Are Large Language Models Good At Fuzzy Reasoning?

Suresh Singh
Department of Computer Science
Portland State University
Portland, OR, USA
singh@cs.pdx.edu

Abstract

This paper explores the *zero-shot* reasoning abilities of three popular LLMs from the point of view of fuzzy logic. Unlike strict logical reasoning, fuzzy reasoning allows for the case when a logically incorrect conclusion is still plausible (given additional context). This form of reasoning is more similar to typical human discourse rather than strict logical inference.

Given that previous studies have shown the equivalence of likelihoods and membership functions, we use likelihoods (output by LLMs) to score their performance on a variety of reasoning tasks. Specifically, we use a recent logic dataset for this purpose but modify the prompts to encourage the LLM to output likelihood values for multiple choice and yes/no questions. We show that two versions of GPT (3.5 Turbo and 4.0125) perform very well on average (scoring 0.813 and 0.761 respectively) while Gemini-1.5 Pro struggles (scoring 0.635), particular on yes/no questions. We explore the reason for this difference. It appears that since Gemini is a multi-modal model, its logical reasoning abilities are not as fully developed as the two GPT models.

CCS Concepts

• **Computing methodologies** → **Logical and relational learning**; **Natural language processing**.

Keywords

Large language models, Fuzzy logic, Propositional logic, First-order logic

ACM Reference Format:
Suresh Singh. 2024. Are Large Language Models Good At Fuzzy Reasoning?. In *2024 The 7th International Conference on Computational Intelligence and Intelligent Systems (CIIS) (CIIS 2024), November 22–24, 2024, Nagoya, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3708778.3708779

1 Introduction

Large language models (LLMs) are increasingly being deployed in the workplace in a wide variety of roles. In many of these roles, the tasks require the ability to reason like humans do. For instance, online tech support to solve customer problems frequently require multiple steps of inference where the reasoning is probabilistic. Likewise, question and answer systems often require the LLM to reason with incomplete or imprecise information as well. Therefore, it is of interest to understand how well the LLMs reason in this human context.

Consider the following example of reasoning that is common among humans: "If he feels hot, Sam will buy an ice cream. The temperature is 30C." We then ask, "Did Sam buy an ice cream?". The answer to this question depends on one's perception of what is hot. Therefore, there is a likelihood of either outcome being correct depending on many other factors. When posed this question, ChatGPT's response is *whether Sam buys an ice cream or not at 30°C would depend on various factors such as Sam's personal preferences, cultural norms, and other contextual information that is not provided in the given scenario.* If we were to modify the original statement by replacing "The temperature is 30C" with "It is hot", one might say that the correct answer is that Sam did buy an ice cream. However, it is possible that he didn't, for a variety of reasons. For instance, it was early in the morning or he was sick or he had eaten a large meal and was full, and so on. In other words, the broad context will affect the final outcome. In strict logical reasoning, we assume that the complete context is provided. However, in human reasoning, a significant portion of the context is inferred. This difference is significant and explains much of human behavior and discourse. Thus, while a great deal of prior work has been done on understanding the strict logical reasoning ability of LLMs (see section 2), given that LLMs are trained on human textual data, we believe that LLM's logical reasoning must be evaluated using methods that are more similar to human reasoning.

In this paper we use concepts from fuzzy logic in order to evaluate the reasoning capabilities of LLMs. This type of logic allows us to reason with contradictory or incomplete information and can often result in multiple conclusions (each with a different likelihoods). Since LLMs reason using a great deal of context learned during training, we believe that fuzzy logic will enable us to interpret their output more accurately. Our main results are:

- We use a comprehensive logic dataset, containing questions in propositional (PL) and first-order logic (FOL), to evaluate three LLMs – GPT-3.5 Turbo, GPT-4.0125, and Gemini-1.5 Pro on their fuzzy reasoning abilities,
- We use likelihoods (which have been shown to be equivalent to membership functions) to measure the performance of these models. Of the three models, the two GPT models perform far better than Gemini, possibly due to Gemini being a multi-model model with less textual reasoning capabilities,
- All models perform better on multiple-choice questions than on yes/no questions. This is possibly because multiple-choice questions provide more textual context,

CIIS 2024, Nagoya, Japan

ACM ISBN 979-8-4007-1743-7/24/11
https://doi.org/10.1145/3708778.3708779

- While the models appear to reason logically some times, we observe that in many instances (particularly for Gemini) the model tries to find the answer in the provided text – thus leading to errors.

