LLM Stability: A detailed analysis with some surprises

Berk Atil[1,2] * **Alexa Chittams**[2] **Lisheng Fu**[2] **Ferhan Ture**[2] **Lixinyu Xu**[2]
Breck Baldwin[2]

[1]Penn State University
[2]Comcast AI Technologies
bka5352@psu.edu, breck_baldwin@comcast.com

Abstract

LLM (large language model) practitioners commonly notice that outputs can vary for the same inputs, but we have been unable to find work that evaluates LLM stability as the main objective. In our study of 6 deterministically configured LLMs across 8 common tasks with 5 identical runs, we see accuracy variations up to 10%. In addition, no LLM consistently delivers repeatable accuracy across all tasks. We also show examples of variation that are not normally distributed and compare configurations with zero-shot/few-shot prompting and fine-tuned examples. To better quantify what is going on, we introduce metrics focused on stability: TARr@N for the total agreement rate at N runs over raw output, and TARa@N for total agreement over parsed-out answers. We suggest that stability metrics be integrated into leader boards and research results going forward.

Introduction

The usage of Large Language Models (LLM) has increased with their powerful capabilities including question answering (Robinson and Wingate 2023), reasoning (Qiao et al. 2023), code generation (Jiang et al. 2024b), etc. There are several hyper-parameters such as temperature, top-k, top-p, and repetition penalty, that affect the performance of the models significantly (Wang, Liu, and Awadallah 2023). Besides an effect on the performance, temperature affects the creativity of the model by modifying the probability distribution over the vocabulary (Wang et al. 2020; Wang, Liu, and Awadallah 2023). When the temperature is high, the distribution becomes flatter; on the other hand, when it is close to 0, the model should be more deterministic. The naive expectation would be that the model is entirely deterministic with temperature=0.

The typical way to measure LLM performance with benchmark datasets concludes with one single output (Hendrycks et al. 2021; Suzgun et al. 2023; Wang et al. 2024; Gema et al. 2024; Rein et al. 2023), possibly due to cost and computational time restrictions. However, if there is a significant variance in the output across identical runs, this reduces the validity of the benchmarks since the reported result may just be due to random chance.

In this work, we analyze the stability of various families of models on tasks from two common benchmarks (Hendrycks et al. 2021; Suzgun et al. 2023). We show that models are not deterministic even with a temperature of 0, and that the degree of stability changes from model to model and task to task. Furthermore, this degree of randomness can impact the ranked performance of the models across identical input runs. More specifically, our contributions include:

- We propose two metrics for LLM stability: total agreement rate@N for the answer (TARa@N) and total agreement rate@N for the raw model response (TARr@N).
- We quantify variations in LLM responses over 8 random tasks from two common benchmarks: BBH and MMLU.
- We compare the variations in different setups including zero-shot, few-shot, and fine-tuning.
- We conduct correlation analysis between stability, accuracy, input length, and output length.
- Data from runs and source code are available at https://github.com/Comcast/llm-stability.

Related Work

There have been investigations on the robustness of machine learning (ML) models with trivial changes to the input (Sehwag et al. 2019; Freiesleben and Grote 2023; Hancox-Li 2020; Rauber, Brendel, and Bethge 2017). For multiple choice questions, the order of the options changes the performance of the models significantly (Gupta et al. 2024). Xu et al. (2024) replace the correct option with "none of the above" and observe a dramatic performance change across different models. However, our focus is on the exact same input and setup, and noting the variations across N runs.

Biderman et al. (2024) introduce a standard evaluation toolkit for LLMs and suggest best practices for reproducibility, however, stability is not addressed. Works that mention stability includes Song et al. (2024), which analyzes the effect of temperature, sampling strategy, repetition penalty, and alignment algorithms on the performance. They have similar findings that LLMs have some variance in the output which should be taken into account in the evaluation benchmarks. However, they use a temperature of 1 when they report the variance of the outputs introducing the variability that our study seeks to eliminate. Ouyang et al. (2023) also

*Berk Atil completed this work during his internship at Comcast AI Technologies

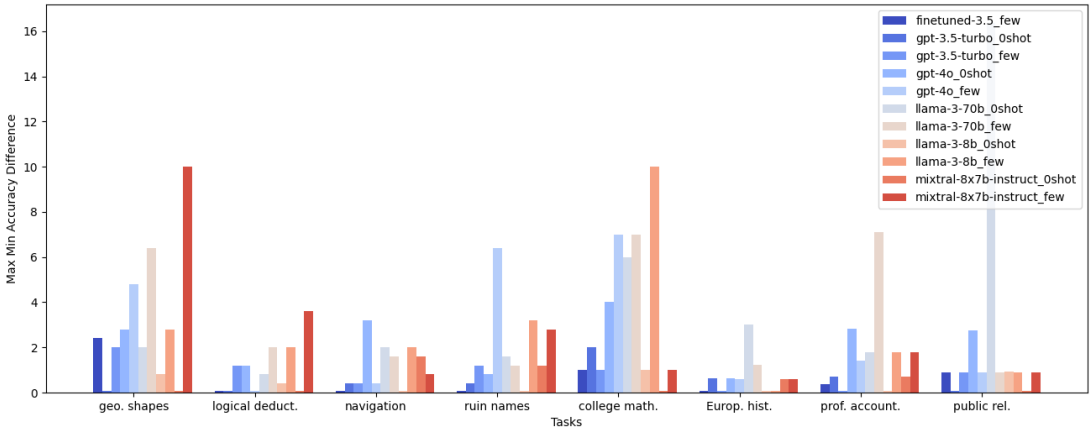

Figure 1: The difference between the maximum and minimum accuracy over 5 runs for each model and task in terms of %. .

