Solving logical puzzles using Large Language Models and Mace4

GIOVANNI MANFREDI SEBASTIANO MENEGHIN
gioman | meneghin @kth.se

November 27, 2023

Abstract

Recent work has shown how approaches based on obtaining a declarative task specification that then can become the input of an automated theorem prover, can improve a Large Language Model reasoning capabilities. In this paper we present a exploratory analysis applying this new approach to solve model generating and checking puzzles, testing the results across different Large Language Models. We are also comparing this new approach with more traditional prompt engineering techniques, namely Zero-shot and Few-shot Chain of Thought.

Missing: Insert results here

Contents

List of Acronyms and Abbreviations

AI Artificial Intelligence

CoT Chain of Thought

LADR Library for Automated Deduction Research

LLM Large Language Model

LLMs Large Language Models

NeurIPS Neural Information Processing Systems

NLP Natural Language Processing

SAT Satisfiability

1 Introduction

1.1 Background

Large Language Models (LLMs) became one of the main area of focus of Natural Language Processing (NLP) thanks to development followed after the discovery of the properties of Transformers. Based on attention, Transformers can train significantly faster than architecture based on recurrent or convolutional layers [1]. This new paradigm became the foundation over which Generative Pre-Training models were developed. This framework allowed to create models with a significant world knowledge, acquired processing text with long-range dependencies [2].

The research over these concepts and the development of models with a ever growing number of parameters brought BERT by Google [3] and GPT-2 by OpenAI [4]. Then, more recently, the whole NLP field became famous in the general public for ChatGPT, chat-bot based on GPT-3 by OpenAI [5].

These LLMs are now known as foundational models, thanks to a Stanford University research that outline this paradigm shift in Artificial Intelligence (AI), outlining its most critical aspects, ranging from a technical perspective to the societal impact of these new technologies [6].

A brief overview of the major LLMs and the companies working on these technology is provided by Figure 1.

