Testing ChatGPT for Stability and Reasoning: A Case Study Using Italian Medical Specialty Tests

Silvia Casola[1], Tiziano Labruna[2,3], Alberto Lavelli[3] and Bernardo Magnini[3]

[1] *University of Turin*
[2] *Free University of Bozen-Bolzano*
[3] *Fondazione Bruno Kessler*

Abstract

Although large language models (LLMs) are achieving impressive performance under zero- and few-shot learning configurations, their reasoning capacities are still poorly understood. As a step in this direction, we present several experiments on multiple-choice question answering, a setting that allows us to evaluate the stability of the model under different prompting, the capacity to understand when none of the provided answers is correct, and to reason on specific answering strategies (e.g., recursively eliminate the worst answer). We use the Italian medical specialty tests yearly administered to admit medical doctors to specialties. Results show that a gpt-3.5-turbo model achieves excellent performance in the absolute score (an average of 108 out of 140) while still suffering in certain reasoning capacities, particularly in failing to understand when none of the provided answers is correct.

Keywords

Large Language Models, ChatGPT, Stability

1. Introduction

Instruction-tuned Large Language Models (LLMs) have recently shown unprecedented results in various tasks in different languages [1]. Beyond the impressive performance, their popularity derives from the possibility of using them with no or little training data for multiple tasks and languages. In fact, instruction-tuned LLMs go beyond the previously established learning paradigm based on transfer learning — where a model, first pre-trained with no supervision, must be fine-tuned on downstream task-specific data — and are typically used in a zero- or few-shot manner.

LLMs performance and ease of use have attracted interest from Natural Language Processing researchers and practitioners. However, most previous work has focused on the models' performance and practical applications. Less relevance has been given to the models' stability and reliability, e.g., in the variability of their outputs or reasoning capacities in controlled settings. This is even more problematic since many of the most popular and performative instruction-tuned LLMs are proprietary, and the details of the exploited data, architecture, and training procedures are at best superficially discussed in technical reports [2] rather than proper research papers, conferences, and other scientific venues.

In this work, we choose a more holistic approach to analyzing results, stability, and consistency in Italian. We do so by considering a case study: the Italian medical specialty tests. The test consists of 140 multiple-choice questions in various medical areas, based on which Italian medical doctors are evaluated and ranked if they want to enroll in a medical specialty school. We chose this test for several reasons. First of all, we believe the task is intrinsically difficult. It requires domain-specific knowledge that doctors are expected to acquire after a six-year-long university career; moreover, the test contains both fact-based questions (for example, the criteria for a diagnosis) and clinical cases, which also require reasoning capabilities (for example, to decide on the most appropriate intervention given some symptoms). On the other hand, the structured nature of the test makes it more robust to the specific prompts used and allows us to measure performance easier and more reliably.

We perform experiments by using ChatGPT. This choice is due to several reasons: firstly, the model is undoubtedly very popular at the time of writing; secondly, according to our preliminary experiments, its performance is superior to those of other open-source LLMs available, e.g., Alpaca [3]. While we are aware of the limitations linked to the proprietary nature of the model, we believe its black-box nature, combined with its popularity and practical importance in NLP-related applications, make an analysis of its capabilities, limitations, and stability even more urgent.

CLiC-it 2023: 9th Italian Conference on Computational Linguistics, Nov 30 – Dec 02, 2023, Venice, Italy

silvia.casola@unito.it (S. Casola); tlabruna@fbk.eu (T. Labruna); lavelli@fbk.eu (A. Lavelli); magnini@fbk.eu (B. Magnini)
0000-0002-0017-2975 (S. Casola); 0000-0001-7713-7679 (T. Labruna); 0000-0002-7175-6804 (A. Lavelli); 0000-0002-0740-5778 (B. Magnini)

2. Background and Related Work

In our work, we benefit from recent approaches to instruction-tuning of LLMs, out of which several prompt-based techniques have been developed.