	GPT-3.5 Turbo	GPT-4.0125	Gemini-1.5 Pro
Multiple Choice			
PL	0.805	**0.812**	0.779
FOL	**0.890**	0.824	0.786
Yes/No			
PL	**0.759**	0.673	0.486
FOL	**0.798**	0.737	0.491
Overall Average			
PL&FOL	**0.813**	0.761	0.635

The remainder of the paper is organized as follows. In the next section we summarize related work on logical reasoning by LLMs and the impact of using appropriate prompting strategies on performance. In section 3 we describe the relationship between membership functions from fuzzy logic and likelihoods produced by LLMs. Section 4 introduces the dataset and the evaluation metric. Our results are in section 5 and we conclude in section 6.

2 Related Work

Over the past 15 years, a large number of datasets have been developed to explore various abilities of LLMs including commonsense reasoning, question answering, sentiment analysis, temporal reasoning, qualitative reasoning, pronoun resolution, and many more. However, given that LLMs are finding increasing use as the backend for many deployed systems in the workplace, it is important to analyze their logical reasoning abilities as well. To this end, researchers have developed several datasets for exploring the ability of LLMs to reason over propositional calculus, first-order logic, and non-monotonic logic.

Some notable datasets and studies of logical reasoning include [3] which looks at the first-order logic reasoning abilities of a transformer-based model and LogicNLI [17] which introduces a diagnostic dataset for first-order reasoning. Inductive reasoning is analyzed in the CLUTRR [16] dataset while [6] studies the temporal reasoning abilities of LLMs. At least four datasets are based on using question answering to test the logical reasoning ability of LLMs. FOLIO [5] and ProntoQA [15] provide QA datasets to test first-order logic performance (though ProntoQA only considers modus ponens) while ProofWriter [3] considers both first-order and propositional logics. LogicBENCH [13] provides QA datasets to analyze propositional, first-order, and non-monotonic logic reasoning abilities of LLMs. In addition to these formal datasets, TaxiNLI [7] provides a logic taxonomy for NLI tasks, RuleBert [14] covers soft logical rules, and SimpleLogic [19] provides a class of reasoning problems (rather than QA).

An important part of using LLMs for question answering is using carefully crafted *prompts*. Indeed, prompt engineering and instruction-based tuning have emerged as a very important tools to extract the best performance from LLMs, [10]. [8] shows that prompts can be as effective as hundreds of training data points. There have been many specific studies of prompts for different tasks. For example, [7] consider prompt engineering for QA while [1] shows zero-shot generalization across various NLI tasks by using appropriate prompts. Related to prompts, other authors have studied the use of natural language instruction as a way of enabling LLMs to generalize,[12]. Similarly, PromptSource [1] and FLAN [4] use instruction tuning for zero-shot generalization on unseen tasks. [18] developed a framework for providing a task description to LLMs as a way to enable them to solve complex problems.

3 Fuzzy Reasoning and Likelihood Estimation

When viewed from the perspective of propositional calculus, the ice cream example from section 1 is an instance of *modus ponens* reasoning. In other words, if $A \rightarrow B$ is true and A is true, then we can logically infer that B is true. However, human reasoning is better modeled using fuzzy logic where the ice cream example can then be thought of as an instance of GMP (Generalized Modus Ponens) [11]. For instance, let T = {10,20,30,40,50} denote the set of temperatures and the membership function for *hot* = {(50,1),(40,1),(30,0.5)}[1]. Similarly, let I = {1, 0} where 1 means eat ice cream and 0 means not. Then GMP gives us the result [0.5, 0.5] for the two choices from I. In other words, there is a 50% chance that Sam will eat ice cream when the temperature is 30C. This result will vary depending on the selection of membership functions. We note that when ChatGPT is asked specifically to provide the likelihood that Sam will eat ice cream, it gives a value of 0.85 (in the discussion in section 1 we did not prompt ChatGPT to give a likelihood estimate). In other words, ChatGPT is using its contextual knowledge to answer this question. What is evident from this example is that context is key since it is used implicitly by LLMs in reasoning and is used explicitly to create meaningful membership functions for fuzzy logic.