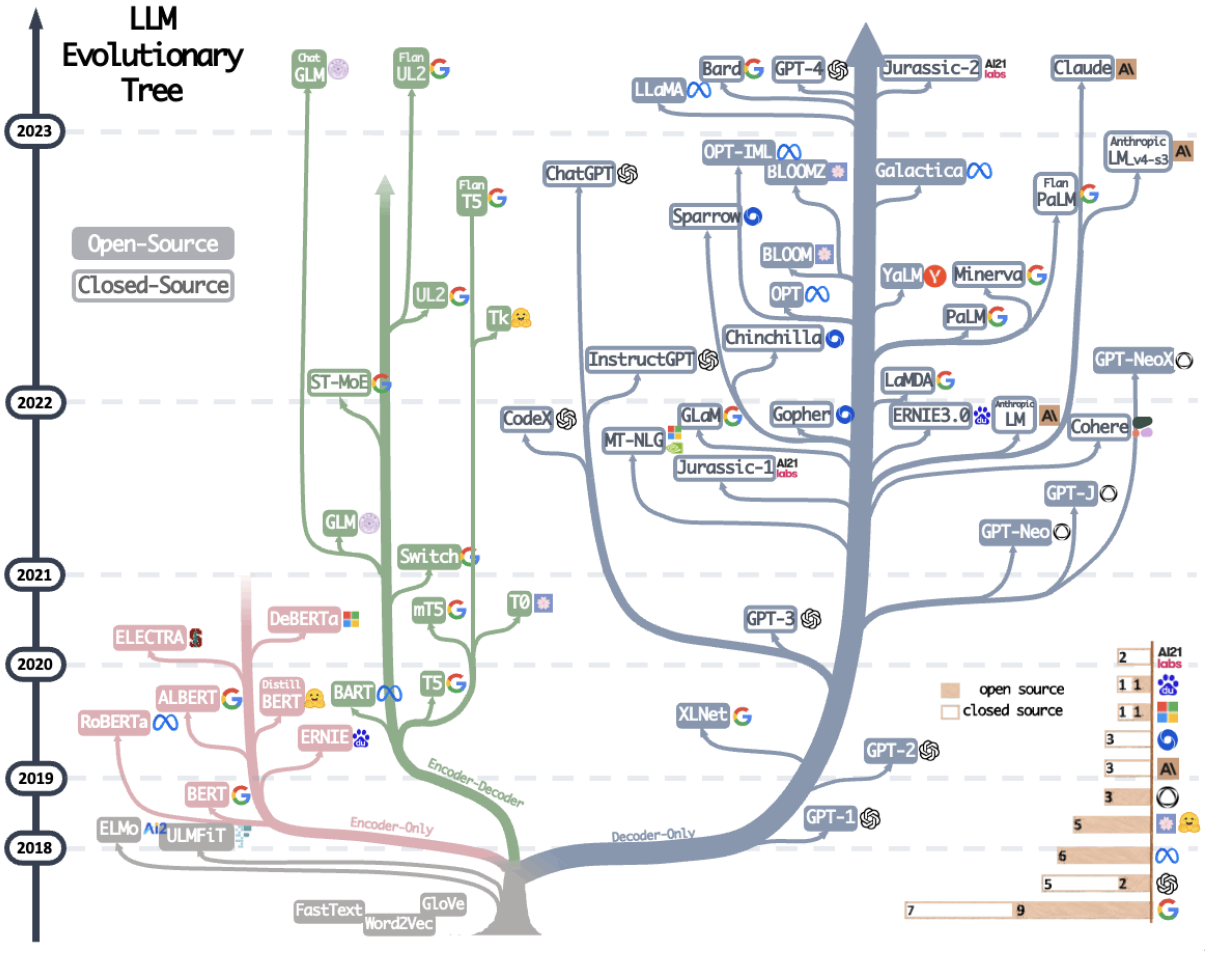

Figure 1: The evolutionary tree of modern LLMs traces the development of language models in recent years and highlights some of the most well-known models. Models on the same branch have closer relationships. Transformer-based models are shown in non-grey colors: decoder-only models in the blue branch, encoder-only models in the pink branch, and encoder-decoder models in the green branch. The vertical position of the models on the timeline represents their release dates. Open-source models are represented by solid squares, while closed-source models are represented by hollow ones. The stacked bar plot in the bottom right corner shows the number of models from various companies and institutions. [7]

LLMs have a large set of capabilities, including text summarization, chatbot behaviour, search, code generation and essay generation [8]. However logical reasoning is an area where these models tend to perform poorly [9].

Recently, many researchers have focused their effort on improving the capabilities of LLMs on complex logical reasoning [10, 11, 12]. One technique that proved to be successful is Chain of Thought (CoT) prompting [13, 14], that allows the Large Language Model (LLM) to take a series of reasoning steps before outputting a prediction. CoT typically improves the performance of the model [13, 14], but when a long sequence of reasoning steps is required [15, 16], the performance of this technique decline.

On this problem, a paper presented at the 37th Conference on Neural Information Processing Systems (NeurIPS) [17], explored a new technique that combines LLMs and automated Satisfiability (SAT) solvers. The technique presented is able to improve the reasoning capabilities of LLMs by using LLMs to generate a declarative task specification for the problem given as input. The specification is then taken by an automated SAT solver that computes the answer. This approach guarantees the correctness of the answer with respect to the specification, limiting the problem at the specification phase [17].

1.2 Problem

This report proposes an exploratory study, taking SAT-LM [17] study as guide, to explore a different logical problem that could benefit from SAT-LM's paper approach. The task to solve is logical puzzles [18] on model generation and model checking taken from Groza's book (Chapter 7, [18]).

The goal is to expand Ye et al. research [17] using a similar approach, but on a different problem (model generating and checking) and with a different program that generates models, namely Mace4 [19]. The results of this investigation are then compared with traditional CoT prompt engineering techniques to benchmark the improvement. This research validates Ye et al. work [17] and verifies the performance of his approach on a different task, overall contributing to the improvement of LLMs in solving logical reasoning tasks.

1.3 Research question

RQ1 : When solving model generation and model checking puzzles through the use of LLMs, which one between the following prompt engineering techniques, Few-shot CoT, Zero-shot CoT and the SAT-LM approach [17], provides a solution that is relevant, accurate, complete, clear and coherent?

RQ1 outlines five key aspects in the solution that this research evaluates to estimate the performance of the prompt engineering techniques above described. These aspects are:

1. **Relevance**: Does the answer directly address the question or topic that was posed? Is it on-topic and not a diversion from the question?

2. **Accuracy**: Is the information provided in the answer factually correct and the result of a deductive reasoning? Does the answer contain any errors or false deductive steps?

3. **Completeness**: Does the answer provide a comprehensive response to the question? Does it cover all aspects of the question, or does it leave out important details?