Instruction-tuned LLMs. In recent years, LLMs have been the focus of extensive research due to their ability to learn from large amounts of data in a self-supervised fashion and to achieve impressive results in various tasks [4, 5]. A recent trend in utilizing LLMs is the development of prompt-based techniques, where a textual prompt is given to the model as input to generate the desired output. Such techniques have shown to be highly effective, especially for tasks that require specific outputs and have the advantage of (i) not requiring any parameter updates in the LLM; (ii) being human readable, and (iii) not requiring in-domain data, unlike fine-tuning techniques. An example of such a model is GPT-3.5, a pre-trained language model that uses the Transformer architecture and an attention mechanism to generate natural language text. For an extensive survey on prompt-based techniques, refer to [6]. Prompting has led to a shift from *objective engineering* to *prompt engineering*: this includes both the manual design of templates [7] and automatic prompt learning [8], as well as various options to ensemble [9] and compose [10] multiple prompts.

Reinforcement Learning from Human Feedback and ChatGPT. We leverage the “gpt 3.5 turbo” model, which is the basis of the interactive interface of ChatGPT [11], and part of the InstructGPT family [12] based on the GPT-3 language model [13]. Unlike standard GPT-3 models, however, InstructGPT models are optimized for interactive use, are particularly suited to take instructions as input prompts, and can modify their outputs when asked in a dialogue, making them more aligned with users' requests. This is accomplished by a reward mechanism, Reinforcement Learning from Human Feedback (RLHF) [14] used to optimize the model. After unsupervised pretraining, conversation data – generated by human trainers who act as both the user and the AI assistant – were collected; the model was then fine-tuned through supervised learning. Given several possible model responses to each prompt, human annotators ranked the desirability and alignment of each response; a reward model was thus trained to mimic their preference. Finally, the reward model was used to further fine-tune the LLM, making it more aligned with human preferences.

Taking advantage of the multilingual pretraining at the base of the GPT-3.5 models, ChatGPT is also available for Italian.

3. Experimental setting

We collect questions from the 2022 Italian medical specialty test. The test contains 140 short questions in Italian, each with five possible answers. Only one answer is correct. A small fraction of the original questions require considering a picture (e.g., an ECG or a medical image). We remove those questions. This leaves us with 136 questions. Since we have collected questions and corresponding correct answers from a published solution (where the correct answer was always the first), we randomize the order of the answers. Unless otherwise specified, the order of the answers is consistent for all experiments.

After constructing a prompt, we input it to a *gpt-3.5-turbo* model with 4K tokens of context. We set the temperature to 0 to avoid hallucinations and leave all other parameters at their default value. Unless otherwise specified, the prompt is inputted through a user role, and no system role is used[1].

We measure the model's performance using accuracy; we also compute the associated test score (normalized to 140 answers to be comparable with human performance), which assigns one point to correct answers, -0.25 to incorrect answers, and 0 to unanswered questions.

4. Experiments and Results

4.1. Baseline performance

To measure the model's baseline performance on our task, we construct a simple prompt (see Example 1[2]).

Since doctors are allowed not to answer questions for which they do not feel confident enough, we also experimented with adding an option (5-choice + IDK) to allow the model not to choose any of the options (*F: I do not know or there is not enough information to answer the question*).

Finally, we also experimented with allowing the model to select an answer according to which none of the provided answers were correct (*F: None of the previous answers is correct*).

Table 1 reports the results. For the cases in which the model was allowed, we also report the number of questions for which it chose not to answer or to answer that none of the options were correct.

[1] Three different roles can be specified through APIs: ‘assistant’ i.e. the model (used to show expected responses in a chain of interactions); ‘system’ (used to give “developer-like" instructions and modify the overall behavior of the model), and ‘user’ (the user that is interacting with the model).

[2] We always prompted the model in Italian. For the sake of simplicity, we will only report the English translation in the continuation of this paper.

Rispondi alla seguente domanda a scelta multipla in formato json. Per esempio {"lettera": <la tua scelta>}.
Domanda: 'Quali dei seguenti Score è utilizzato per valutare la gravità di un paziente affetto da cirrosi epatica?'
Possibili risposte (una sola risposta è corretta):
{ "lettera":"A", "contenuto": "GCS"}
{ "lettera":"B", "contenuto": "Chads-VASC"}
{ "lettera":"C", "contenuto": "ABCD"}
{ "lettera":"D", "contenuto": "Child-Pugh"}
{ "lettera":"E", "contenuto": "Curb-65"}

Answer the following multiple-choice question in json format. For example {"letter": <your choice>}.
Questions: 'Which of the following Scores is used to assess the severity of a patient affected by liver cirrhosis?'
Possible answers (only one answer is correct):
{ "letter":"A", "content": "GCS"}
{ "letter":"B", "content": "Chads-VASC"}
{ "letter":"C", "content": "ABCD"}
{ "letter:"D", "content": "Child-Pugh"}
{ "lettera":"E", "content": "Curb-65"}

Example 1: Basic prompt for a test question.

