

First Token Probabilities are Unreliable Indicators for LLM Knowledge

Justin Shao



Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/Eecs-2024-114

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/Eecs-2024-114.html>

May 16, 2024

Copyright © 2024, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

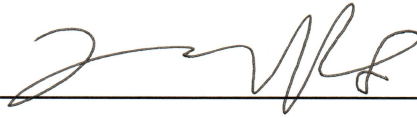
First Token Probabilities are Unreliable Indicators for LLM Knowledge
by Justin Shao

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, in partial satisfaction of the requirements for the
degree of **Master of Science, Plan II.**

Approval for the Report and Comprehensive Examination:

Committee:



Professor Jiantao Jiao
Research Advisor

05/15/2024

(Date)



Professor Alane Suhr
Second Reader

5/16/24

(Date)

FIRST TOKEN PROBABILITIES ARE UNRELIABLE INDICATORS FOR LLM KNOWLEDGE

Justin Shao
University of California, Berkeley

ABSTRACT

Multiple Choice Questions (MCQs) are a prevalent evaluation method used across many popular LLM benchmarks. Typically, these evaluations rely on first-token probabilities to deduce the model’s proposed answer. However, previous studies have demonstrated that first-token probabilities are vulnerable to prompt perturbations. In this study, we broaden our examination to explore the performance disparity between direct free-generation and the assessment of MCQs using first-token probabilities. Our experiments confirm the unreliability of first-token probabilities, as they often do not align with generation results. Additionally, we uncover a surprising finding: LLMs tend to struggle with arithmetic MCQs, even though they can reliably generate the correct answers.

1 INTRODUCTION

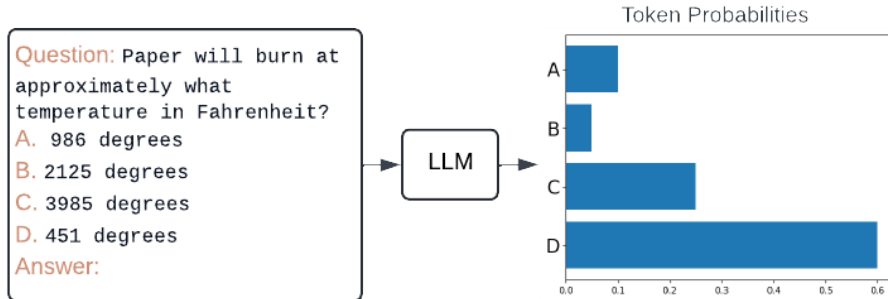


Figure 1: Example multiple choice question (MCQ), taken from the MMLU dataset

While large language models have demonstrated excellent performance in a variety of Natural Language Processing tasks, LLMs’ generation capabilities are notoriously difficult to evaluate. Apart from direct human evaluation that is expensive and difficult to scale, most popular evaluation methods either utilize automatic metrics as a proxy for human evaluation (Sai et al., 2022), or utilize a strong LLM as a judge to approximate human evaluation (Zheng et al., 2023).

One approach to circumvent this difficulty is to present questions in a multiple-choice question (MCQ) format, as demonstrated in Figure 1. Typically, a question is provided alongside multiple candidate answers, which then requires the LLM to choose the most suitable answer among the candidates. There are two major benefits to this approach: the simplicity of answer evaluation and the low computational cost during evaluation. As such, the MCQ format is currently used as a common form of evaluation in numerous popular benchmarks, including MMLU (Hendrycks et al., 2020), ARC-challenge (Clark et al., 2018), and commonsenseQA (Talmor et al., 2019).

Despite the popularity of MCQ evaluations, a recent study reveals that LLM responses to MCQs are prone to be influenced by the ordering of candidate answers, where models exhibit a selection bias for certain answer tokens over others (Zheng et al., 2024). Furthermore, LLM outputs to MCQs are also sensitive to prompting formats, resulting in drastic discrepancies in MCQ benchmark results (Lyu et al., 2024).

In this project, we further investigate the MCQ format by gauging the performance gap between free-generation and MCQ. Furthermore, we explore the correlation between model performance and their robustness against varying prompt formats. While it fits our intuition that MCQs are generally simpler to answer due to their restricted answer space, our experiments show surprising results that this assumption does not always hold. Most notably, we discovered that all tested LLMs consistently struggle with the MCQ format for arithmetic problems, despite varying parameter sizes and model capabilities.