4. **Clarity**: Is the answer easy to understand? Is it written in clear, concise language without being overly complex or filled with jargon? If the answer is code, is it easy to read?

5. **Coherence**: Does the answer flow logically? Is the answer following a correct logical reasoning from the beginning to the end?

These aspects are used as metrics to qualitatively evaluate the LLMs answers and being able to compare them. The result collected will be composed of the answer provided by the LLM and a comment of the authors, that provides an answer for each category (relevance, accuracy, ...) on that specific answer.

2 Theoretical framework

2.1 Model generation and model checking problem

Model checking is an automatic technique for verifying models of software or hardware systems against their specification [20]. In other words, once a specification, i.e. a set of logical formulas, is defined, not all models might fit that set of rules.

In the puzzles [18] we are going to solve in this paper, the problem will ask to find the limited number of models that fit with given specification. This task is called model generation, but to be able to find the model that fits with the specification, solving model checking is necessary.

2.2 Mace4

Mace4 [19] is one of the most powerful model generators developed by William "Billy" McCune [21]. Mace4 is the first released program that has been constructed with the Library for Automated Deduction Research (LADR) [22], a library of C routines for building automated deduction tools and a larger project developed by McCune.

In this research, the Mace4 setup present in Chapter 1 of [18] will be utilised to start up the environment and use Mace4 locally.

2.3 Zero-shot and Few-shot prompting

Few shots prompting consists in querying a LLM asking it to perform a task given one or more examples of how the task is computed [23].

The opposite would be Zero shot prompting, in which we ask the answer straight away. In Figure 2 there is an example.

In this research, Zero-shot and Few-shot prompting are only utilized in their CoT equivalent. This is done to focus the research on the approach presented in Ye et al. paper [17]. Furthermore, non-CoT prompting approaches were recently proven to be less effective in solving logical problems [13, 14].

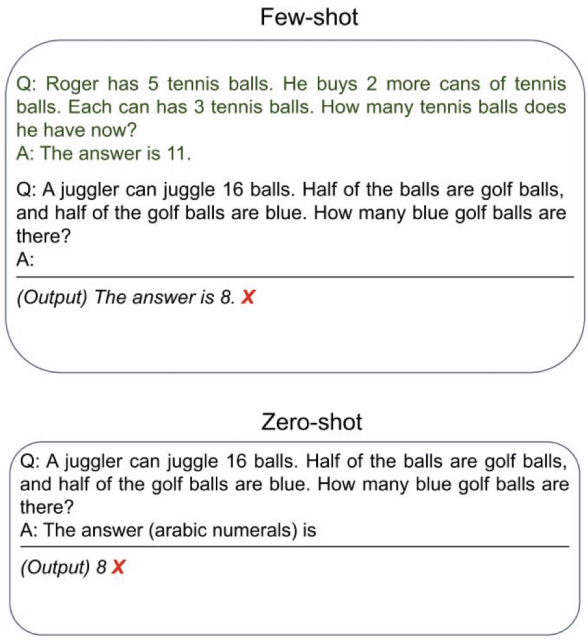

Figure 2: Comparison between Few-shot and Zero-shot prompting [23]

2.4 Chain of thought prompting

Chain of thought is a series of intermediate reasoning steps. If this is prompted in a LLM, it was shown to significantly improve the ability of large language models to perform complex reasoning [13, 14]. This type of prompting technique was presented at NeurIPS 2022, in two papers exploring both Few-shot and Zero-shot prompting [13, 14].

Also in this case the difference between Few-shot and Zero-shot prompting is that the former provides examples in the input, while the latter does not, asking for a CoT using a sentence along the lines of ”Explain you reasoning step by step” in the prompt. In Figure 3 there is an example.

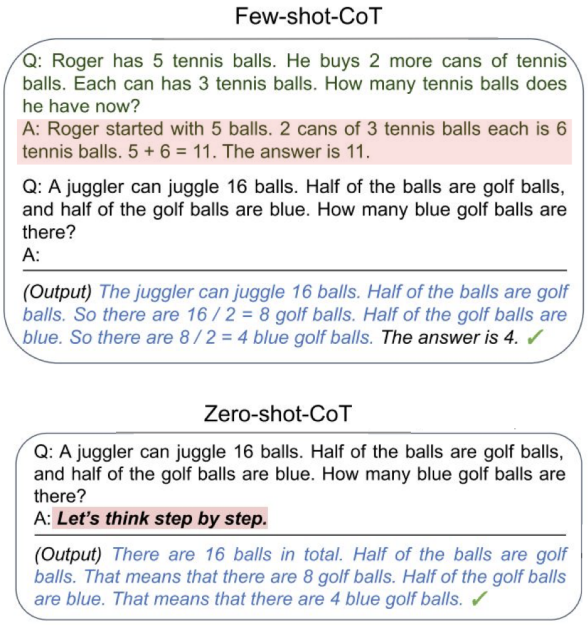

Figure 3: Comparison between Few-shot CoT and Zero-shot CoT prompting [23]

