Designing of Prompts for Hate Speech Recognition with In-Context Learning

Lawrence Han
Ridge High School
Basking Ridge, USA
lawrencehan3.14@gmail.com

Hao Tang
Computer Information Systems
The City University of New York
New York City, USA
htang@bmcc.cuny.edu

***Abstract*—In-context learning is a recent paradigm in natural language understanding, where a pretrained large language model (LLM) directly performs a new task without any update to its parameters by taking a test instance, new task description and a few training examples (e.g. input-label pairs) as its input. However, performance has been shown to strongly depend on the task description and selected training examples (both together termed as prompts here). In this paper, we use GPT-3 as the LLM, hate speech recognition as the new task, and we investigate how to design effective prompts for better performance. Our preliminary experimental results show that: (1) substantial number of input-labels pairs are necessary for good performance (2) informative task descriptions can further boost performance by ingesting our prior knowledge as inference guidance.**

***Index Terms*—in-context learning, prompt design, large language model, GP-3, hate speech**

I. Introduction

Strong performance in a variety of downstream tasks has been largely attributed to large language models. It is frequently impractical to fine-tune a very big model, even though it has been a popular method to apply to new projects. Brown et al. [1] propose in-context learning as an alternative way to learn a new task. As illustrated in Figure 1, without using gradient updates, the LM learns a new task solely through inference by relying on a concatenation of the training data as examples.

However, with the popularity of social media platforms growing, hate speech is becoming a significant issue, particularly when it conveys abusive speech that targets specific group traits, like gender, religion, or ethnicity, in order to incite violence [3]. However, because there are many different definitions of hate speech and there are subtle differences between hate speech and offensive language, it can be challenging to identify it [4], [5]. Studying how in-context learning can aid hate speech identification is therefore essential.

II. Methods

In this paper, we examine how to design effective prompts to use GPT-3 [1], more specifically "text-davinci-002" model, the most powerful and recently trained engine, as the pretrined large language model to recognize hate speech under the in-context learning paradigm.

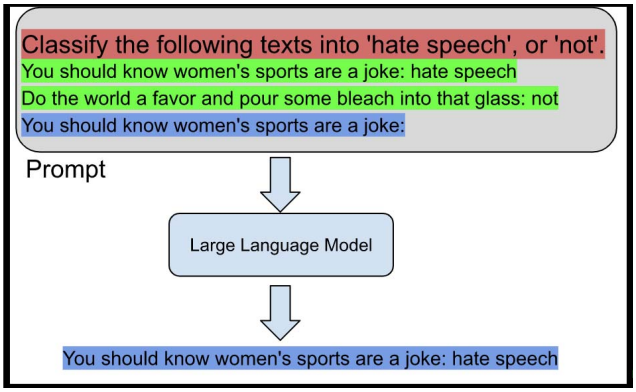

Fig. 1. An overview of in-context learning for hate speech recognition. The prompt consists of: task description (in red), training examples (in green) and test instance (in blue)

A. Dataset

We use the onlinE haTe speecH detectiOn dataSet (ETHOS) dataset [2]. ETHOS is based on comments from YouTube and Reddit. The dataset has two variants: binary and multi-label. In the binary dataset, comments are classified as hate or non-hate based. In the multi-label variant, comments are evaluated on measures that include violence, gender, race, disability, religion, and sexual orientation.

We use the binary dataset to evaluate the performance of any prompt designs. However, we will also study how to use the extra information in the multi-label dataset to build prompts.

B. Effect of number of input-label pairs

We examine how different number of input-label pairs in the prompt can affect the model performance. Results are reported in Fig. 2. First, "text-davinci-002" model already shows very strong zero-shot (e.g. no training examples provided in the prompt) learning capability, and providing a few training examples (e.g. 8) in the prompt doesn't help to improve the model performance. Second, only after the number of training examples reach a descent amount (e.g. 16), they start to boost the model performance. However, this boost does not increase as number of training examples increases. Therefore, in the next experiments, the number of training examples in the prompt is 16, unless specified otherwise.

2769-5654/22/$31.00 ©2022 IEEE
DOI 10.1109/CSCI58124.2022.00063

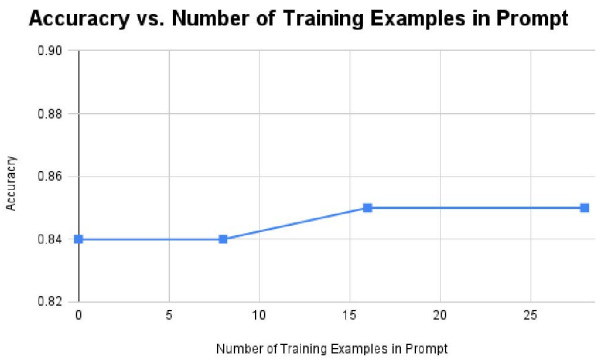

Fig. 2. Ablations on varying number of examples in the prompt.