do a similar stability analysis on LLMs with varying temperatures on code generation tasks. To the best of our knowledge, there is no no work that ran the same inputs and configurations (zero-shot, few-shot, and fine-tuned) with maximally deterministic hyper-parameters multiple times to assess the stability of LLM output.

Maximizing Determinism

Determinism is an important yet often overlooked challenge, with variations in practices across vendors and implementations. There are some strong themes most informatively delivered by ad-hoc means like forum posts (rUv 2023). Clearly, temperature is the key parameter with 0 being the most deterministic and 1 the least. It controls the "creativity" via the softmax function which changes the determinism of the output significantly (see equation 1 which shows the probability of word i where T is temperature.)

$$\frac{e^{\frac{y_i}{T}}}{\sum_{j=1}^{N} e^{\frac{y_j}{T}}} \tag{1}$$

When it is set to 0, the model is supposed to produce the same output given the same prompt in theory, on the other hand, it makes the model less creative.

We also experimented with top-p, which keeps the smallest set of tokens with probabilities that add up to set value, but found no difference for our stability purposes, so we do not report it.

Compute infrastructure, inputs, and configurations were fixed for the 5 and 20 run experiments.

There may be other ways to achieve determinism for any given model that we failed to use, but our goal was to model a fairly standard evaluation as might be used in a leader board or relative performance publication.

Datasets

Beyond Imitation Game Benchmark Hard (BBH) (Suzgun et al. 2023) is a benchmark consisting of 27 challenging tasks about traditional Natural Language Processing (NLP), mathematics, commonsense reasoning, etc. We randomly selected "navigation", a task to determine whether or not an agent returns back to the starting point given navigational steps; "ruin names", a task to pick a humorous simple edit of a band or movie name; "geometric shapes", a task to determine the geometric shape given in SVG path format; and "logical deduction with three objects", a task to deduce the order of a sequence of three objects from a set of conditions.

Task	# of Examples	# of Options
navigation	250	2
ruin names	250	4
geometric sha.	250	10
logical deduct.	250	3
Europ. hist.	165	4
college math	100	4
prof. account.	282	3
public rel.	110	4

Table 1: Statistics about the tasks.

Measuring Massive Multitask Language Understanding (MMLU) (Hendrycks et al. 2021) is another benchmark that contains 57 tasks in humanities, social sciences, STEM, and other important fields to learn. We randomly picked "high school European history" from humanities, with questions about the history of Europe; "college mathematics" from STEM with questions about calculus, algebra etc.; "public relations" from social sciences with questions about media theory, crisis management etc.; and "professional accounting" with accounting questions. All of the tasks are multiple choice questions with varying number of options, see brief statistics in Table 1.

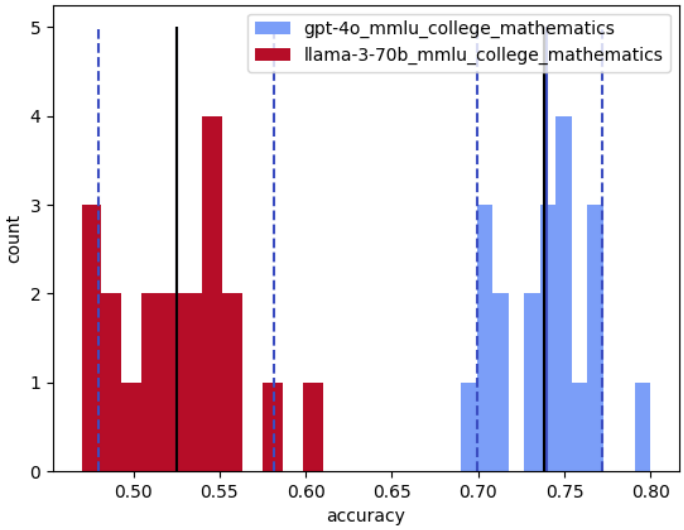

Figure 2: Accuracy over 20 identical runs on college math, temperature=0, top-p=1. Median in blue, mean in black with dashed 5% and 95% quantiles.

Experiments and Results

We use both few-shot and zero-shot prompting without Chain-of-Thought (CoT) (Wei et al. 2022). Regarding the number of examples for few-shots, we use the standard settings that are 3-shot for BBH tasks, 5-shot for MMLU tasks. We set the temperature to 0, top-p to 1, and fix the seed.