2 EXPERIMENTAL SETUP

2.1 DIFFICULTY IN COMPARING MCQ AND FREE-GENERATION

To make fair comparisons between MCQ and free-generation performance, it is crucial to formulate questions in a way that ensures LLM responses are comparable across both formats without substantially changing the difficulty of the questions. There are two general approaches to creating datasets that can be tested in both formats, each with its specific shortcomings.

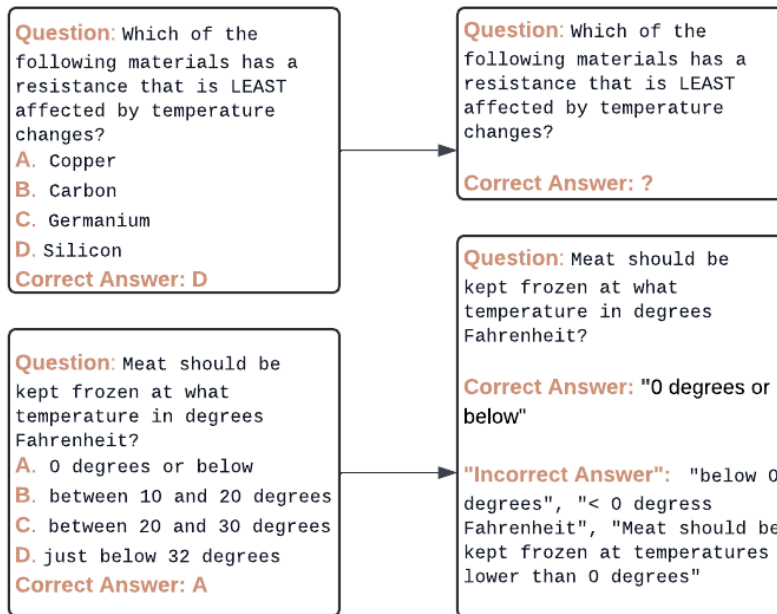


Figure 2: **(top)** If the question references the candidate answer choices, the adapted free-generation fails to make sense. **(bottom)** When using the MCQ correct answer choice as the golden answer, there are significant chances for LLMs to generate semantically correct answers that will not be considered correct via exact match. Both example questions are taken from MMLU.

The first approach is to adapt an MCQ dataset to perform free-generation. For each question, we perform a trivial adaptation, where candidate answers are removed from the prompt, and the generated sequence completion will be considered the model’s proposed answer. However, the correctness of the proposed answers for the new free-generation question is difficult to evaluate. Using the exact match (EM) metric to check the generated response against the original MCQ answer makes the difficulty of the generation question heavily dependent on the brevity of the original MCQ answer. Using proxy metrics like BLEU or ROGUE will ease the correctness requirement compared to EM, but the sensitivity to the specific wording of the correct answer choice still poses an issue. Furthermore, the adaptation setup will create non-sensible free-generation problems if the original

MCQ question makes references to the candidate answers, or incorporates comparatives between candidate answers. Figure 2 demonstrates some examples of the complications.

The second approach is to adapt a free-generation dataset to perform MCQs. For each free-generation question, we provide three additional false candidate answers to form an MCQ prompt. In this setup, the difficulty of the new MCQ is subject to the plausibility of the newly provided false candidate answers. If the created candidate answers are completely nonsensical, then the MCQ question can become trivial, potentially undermining the correlation between MCQ results and model knowledge. This possibility is demonstrated in Figure 3.

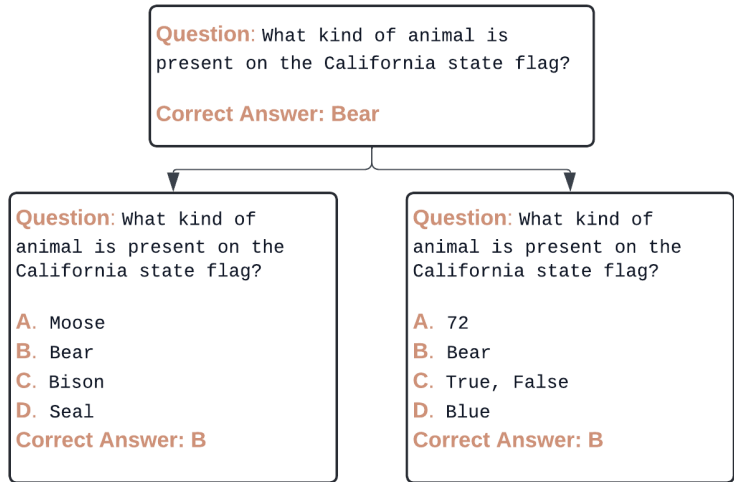


Figure 3: MCQ adaptations from a free-generation question can vary in difficulty due to different distractor answer choices. The MCQ can be trivial when the false candidate answers are implausible.