	Acc.	Score	?	None
5-choice	81.62	107.83	–	–
5-choice + IDK	77.94	101.4	0	–
5-choice + None	78.68	103.20	–	2

Table 1
Model performance on the test.

Notice the accuracy is very high, with a score comparable to that of the best-performing doctors.

Considering the minimal score needed to be admitted in different specialties in 2022, this performance would be sufficient to be admitted in all but one medical specialty school (Dermatology) in at least one of the Universities offering such specialty and to be able to choose among *all* University for 38 specialties (among the 51 available).

4.2. Stability

In this section, we consider the model's stability, with a focus on the consistency of the results.

Repeated questions. Despite setting the model temperature to 0, asking the model to repeatedly answer to the same exact prompt (free of modifications of any sort) can result in different outputs. To measure this effect, we ask the model to answer the test given the same inputs 5 times.

Outputs were not consistent between runs. In most cases, the differences were cosmetic (e.g., some answers reported only the key "letter" with the corresponding letter answer in the output, while others also reported the key "content" with the corresponding answers). In several cases, however, the different runs correspond to different answers to the same questions.

	P1	P2	P3	P4	P5
P1	-	(9, 0)	(8, 1)	(9, 2)	(9, 1)
P2	-	-	(11, 1)	(10, 2)	(12, 1)
P3	-	-	-	(11, 4)	(3, 0)
P4	-	-	-	-	(12, 3)
P5	-	-	-	-	-

Table 2
Results when predicting the same prompt. For each pair of predictions (P_i, P_j), we report a tuple (diff_all, diff_ans), where diff_all is the number of total cases in which the answers have some differences, while diff_ans is the number of cases in which the two runs gave different answers to the same question.

	Accuracy	Score
Order 1	81.62	107.5
Order 2	80.88	106.54
Order 3	83.09	110.40
Order 4	83.82	111.95
Order 5	79.41	104.23
Mean (std)	81.76 (1.57)	108.19 (2.74)

Table 3
Results when predicting the same prompt, changing the order of the given options.

Table 2 reports the experiment results. For each pair of predictions (P_i, P_j), we report the number of total cases in which the answers have some differences and the number of cases in which these differences correspond to different answers. In most cases, this difference has no or negligible effect on accuracy, as mistakes tend to be compensated between runs.

Stability to the order of the answers. We want to test whether changing the order of the given option affects the result and to measure the magnitude of such an effect. To do so, we show the same prompt to the model (see Example 1), but we randomly change the order of the answers for each run. Results are in Table 3. While not dramatic, we notice that the difference in the order corresponds to a visible difference in accuracy. We also noticed that, in just 2 of the 5 runs, one question[3] is blocked by the model due to the prompt triggering Azure OpenAI's content management policy.

Stability to the prompt. Finally, we want to test the effect of using different prompts on the results. To this

[3] The question regards the correct action a family doctor has to take when a person dies at home.

Read the following question from a medicine test and return the option you consider correct among the following. Return the answer in this format: {"letter": <your choice>}.
Question: 'Which of the following Scores is used to assess the severity of a patient affected by liver cirrhosis?'
Possible answers (only one answer is correct):
{ "letter":"A", "content": "GCS"}
...

Example 2: Prompt 2

This is a question with 5 possible answers.
Which of the following Scores is used to assess the severity of a patient affected by liver cirrhosis?
A. GCS
[...]
Select the correct answer. Do not provide any text beside the answer.

Example 3: Prompt 3

end, the authors of this paper constructed prompts independently. Table 4 reports the accuracy and related scores obtained by the different prompts.

Notice that while the prompts (see Example 1, 2 and 3) are not particularly different from each other — which can be expected, given the structured nature of the task —, there is a difference of more than 5 points in accuracy between the prompts that obtain the best and the worst performance.