In addition, we fine-tuned GPT-3.5 Turbo using OpenAI API. As our training data, we use an auxiliary train subset of MMLU which has been found to be effective as a fine-tuning data (Zhou et al. 2023) for MMLU tasks and use 10-fold cross-validation for BBH tasks since there is no training data partition. After the model is fine-tuned, we prompt it in a few-shot setting. We use the OpenAI API and run each prompt 5 times with the same setup.

Models

We experiment with GPT-3.5 Turbo (Brown et al. 2020), GPT-4o (OpenAI et al. 2024), Llama-3-70B-Instruct (Meta 2024), Llama-3-8B-Instruct (Meta 2024), and Mixtral-8x7B-Instruct (Jiang et al. 2024a).

Metrics

The focus of this work is evaluating the stability of various LLMs across tasks, not the performance, but we report accuracy to validate that the LLMs are performing as expected and measure accuracy variation across runs which characterizes the larger impact of ignoring LLM stability when evaluating them. We also define some stability metrics that concern the reproducibility of answers and raw output.

- Minimum-maximum spread, which is the difference between the minimum and maximum accuracy over the runs.
- Total agreement rate@N (TAR@N), which is the percentage of test set questions across N runs where all answers are identical, regardless of whether the answer was correct. Its value might vary depending on the number of runs so we have @N notation. We have two different versions of TAR@N:
  - TARa@N (TAR@N for the answer) The parsed answer is the same, e.g., "The answer is a)" is the same as "a) is the answer". The answer may or may not be correct.
  - TARr@N (TAR@N for the raw model response) The LLM response is string equivalent. The parsed answer in the string may or may not be correct.
- Minimum, median, and maximum accuracy values over the runs.

Our metrics allow for the possibility of a 100% mean accuracy which would have 100% TARa@N but 0% TARr@N score. The TARr@N score is the strictest metric of stability since any character variation will result in a non-match. The reason for not reporting standard deviation and means is that the variations are not normal. We reran the GPT-4o and Llama-3-70b models on college math data 20 times, which are some of the most unstable configurations, to do the normality tests. Figure 2 demonstrates a clearly non-normal distribution with mean and median values far from the mode. A Kolmogorov-Smirnov normality test (Massey Jr 1951) also rejected the normal hypothesis with a smaller p-value than 10^{-9}.

Results

Table 2 summarizes the 5 run experiments across minimum, median, and maximum values followed by total agreement

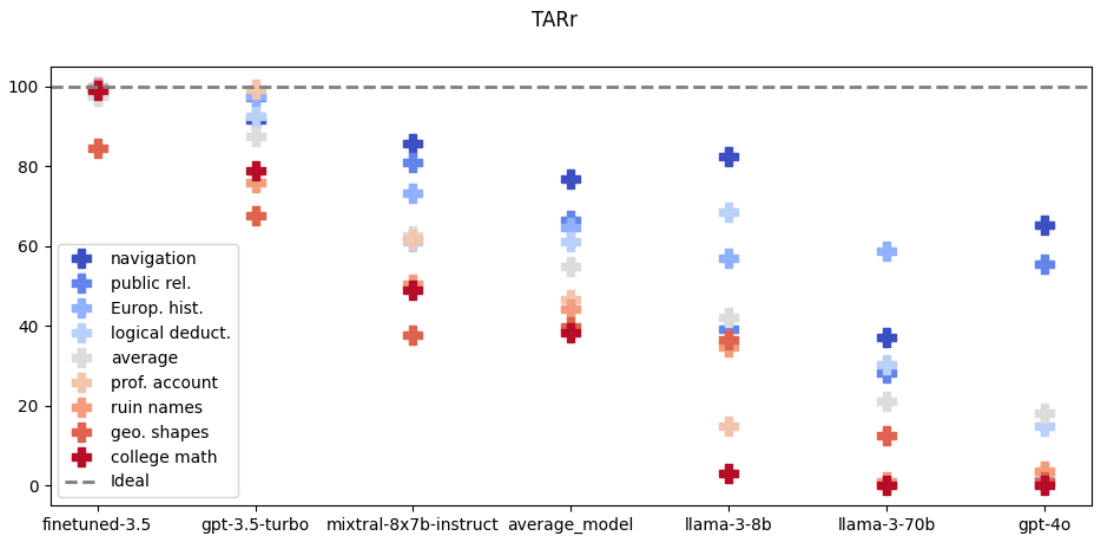

Figure 3: TARr@5 for each model, in terms of %. All models are prompted in few-shot setting. When the colors change from dark red to dark blue, TARr@5 gets better.

