Can Instruction Tuning Enhance the Predictive Capability of ChatGPT for Classifying Political Messages?

Saklain Zaman, Chinmay Ashok Maganur, Sampada Regmi and
Jennifer Stromer-Galley
Syracuse University, Syracuse NY 13244, USA

Abstract. ChatGPT's generative capabilities enable researchers to translate a diverse range of problems into language modeling problems. Though classifying text messages into multiple categories can be achieved by supervised classifiers, the issue arises when there's a scarcity of large, labeled domain-specific dataset which can be referred as the ground truth. ChatGPT has already demonstrated its ability to address various downstream tasks including classification problems. In this article, we aimed at classifying the political messages from Facebook during the 2020 presidential election into five predefined categories: Advocacy, Issue, Attack, Image/Persona, and Call to Action by prompting ChatGPT. According to the literature, obtaining the best set of results from ChatGPT requires careful tuning of the instructions which are fed into ChatGPT, so we identified three checkpoints while tuning the instructions for ChatGPT. We addressed the shortage of gold standard dataset and utilized a small subset of 100 labeled instances to both guide classification and evaluate the performance of ChatGPT. We also transformed our multi-class classification task into a series of binary classification tasks to make a comparative analysis of ChatGPT's annotation performance compared to trained human annotators.

Keywords: Large Language Models, Annotation, Supervised Machine Learning.

1 Introduction

Large language models (LLM) can be used to solve various tasks using appropriate instruction tuning, which unlocks their enhanced capabilities. The recent advancement of transformer-based large language models like ChatGPT has shown immense potential in solving various tasks, such as question-answering, machine translation, knowledge and mathematical reasoning, code generation, etc. [1]

Instruction tuning refers to the strategy of fine-tuning pre-trained LLMs by providing a task description and a set of example instances formatted as natural language [2], and the process is highly similar to supervised fine tuning [3]. Instruction tuning enables the LLMs to exhibit superior potential in solving unseen tasks [2, 4, 5], even for multilingual contexts [6]. Designing appropriate prompts are necessary to obtain desirable outcomes from the LLMs for adopting downstream tasks [7, 8, 9, 10].

Our primary objective is to train predictive models on predetermined categories in political campaign texts from social media, such as Facebook. The conventional technique to approach this classification problem is to train supervised learning models on a large, annotated dataset. But doing so can be labor-intensive and time-consuming. Our research question that motivated this research project was to determine the efficacy of ChatGPT to develop training data for supervised learning models using appropriate prompting strategies or instruction tuning. Compared to supervised learning techniques, in theory this process only demands a well-formatted task instruction and a few labeled instances to guide ChatGPT.

We employed several instruction tuning strategies discussed in the literature [1], such as In-Context learning and Chain-of-Thought (CoT) to address this predictive task and adjusted our prompts to achieve better classification results. We observed that when ChatGPT was instructed to classify the texts into multiple classes at once, it got confused about the target labels, and generated new categories as the target class. Then, we transformed our multi-class classification task into a series of binary classification tasks to evaluate the comparative performances for the predictions.

The overall contributions of this work are the following:

- Our project addresses the issue of gold standard data scarcity when classifying social media messages in the political domain leveraging the public interface of ChatGPT.
- We identify the common errors encountered while performing straightforward classification tasks in a niche domain using the iterative tuning of the prompting techniques.
- Based on our experiments, we suggest the best prompting practices for ChatGPT for binary classification, which can be utilized by the researchers to replicate manual data annotation process to obtain quality synthetic labeled data for further supervised learning.

2 Related Research

ChatGPT and other LLMs offer substantial advances in a variety of contexts given its ability to generate natural-language text as well as programming code. The challenge is devising the right prompts to get the desired results. Research in this area is nascent, but a few studies have provided insights on effective techniques.

One technique is in-context learning (ICL). A pre-trained LLM is provided with the task description and a few demonstration examples, and the test instances are appended to form the prompt. The examples can help the LLMs to identify and perform a new task without explicitly updating the model weights (or, gradient) [1]. Fine-tuning prompt instructions is required to adapt to a new task using this in-context learning technique.

The selection of demonstration examples shows a large variance with in-context learning tasks [11]. Previous works have employed k-NN based approaches to sample the most semantically relevant queries as demonstration sets [12, 13]. While selecting the demonstration set, both diversity and relevance of the examples should be considered to form the prompts for in-context learning tasks [14]. Several studies have leveraged LLMs to create the demonstration set without human intervention utilizing the generative capability of LLMs [15, 16]. Due to the recency bias that LLMs exhibit, they generate responses that are closer to the end of the demonstrations [17]. The task examples should be organized in an order that puts the more relevant examples toward the end [18].