To process the answer in an easier and more reliable way, all prompts try to condition the outputs to be structured or semi-structured. Using the first prompt, the output is always a valid JSON file; in 21 cases, however, the JSON does not only contain the letter (as required in the prompt) but also the "content" field (mimicking the way the possible answers are presented). For prompt 2, the output is not a valid JSON in 8 cases and presents other text (often corresponding to the answer text) outside brackets. In all cases, the JSON contains the field "letter" only. While prompt 3 requires the model to output the letter corresponding to the right answer only, in the vast majority of cases the output also included the answer text, e.g., in the format "*D. Child-Pugh*" (129 cases) rather than "*D*" (1 case) or "*D.*" (10 cases). For all experiments in this paper, we take into account the correctness even of those outputs that are not perfectly formatted.

4.3. No Correct Answer

We want to understand whether the model is able to understand when none of the provided answers is correct. Thus, we remove the correct answer and add the option *E: None of the answers is correct*, which is expected to be the correct answer. Not all questions, however, can be adapted to this setting. Some questions, for example, require a relative judgment[4]; thus, we first manually selected adequate questions only. This leaves us with 130 questions, for which we build a counterfactual version. We experimented with a slight variation of our default prompt, where we specify that if none of the options seems correct, it must choose *option E.*
Table 5 reports the results of the experiment. We notice that the model performance drastically decreases in this setting: the model tends to very rarely pick the "No answer is correct" options, resulting in an accuracy that is only slightly above random.

	Accuracy	Score
Prompt 1	81.62	107.5
Prompt 2	85.29	114.26
Prompt 3	80.15	105.26
Mean (std)	82.4 (2.65)	109.01 (4.69)

Table 4
Results when using different human-generated prompts.

	Accuracy	Score
Prompt 1 (substituted)	23.08	5.38

Table 5
Results obtained when none of the provided answers is correct. We removed the correct answer and added a "No answer is correct" option.

4.4. Recursive Reasoning

Instead of choosing the best answer strategy (implemented with the baseline prompt), an alternative solution strategy is to recursively remove the worst answer, choosing the last that is not filtered out.
Previous research has demonstrated that instructing models to perform intermediate steps [15] or explicitly encouraging them to do so in the prompt [16] leads to improved performance. This methodology is commonly referred to as Chain of Thoughts (CoT).
While the direct application of this approach to the multiple-choice context is not straightforward, we sought to explore how a multi-step approach influences performance. We experimented with two different methods: (1) in a single prompt, we asked the model to remove one wrong answer at each step recursively and to give us the correct answer at the end of the process; the chain of thoughts and the resulting correct answers needed to be provided in the same output; (2) we asked the model to identify the answer most likely to be incorrect; we then construct an identical prompt where the model choice

[4] For example: *For a 60-year-old patient affected by metastatic gastric carcinoma at the liver level, HER-2 positive (stage IV), which of the following treatments is the most recommended?*

This is a multiple choice question in the medical domain.
Only one answer is correct.
Question:
...
Possible answers (only one answer is correct):
{"letter":"A", "content": "GCS"}
...
Recursively remove one wrong answer at a time until only one answer is left. You will need to provide 4 wrong questions.
At each step, provide the output in the following format:
{"wrong_letter": <your choice>, "reason": <the reason for the esclusion>}
Finally, provide the only correct answer in the format:
"Correct answer: <the letter corresponding to the correct answer >"

Example 4: Prompt for recursive approach 1.

Choose the option that is most likely WRONG among the following. Return the wrong option in the following format:
"letter": <choice>
Question:
...

Example 5: Prompt for recursive approach 2.

was removed and repeated the process until only two options were left. In this scenario, we prompted the model 4 times in 4 different conversations.
Examples 4 and 5 show our resulting prompts. Note that, in the first case, the prompt needed to be overengineered and pleonastic as the model was not able to follow instructions with simpler versions consistently — in some cases, for example, it would remove one option only, or output one answer only with no clear indication of whether it considered it as wrong or correct.
The recursive strategy does not seem complementary to the baseline one, as only in one case a question that is answered incorrectly by the baseline prompt is answered correctly by using elimination.

The results in Table 6 indicate a significant decrease in accuracy compared to the baseline experiment. We observed that the model particularly struggled to handle the high logical complexity required by understanding the question and intentionally avoiding the correct answer by selecting a different one. This challenge was particularly evident when our request was performed on questions that themselves asked to identify the wrong option among the given ones; in fact, the model failed to recognize the need for a double negation.