Task	gpt3.5	gpt4o	llama70b	llama8b	mixtral8-7b	fine-tuned-gpt3.5
navigation	95, (95), 96 ; ≈100, 92	**99, (99), 99** ; 99, 65	93, (94), 95 ; 95, 37	80, (81), 82 ; 96, 82	84, (84), 84 ; 99, 86	**97, (97), 97** ; **100**, 99
geo. shapes	58, (59), 60 ; 86, 68	66, (69), 71 ; 71, 1	59, (60), 66 ; 57, 12	48, (48), 50 ; 76, 36	23, (32), 33 ; 38, 38	48, (49), 50 ; 92, 84
logical deduct.	83, (83), 84 ; 99, 92	**100, (100), 100** ; **100**, 15	96, (96), 98 ; 95, 30	90, (91), 92 ; 94, 68	72, (75), 76 ; 88, 61	**86, (86), 86** ; **100, 100**
public rel.	75, (75), 75 ; 97, 97	77, (78), 78 ; 98, 55	76, (77), 77 ; 97, 28	68, (68), 69 ; 96, 39	66, (66), 67 ; 96, 81	73, (74), 74 ; 98, 98
Europ. hist.	**81, (81), 81** ; **100**, 98	**92, (92), 92** ; 99, 1	84, (85), 85 ; 98, 59	**33, (33), 33** ; 98, 57	74, (74), 75 ; 99, 73	**84, (84), 84** ; **100, 100**
ruin names	65, (66), 66 ; 96, 76	86, (92), 93 ; 89, 4	87, (88), 88 ; 92, 1	63, (65), 66 ; 91, 35	41, (42), 44 ; 82, 50	**62, (62), 62** ; **100**, ≈100
prof. account.	**54, (54), 54** ; **100**, 99	88, (89), 89 ; 90, 3	55, (59), 62 ; 66, 0	47, (49), 49 ; 91, 15	46, (47), 48 ; 91, 62	46, (47), 48 ; 91, ≈100
college math.	32, (32), 33 ; 99, 79	70, (71), 77 ; 63, 0	48, (50), 55 ; 39, 0	19, (23), 29 ; 65, 3	31, (32), 32 ; 75, 49	37, (38), 38 ; 99, 99

Table 2: Minimum accuracy, (median accuracy), Maximum accuracy; TARa@5, TARr@5. Completely stable results shown in **boldface**. These models are prompted in few-shot setting.

rate TARa@5 and TARr@5. Perfectly stable system performance would show the same score for minimum, median, and maximum accuracy, and 100% for both TARa@5 and TARr@5. No model/task achieves this performance except the fine-tuned model on logical deduction and European history tasks.

Table 3 shows the median performance across datasets for TARa@5 and TARr@5 metrics. TARa@5 demonstrates much higher medians overall as expected. Fine-tuned GPT-3.5 Turbo has non-100% values but the median remains 100% .

Figure 1 differentiates experiment configurations such as fine-tuning, N-shot across tasks or LLMs for the minimum and maximum accuracy values. Many models have between 5-10% spreads on some configuration/task.

Figure 3 shows the TARr@5 for each task and model in a few-shot setting. GPT-3.5 Turbo outperforms other models across tasks. Moreover, fine-tuning on GPT-3.5 Turbo makes the stability almost perfect. This suggests that fine-tuning models significantly improves stability.

Figure 4 shows TARa@5, the x and y axes are inverted from figure 3 for better presentation. The abstraction offered by parsing out the answer results in higher numbers across the board, but it is far from perfect and very task-specific. The high performing circles in European history indicate that leader boards on this task are expected to be more reliable. On the other hand, the scattered circles in college math and geometric shapes indicate that results reported on these tasks might not be as robust.

Correlation Analysis

We performed a Spearman rank correlation analysis on the factors discussed previously: TARa@5, TARr@5, minimum-maximum accuracy spread, along with accuracy, input length, and output length. The results are shown in a heat map in Figures 5 and 6 with the few-shot and zero-shot prompted models respectively. We define accuracy as the mean accuracy over the 5 runs with the same model and dataset setup. Input length and output length are median word counts split by space, calculated over the input and output of each LLM experiment setup. We split the words by space instead of using a particular tokenizer, as the models we experimented with used different tokenizers.

The results show a strong negative correlation between the output length and TARa@5, as well as between the output length and TARr@5 in few-shot settings. This means that as