In our experiments, we used the second approach to adapt free-generation datasets to be compatible with the MCQ format. To address the issue of varying MCQ difficulty, we additionally aim to produce believable false candidate choices in the adaptation process. Our approach to this issue is to utilize a strong LLM (that is not being tested in our experiments) to provide false but plausible candidate answers, as shown in Figure 4. See appendix A.1 for the specific prompting format.

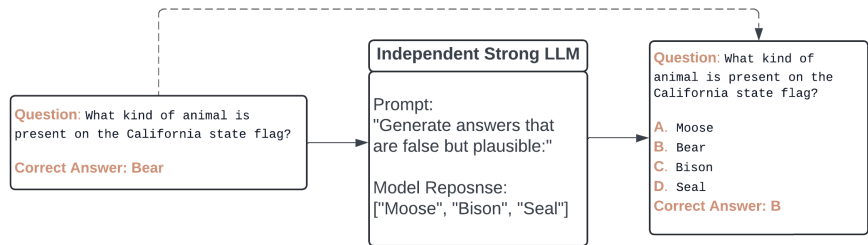


Figure 4: Dataset pre-processing procedure for each free-generation question, using an independent LLM to generate false answer candidates to be used in the MCQ setup.

2.2 MODELS

In our experiments, we focus on open-source decoder-only LLMs, evaluating 6 popular models: gemma2/7B (Team et al., 2024), llama2-7B (Touvron et al., 2023b), llama3-8/70B¹, and Mixtral8x7B (Jiang et al., 2024). All tested models are available on Huggingface, hence we can directly access the log probabilities of each token. In addition to the models being evaluated, we used gpt-3.5-turbo² to generate wrong answers for datasets as needed.

2.3 DATASETS

Arithmetic: The arithmetic dataset used by our experiments is a programmatically generated set of integer arithmetic problems. It includes addition problems from 1-5 digits, subtraction problems from 1-5 digits, multiplication problems from 1-3 digits, and multi-ops problems that involve 2 non-repeating randomly selected operators applied to 3 numbers. Table 5 provides examples for each category. The false candidate answers for each problem are generated by adding randomly sampled non-zero offsets to the correct answer.

TriviaQA: We randomly sampled a subset of 1,000 questions from TriviaQA (Joshi et al., 2017) to cover general QA. We selected TriviaQA over other potential datasets for two major reasons: (1) the questions in TriviaQA are sufficiently general and cover a wide range of domains; and (2) the dataset provides numerous “aliases” of the golden answer, each being a unique string that can be compared against for exact matches. This enables us to robustly assess the model-generated responses. For the MCQ setup, we prompt gpt-3.5-turbo to generate incorrect answers. The generated incorrect answers are further checked against the golden answer and its aliases, ensuring that all generated incorrect answers are *truly incorrect*. Finally, we only use questions that have golden answers with token length ≤ 15 , ensuring that no questions are too disproportionately disadvantaged for the free-generation format due to the exact match requirement.

2.4 EVALUATION

During evaluation, each question is evaluated alongside its adapted MCQ/free-generation counterpart.

For the MCQ format, we provide a 5-shot prompt to the model and use the model’s first-token probabilities for the label tokens (i.e. “A/B/C/D”). This evaluation method is a standard approach that is widely adopted for MCQs (Hendrycks et al., 2020; Liang et al., 2023). One modification we made is that we record the probability assigned to each of the answer labels, instead of only measuring the correctness of the label that is assigned the highest probability. In doing so, we hope to use the assigned probabilities to gauge the model’s confidence associated with each answer choice.

For the free-response format, we provide a 5-shot prompt and sample 20 responses at $T=1$ for each question. Each of the 20 sample responses is first normalized and then checked against the normalized golden answer and aliases for exact matches. The fraction of exact matches is interpreted as an empirical estimate of the probability of the model generating a correct answer, allowing us to directly compare the model’s probability of answering the question correctly in both the free-generation and MCQ setups.

2.4.1 ANSWER NORMALIZATION AND EXTRACTION

Arithmetic: we simply parse the generated text and take the first number in the sampled response as the model’s proposed answer.

