Instruction-tuning LLMs for Annotating Answer Helpfulness

Zhuge Gao
4215451
University of Tübingen
{name.surname}@student.uni-tuebingen.de

Abstract

Could ChatGPT 3.5 and LLma 2 be used as annotator? If they can understand the principle of helpfulness, follow annotation guideline and generate output annotation in the correct format. It could act as a copilot for annotators, saving major time and headache for the annotator. No more copy paste and fact checking! In this work, a small set of instruction following data with the correct annotation is used to instruction tuning for LLMs. ChatGPT 3.5 would work as a timely reliable assistant, but Llama 2 is not ideal as it is slow to learning, response and not as reliable.

1 Introduction

Large Language Models(LLMs) have gained significant attention in various research fields due to their capability to handle a broad range of applications. It is important to know if LLMs could be trusted as assistants. The principles of helpfulness, honesty, and harmlessness (HHH) (Askell et al., 2021) is one of the evaluation frame work used to assess the performance of language models. Harmlessness is usually the focus in previous works, while helpfulness and honesty are often ignored (Rauh et al., 2022).

The preceding work that inspired this research decided to focus on the principle of helpfulness and wanted to investigate "Do LLMs react to different types of instructions as well as Humans?". We annotate part of the QA `databricks-dolly-15k` dataset and some LLM generated responses using the same questions with a set of components that represents the principle of helpfulness. The components are based on Grice's maxims (Grice, 1975). investigated the components would further investigate helpfulness. The previous analysis concludes for the small scope of our selected data, LLMs are as helpful as human at simple instructions, sometime slightly better, but creative tasks does pose some difficulty. More research are needed to better evaluate LLM's ability and usage in data annotation.

In this work, the focus would be using instruction tuning and see if LLM could use the annotation guideline, a few example JSON data and corresponding csv annotations as input, then learn to annotate data points based on the helpfulness components by output csv format annotations like the human annotator. It is intended to observe if LLM could perform in a more desired way with instruction tuning without the fine-tuning, which is time and resource consuming.

2 Data and Annotation Guideline

In the preceding work, three categories were selected from the databricks-dolly-15k corpus: closed QA, summarization and brainstorming. The principle of helpfulness is divided into a set of components based on Grice's maxims: structure, informativity, on-topic, and correctness. 1308 human generated answers and 150 answers generated by the Llama-13b-chat model are annotated manually for the helpfulness components on a nominal scale from 0 to 4. Irrelevant component would get a `0` (= not relevant). If the component is relevant, then the answer's performance that componentwould be evaluated on a nominal scale ranging from: 1 (= very poor), 2 (= rather poor), 3 (= rather good), 4 (= very good). The annotation results indicate that 4 is most frequent score, which aligns with the data being gold standard training for instruction following tasks. A general annotation guideline was created after group discussions, but each individual does have their own preferences and ways of annotating which shown in the inter-annotator agreement of 0.37.

3 Method: The Process of Instruction Tuning for Annotation Tasks

Model used in this are ChatGPT 3.5(OpenAPI, 2024) and LLama 2 13b chat(Touvron et al., 2023), both free and widely used LLMs worth exploring and evaluating their ability to annotating data with few-shot prompt instruction tuning method. There is no need for fine-tuning, and no funds for extra GPU time. ChaptGPT 3.5 is accessed using their free web interface. Llama 2 is running with Ollama [1] in a local terminal, also in a interactive chat like manner, which is the only way to have a Llama 2 running at its full capacity for the author.

Since only the author's annotations are used as training examples for few-show prompting, the LLMs used are first given a more specific version of the annotation guidelines written by the author. It is a draft and will not be presented in the article. It is a prompt that introduce the tasks and the components needed for annotations to the LLMs.

Learning is initiated with gold label data points. The data points are annotated by the same annotator. The prompts specifically states that JSON format data will be provided, and the csv format annotation is gold standard and instruct the model to learn from that. The specific instruction and gold standard annotations are requirements for instruction tuning.

Following a few-shot prompting method(Li et al., 2006), after around 5 in-context learning examples, the model will be asked to use what it has learned to annotate a new unseen JSON data point and output the annotation in csv format. If a desired format output is generated, the model will get feedback if the new annotations is not the same as the gold label. This method will allow us to see if LLM can improve its performance with such iterative interaction, and learn to adapt to the annotation pattern from the same annotator.

4 Result and Evaluation

Evaluating the Performance of ChatGPT 3.5 and LLama 2 as data annotators on the generated data annotations by generating a progressive inter annotator agreement comparison for the two models and the human annotator to see if it gets better with adaptive learning. For ChatGPT 3.5, there are about 40 data annotations, comparing to Llama 2's 24, which has to terminate due to deteriorating performance and inability to generate a valid annotation in desired format.