Chain-of-thought (CoT) prompting strategy can be utilized to improve performance to derive multi-step reasoning while solving complex reasoning tasks, such as symbolic [19], common-sense [20], or arithmetic [21]. CoT can be incorporated with ICL to boost performance in two notable settings: few-shot and zero shot. Few-shot CoT is a variant of ICL where the prompts follow the format of <input, CoT, output> instead of <input, output> format followed by ICL to augment the intermediate reasoning steps [1]. The performance of CoT can be further improved by listing diverse reasoning paths for a specific problem [22]. However, as these techniques rely on labeled datasets for CoT, their use cases in practice are restricted to an extent. To address this limitation, zero-shot CoT was introduced where the models are not provided with human-annotated demonstrations. When zero-shot CoT was first introduced, the LLM was prompted with "Let's think step by step" to induce the reasoning steps and finally prompted with "Therefore, the answer is" to obtain the final answer [23]. Auto-CoT [24] also utilizes zero-shot CoT where the LLMs are prompted to generate reasoning paths without manual intervention. Furthermore, the questions in the training dataset are divided into different clusters, and the questions which are closer to each cluster's centroid are chosen to well-represent the training data.

While these two approaches provide helpful insights, we aimed to explore the efficacy of tuning the instructions for ChatGPT to classify multiple categories at once as compared with the binary classification approach. We also wanted to understand if more complex and detailed instructions were more effective than simplified explanations of categories in the instruction tuning process.

3 Research Context

The main objective for our project was to ascertain the utility of LLMs to annotate social media messages and the text of political advertisements from U.S. presidential candidates for the purpose of creating training data for human-supervised machine-learning model development. Obtaining a high-quality labeled dataset demands adequately trained human annotators, but this often is a bottleneck for preparing a large gold standard dataset. It takes time to train annotators and to ensure that they sufficiently understand the rules that guide the classification task. Additionally, this process can be expensive, as training and annotation are often done by paid workers.

We relied on an existing project that has developed an annotation guide for classifying social media messages in the political domain. The guidebook includes categories that identify different types of campaign messages:

- attack - criticize an opponent, group, or institution on policy or character.
- advocacy - highlight positively some aspect of the candidate
- image/persona - characterize the candidate or the opponent on their character, background, or ability to lead
- issues - advance policy positions
- call-to-action - ask people to do something, such as give money, vote, or attend a campaign event

The data collected for the project is from U.S. presidential political candidates that competed in the primaries of the Republican and Democratic Parties in the 2020 presidential election. The corpus includes Facebook Posts from the official Facebook accounts of the candidates. The samples that we experimented with to test different prompt strategies were derived from gold-labeled training data created by trained annotators.

We chose to experiment with the public-access version of ChatGPT (version 3.5). It is the most sophisticated and usable model that is freely available. We also experimented initially with Google's BARD, but after several trials we found it unable to complete even the simplest tasks without substantial effort. The experimentation work was completed between July and August 2023. We note this because the LLM is continuously adapting based on the public's interactions with it, rendering replication challenging. Another important point to note is that we deliberately kept 'Chat History and Training' disabled in ChatGPT settings during the entire experimentation so that ChatGPT doesn't train itself on our testing data and we can utilize the same testing set for different phases of the experiment.

4 Experimentation

To determine the utility and accuracy of ChatGPT in an annotation task, we adopted the annotation guide used to train human annotators and the examples used in that guidebook. Examples for human training included cases that clearly exemplify the category as well as edge cases that help identify the boundary conditions of a category. Our process was iterative and exploratory, experimenting with various prompt strategies that would generate the fewest errors from ChatGPT while also providing the most accurate results.

We started our experimentation by instructing ChatGPT on the 5 defined categories (Advocacy, Issue, Image/Persona, Attack, Call to Action) using the detailed human annotation guide. The original annotation guide used to train the human annotators included a great deal of information about inclusion and exclusion criteria as well as demonstration examples, which in some cases had their own sub-category labels. After feeding the raw instructions from the original codebook, we asked ChatGPT to classify 100 randomly selected messages into the 5 overarching categories. While classifying multiple categories at once, we observed that ChatGPT frequently got confused about the categories, assigning sub-categories to the messages, and creating its own categories when unable to identify the target category.

> *Initial Instruction Prompt: I will give you definitions of categories, rules, and examples of categories. I want you to remember the rules and examples. You will use them to categorize messages from political candidates. Each category is not mutually exclusive. This means that each message can receive more than one of the categories that I give to you.*
>
> *Response from ChatGPT: I understand your instructions. Please provide me with the messages you'd like me to classify into the specified categories: Advocacy, Persona, Attack, Issue, and Call to Action. Once you provide the messages, I'll categorize them accordingly based on the rules and examples you've provided.*

To obtain better annotation results, we transformed our multi-class classification task into a series of binary classification tasks to determine the target categories for the messages. We designed the instructions for ChatGPT to identify one class at a time. For example, while working with the *advocacy* category, we provided instructions and examples only for this category and asked ChatGPT to categorize the message into "Advocacy" and "Not_Advocacy" depending on the inclusion and exclusion criteria.