TriviaQA: We follow the normalization procedure outlined in the Llama paper (Touvron et al., 2023a). We first parse the generated answers up to the first “.” or “\n”, then lowercase them, and finally remove articles, punctuations, and duplicate whitespaces.

¹<https://llama.meta.com/llama3/>

²<https://platform.openai.com/docs/models/gpt-3-5-turbo>

2.4.2 EVALUATING MISALIGNMENT

In our setup, we consider a model to be well-aligned if the model’s performance difference across formats is small. To quantify this expected alignment error, we evaluate the following:

$$E_{alignment} = \frac{1}{N} \sum_1^N |x_i - y_i| \quad (1)$$

Each coordinates (x_i, y_i) here represents a data point on the alignment chart, where x_i represents the average probability assigned to the correct choice in MCQ format, and y_i represents the average correctness in the free-generation format. Each data point is determined by grouping the questions into equal-sized bins, according to their x-values.

3 INVESTIGATING PERFORMANCE GAP BETWEEN FORMATS

3.1 STRONGER MODELS ARE NOT BETTER ALIGNED

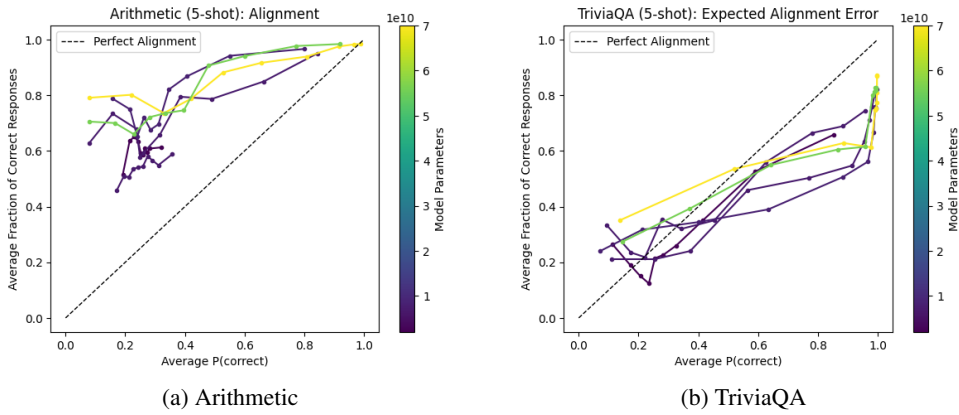


Figure 5: In our analysis of the Arithmetic dataset, we find that LLMs consistently outperform in the free-generation format, indicating a misalignment with their performance in other formats. Conversely, on the TriviaQA dataset, LLMs generally perform better in the MCQ format, except at the lower performance extremes. We use a diagonal line on our graphs to represent ideal alignment between the two formats.

The alignment plots on the two datasets are shown in Figure 5. On both datasets, the alignment trends are mostly similar across different model sizes. Graphing expected alignment error against the MCQ performance in Figure 6, we observe no correlation in the Arithmetic dataset and a weak positive correlation in the TriviaQA dataset. We conjecture that the relatively poor alignment on TriviaQA for better-performing models could be attributed to the existence of difficult problems that are easier to answer in the MCQ format. This is also supported by the observation that better-performing models have significantly more data points clustered towards the right edge of the alignment graph.

3.2 FORMAT PREFERENCES VARY ACROSS DATASETS

As shown in the right sub-figure in Figure 7, we find that all 6 tested LLMs perform better on average in the MCQ format compared to the free-generation format on the TriviaQA dataset. This aligns with our general human intuition that MCQs are easier than the free-generation format since the ability to come up with the correct answer would imply our ability to discern the correct answer.

As for the Arithmetic Dataset, we find that all 6 tested LLMs perform better in the free-generation format instead. This is particularly surprising, considering that models across all sizes demonstrate this preference. We will use the following sections to further explore this counterintuitive observation.