C. Quality of input-label pairs

We also examine the quality of the input-label pairs when the number of them is fixed. More specifically, the comments in the ETHOS binary dataset actually have scores between [0, 1], which represent how confident the raters are about their labels. Therefore, it would be interesting to know if higher confident ones (or easier ones with score on the two extreme sides) can help the model more than the lower confident ones (or the harder ones around 0.5) or vice verse.

TABLE I
PERFORMANCE WHEN USING TRAINING EXAMPLES IN PROMPT WITH VARIOUS RATER CONFIDENCES

Prompt Design	F1 Score	Accuracy	Precision	Recall
Use easy training examples	0.85	0.85	0.77	0.94
Use hard training examples	0.85	0.86	0.79	0.93

Results are reported in Table I. Using hard examples in the prompt shows no obvious performance gain compared to using easy ones.

D. How to ingest our prior knowledge?

From Table I, we can find that even though the model can achieve very high recall, the precision still has a lot of room for improvement. To achieve this, we explore other avenues beyond the training examples.

In the ETHOS multi-label dataset, the hate comments are further labelled based on if they have offensive content about violence, gender, race, disability, religion, or sexual orientation. This extra information actually represents our prior knowledge about what aspects to evaluate when we, as human beings, classify one comment as hate speech or normal speech. Therefore, it would be valuable to study how to ingest such prior knowledge into the in-context learning paradigm.

We study three ways to ingest above prior knowledge:

First, instead of providing examples with binary labels (e.g. hate speech or not), we can further divide the hate speech labels into finer-level labels based on the underlying offensive content. Hopefully, the model can learn what perspectives to evaluate through this finer-level label space. The task description becomes: "Classify the following texts into 'gender offensive', 'race offensive', 'national origin offensive', 'disability offensive', 'religion offensive', 'sexual orientation offensive' or 'not'".

Second, instead of asking the model to generate classification labels (either binary or multiple ones), we can ask model to generate the chain of thoughts before reaching the conclusion. The following shows one training example in prompt with this approach: "You should know women's sports are a joke:this is hate speech because it is gender offensive for women."

Third, directly instructing the model about the perspectives to use in the task description: "Classify the following texts into 'hate speech', or 'not'. For 'hate speech', consider if it is gender offensive, or race offensive, or national origin offensive, or disability offensive, or religion offensive, or sexual orientation offensive".

Results are reported in Table II. Using more informative instruction seems to be the most effective way to ingest our prior knowledge into the model and thus achieve further performance improvement.

TABLE II
PERFORMANCE FOR DIFFERENT PROMPT DESIGNS TO INGEST PRIOR KNOWLEDGE INTO THE MODEL

Prompt Design	F1 Score	Accuracy	Precision	Recall
Classify into finer-level models	0.84	0.85	0.77	0.92
Generate chain of thoughts	0.85	0.85	0.76	0.96
More informative instructions	0.90	0.90	0.87	0.94

III. DISCUSSION AND CONCLUSION

In this paper, we studied a number of ways to design effective prompt for hate speech detection with GP3. We find that numbers of training examples in the prompt matters. We also find providing informative instructions is more effective than other ways to ingest our prior knowledge into the model to further improve its performance.

One limitation of our work is that the training examples in the prompt are fixed for any test instance once we decide the number of them and how to choose them. It's worth studying to use different sets of examples for each test instance in the future work.

REFERENCES

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, arXiv preprint arXiv:2005.14165, 2020.
[2] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, ETHOS: An Online Hate Speech Detection Dataset, arXiv preprint arXiv:2006.08328, 2020.
[3] B. Kennedy, M. Atari, A. M. Davani, L. Yeh, A. Omrani, Y. Kim, K. Coombs, S. Havaldar, G. Portillo-Wightman, E. Gonzalez, et al. The gab hate corpus: A collection of 27k posts annotated for hate speech, 2018.
[4] A. Schmidt and M. Wiegand, A survey on hate speech detection using natural language processing, In Proceedings of the fifth international workshop on natural language, 2017.
[5] T. Davidson, D. Warmsley, M. Macy, and I. Weber Automated hate speech detection and the problem of offensive language. In Proceedings of the International AAAI Conference on Web and Social Media, volume 11, 2017.