3 Method

3.1 Choice of method

The approach taken to develop this explorative research is a qualitative evaluation of the LLMs responses, comparing different prompting methods, namely Few-shot CoT, Zero-shot CoT and a SAT-LM [17] approach, across two different LLMs, namely *gpt-3.5-turbo-1106* and *gpt-4-turbo* using Chatbot Arena platform [24].

We decided to conduct an experiment because the experimental analysis was a crucial point of our findings, enabling us to compare how different prompt engineering techniques perform on our specific case.

We opted for a qualitative approach over a quantitative approach due to the nature of the data we were going to analyse. In fact, if we were to evaluate the performance of the LLM and prompting techniques as just ”successful” or ”not successful” we would have glossed over the improvement on the reasoning capabilities of the model. We can be more easily evaluate the output of the LLM through a qualitative analysis that compares it to human reasoning. A clear alternative approach would have been to use only correctness and perform a quantitative analysis. In the case of this research this would have been unfeasible, due to the lack of a large set of solved logical puzzles.

To evaluate qualitatively the LLMs answers, five key aspects were identified: relevance, accuracy, completeness, clarity and coherence. These metrics were selected to provide a more accurate picture of the model performance that goes beyond the simple correctness of the solution. This allows the authors to

compare answers from different LLMs more easily, providing a framework of the key characteristics of a good answer. In the experiments, defining relevance, accuracy, completeness, clarity and coherence using questions, allowed the authors to have a series of questions to answer for each LLM answer, that then could be compared with other answers. The definition of these aspects is here reported for completeness:

1. **Relevance**: Does the answer directly address the question or topic that was posed? Is it on-topic and not a diversion from the question?

2. **Accuracy**: Is the information provided in the answer factually correct and the result of a deductive reasoning? Does the answer contain any errors or false deductive steps?

3. **Completeness**: Does the answer provide a comprehensive response to the question? Does it cover all aspects of the question, or does it leave out important details?

4. **Clarity**: Is the answer easy to understand? Is it written in clear, concise language without being overly complex or filled with jargon? If the answer is code, is it easy to read?

5. **Coherence**: Does the answer flow logically? Is the answer following a correct logical reasoning from the beginning to the end?

For conducting the experiments we used a collection of puzzles already modelled in Mace4, including a comprehensive explanation of the ”human” approach to the problem [18]. This book chapter was chosen as it allowed us to have a solid base to which be able to compare the outputs of the LLMs both in the CoT approach and the SAT-LM approach, thanks to the solutions of the puzzles being in both code and natural language. The alternative were scarce, as it would have created the need to design a set of puzzles on model generation and model checking complete with answers.

The platform Chatbot Arena was used to query the two selected LLMs. *gpt-3.5-turbo-1106* and *gpt-4-turbo* were selected as they are the are currently the most prominent LLMs that are being analysed. Chatbot Arena was used as testing environment as it provided a standardised way of conducting experiments in parallel on two LLMs. It also allowed us to have access to GPT-4, to which we would not be able to access freely using OpenAI platform. The fact that how the platform behaves is explained in the related paper was also considered beneficial [24]. Alternatives to this approach mainly consist in using platforms of companies developing LLMs, such as OpenAI and its ChatGPT, but this would bind the researchers on the accessibility of the platform, that might require a monthly payment, as GPT-4 does.

Only the authors were involved in conducting the project. Neither surveys nor interviews were led, thus removing the need for participants, being those humans or animals. This choice was taken to reduce the time needed to conduct the experiments, a necessity to provide a report of value within the allotted time for the project.

3.1.1 Ethical considerations

The data used for creating the queries to the LLMs mainly comes from the book *Modelling Puzzles in First Order Logic*, Chapter 7, Island of Truth [18]. The access to the document was granted by Politecnico di Milano, that bought the ebook, and with this purchase the authors, being Politecnico di Milano’s students have the rights to read and cite the book as resource. This right was confirmed by a Springer representative through a brief e-mail conversation. More information can be found on this website. To extend this possibility also to KTH students and employees, KTH as institution needs to buy the e-book from the publisher Springer.