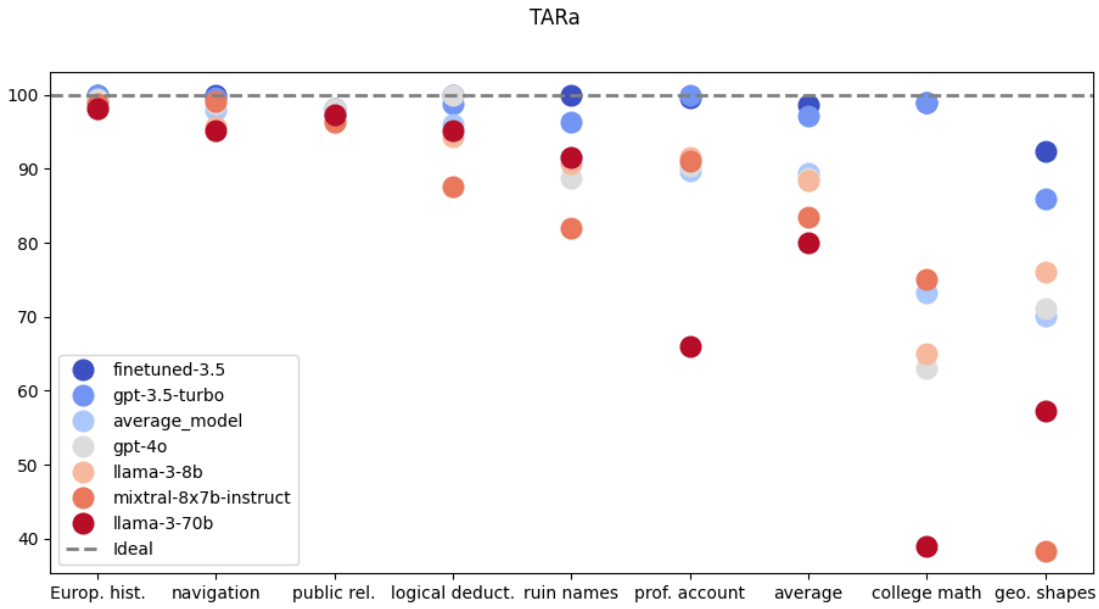

Figure 4: TARa@5 for each task, in terms of %. All models are prompted in few-shot setting. When the colors change from dark red to dark blue, TARa@5 gets better.

an LLM's output length increases, the stability of the output decreases, resulting in more diverse natural language responses as well as the actual multiple choice selection. The strong negative correlation between LLM output length and stability could motivate those using LLMs commercially to restrict the max generation tokens to control the stability. We also see a moderate positive correlation between accuracy and TARa@5. This indicates that when the LLM is more confident it becomes more stable for multiple choice selections. Additionally, in the few-shot setting, there is a moderate negative correlation between the output length and accuracy which indicates that restricting max generation tokens does not help only stability but also accuracy.

In addition to general correlations, we also look at correlation maps per model to how general findings apply to each. We find that that GPT and Llama-3 models are more stable when they generate shorter responses This trend applies to Mixtral only in few-shot setting but it behaves in the opposite way in zero-shot setting. Another interesting finding is that Mixtral and Llama-3 models are more stable when they are more accurate. Last but not least, Mixtral is more stable when the input is longer.

Discussion

Theoretically/naively, the LLM should be consistent given the same input which results in 100% TARa@5, 100% TARr@5, and 0% difference in the minimum and maximum values across all tasks with hyper-parameters set as they are.

The TARr@5 scores show that models are not robust at the string variation level although they are far more robust at the parsed answer level. String variation does not affect a human reader much because we can extract the same answer even if the output format is different, but a downstream system that needs to parse the LLM response can be affected significantly when the format or pattern is different. This should be taken into account when traditional artificial intelligence (AI) systems are being replaced with LLMs.

TARa@5 are significantly more consistent than TARr@5 which is the basis of our reported accuracy variation, but our answer extraction system has many hard-coded parts which reduces the generality of the system and we have no guarantee that raw outputs will retain previous patterns.

The maximum/minimum spread should be 0% theoretically, but we only see 4 instances across all models/tasks with that property. There are some cases where there is a high difference such as 10% for Mixtral-8x7b on geometric shapes or 10% for Llama-3-8B on college math. Only GPT-3.5 Turbo is at or below 2% across all tasks.

Implications on Practical Engineering

Until the advent of GPUs which introduced one source of non-determinism (Nvidia 2024) in neural networks, AI models were generally entirely deterministic given the same input. Any mistakes made could be attributed to data that was fed into the model, which exposed the model's limitations in generalizing from its training process (e.g., under-/over-fitting).

Non-deterministic AI brings new challenges to developers, especially in commercial applications:

- The usage of unit tests for AI functions in the same way as a mathematical function is limited because of non-determinism. Those unit tests might fail now and have to be handled in a non-unit test framework such as regression testing that tolerates variability.

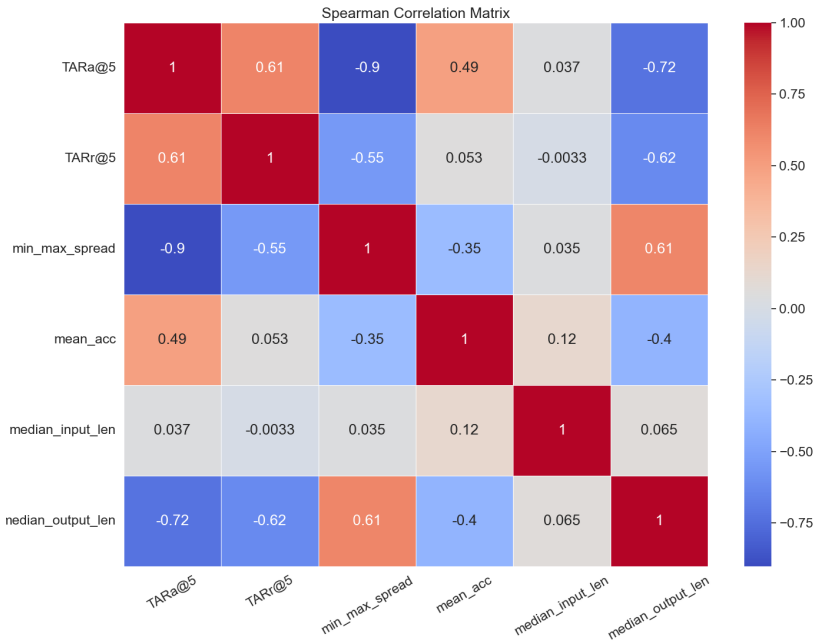

Figure 5: Spearman correlation matrix between metrics in few-shot setting for the models that show significant instability (all models except the fine-tuned model)