One noteworthy problem we encountered with this approach pertained to the "Issue" category. The messages in the original annotation guide have sub-categories such as *social*, *economic*, *governance*, etc. This issue guided us to fine-tune the instructions for ChatGPT in the next phase of our experimentation where we removed the sub-categories information in the instruction tuning materials to make the prompts less complex and more straightforward.

As ChatGPT gets overwhelmed with the nuances and details in the original codebook, we then experimented with a simpler training guide. We devised a simpler set of prompting instructions with basic rules that applied to all messages regardless of category, and then a simple definition of each category, a brief elaboration of that category, and 4 example messages for that category. The basic rules gave guidance on handling messages with URLs or hashtags. The examples chosen were meant to provide illustrations of the category and the range of message styles.

This instruction-tuning phase follows in-context learning (ICL) where the prompts are formatted as task descriptions and a set of examples for each task. We performed both the multi-class classification and the binary classification focusing on each category separately using this technique and contrasted the annotation performance for the set of 100 randomly selected messages against gold label human annotation.

Finally, we further tuned our instructions for ChatGPT to follow the chain-of-thought (CoT) prompting strategy which is an extension of the ICL technique. We included explanations for each of the examples, which represent the reasoning for the examples to be included in the relevant category. One additional complexity we encountered is with the original *image* category. ChatGPT tended to classify every message as *image*. So, we changed the label to *persona* and refined the instructions.

We evaluated ChatGPT's annotation performance for these three approaches for both multi-class and binary classification for the five categories and contrasted it human annotation. We used Cohen's Kappa to measure the annotation agreement between human and ChatGPT to evaluate inter-coder reliability. We also utilized metrics including F1-score, precision, recall and accuracy to capture the annotation quality, such as understanding the trade-off between false positive and false negative cases. The combination of the metrics provides a comprehensive assessment of both annotation quality and agreement. From Table 6, we can determine that the final instruction guide for binary classification yields better annotation results compared to previous phases based on these metrics.

Table 1. Multi-class classification using raw codebook.

Category	F1-Score	Accuracy	Recall	Precision	Cohen's Kappa
Advocacy	0.705	0.65	0.656	0.763	0.28
Attack	0.705	0.8	0.6	0.857	0.561
Persona	0.48	0.74	0.387	0.631	0.32
Issue	0.8	0.74	0.825	0.77	0.43
Call to Action	0.769	0.88	0.714	0.833	0.689

Table 2. Binary classification using raw codebook.

Category	F1-Score	Accuracy	Recall	Precision	Cohen's Kappa
Advocacy	0.780	0.64	1	0.64	0
Attack	0.815	0.86	0.775	0.861	0.703
Persona	0.473	0.31	1	0.31	0
Issue	0.779	0.74	0.730	0.836	0.466
Call to Action	0.766	0.86	0.821	0.718	0.667

Table 3. Multi-class classification using task descriptions and examples (in-context learning)

Category	F1-Score	Accuracy	Recall	Precision	Cohen's Kappa
Advocacy	0.666	0.6	0.625	0.714	0.172
Attack	0.526	0.73	0.375	0.882	0.378
Persona	0.521	0.67	0.580	0.473	0.274
Issue	0.824	0.77	0.857	0.794	0.492
Call to Action	0.716	0.85	0.633	0.826	0.617

Table 4. Binary classification using task descriptions and examples (in-context learning)

Category	F1-Score	Accuracy	Recall	Precision	Cohen's Kappa
Advocacy	0.784	0.650	0.984	0.652	0.052
Attack	0.805	0.857	0.763	0.852	0.693
Persona	0.495	0.377	1	0.329	0.066
Issue	0.805	0.704	0.967	0.689	0.255
Call to Action	0.707	0.806	0.851	0.605	0.569

Table 5. Multi-class classification with task descriptions and examples with explanations (Chain-of-Thought)

Category	F1-Score	Accuracy	Recall	Precision	Cohen's Kappa
Advocacy	0.734	0.66	1	0.64	0.262
Attack	0.686	0.8	0.55	0.916	0.554
Persona	0.447	0.58	1	0.377	0.127
Issue	0.771	0.71	0.777	0.765	0.374
Call to Action	0.711	0.87	0.571	0.941	0.634

Table 6. Binary classification with task descriptions and examples with explanations (Chain-of-Thought)

Category	F1-Score	Accuracy	Recall	Precision	Cohen's Kappa
Advocacy	0.816	0.74	1	0.64	0.38
Attack	0.724	0.81	0.625	0.862	0.585
Persona	0.549	0.59	1	0.31	0.238
Issue	0.870	0.83	0.904	0.838	0.625
Call to Action	0.836	0.91	0.821	0.851	0.774