From an ethical point of view on data regulation, the project does not collect or store personal or sensitive data. All data used in the project was already present and available online.

The authors declare that no conflict of interest was present while conducting this research.

3.2 Applying the method

3.2.1 Dataset

The data set of puzzles on model generating and model checking comes from the book *Modelling Puzzles in First Order Logic*, Chapter 7, Island of Truth [18]. The chapter provides twelve logical puzzles solved with code snippets and natural language. The problem is mapped in both Mace4, a model generator and checker and Prover9, a SAT-theorem prover. For the purpose of this research, only Mace4 code will be considered, as Prover9 verifies Mace4 results, providing steps, that are mostly described in the natural language solution.

The puzzles are numbered from 61 to 72, with the first eight puzzles that have a setting based on knights and knaves, and the last four that have an additional character: a spy. In the puzzle settings:

- A knight tells always the truth.
- A knave tells always a lie.
- A spy tells sometimes the truth and sometimes a lie.

The puzzle is to determine among a number of unknown individuals who is a knight, who is a knave and who a spy based on what they say. For the purpose of the experiment, the book content was adapted to have similar but independent questions can could be run without needing contextual information. The twelve puzzles were paired to allow the construction of queries using Few-shot learning, i.e. providing a solved puzzle and asking the solution of a similar puzzle. The complete set of queries can be found in the Appendix A.

3.2.2 Large Language Models

Developed for the paper *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena* [24], Chatbot Arena is a crowd-sourced platform built to collect users opinions on LLMs answers and to test the human judgement of telling a more powerful LLM from a less powerful one. This is done thorugh a simple blind test.

In the case of this research, we are taking advantage of another part of the platform that allows the user to select two LLMs to compare against one another. After the prompt is inputted, the responses are computed and displayed to the user.

The LLMs selected for this study are *gpt-3.5-turbo-1106* and *gpt-4-turbo* but the platform offers a plethora of other LLMs. This could be an aspect on which a future research could expand upon.

3.2.3 Materials

- **Software**:
  - **Safari 17.0** : default browser on the laptop we used for the experiments.
  - **Mace4** : install and setup conducted following *Modelling Puzzles in First Order Logic* book, Chapter 1, Getting Started with Prover9 and Mace4 [18].
  - **iTerm2** : terminal used to submit Mace4 queries.
  - **Overleaf** : browser app used for creating the report.
- **Hardware**: all experiments were conducted on a 2021, 16-inch MacBook Pro with Apple M1 Pro chip, 16 GB of RAM and running macOS Sonoma 14.0 .

3.2.4 Procedure

1. **Data gathering**: open the *Modelling Puzzles in First Order Logic* book, Chapter 7, Island of Truth [18] and retrieve each puzzle and its respective answer.

2. **Query preparation**: for each pair of puzzle (12 puzzles, 6 pairs) create a prompt for Zero-shot prompting, Few-shot prompting and SAT-LM approach for each pair of puzzles (6 pairs of puzzles, 3 queries each for a total of 18 queries).

3. **Querying the LLMs**: using the Chatbot Arena platform [24] query the two selected models, namely *gpt-3.5-turbo-1106* and *gpt-4-turbo* and collect the results from the platform (18 queries in parallel on two LLMs for a total of 36 results).

4. **Compute on Mace4**: use the output of each SAT-LM approach query to look for a model that matches the specification with Mace4 and collect the results.

5. **Results and Considerations**: collect the result and compare the performance of the different prompt engineering techniques and of the different LLMs.

4 Results

Preliminary: this section is currently a work in progress. It currently presents the results of the first 3 experiments carried out (out of a total of 6). In the near future it will completed, integrating the next experiments. Any feedback is welcome

The research conducted is composed of six experiments (available at https://github.com/Silemo/rmsw-2023-manfredi-meneghin) that were carried out using three plus one prompt engineering techniques on two different LLMs. In this section, the experiments results are presented, giving the reader an overall perspective on how the LLMs behaved when tacking this logical questions.

The experiment were divided into two main groups:

- **Knights and knaves** : the inhabitants of the island of truth are knights, who always tell the truth and knaves, who always lie.

