8-Puzzle** and **Pocket Cube**, achieving surpassing many LLM techniques, including CoT, CoT-SC, ToT and GoT. Additionally, Rozner et al. (2021) besides fine-tuning T5 for solving cryptic crosswords, have also used non-neural baselines including a WordNet-based heuristic model, a K-Nearest Neighbours bag of words model and a rule-based model, showing that the fine-tuning of T5 had the best results among them.

Finally, Studholme (2001) proposed a method for solving **Minesweeper** by considering it as a constraint satisfaction problem (CSP). The core strategy involves transforming the game's challenges into a set of logical constraints that must be satisfied to avoid mines effectively.

In conclusion, most conventional methods used to solve rule-based puzzles employ deterministic approaches that reliably produce solutions, in stark contrast to the unpredictable nature of LLMs. Another advantage of these traditional methods is their explainability and interpretability, crucial attributes for thoroughly evaluating algorithms and understanding their decision-making processes. However, as demonstrated in the study by Takano (2023), these methods can sometimes exhibit increased time complexity, indicating a potential trade-off between reliability and efficiency.

A.3 Tables

Table 1 delineates the various methods leveraged for puzzle-solving based on the datasets we have collected, illustrating the landscape of current LLM research in this domain. It particularly highlights the extensive methods applied to rule-based deterministic and rule-less commonsense puzzles. The absence of neuro-symbolic techniques and selection inference prompting indicates potential areas for expansion, especially considering their prospective benefits for LLMs grounded in logical reasoning datasets. The table further reflects the adaptability of certain methods like Chain-of-Thought, few-shot learning and fine-tuning, which are utilized across multiple puzzle types, hinting at their effectiveness. Based on this information, we not only catalogue the current state of method applications in puzzle-solving with LLMs but also highlight opportunities for innovative research in areas yet to be explored.

Table 2 summarizes the curated datasets and tasks associated with each category within our taxonomy of puzzles. A detailed examination reveals a substantial number of datasets for rule-based deterministic puzzles, such as Sudoku and Rubik's Cube, and a variety of rule-less riddles, indicating a strong research interest and resource availability in these areas. However, there appears to be a scarcity in the collection of rule-based stochastic puzzles and rule-less programming puzzles. This gap points to an opportunity for further research and dataset creation that could provide more diverse challenges for advancing the problem-solving capabilities of Large Language Models. Addressing this gap could lead to a more balanced and comprehensive set of benchmarks that reflect a wider spectrum of puzzle-solving scenarios, potentially catalyzing advancements in LLMs' abilities to handle uncertainty and complex logic-based problem-solving.

Methods	Rule-based Puzzles		Rule-less Puzzles		
	Deterministic	Stochastic	Riddles	Programming	Commonsense
Prompting	-	-	-	-	-
Few-shot	✓	✓	✓	✓	✓
Chain-of-Thought	✓	✓	✓	✓	✓
Self-refine	✓				
Auto-CoT					✓
Complexity CoT					✓
Plan & Solve					✓
Detective Thinking					✓
Self-Consistency	✓				✓
Tree-of-Thoughts	✓				
Tree-of-uncertain-Thoughts	✓				
Inferential Exclusion Prompting			✓		✓
Graph-of-Thoughts	✓				
Everything-of-thoughts	✓				
Hints			✓		✓
Introduction/Summarization	✓	✓	✓	✓	✓
Puzzle Translation	-	-	-	-	-
Logic	✓				
Code					
Fine-Tuning	✓	✓	✓	✓	✓

Table 1: Methods used by each category of our taxonomy based on the puzzle benchmarks we collected

Category	Type	Datasets
Rule-based	Deterministic	BoardgameQA (Kazemi et al., 2023), Sudoku (Noever and Burdick, 2021; Long, 2023; Ishay et al., 2023), Rubik's Cube (Noever and Burdick, 2021; Ding et al., 2023), Maze (Noever and Burdick, 2021), Crossword (Yao et al., 2023; Rozner et al., 2021; Efrat et al., 2021; Kulshreshtha et al., 2022), 8-puzzle (Ding et al., 2023), Game of 24 (Ding et al., 2023; Yao et al., 2023), Chess (Ishay et al., 2023; Feng et al., 2023b)
	Stochastic	Minesweeper (Li et al., 2023), BoardgameQA (Kazemi et al., 2023), Card Games (Huang et al., 2024; Gupta, 2023), Social Deduction Games (Wang et al., 2023b; Xu et al., 2023b; Lan et al., 2023)
Rule-less	Riddles	BrainTeaser (Jiang et al., 2023), RiddleSense (Lin et al., 2021), BiRdQA (Zhang and Wan, 2021), CC-Riddle (Xu et al., 2022), PUZZLEQA (Zhao and Anderson, 2023), MARB (Tong et al., 2023)
	Programming	P3 (Schuster et al., 2021), (Savelka et al., 2023)
	Commonsense	LatEval (Huang et al., 2023b), True Detective (Del and Fishel, 2022), DetectBench (Gu et al., 2023), MARB (Tong et al., 2023)

Table 2: Collected Datasets and Tasks for each Category