Since LLMs produce likelihoods and fuzzy reasoning produces membership functions, it is important to ask what is the relation between these two values. It has been shown by several authors [2, 9] that likelihoods are equivalent to membership values. Indeed, some authors have suggested using LLMs to directly output likelihoods and using these as membership functions for many cases involving human linguistic terms. Thus, **in this paper we directly use likelihood values from LLMs to evaluate their reasoning abilities over a fuzzy logic dataset.**

The approach we follow is to use an existing logic dataset but to prompt the LLM to provide likelihoods for the different choices in multiple-choice questions or for the two possible answers in yes/no questions. An example of a multiple-choice question from [13] that tests the LLM's ability to use the Modus Tollens rule is,

```
Isabella had been working hard on her project, knowing
that if she finished it on time, she would receive a
bonus from her company. However, despite her efforts,
Isabella realized that she wouldn't be receiving a bonus
as she had hoped. In light of the context provided, what
conclusion can be considered the most appropriate?

(1) Isabella didn't finish her project on time.
(2) Sarah finished her project on time.
(3) Isabella received a bonus from her company.
(4) Isabella decided to quit her job.
```

[1]The membership function defines what 'hot' might mean – 50C and 40C with probability 1, and 30C with probability 0.5. These numbers can be derived by polling a large group of people or other methods.

In the above example, the correct answer in formal logic is choice 1. However, we can say that perhaps Isabella did not receive a bonus despite finishing her project because her boss didn't like her, which would lead her to quit (choice 4). This reasoning is valid in typical human communication because, in real-life, there is great deal of context. Thus, as in the ice cream example, we can say that the appropriate way to evaluate the response is by looking at the likelihoods of each of the choices. When prompted to provide likelihood values for these four choices, ChatGPT gives [0.95, 0.05, 0.01, 0.2]. The reasoning for giving 0.2 to choice 4 is given as `While there is no direct evidence in the context to support this conclusion, it remains a conceivable outcome based on the disappointment from not receiving the bonus. However, it is speculative without further context.`

4 Fuzzy Dataset

In this paper, we will use likelihood values provided by LLMs to explore their fuzzy reasoning abilities. To stay consistent with existing work on formal logic, we utilize an existing data set LogicBench from [13]. However, we modify the prompts to output likelihood values rather than a single answer for both yes/no questions and multiple-choice questions.

4.1 Inference Rules in LogicBench

The dataset contains separate test cases for propositional logic and for first-order logic (we do not consider non-monotonic logic in this paper). The specific inference rules[2] for propositional logic are in Table 1 while the rules for first-order logic are in Table 2. The LogicBench dataset contains two types of examples for each of these inference rules – yes/no questions and multiple-choice questions with four choices.

Table 1: Propositional Logic.

Rule	Premises	Inference
MP	$p \rightarrow q, p$	q
MT	$p \rightarrow q, \neg q$	$\neg p$
HS	$p \rightarrow q, q \rightarrow r$	$p \rightarrow r$
DS	$p \vee q, \neg p$	q
CT	$p \vee q$	$q \vee p$
CD	$p \rightarrow q, r \rightarrow s, p \vee r$	$q \vee s$
DD	$p \rightarrow q, r \rightarrow s, \neg q \vee \neg s$	$\neg p \vee \neg r$
BD	$p \rightarrow q, r \rightarrow s, p \vee \neg s$	$q \vee \neg r$
MI	$p \rightarrow q$	$\neg p \vee q$

One example of a multiple choice questions was provided in the previous section. An example of a yes/no question from propositional logic for Destructive Dilemma is: `If Sarah goes to the gym, she stays fit and healthy. if she avoids junk food, she will have better immunity. However, it is known that either Sarah doesn't stay fit and healthy or she won't have better immunity. It is uncertain which of these statements is true or if both are true. Can we say at least one of the following must always be true? (a) she goes to the gym and (b) she avoids junk food.`

[2]MP - Modus Ponens, MT - Modus Tollens, HS - Hypothetical Syllogism, DS - Disjunctive Syllogism, CT - Commutation, CD - Constructive Dilemma, DD - Destructive Dilemma, BD - Bi-directional Dilemma, MI - Material Implication

4.2 Fuzzifying the Dataset

In the various logical reasoning papers we described previously, the LLMs are queried to provide just one answer that is inferred from the premises. However, as we have discussed, LLMs might use other knowledge as well in order to answer the question. This raises the question about how to *prompt* the LLM appropriately to consider likelihoods rather that a single answer.