- **Knights, knaves and spies** : the inhabitants of the island of truth are knights, who always tell the truth, knaves, who always lie and spies that can say the truth or lie at will.

4.1 A new technique - Few-shot SAT-LM

During the experiments, the SAT-LM technique [17] initially proved unsuccessful resulting in many outputs which were Mace4 pseudo-code and could not be computed directly. An idea came to mind and a new technique was proposed. The new technique gives in input to the LLM a similar question that is already answered. Then it poses the LLM the actual question. This is a similar approach to ”Few-shot CoT”, but applied on Mace4 scripts. In the experiments the SAT-LM original technique was renamed ”Zero-shot SAT-LM” and the new technique was named ”Few-shot SAT-LM”.

4.2 Experiments on Knights and Knaves

The scenario most explored is the setting of the island of truth where only knights, that only tell the truth and knaves, that always lie. This scenario is the setting of experiments from one to four. The setting is generally understood by the LLMs, with nearly all scripts recognising the difference between a knight and a knave.

4.2.1 Relevance

The answers are generally relevant by both by GPT-3.5 and GPT-4, with only one exception by GPT-3.5 in experiment 3 with the Few-shot SAT-LM technique, where it is incapable of providing any sort of answer.

4.2.2 Accuracy

The responses accuracy greatly varies between models, experiments and techniques. GPT-4 accuracy is very high in the first two experiments, lacking only in the Zero-shot SAT-LM technique because it is not able to generate a correct Mace4 script without an example. But it plummets in experiment 3, where the LLM is never able to provide a correct solution. It should be noted that experiment 3 has an higher complexity than the first two experiments, asking a question that could be answered with two different models.

GPT-3.5 in the first two experiments is never able to provide a correct response with the exception of the Few-steps SAT-LM technique, that allows it to compute a correct Mace4 script. In the third experiment, GPT-3.5 is able once to provide a correct answer, even beating GPT-4, but since the answer still contains errors in the logical deduction steps, it could not be relevant.

It should be noted that in all experiments the use of a Few-shot script, provided more accurate results than its Zero-shot counterpart.

4.2.3 Completeness

The responses have a generally high completeness, exploring most scenarios in the logical setting, but we could clearly see a more complete approach in GPT-4 answers than in GPT-3.5. This corresponds to the model not having considered all hypothesis, or all combinations of inhabitants being knights and knaves.

It should be noted that even if the question explored all scenarios of the question, the incorrectness of its logical steps could hinder the LLM ability to have the full picture of the problem.

4.2.4 Clarity

The responses are generally clear and understandable, but errors in the deductive reasoning can make the text difficult to follow and understand. This is often the case in GPT-3.5 responses, that in particular in CoT techniques, often fail simple deductive reasoning and implications.

GPT-4 is typically more clear, with the exception of experiment 3, where the logical reasoning errors also make the text more difficult to read.

4.2.5 Coherence

The coherence, intended as the capability of following a logical flow, is greatly diverse between the two LLMs considered.

While GPT-4 provides a reasoning which is usually coherent, with the exception of experiment 3, GPT-3.5 is never able to provide a full deductive reasoning, failing at that even when the answer is actually correct (experiment 3, Zero shot CoT).

4.3 Unexpected results

During the first experiment, the Few-shot SAT-LM technique was proposed and added. The objective of this addition, was to verify, if with some additional knowledge about Mace4, GPT-4 could still solve the problem, as it had done in CoT prompts. It was a surprise when also GPT-3.5 with this script was able to compute a correct Mace4 code.

Another unexpected outcome of the experiments are the poor results of both LLMs in the experiment 3. The presence of three different inhabitants and the possibility of two models as solution, consists of an increase in difficult that even GPT-4 could not overcome. So the problem, even if very similar to the previous, cannot be solved by the LLM.

5 Discussion

Preliminary: this section is currently a work in progress. Any feedback is welcome

Overall GPT-4 shows a greater capability at solving logical deduction tasks, probably mainly thanks to the greater complexity of the model, and its capability of logical thinking.

SAT-LM proved successful, as long as the LLM considered knew how to write Mace4 code. Another few step approach could have been possible, by providing the model with the manual of Mace4. An observation that can be taken here is the Few-shot SAT-LM prompt engineering technique, enabled even LLM less powerful, such as GPT-3.5 to solve a problem, reducing the problem and still obtaining a valid solution.