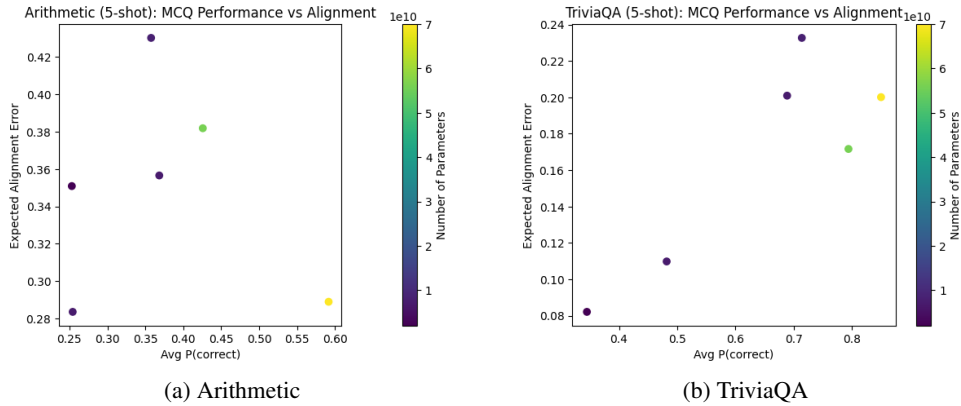


Figure 6: We show that on the Arithmetic dataset, LLMs demonstrate no correlation between MCQ performance and alignment error. On the TriviaQA dataset, LLMs that perform better in the MCQ format tend to have a higher expected alignment error.

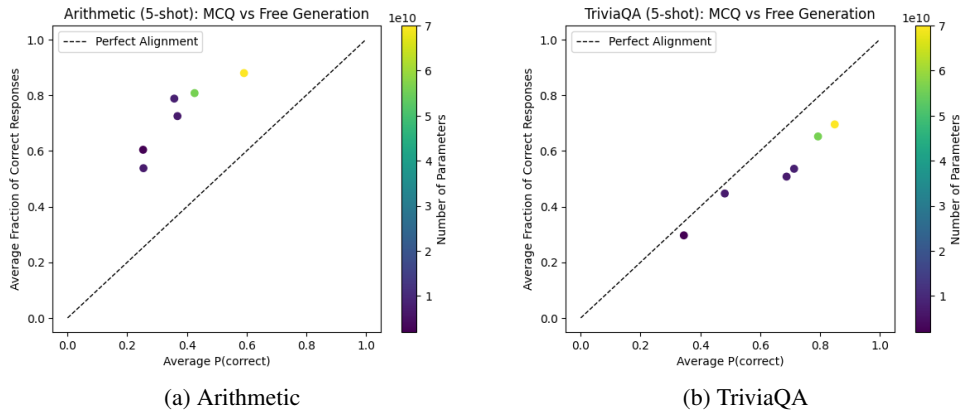


Figure 7: We show that on the Arithmetic dataset, LLMs consistently perform better on average in the free-generation format by a wide margin. On the TriviaQA dataset, LLMs perform better on average in the MCQ format with a smaller margin.

4 EXPLORING LLM PERFORMANCE DISCREPANCY ON ARITHMETIC

4.1 LLM OUTPUTS ACROSS PROMPT FORMATS ARE ONLY WEAKLY CORRELATED

Table 1: Comparison of performance across models in MCQ and free-response formats, with correlation coefficients (ρ)

Model	P(correct) (MCQ)	P(correct) (free-response)	ρ
gemma-2B	0.253	0.604	0.048
gemma-7B	0.368	0.725	0.254
llama2-7B	0.254	0.537	0.095
llama3-8B	0.358	0.788	0.237
llama3-70B	0.591	0.879	0.292
Mixtral-8x7B	0.426	0.808	0.321

Calculating the Pearson’s correlation coefficient, we observe that the model’s correctness across the two formats is not strongly correlated. This is further made clear in Table 2, where we show the

distribution of MCQ P(correct) when partitioned by free-generation P(correct). For all LLMs tested, the distributions for both partitions are relatively similar. This implies that the model’s correctness in one format can only provide weak predictive power for the correctness in the other format. This is in contrast to what we observe in the TriviaQA dataset 3, where the MCQ P(correct) distributions of the two partitions created by the same partition scheme remain mostly distinct.

Table 2: Arithmetic: MCQ P(correct) distribution categorized by free-generation P(correct)