The highest inter annotator agreement is between ChatGPT 3.5 and Llama 2 with 0.39. The human annotator and ChatGPT 3.5 has a 0.35 agreement score which is about the same as the agreement between human annotators. Llama 2 and the human annotator has the lowest agreement of 0.27, which is consistent with Llama 2's unstatisfying performance.

4.1 Comparative Analysis of ChatGPT 3.5 and LLama 2

First thing the user will notice is the difference is the time latency. ChatGPT response time is seconds after the input, while Llama running locally will take about 30 seconds to start response generation. The time difference over time would accumulate to a significantly large for the amount of annotation work, which are usually hundreds of entries. Waiting for the response would interrupt the annotation process and breaks up the work flow.

Performance will be discussed in the speed of adaptation and stability. Llama 2 does not learn as fast as GPT3.5 to instruction and feedback correction. They both respond they understand the task after the instruction including the annotation guideline which explains the tasks and what they have to do. ChatGPT adapts very quickly, will only output the csv format annotations after the first data annotation example prompt. Llama always include additional sentences at the beginning and end of the response, like:

```
Certainly! Here's the annotated version
of the text based on the human
annotator's labels and formatting:
723,3,4,4,4
I hope this helps! Please let me know
if there's anything else I can
assist you with.
```

The sentences generally express that Llama 2 understand the task and would like to be helpful. However, by including such sentences it is actually being not helpful because the instruction ends with "Please only output the annotation in csv format: ". This shows that Llama 2 does not understand or follow the instruction.

Even with explicit prompts, Llama 2 still includes additional information in the answer, unlike ChatGPT 3.5 which learned to only output csv format annotations after the first prompt.

[1] https://github.com/ollama/ollama

ChatGPT 3.5 is more stable in terms of sticking to the desired csv output format, even though it is implied in the prompt. Llama received specific instructions to output csv format many times, but still includes the additional sentences before and after the csv annotations, which is also quite stable. Llama is slow adapting. If explicit instruction on output format is not given every time, right at the end before getting a response. Llama will not include a csv annotation. ChatGPT 3.5 could perform the annotation task without additional instruction. If only the JSON data is given, it will correctly output a csv format annotaiton. It does tend to "lost memory" and deviate from the csv format output after multiple of non-instructive input. It will remember or switch back after one properly structured prompt with instructions.

Both model are not very consistent when you ask it to output the annotation in the required format for the same JSON data. Sometimes different annotation scores will be given.

There are some common behavior that shows the challenges of using LLMs as data annotators, which will be discussed in the next section.

5 Challenges and Solutions in Tuning LLMs for Specific Annotation Functions

In the annotation task, there are some challenges and difficulties encountered when trying to use LLMs to annotate answer for helpfulness.

Both models would insisting on its own annotations and refuse to change even though correction are given and instructed to modify annotation and learn from it.

Both model also output invalid annotation. Annotation score may not on the requested nominal scale from 0 to 4. ChatGPT had output incomplete annotations with less than the required 4 components. But both model would corrected its behavior after being told that it's not a valid annotation.

Both models will diverge and give different kind of response, sometimes answering the question directly instead of annotating the answer, saying random things, or fail to repeat the gold label annotation correctly.

Repeating the instruction may make it worse. Usually after one to three iterations of repeating the instruction, it will learn and adapt, or "give up the fight" and repeat the correct annotation. ChatGPT 3.5 is less stubborn than Llama 2. However, they both have refuse to cooperate and do not respond to instruction. User will have to move on, or go back to previous interaction where models was following instruction and start there.

5.1 Llama 2 issues and observation

There are some issues specific to the Llama 2 model caused by longer prompts and system default prompt used in the Llama 2 13b chat model.

Longer prompts could cause issue with Llama 2 but not for GPT 3.5. Data point id 2621 is a lengthy outlier, including the instruction the prompt is 1259 words long. The Llama 2 response could not follow the desired csv format output with explicit prompt that works with other data points. It outputs the headers of the csv followed by part of the answer provided in the data.

Output would have additional tags like "[Inst[Inst[Inst" right after the annotation csv output. When questioned about the "[Inst]" at end of sequence string, the model respond it is a typo but correct the multiple [Inst] to just one complete [Inst]. Given separate instruction to remove it and told the model it is not necessary but the tag still remains. After further investigation, it likely caused by the default system prompt template for Llama 2 chat:

```
[INST] <<SYS>>{{ .System }}<</SYS>>
{{ .Prompt }} [/INST]
```

To improve the adaptation and performance of LLMs like Llama 2, explicit instruction and correction about the desired output format are necessary. Set the template to a single character and see if that improves the output. But it does not seems to have an effect on the output format.