6 Conclusions

Preliminary: this section is currently a work in progress. Any feedback is welcome

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[2] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018. [Online]. Available: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019. [Online]. Available: https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[6] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang, "On the opportunities and risks of foundation models," 2021. [Online]. Available: https://arxiv.org/abs/2108.07258

[7] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, B. Yin, and X. Hu, "Harnessing the power of llms in practice: A survey on chatgpt and beyond," 2023. [Online]. Available: https://arxiv.org/abs/2304.13712

[8] A. Tamkin, M. Brundage, J. Clark, and D. Ganguli, "Understanding the capabilities, limitations, and societal impact of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2102.02503

[9] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young, E. Rutherford, T. Hennigan, J. Menick, A. Cassirer, R. Powell, G. v. d. Driessche, L. A. Hendricks, M. Rauh, P.-S. Huang, A. Glaese, J. Welbl, S. Dathathri, S. Huang, J. Uesato, J. Mellor, I. Higgins, A. Creswell, N. McAleese, A. Wu, E. Elsen, S. Jayakumar, E. Buchatskaya, D. Budden, E. Sutherland, K. Simonyan, M. Paganini, L. Sifre, L. Martens, X. L. Li, A. Kuncoro, A. Nematzadeh, E. Gribovskaya, D. Donato, A. Lazaridou, A. Mensch, J.-B. Lespiau, M. Tsimpoukelli, N. Grigorev, D. Fritz, T. Sottiaux, M. Pajarskas, T. Pohlen, Z. Gong, D. Toyama, C. d. M. d'Autume, Y. Li, T. Terzi, V. Mikulik, I. Babuschkin, A. Clark, D. d. L. Casas, A. Guy, C. Jones, J. Bradbury, M. Johnson, B. Hechtman, L. Weidinger, I. Gabriel, W. Isaac, E. Lockhart, S. Osindero, L. Rimell, C. Dyer, O. Vinyals, K. Ayoub, J. Stanway, L. Bennett, D. Hassabis, K. Kavukcuoglu, and G. Irving, "Scaling language models: Methods, analysis & insights from training gopher," 2021. [Online]. Available: https://arxiv.org/abs/2112.11446

[10] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," 2017. [Online]. Available: https://arxiv.org/abs/1703.05175

[11] Z. Chen, M. M. Balan, and K. Brown, "Language models are few-shot learners for prognostic prediction," 2023. [Online]. Available: https://arxiv.org/abs/2302.12692

[12] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, "Opt: Open pre-trained transformer language models," 2022. [Online]. Available: https://arxiv.org/abs/2205.01068

[13] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2022. [Online]. Available: https://arxiv.org/abs/2201.11903

[14] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2205.11916

[15] A. Creswell, M. Shanahan, and I. Higgins, "Selection-inference: Exploiting large language models for interpretable logical reasoning," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=3Pf3Wg6o-A4

[16] A. Saparov and H. He, "Language models are greedy reasoners: A systematic formal analysis of chain-of-thought," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=qFVVBzXxR2V

[17] X. Ye, Q. Chen, I. Dillig, and G. Durrett, "Satlm: Satisfiability-aided language models using declarative prompting," 2023. [Online]. Available: https://arxiv.org/abs/2305.09656

[18] A. Groza, *Modelling Puzzles in First Order Logic*. Springer International Publishing, 2021. [Online]. Available: https://doi.org/10.1007/978-3-030-62547-4

[19] W. McCune, "Mace4 reference manual and guide, technical memorandum no. 264, argonne national laboratory," http://www.cs.unm.edu/~mccune/prover9/, August 2003.

[20] D. Peled, P. Pelliccione, and P. Spoletini, "Model checking," pp. 1904–1920, Mar. 2009. [Online]. Available: https://doi.org/10.1002/9780470050118.ecse247

[21] M. P. Bonacina and M. E. Stickel, Eds., *Automated Reasoning and Mathematics*. Springer Berlin Heidelberg, 2013. [Online]. Available: https://doi.org/10.1007/978-3-642-36675-8

[22] W. McCune, "Library for automated deduction research," http://www.cs.unm.edu/~mccune/prover9/, 2009.

[23] Z. Ding and Z. Zhang, "Chain of thought prompting for large language model reasoning, lecture 9, cos 597g," https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec09.pdf, Fall 2022.

[24] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging llm-as-a-judge with mt-bench and chatbot arena," 2023. [Online]. Available: https://arxiv.org/abs/2306.05685