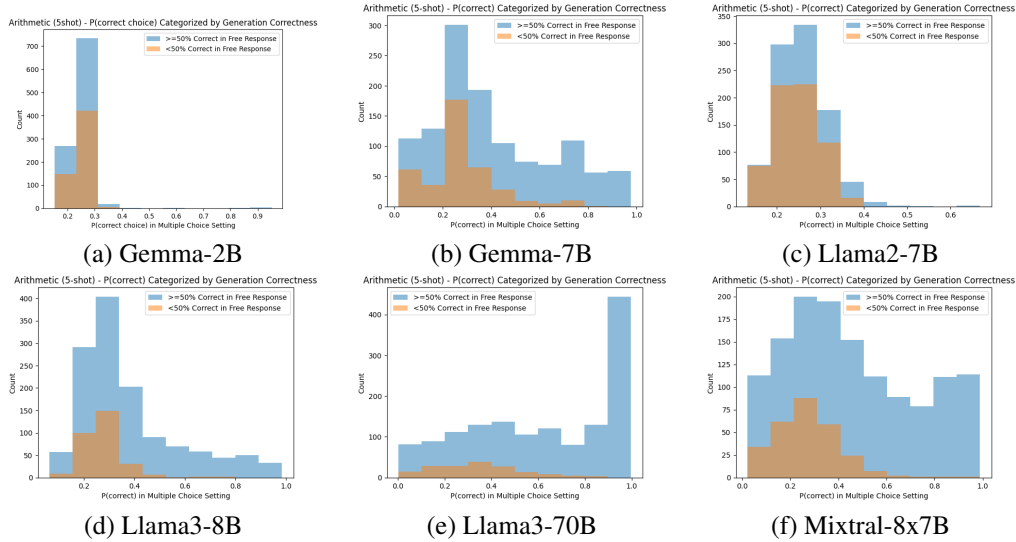
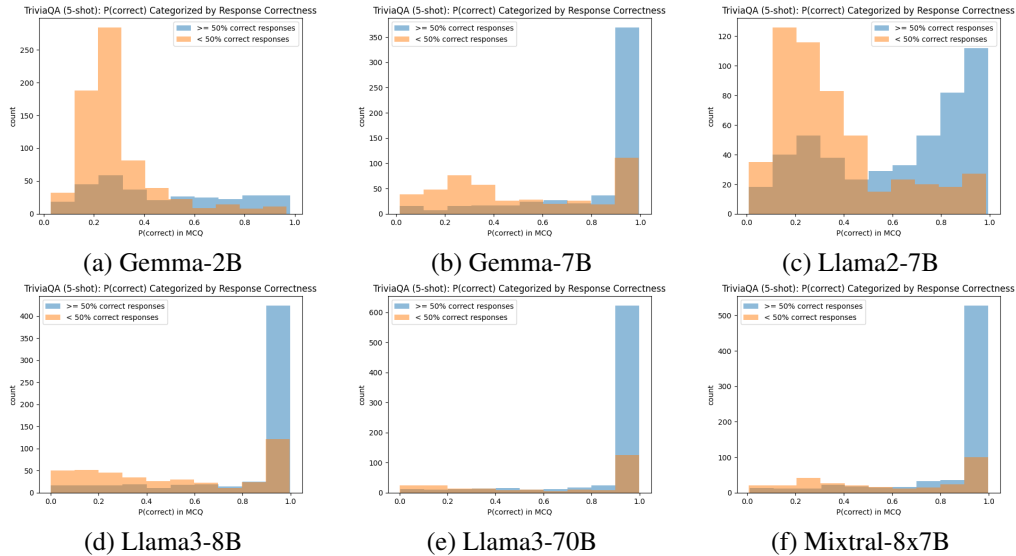


Table 3: TriviaQA: MCQ P(correct) distribution categorized by free-generation P(correct)