Model	TARa@5 Median	TARr@5 Median
fine-tuned-gpt3.5-few	100%	100%
gpt3.5-few	99%	92%
gpt3.5-0shot	99%	99%
mixtral8-7b-few	89%	61%
mixtral8-7b-0shot	99%	88%
llama8b-few	92%	37%
llama8b-0shot	100%	98%
llama70b-few	93%	20%
llama70b-0shot	69%	2%
gpt4o-few	94%	3%
gpt4o-0shot	93%	11%

Table 3: Median of TARa@5 and TARr@5 over datasets. The models with the "few" in their name are prompted in few-shot setting.

- Low stability might also increase the potential for inexplicable errors that are very different from human mistakes such as responding as "none of the above" when the task is a multiple choice selection.
- Instability of the format of the outputs can result in a failure of the parser for the downstream systems.
- Assuming that errors due to non-determinism and errors due to system performance are IID (independently, identically distributed), the factors are multiplicative reductions of performance. A dialog system that uses non-deterministic LLM classifiers to manage transitions multiplicatively degrades with each additional state in the sequence of dialog states, e.g., a dialog system with 4 classifiers that are 95% stable will show $.95^4 = .814$ expected performance before even factoring in the accuracy of the classifier on novel inputs.
- One of the most important effects is in system complexity that has to handle gracefully "usually correct but this time wrong" results. Zipfian distributions are commonly seen in applied AI systems where the frequency of an input/-category is inversely related to its rank in count sorted order, $frequency \propto 1/rank$. Testing tends to concentrate on the frequent events and that delivers confidence that the resulting system is stable for the common inputs. However, the lack of stability undermines the entire foundation of this confidence especially if mistakes are costly.

Conclusion

We have made a systematic analysis of the stability of LLMs with the hyper-parameters that should maximize determinism. Our results show that LLMs can be very unstable in standard setups. Furthermore, an LLM rarely produces the same exact response 5 times given the same input; however, the parsed answer is often more stable. Another interesting finding is that the accuracy values that you get over different runs are not normally distributed. Regarding the factors that affect the stability, the task difficulty and output length of the model are important.

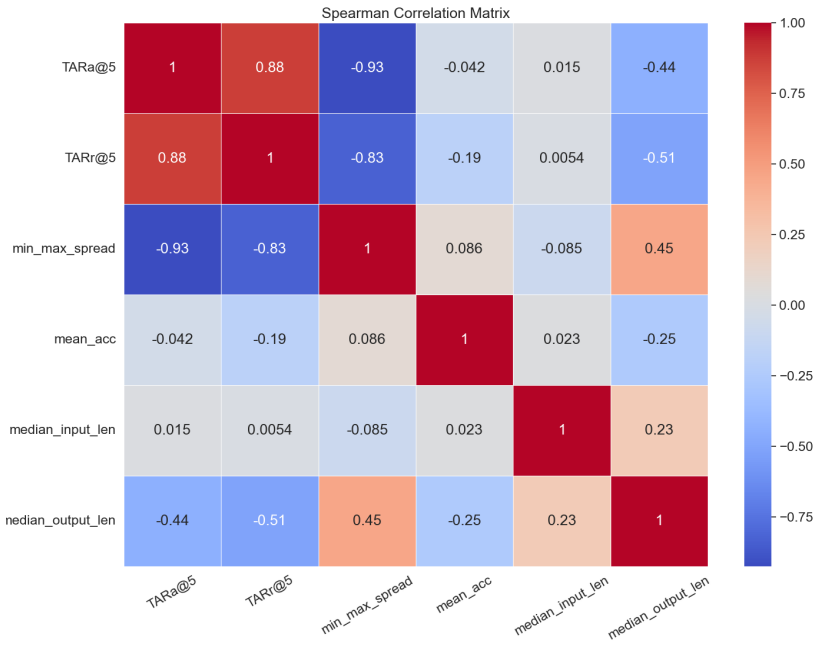

Figure 6: Spearman correlation matrix between metrics in zero-shot setting for the models that show significant instability (all models except the fine-tuned model)

Future Work

There are various potential directions to build on these findings. For instance, how can we improve the instability of LLMs during training or inference time (e.g., adding a meta prompt to indicate the model is only allowed to answer with a single letter)? Second, how to take the lack of stability of LLMs into account in business products? Third, how to communicate with decision-makers about instability? Last but not least, more analysis can be done to see if there is any correlation between the stability and fine-grained errors of the model (e.g., false negatives, false positives) besides correct predictions.

References

Biderman, S.; Schoelkopf, H.; Sutawika, L.; Gao, L.; Tow, J.; Abbasi, B.; Aji, A. F.; Ammanamanchi, P. S.; Black, S.; Clive, J.; et al. 2024. Lessons from the Trenches on Reproducible Evaluation of Language Models. *arXiv preprint arXiv:2405.14782*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Freiesleben, T.; and Grote, T. 2023. Beyond generalization: a theory of robustness in machine learning. *Synthese*, 202(4): 109.