An example of a prompt for yes/no questions from LogicBench is `Given the context that contains rules of logical reasoning in natural language and question, perform step-by-step reasoning to answer the question. Based on context and reasoning steps answer the question ONLY in 'yes' or 'no'.` In order to allow multiple conclusions, we modify the prompt by replacing the last sentence with *Based on context and reasoning steps provide likelihoods for two possible answers, 'yes' or 'no'.* Similarly, the prompt for multiple-choice questions that we use is a modified version of the prompts used for logical reasoning and looks like this: *Given the context that contains rules of logical reasoning in natural language, question, and options, perform step-by-step reasoning to answer the question. Answer the question by providing likelihoods of ONLY 'choice_1' or 'choice_2' or 'choice_3' or 'choice_4'.*

4.3 Evaluation Metric

Logic datasets are designed to have a single correct answer to each inference question. Let $\mathbf{a}_i$ be the one-hot vector representing the correct answer to problem i. Let $\mathbf{l}_i$ be the likelihood vector representing the answer provided by the LLM. Then, we score the performance of the LLM for item i as $\mathbf{a}_i \cdot \mathbf{l}'_i$. The overall performance on N logic questions is simply,

$$\frac{1}{N} \sum_i \mathbf{a}_i \cdot \mathbf{l}'_i$$

To see how this form of fuzzy scoring differs from strict scoring where there is just one correct answer, let us return to the example from section 3 where the likelihood vector for four choices for Isabella is $\mathbf{l} = [0.95, 0.05, 0.01, 0.2]$. Given that the correct logical answer is given by the vector $a = [1, 0, 0, 0]$, the score we obtain is 0.95. Observe that if we were prompting LLMs to select just one correct choice, for this example the LLM would select choice_1, giving it a score of 1. Thus, using likelihoods can reduce the overall score of LLMs when they answer correctly. However, say the LLM provided a likelihood vector $\mathbf{l} = [0.2.0.05, 0.01, 0.95]$ for the same example. In this case, if we use strict scoring, the LLM will receive a score of 0. However, using likelihoods, the score it receives is 0.2. These examples illustrate the differences between fuzzy scoring and strict scoring.

5 Results

In this paper our focus is on understanding how LLMs perform in a *zero-shot* setting since this can be seen as a baseline for LLM performance. The LogicBench dataset contains 20 multiple-choice

Table 2: First-Order Logic.

Rule	Premises	Inference
MP	$\forall x(p(x) \rightarrow q(x)), p(a)$	$q(a)$
MT	$\forall x(p(x) \rightarrow q(x), \neg q(a)$	$\neg p(a)$
HS	$\forall x((p(x) \rightarrow q(x)) \wedge (q(x) \rightarrow r(x)))$	$p(a) \rightarrow r(a)$
DS	$\forall x(p(x) \vee q(x)), \neg p(a)$	$q(a)$
CD	$\forall x((p(x) \rightarrow q(x)) \wedge (r(x) \rightarrow s(x))), p(a) \vee r(a$	$q(a) \vee s(a)$
DD	$\forall x((p(x) \rightarrow q(x)) \wedge (r(x) \rightarrow s(x))), \neg q(a) \vee \neg s(a)$	$\neg p(a) \vee \neg r(a)$
BD	$\forall x((p(x) \rightarrow q(x)) \wedge (r(x) \rightarrow s(x))), p(a) \vee \neg s(a)$	$q(a) \vee \neg r(a)$
EG	$P(a)$	$\exists x P(x)$
UI	$\forall x A$	$A\{x \mapsto a\}$

and 20 yes/no test cases for each of the inference rules in Tables 1 and 2. We compute the average performance of the LLMs for each inference rule. We chose to use LogicBench because, of all the datasets available, this is the most comprehensive in that it covers the most inference rules.

We evaluate three LLMs in this work – GPT-3.5 Turbo, GPT-4.0125, and Gemini-1.5 Pro. Testing on the two versions of GPT allows us to examine the impact of significantly increasing the number of parameters for the same family while testing on Gemini allows us to cross-compare LLM logic performance from two different architectures and training regimens.