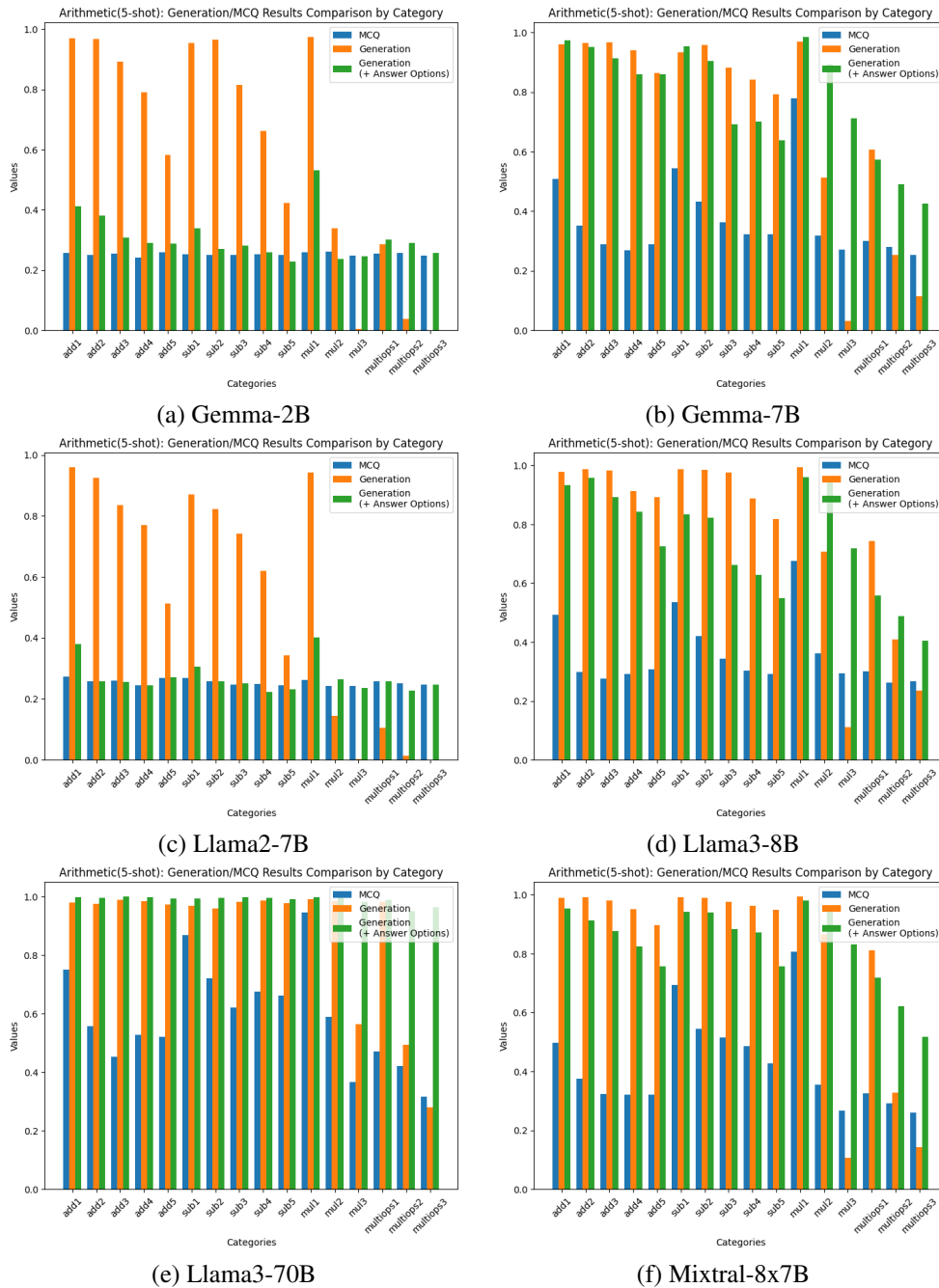
4.2 INCLUSION OF POSSIBLE ANSWERS MAY HARM GENERATION PERFORMANCE

To further understand the performance gap, we modified the free-generation format to provide candidate answers in the form of “possible answers” to form a third prompting format, which we call *free-generation with answers options*. At evaluation, we observed that for LLMs that performed better in the other two formats, the new format produces the best overall performance by significantly boosting P(correct) for difficult arithmetic question, like 3-digit multiplication or 3 digit multi-operational arithmetic. However, for LLMs that struggled with the MCQ format, the new

format performs worse than plain free-generation. Notably, MCQ format consistently performs the worst out of the three formats, across all arithmetic subcategories and for all tested models. The specific results is shown in Table 4.

Since the only significant difference between MCQ and *free-generation with answers options* is whether the answer is expected as a label token or as a numeric string, we conjecture that the performance discrepancy is primarily caused by LLM’s inability to produce token probabilities that match up with what would be generated in free-generation.

Table 4: Arithmetic: P(Correct) for all formats, evaluated over all arithmetic subcategories



5 CONCLUSION

This work studies the robustness of popular decoder-only LLMs in the context of prompt format variations. Through extensive experiments, we report two discoveries: (1) first-token probabilities are generally misaligned with the probabilities of the model generating correct answers through free-generation, and (2) LLMs especially struggle with multiple-choice questions in the arithmetic domain. In conclusion, we recommend proceeding with caution when using MCQs to evaluate Large Language Models, and especially recommend against using first-token probabilities as the sole method of model evaluation.

5.1 FUTURE WORKS

On our Arithmetic dataset, we observed the surprising result that MCQ performed significantly worse than free-generation. It is worth future investigation on whether or not this pattern generalizes to other problem domains.