After the lengthy prompt with data id 2621, the issue with Llama 2's output format with dangling tags from detault system prompt is gone. The reason unknown. The two events may not have a causal relationship, but the additional sentences before and after the csv annotation output disappeared after the second time model choose to answer the instruction question in JSON data by outputting part of the response from the data point(first time is for id 2621). This may caused by the explicit instruction to learn and annotate like the annotator which is repeated every prompt between these two times, but other data point annotation shows no such change. The [Inst] shows up again after multiple iterations.

Llama 2 repeated the annotation for the last training prompt 579 instead of annotating for id 1982 like asked. When the mistake is pointed out. It simply replaced 579 with 1982. When the model is given JSON data for 1982 and asked to annotate it the second time. It does not output csv format annotation, but rather give a part of the response from the 1982 data point, like how it reacted with prompt that is too long. This shows that the model does understand it's doing a instruction following and question answering kind of task. The point is the answer, but not always remember what it should do is to give scores on helpfulness of the answer. This again shows the problem of slow adaptation and being stubborn.

Finally, Llama 2 refuses to "learn" the correct annotation. It will not repeat the correct one, insisting on its own annotation or partially change it, not learning the training annotation correctly. For both models, if they make this kind of inaccurate mistakes. They make it in a row. Appears to be not cooperative like a frustrated person would do.

Data id 4312 is particularly difficult for Llama 2. It refuse to modify and repeat the correct annotation after being corrected more than three times. Finally changed to the correct after very specific instruction: " please modify your annotation to 0,4,4,0", just telling it to "modify the annotation" would not work. After this correction, and then ask it to annotate a new data point. Llama will repeat the training annotation. Repeat the last annotation when asked to annotate a new data point. Repeated four times and a correction to get the correct annotation for the correct id. The model does remember the correct annotation. The problem continues. It gave same annotation 3,1,1,0 four times in a row. Invalid annotation value shows up again as 3,1,1,6. After telling it is in valid because it's not within the range from 0-4, Llama changed the annotation to 4,4,4,6. This is again, invalid. This problem happened before but it was corrected after being told.

On the 25th data point to be annotated, Llama 2 refuse to cooperate and instruction would not work. The author decide to stop the experiment with Llama 2.

6 Conclusion

In this paper, ChatGPT 3.5 and LLma 2 are used as annotator for helpfulness in the answer. The hypothesis is to test if LLM could understand the task, understand the principle of helpfulness, follow instruction and consistently output annotation in the requested format. If it is successful, maybe LLM could act as a copilot for annotators, saving time and effort. The annotator average 4 minutes per data point, but majority of the time is going back and forth between the text editor and the browser and copy pasting sentences. ChatGPT 3.5 would work pretty good as a help for suggestion and fact checking. Llama 2 has a disadvantage to annotate data in a timely and efficient annotation, not ideal for a co-pilot, but would work for replacing annotator. Ideally LLM would be able to explain its decision and the human annotator only need to check if it holds up or make sense.

Instruction tuning is used with a set of human generated and annotated dataset. The data and gold labels are included with specific instruction to perform the annotation task. The model is learning in a few-shot prompting manner. However, Llama 2 does not work as a data annotator and had to stop the experiment early. It is slow adapting and stubborn to change. ChatGPT 3.5 is a much promising choice.

Possible reason for the 60 example in-context learning not working as shown in previous researches. In-context examples are shown to lead to better performance in language models for many types of tasks(Liang et al., 2023), especially the first example will increase performance significantly but subsequent examples marginal increase are very low, sometimes counter productive. It was also shown that maybe demonstration using example and gold labels may not affect LLM's in-context learning performance as much(Min et al., 2022). There are other aspects like the distribution of input text, label space and format sequence. These are possible improvement for further research.

Acknowledgments

I would like to thank my previous teammate and fellow annotators Giulio Posfortunati, Ching Nei Pun, Giulio Cusenza, Darja Jepifanova.

References

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. A general language assistant as a laboratory for alignment. *Preprint*, arXiv:2112.00861.

H. P. Grice. 1975. Logic and conversation. In Donald Davidson and Gilbert Harman, editors, *The Logic of Grammar*, pages 64–75.

Fei-Fei Li, Robert Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):594–611.

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 Ré, 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. 2023. Holistic evaluation of language models. *Preprint*, arXiv:2211.09110.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *Preprint*, arXiv:2202.12837.

Maribeth Rauh, John Mellor, Jonathan Uesato, Po-Sen Huang, Johannes Welbl, Laura Weidinger, Sumanth Dathathri, Amelia Glaese, Geoffrey Irving, Iason Gabriel, William Isaac, and Lisa Anne Hendricks. 2022. Characteristics of harmful text: Towards rigorous benchmarking of language models. *Preprint*, arXiv:2206.08325.

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. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

A Example Appendix

This is an appendix.