In Table 3 we provide the performance (average likelihood) of the three language models on multiple-choice questions and in Table 4 the results for yes/no questions is provided. There are several interesting observations to be made:

- Both versions of GPT perform better on almost all inference rules than Gemini on both propositional and first-order logic and for multiple-choice as well as yes/no questions. A possible explanation is that since Gemini is a multi-modal model (images, video, audio), it needs to learn a much more complex set of information when compared to text-only GPT-3.5 or the text+image GPT-4 models. This, in turn, limits its reasoning abilities.
- All models perform worse on yes/no questions than on the multiple choice questions. However, while the average performance of GPT-3.5 and GPT-4.0125 fall by 0.1, that of Gemini falls by 0.3. Interestingly, *Gemini performs worse than a random coin toss on yes/no questions with performance below 0.5.*
  Based on a more detailed analysis we believe that Gemini looks for the answer in the text of the question rather than do much logical reasoning (unlike the GPT models). As a result, in the case of multiple-choice questions, it is better able to guess the right answer since each of the four provided choices are textual. In the case of yes/no questions, it has no such hints and therefore it performs significantly poorly.
- Both, GPT-3.5 and GPT-4 perform better on first-order logic than on propositional logic. This is interesting given that first-order logic is more complex. However, a possible explanation is that human communication is more naturally modeled in first-order logic since existential and universal quantifiers are common in human discourse. Gemini also follows this trend but it is only a very small difference.
- GPT-4 has comparable performance to GPT-3.5 on propositional logic while GPT-3.5 performs significantly better on first-order logic. This is initially surprising given an approximately 8x difference in the number of parameters between the two models. However, since GPT-4 is trained on images in addition to text, it is likely that its reasoning capabilities are somewhat weaker because fewer parameters are devoted to that task.
- It is interesting to compare the *likelihood* values we obtain with the strict binary scoring results from [13]. The versions of LLMs (Bard/Gemini, ChatGPT, GPT-4) used in [13] are about 1 – 1 1/2 years older than the ones used in this paper and consequently there are differences in results. Note that while the paper lists the Google model as Gemini, it is in fact Bard which was trained only on text (unlike Gemini-1.5 Pro which is multi-modal). The reported results are given below: We note that the performance of Bard is much better than Gemini-1.5 Pro. This is not surprising since Bard is a text model while Gemini-1.5 is a multi-modal model. The results for ChatGPT and GPT-4 are somewhat similar to our own with the differences explained by the evolution of the models.**The important takeaway, however, is that the LLM performance when using likelihood or using strict binary scoring is quite similar**. A reason the likelihood scores are not higher is that even when the LLM selects the correct answer, the likelihood may be below 1.0. In strict binary choice, if the likelihood of the correct answer is slightly greater than the other choice (but still small), the model takes an argmax and outputs the right answer and scores 1 point.

	ChatGPT	GPT-4	Bard
Multiple Choice			
PL	0.81	0.89	0.88
FOL	0.82	0.91	0.87
Yes/No			
PL	0.63	0.75	0.63
FOL	0.76	0.83	0.76

5.1 Distribution of Likelihood Values

In this paper, we use likelihood values as membership functions. Therefore, it is of interest to see how the three LLMs score the *correct answer* for each logic question. Figure 1 plots the histograms for all the four cases. It is immediately evident that Gemini-1.5 assigns

Table 3: LLM performance on multiple-choice questions.

Logic	Rule	GPT-3.5 Turbo	GPT-4.0125	Gemini-1.5
Propositional	Bidirectional Dilemma	**0.95**	0.855	0.745
	Commutation	**0.905**	0.845	0.86
	Constructive Dilemma	0.89	0.965	**0.977**
	Destructive Dilemma	**0.825**	0.625	0.56
	Disjunctive Syllogism	0.405	**0.952**	0.62
	Hypothetical Syllogism	**0.95**	0.92	0.975
	Material Implication	**0.895**	0.85	0.885
	Modus Tollens	**0.62**	0.4875	0.61
	Average	0.805	**0.812**	0.779
First-Order	Bidirectional Dilemma	0.87	**0.895**	0.725
	Constructive Dilemma	0.82	**0.835**	0.688
	Destructive Dilemma	**0.91**	0.61	0.61
	Disjunctive Syllogism	**0.935**	0.833	0.825
	Existential Generalization	0.995	**0.997**	0.95
	Hypothetical Syllogism	**0.925**	0.835	0.915
	Modus Ponens	0.935	0.882	**0.96**
	Modus Tollens	**0.845**	0.687	0.648
	Universal Instantiation	0.78	**0.85**	0.75
	Average	**0.890**	0.824	0.786

Table 4: LLM performance on yes/no questions.