5 Discussion and Lesson Learned

The codebook used for training human annotators evolved over multiple iterations to include lengthy and detailed explanations and a variety of examples to help the annotators understand inclusion and exclusion criteria. They requested these details to help them fully understand the category and deal with the complexity and nuance of human communication. ChatGPT, rather than performing better with more detailed explanations, were more accurate with simpler definitions and examples. It seemed from our experiments that the more complex instruction tuning materials confused the LLM rather than clarifying the inclusion criteria. Relatedly, ChatGPT did not handle the task instructions that required it to annotate for a broad category, like "issue", while being given examples of sub-categories, such as "military" or "crime and safety". It conceptually seemed unable to hold the overarching category in memory once sub-categories were introduced in the instructions. We were hoping that ChatGPT could handle the task of annotating multiple classes at once, as that would be a faster approach than annotating each binary category. Unfortunately, ChatGPT performed worse when instructed to annotate multiple categories.

6 Conclusion

Our project aimed at leveraging the advanced capabilities of ChatGPT to annotate political messages into predefined categories with the goal of obtaining high quality labeled data. This explorative study focused on the iterative tuning of the instructions for ChatGPT to extract superior annotation results. Based on our experiments, ChatGPT illustrated better annotation quality and agreement with trained human annotators with simple and straightforward set of instructions along with examples with explanation, and when ChatGPT was instructed to classify one category at a time. Our study can serve as a guide to address the issue of data scarcity for supervised learning and how this issue can be addressed with proper tuning of the instructions for ChatGPT, which calls for more investigation.

References

1. Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z. and Du, Y.: A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023)
2. Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M. and Le, Q.V.: Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021)
3. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A. and Schulman, J.: Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, *35*, 27730-27744 (2022)
4. Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T.L., Raja, A. and Dey, M.: Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207* (2021)
5. Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S. and Webson, A.: Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022).
6. Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Scao, T.L., Bari, M.S., Shen, S., Yong, Z.X., Schoelkopf, H. and Tang, X.: Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786* (2022)
7. Tang, T., Li, J., Zhao, W. X., & Wen, J. R.: Mvp: Multi-task supervised pre-training for natural language generation. *arXiv preprint arXiv:2206.12131* (2022)
8. Liu, X., He, P., Chen, W., & Gao, J.: Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504* (2019)
9. Aghajanyan, A., Gupta, A., Shrivastava, A., Chen, X., Zettlemoyer, L., & Gupta, S.: Muppet: Massive multi-task representations with pre-finetuning. *arXiv preprint arXiv:2101.11038* (2021)
10. Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Arunkumar, A., Ashok, A., Dhanasekaran, A.S., Naik, A., Stap, D. and Pathak, E.: Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705* (2022)
11. Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., & Chen, W.: What Makes Good In-Context Examples for GPT-$3 $?. *arXiv preprint arXiv:2101.06804* (2021)
12. Levy, I., Bogin, B., & Berant, J.: Diverse demonstrations improve in-context compositional generalization. *arXiv preprint arXiv:2212.06800* (2022)
13. Su, H., Kasai, J., Wu, C.H., Shi, W., Wang, T., Xin, J., Zhang, R., Ostendorf, M., Zettlemoyer, L., Smith, N.A. and Yu, T.: Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975* (2022)
14. Ye, X., Iyer, S., Celikyilmaz, A., Stoyanov, V., Durrett, G., & Pasunuru, R.: Complementary explanations for effective in-context learning. *arXiv preprint arXiv:2211.13892* (2022)
15. Gilardi, F., Alizadeh, M., & Kubli, M.: Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056* (2023)
16. Kim, H. J., Cho, H., Kim, J., Kim, T., Yoo, K. M., & Lee, S. G.: Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator. *arXiv preprint arXiv:2206.08082* (2022)
17. Zhao, Z., Wallace, E., Feng, S., Klein, D., & Singh, S.: Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning* (pp. 12697-12706). PMLR (2021)

18. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, *55*(9), 1-35 (2023)
19. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V. and Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, *35*, pp.24824-24837 (2022)
20. Talmor, A., Herzig, J., Lourie, N., & Berant, J.: Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937* (2018)
21. Miao, S. Y., Liang, C. C., & Su, K. Y.: A diverse corpus for evaluating and developing English math word problem solvers. *arXiv preprint arXiv:2106.15772* (2021)
22. Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J. G., & Chen, W.: On the advance of making language models better reasoners. *arXiv preprint arXiv:2206.02336* (2022)
23. Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y.: Large language models are zero-shot reasoners. *Advances in neural information processing systems*, *35*, 22199-22213 (2022)
24. Zhang, Z., Zhang, A., Li, M., & Smola, A.: Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493* (2022)