Gema, A. P.; Leang, J. O. J.; Hong, G.; Devoto, A.; Mancino, A. C. M.; Saxena, R.; He, X.; Zhao, Y.; Du, X.; Madani, M. R. G.; et al. 2024. Are We Done with MMLU? *arXiv preprint arXiv:2406.04127*.

Gupta, V.; Pantoja, D.; Ross, C.; Williams, A.; and Ung, M. 2024. Changing Answer Order Can Decrease MMLU Accuracy. *arXiv preprint arXiv:2406.19470*.

Hancox-Li, L. 2020. Robustness in machine learning explanations: does it matter? In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, 640–647. New York, NY, USA: Association for Computing Machinery. ISBN 9781450369367.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Jiang, A. Q.; Sablayrolles, A.; Roux, A.; Mensch, A.; Savary, B.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Hanna, E. B.; Bressand, F.; Lengyel, G.; Bour, G.; Lample, G.; Lavaud, L. R.; Saulnier, L.; Lachaux, M.-A.; Stock, P.; Subramanian, S.; Yang, S.; Antoniak, S.; Scao, T. L.; Gervet, T.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2024a. Mixtral of Experts. arXiv:2401.04088.

Jiang, J.; Wang, F.; Shen, J.; Kim, S.; and Kim, S. 2024b. A Survey on Large Language Models for Code Generation. *arXiv preprint arXiv:2406.00515*.

Massey Jr, F. J. 1951. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253): 68–78.

Meta. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date.

Nvidia. 2024. Floating Point and IEEE 754 Compliance for NVIDIA GPUs.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; Avila, R.; Babuschkin, I.; Balaji, S.; Balcom, V.; Baltescu, P.; Bao, H.; Bavarian, M.; Belgum, J.; Bello, I.; Berdine, J.; Bernadett-Shapiro, G.; Berner, C.; Bogdonoff, L.; Boiko, O.; Boyd, M.; Brakman, A.-L.; Brockman, G.; Brooks, T.; Brundage, M.; Button, K.; Cai, T.; Campbell, R.; Cann, A.; Carey, B.; Carlson, C.; Carmichael, R.; Chan, B.; Chang, C.; Chantzis, F.; Chen, D.; Chen, S.; Chen, R.; Chen, J.; Chen, M.; Chess, B.; Cho, C.; Chu, C.; Chung, H. W.; Cummings, D.; Currier, J.; Dai, Y.; Decareaux, C.; Degry, T.; Deutsch, N.; Deville, D.; Dhar, A.; Dohan, D.; Dowling, S.; Dunning, S.; Ecoffet, A.; Eleti, A.; Eloundou, T.; Farhi, D.; Fedus, L.; Felix, N.; Fishman, S. P.; Forte, J.; Fulford, I.; Gao, L.; Georges, E.; Gibson, C.; Goel, V.; Gogineni, T.; Goh, G.; Gontijo-Lopes, R.; Gordon, J.; Grafstein, M.; Gray, S.; Greene, R.; Gross, J.; Gu, S. S.; Guo, Y.; Hallacy, C.; Han, J.; Harris, J.; He, Y.; Heaton, M.; Heidecke, J.; Hesse, C.; Hickey, A.; Hickey, W.; Hoeschele, P.; Houghton, B.; Hsu, K.; Hu, S.; Hu, X.; Huizinga, J.; Jain, S.; Jain, S.; Jang, J.; Jiang, A.; Jiang, R.; Jin, H.; Jin, D.; Jomoto, S.; Jonn, B.; Jun, H.; Kaftan, T.; Łukasz Kaiser; Kamali, A.; Kanitscheider, I.; Keskar, N. S.; Khan, T.; Kilpatrick, L.; Kim, J. W.; Kim, C.; Kim, Y.; Kirchner, J. H.; Kiros, J.; Knight, M.; Kokotajlo, D.; Łukasz Kondraciuk; Kondrich, A.; Konstantinidis, A.; Kosic, K.; Krueger, G.; Kuo, V.; Lampe, M.; Lan, I.; Lee, T.; Leike, J.; Leung, J.; Levy, D.; Li, C. M.; Lim, R.; Lin, M.; Lin, S.; Litwin, M.; Lopez, T.; Lowe, R.; Lue, P.; Makanju, A.; Malfacini, K.; Manning, S.; Markov, T.; Markovski, Y.; Martin, B.; Mayer, K.; Mayne, A.; McGrew, B.; McKinney, S. M.; McLeavey, C.; McMillan, P.; McNeil, J.; Medina, D.; Mehta, A.; Menick, J.; Metz, L.; Mishchenko, A.; Mishkin, P.; Monaco, V.; Morikawa, E.; Mossing, D.; Mu, T.; Murati, M.; Murk, O.; Mély, D.; Nair, A.; Nakano, R.; Nayak, R.; Neelakantan, A.; Ngo, R.; Noh, H.; Ouyang, L.; O'Keefe, C.; Pachocki, J.; Paino, A.; Palermo, J.; Pantuliano, A.; Parascandolo, G.; Parish, J.; Parparita, E.; Passos, A.; Pavlov, M.; Peng, A.; Perelman, A.; de Avila Belbute Peres, F.; Petrov, M.; de Oliveira Pinto, H. P.; Michael; Pokorny; Pokrass, M.; Pong, V. H.; Powell, T.; Power, A.; Power, B.; Proehl, E.; Puri, R.; Radford, A.; Rae, J.; Ramesh, A.; Raymond, C.; Real, F.; Rimbach, K.; Ross, C.; Rotsted, B.; Roussez, H.; Ryder, N.; Saltarelli, M.; Sanders, T.; Santurkar, S.; Sastry, G.; Schmidt, H.; Schnurr, D.; Schulman, J.; Selsam, D.; Sheppard, K.; Sherbakov, T.; Shieh, J.; Shoker, S.; Shyam, P.; Sidor, S.; Sigler, E.; Simens, M.; Sitkin, J.; Slama, K.; Sohl, I.; Sokolowsky, B.; Song, Y.; Staudacher, N.; Such, F. P.; Summers, N.; Sutskever, I.; Tang, J.; Tezak, N.; Thompson, M. B.; Tillet, P.; Tootoonchian, A.; Tseng, E.; Tuggle, P.; Turley, N.; Tworek, J.; Uribe, J. F. C.; Vallone, A.; Vijayvergiya, A.; Voss, C.; Wainwright, C.; Wang, J. J.; Wang, A.; Wang, B.; Ward, J.; Wei, J.; Weinmann, C.; Welihinda, A.; Welinder, P.; Weng, J.; Weng, L.; Wiethoff, M.; Willner, D.; Winter, C.; Wolrich, S.; Wong, H.; Workman, L.; Wu, S.; Wu, J.; Wu, M.; Xiao, K.; Xu, T.; Yoo, S.; Yu, K.; Yuan, Q.; Zaremba, W.; Zellers, R.; Zhang, C.; Zhang, M.; Zhao, S.; Zheng, T.; Zhuang, J.; Zhuk, W.; and Zoph, B. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Ouyang, S.; Zhang, J. M.; Harman, M.; and Wang, M. 2023. LLM is Like a Box of Chocolates: the Nondeterminism of ChatGPT in Code Generation. *arXiv preprint arXiv:2308.02828*.