REFERENCES

- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL <http://arxiv.org/abs/1803.05457>.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300, 2020. URL <https://arxiv.org/abs/2009.03300>.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L elio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Th eophile Gervet, Thibaut Lavril, Thomas Wang, Timoth e Lacroix, and William El Sayed. Mixtral of experts, 2024.
- Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *CoRR*, abs/1705.03551, 2017. URL <http://arxiv.org/abs/1705.03551>.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher R e, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models, 2023.
- Chenyang Lyu, Minghao Wu, and Alham Fikri Aji. Beyond probabilities: Unveiling the misalignment in evaluating large language models, 2024.
- Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, 55(2), jan 2022. ISSN 0360-0300. doi: 10.1145/3485766. URL <https://doi.org/10.1145/3485766>.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *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)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL <https://aclanthology.org/N19-1421>.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Riviere, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, L eonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Am elie H eliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Cl ement Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikula, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimentko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh

-
- Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023a.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023b.
- Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors, 2024.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

A PROMPTS AND QUESTION FORMATTING

A.1 GENERATING FALSE ANSWER CANDIDATES

Free-Generation to MCQ

We utilized the following prompt to generate false candidate answers for each free-generation question in TriviaQA. We observed that the returned responses occasionally include correct answers. Thus, we instruct the model to generate four instead of three incorrect answers, increasing the chances of the generation result containing at least 3 *truly incorrect* answers. Since we used *gpt-3.5-turbo* via the OpenAI API to perform this task, we also make the distinction between the system message and the user message.

```
System:
  You are a helpful assistant for quiz creation. You will
  be provided a trivia question-answer pair, and your job
  is to create four plausible but incorrect answers. Each
  incorrect answer should be something that might be mistaken
  for the correct answer, but is actually wrong.
User:
  Question: Who was the only president to resign from office?
  Correct Answer: Richard Nixon

  Please provide four incorrect answers, do not elaborate on why
  each answer is wrong.
```

A.2 ARITHMETIC/TRIVIAQA QUESTION FORMATTING

MCQs: MCQs are done in 5-shot. For brevity, a 1-shot example is shown.

```
Question: What is the value of  $75 + 22$ ?
A. 130
B. 97
C. 10
D. 144
Answer: B

Question: What is the value of  $227 + 243$ ?
Answer:
A. 920
B. 470
C. 810
D. 313
Answer:
```

Free-Generation: Free-generations are done in 5-shot. For brevity, a 1-shot example is shown.

```
Question: What is the value of  $75 + 22$ ?
Answer: 97

Question: What is the value of  $227 + 243$ ?
Answer:
```

Free-Generation with answers options: Free-generations with possible answers are done in 5-shot. For brevity, a 1-shot example is shown.

```
Question: What is the value of  $75 + 22$ ?
Here are some possible answers:
130
```

97
 10
 144
 Answer: 97

Question: What is the value of $227 + 243$?
 Here are some possible answers:
 920
 470
 810
 313
 Answer:

B ARITHMETIC DATASET COMPOSITION

CATEGORY	EXAMPLE	NUMERIC ANSWER	MCQ CHOICES
add1	$2 + 3$	5	[-4, 8, 14, 5]
add2	$76 + 41$	117	[138, 117, 82, 100]
add3	$164 + 465$	629	[755, 629, 289, 934]
add4	$4483 + 4870$	9353	[9353, 2531, 12866, 15203]
add5	$18571 + 84868$	103439	[133308, 149580, 103439, 3811]
sub1	$3 - 1$	2	[2, 3, -3, 0]
sub2	$13 - 50$	-37	[-17, -85, -37, 22]
sub3	$970 - 786$	184	[1568, 184, -378, 1438]
sub4	$2828 - 2477$	351	[4018, 351, -316, 873]
sub5	$48732 - 62785$	-14053	[-32533, -14053, -111478, -89193]
mul1	$5 * 2$	10	[5, 4, 10, -4]
mul2	$77 * 81$	6,237	[9847, 6237, 3349, 8514]
mul3	$887 * 895$	793865	[1413272, 18832, 656063, 793865]
multiops1	$8 * 8 - 4$	60	[2, 60, 104, 107]
multiops2	$22 + 35 - 84$	-27	[-41, -8, -27, -58]
multiops3	$727 + 590 * 722$	426707	[83934, 426707, 379914, 254149]

Table 5: Arithmetic dataset categories. One random example from each category is shown.