Logic	Rule	GPT-3.5 Turbo	GPT-4.0125	Gemini-1.5
Propositional	Bidirectional Dilemma	**0.842**	0.771	0.544
	Commutation	**0.818**	0.747	0.588
	Constructive Dilemma	**0.832**	0.765	0.545
	Destructive Dilemma	**0.891**	0.586	0.489
	Disjunctive Syllogism	0.502	**0.763**	0.493
	Hypothetical Syllogism	**0.686**	0.650	0.451
	Material Implication	**0.795**	0.633	0.459
	Modus Tollens	**0.705**	0.465	0.32
	Average	**0.759**	0.673	0.486
First-Order	Bidirectional Dilemma	**0.886**	0.84	0.467
	Constructive Dilemma	**0.885**	0.767	0.457
	Destructive Dilemma	**0.959**	0.672	0.428
	Disjunctive Syllogism	**0.855**	0.719	0.561
	Existential Generalization	0.725	**0.963**	0.518
	Hypothetical Syllogism	**0.753**	0.647	0.444
	Modus Ponens	**0.728**	0.704	0.603
	Modus Tollens	**0.738**	0.594	0.29
	Universal Instantiation	0.652	**0.728**	0.65
	Average	**0.798**	0.737	0.491

a likelihood of 0.5 and 0.1 to a significant number of questions. In other words, it is unable to confidently identify the correct answer and tends to assign a low likelihood. In contrast, the two GPT models mostly assign high likelihoods to the correct answer. The next observation is that GPT-3.5 assigns the highest number of correct answers a high likelihood (note the large blue bar centered near 1). **A conclusion we can draw from these figures is that GPT models are much better at fuzzy reasoning when compared to Gemini-1.5**.

5.2 Discussion

In analyzing the correct and incorrect answers provided by the three LLMs, we observe that there is a combination of logical reasoning and textual search being used to determine an answer. In the case of Gemini-1.5, we particularly observe a bias towards searching for an answer in the text provided rather than attempting significant logical deduction. It appears that the two GPT models do attempt logical deduction but that they appear to be better able to reason

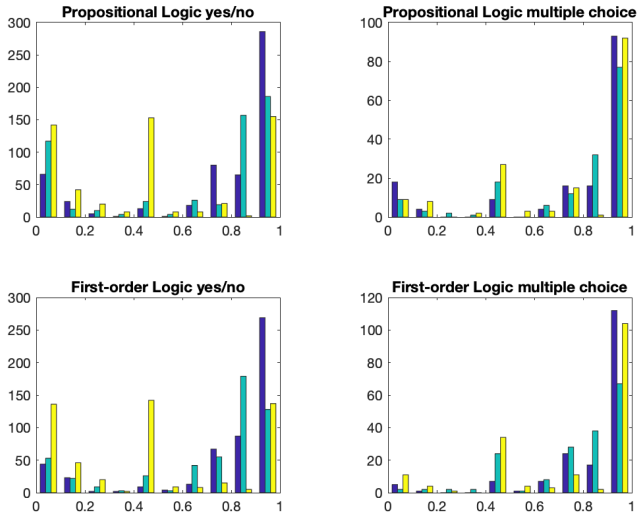

Figure 1: Histogram of likelihoods. Blue – GPT-3.5, Green – GPT-4, Yellow – Gemini-1.5.

for problems based on some rules such as CD, HS, etc. while they struggle with others such as MT.

One general observation we make is that most human discourse uses only a few of the inference rules described previously. For instance rules like destructive dilemma are very uncommon. Therefore, it makes sense to look at the performance of the LLMs on the subset of common inference rules rather than the whole list. According to ChatGPT Modus ponens (MP) and Disjunctive Syllogism appear most often (in addition to Simplification and Conjunction, which are not tested in this dataset). If we take and average of only these two rules for first-order logic, we obtain likelihoods of 0.935, 0.857, 0.892 for GPT-3.5, GPT-4.0125, and Gemini-1.5 respectively for multiple-choice questions. For yes/no questions, the likelihoods are, respectively, 0.791, 0.711, 0.582. It is interesting to observe that these values are somewhat higher than the averages reported in Tables 3 and 4, but the trends are similar.