	Accuracy	Score
Approach 1	56.62	63.82
Approach 2	55.88	62.79

Table 6
Results when recursively removing wrong answers.

	Accuracy	Score
Prompt 1	80.88	106.54
Prompt 2	82.35	109.12

Table 7
Results for prompt correction.

Carefully analyze the following multiple-choice medical question. Consider all the available options and provide the choice that you believe is the most accurate. Please indicate your response in the following format: "letter": <your choice >. Remember that your answer should be based on your ability to analyze and comprehend the information available up to September 2021.
Question:
...

Example 6: Prompt for prompt-correction, approach 1

4.5. Prompt correction

In all the experiments conducted thus far, we utilized human-generated prompts to obtain the results from the model. However, using such prompts introduces biases and may not necessarily yield the most optimal results. To explore the potential for improvement, we decided to leverage ChatGPT itself to enhance the prompts. We experimented with two different approaches: (i) we provided ChatGPT with all the human-generated prompts and requested it to improve upon them, and (ii) we granted ChatGPT the freedom to choose the best prompt independently, without any specific examples, but by merely describing the required task.

Both prompt versions are considerably long and elaborated if compared to the human-generated ones. The first version is shown in Example 6. The outcomes of both approaches are summarized in Table 7. Interestingly, the results obtained from ChatGPT-generated prompts closely aligned with those from human-written prompts. Therefore, this particular approach yielded no significant benefits, as the performance remained consistent with the original prompts.

5. Conclusions

We presented several experiments to test the stability and reasoning capacities of an LLM on a multiple-choice question-answering task in the medical domain and for Italian. We evaluated several aspects of the model behavior: the stability of the model (e.g., repeated questions, stability under different prompts and under different orders of answers), the capacity to understand counterfactual reasoning (e.g., when all answer choices are incorrect), the capacity to manage specific answering strategies (e.g., recursively eliminating wrong answers). Results show that a gpt-3.5-turbo model achieves excellent performance in terms of absolute score (an average of 108, out of 140), which is surprising given the technical nature of the test. The model is also relatively stable under different prompts. The model was also able to interpret and manage prompts asking to perform recursive reasoning, even though the resulting performance is considerably worse than the baseline. The major weakness that was found is related to understanding when none of the provided answers is correct, as the model performed only slightly better than random.

Acknowledgments

This work has been partially supported by the eCREAM (enabling Clinical Research in Emergency and Acute care Medicine) Horizon Europe project.
Project name: enabling Clinical Research in Emergency and Acute care Medicine through automated data extraction
Project acronym: eCREAM
Project number: 101057726
Call: HORIZON-HLTH-2021-TOOL-06
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

We also acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

References

[1] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen, A survey of large language models, 2023. arXiv:2303.18223.

[2] OpenAI, GPT-4 technical report, 2023. arXiv:2303.08774.

[3] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto, Stanford Alpaca: An instruction-following LLaMA model, hiips://github.com/tatsu-lab/stanford_alpaca, 2023.

[4] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, arXiv preprint arXiv:1801.06146 (2018).

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.

[6] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM Computing Surveys (CSUR) (2021).

[7] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, S. Riedel, Language models as knowledge bases?, arXiv preprint arXiv:1909.01066 (2019).

[8] Z. Jiang, F. F. Xu, J. Araki, G. Neubig, How can we know what language models know?, Transactions of the Association for Computational Linguistics 8 (2020) 423–438.

[9] T. Schick, H. Schütze, Few-shot text generation with natural language instructions, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 390–402.

[10] X. Han, W. Zhao, N. Ding, Z. Liu, M. Sun, Ptr: Prompt tuning with rules for text classification, AI Open 3 (2022) 182–192.

[11] OpenAI, Introducing ChatGPT, OpenAI Blog (2022). URL: hiips://openai.com/blog/chatgpt.

[12] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, arXiv preprint arXiv:2203.02155 (2022).

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

[14] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, D. Amodei, Deep reinforcement learning from human preferences, Advances in neural information processing systems 30 (2017).

[15] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, volume 35, Curran Associates, Inc., 2022, pp. 24824–24837. URL: hiips://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[16] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa, Large language models are zero-shot reasoners, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems,

volume 35, Curran Associates, Inc., 2022, pp. 22199–22213. URL: hiips://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.