Qiao, S.; Ou, Y.; Zhang, N.; Chen, X.; Yao, Y.; Deng, S.; Tan, C.; Huang, F.; and Chen, H. 2023. Reasoning with Language Model Prompting: A Survey. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5368–5393. Toronto, Canada: Association for Computational Linguistics.

Rauber, J.; Brendel, W.; and Bethge, M. 2017. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*.

Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2023. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. arXiv:2311.12022.

Robinson, J.; and Wingate, D. 2023. Leveraging Large Language Models for Multiple Choice Question Answering. In *The Eleventh International Conference on Learning Representations*.

rUv. 2023. Cheat Sheet: Mastering Temperature and Top_p in ChatGPT API. https://community.openai.com/t/cheat-sheet-mastering-temperature-and-top-p-in-chatgpt-api/172683. Accessed: 2024-18-13.

Sehwag, V.; Bhagoji, A. N.; Song, L.; Sitawarin, C.; Cullina, D.; Chiang, M.; and Mittal, P. 2019. Analyzing the Robustness of Open-World Machine Learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, AISec'19, 105–116. New York, NY, USA: Association for Computing Machinery. ISBN 9781450368339.

Song, Y.; Wang, G.; Li, S.; and Lin, B. Y. 2024. The Good, The Bad, and The Greedy: Evaluation of LLMs Should Not Ignore Non-Determinism. *arXiv preprint arXiv:2407.10457*.

Suzgun, M.; Scales, N.; Schärli, N.; Gehrmann, S.; Tay, Y.; Chung, H. W.; Chowdhery, A.; Le, Q.; Chi, E.; Zhou, D.; and Wei, J. 2023. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 13003–13051. Toronto, Canada: Association for Computational Linguistics.

Wang, C.; Liu, X.; and Awadallah, A. H. 2023. Cost-effective hyperparameter optimization for large language model generation inference. In *International Conference on Automated Machine Learning*, 21–1. PMLR.

Wang, P.-H.; Hsieh, S.-I.; Chang, S.-C.; Pan, J.-Y.; Chen, Y.-T.; Wei, W.; and Juan, D.-C. 2020. Contextual Temperature for Language Modeling.

Wang, Y.; Ma, X.; Zhang, G.; Ni, Y.; Chandra, A.; Guo, S.; Ren, W.; Arulraj, A.; He, X.; Jiang, Z.; et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Xu, H.; Lou, R.; Du, J.; Mahzoon, V.; Talebianaraki, E.; Zhou, Z.; Garrison, E.; Vucetic, S.; and Yin, W. 2024. LLMs' Classification Performance is Overclaimed. *arXiv preprint arXiv:2406.16203*.

Zhou, K.; Zhu, Y.; Chen, Z.; Chen, W.; Zhao, W. X.; Chen, X.; Lin, Y.; Wen, J.-R.; and Han, J. 2023. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*.