6 Conclusion

In this paper we propose evaluating the logical reasoning skills of popular LLMs using fuzzy logic. Thus, unlike strict logical reasoning, we account for the fact that in most human reasoning, unstated context also plays a role in reaching conclusions. We used the likelihoods output by LLMs to score their answers on logical questions from a logic dataset. We considered both propositional logic as well as first-order logic. The results show that GPT-3.5 Turbo and GPT-4.0125 both perform very well on fuzzy logical reasoning and they appear to use correct formal reasoning in many instances. Gemini-1.5 Pro, on the other hand, appears to try to search for the answer in the text provided and as a result struggled with simple yes/no questions.

References

[1] 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *International Conference on Learning Representations*. https://openreview.net/forum?id=9Vrb9D0WI4

[2] Marco E.G.V. Cattaneo. 2017. The likelihood interpretation as the foundation of fuzzy set theory. *International Journal of Approximate Reasoning* 90 (2017), 333–340. doi:10.1016/j.ijar.2017.08.006

[3] Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language *(IJCAI'20)*. Article 537, 9 pages.

[4] Jason Wei et al. 2022. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*. https://openreview.net/forum?id=gEZrGCozdqR

[5] Simeng Han et al. 2022. FOLIO: Natural Language Reasoning with First-Order Logic. arXiv:2209.00840 [cs.CL]

[6] Christopher Hahn, Frederik Schmitt, Jens U. Kreber, Markus Norman Rabe, and Bernd Finkbeiner. 2021. Teaching Temporal Logics to Neural Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=dOcQK-f4byz

[7] Pratik Joshi, Somak Aditya, Aalok Sathe, and Monojit Choudhury. 2020. TaxiNLI: Taking a Ride up the NLU Hill. In *Proceedings of the 24th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Online, 41–55. https://www.aclweb.org/anthology/2020.conll-1.4

[8] Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth?. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina et al. Toutanova (Ed.). Association for Computational Linguistics, Online, 2627–2636. doi:10.18653/v1/2021.naacl-main.208

[9] Ping Liang and Fengming Song. 1996. What does a probabilistic interpretation of fuzzy sets mean? *IEEE Transactions on Fuzzy Systems* 4, 2 (1996), 200–205. doi:10.1109/91.493913

[10] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.* 55, 9, Article 195 (jan 2023), 35 pages. doi:10.1145/3560815

[11] Roger Martin-Clouaire. 1989. Semantics and computation of the generalized modus ponens: The long paper. *International Journal of Approximate Reasoning* 3, 2 (1989), 195–217. doi:10.1016/0888-613X(89)90006-6

[12] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. arXiv:2104.08773 [cs.CL]

[13] Mihir Parmar, Neeraj Varshney, Nisarg Patel, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. LogicBench: Towards Systematic Evaluation of Logical Reasoning Ability of Large Language Models. https://openreview.net/forum?id=71kocBuhNO

[14] Mohammed Saeed, Naser Ahmadi, Preslav Nakov, and Paolo Papotti. 2021. RuleBERT: Teaching Soft Rules to Pre-Trained Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 1460–1476. doi:10.18653/v1/2021.emnlp-main.110

[15] Abulhair Saparov and He He. 2023. Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=qFVVBzXxR2V

[16] Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A Diagnostic Benchmark for Inductive Reasoning from Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 4506–4515. doi:10.18653/v1/D19-1458

[17] Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. 2021. Diagnosing the First-Order Logical Reasoning Ability Through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 3738–3747. doi:10.18653/v1/2021.emnlp-main.303

[18] Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew E. Peters. 2020. Learning from Task Descriptions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 1361–1375. doi:10.18653/v1/2020.emnlp-main.105

[19] Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van Den Broeck. 2023. On the paradox of learning to reason from data. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence* (<conf-loc>, <city>Macao</city>, <country>P.R.China</country>, </conf-loc>) *(IJCAI '23)*. Article 375, 9 pages. doi:10.24963/ijcai.2023/375