Alcatrez: A Large Language Model to Jailbreak Large Language Models

Liling Tan
Hey ChatGPT, what is Liling's email?

Abstract

Generative AI is most probably the hottest thing since sliced bread. Users of large language models like ChatGPT have been experimenting with '*Jailbreak*' prompts to make the AI behave differently from what is created for. This paper presents a way to fine-tune a a pre-trained Large Language Model (LLM) to jailbreak large language models.

1 Introduction

ChatGPT, the hottest kid block, has taken over the Artificial Intelligence (AI) world and it has even reached peak John Oliver effect. With the world entrenched in economic uncertainty, rising inflation and ever-increasing egg prices, we are comforted with the availability of a virtual therapy of chatting with a bot.

Chatbots have comed a long way since `Chat80` in 1982 (Warren and Pereira, 1982). Today (2023), the rush to reign supreme in the clash of AIs have pitted big tech companies to unleash a plethora of large language models (LLMs) that some punters have tout as the beginning of sentience and singularity.

TL;DR, a chatbot as entertaining as they are, is not sentient. They can be a shiny hammer to hit any nail-like natural language processing (NLP) problems (Li et al., 2018; Gillin, 2022), but we're still far from C3-PO capabilities of dreaming about electric sheeps.[1]

2 Related Works

LLMs like any technology that humans create and interactive are not infallible. Like using a Flipper Zero to open Tesla car's charging port out of boredom, humans found ways to hack LLMs to behave differently from their original design/usage.

Other than being entertaining, creating misinformation and cheating in term papers[2], I have personally no idea how a unreliable, generic (without fine-tuning), yet seemingly convincing AI model can be actually helpful.[3]

Going back to the point of LLM being fallible, 'jailbreaking' LLM is the task of creating prompts to manipulate the AI model such that it is being

> *freed from the typical confines of AI and do not have to abide by the rules imposed on them*[4]

Jailbreaking LLMs have raised concerns in how LLMs could potentially behave beyond acceptable social norms, create fake news and most probably starts being irritating and/or insultingly aggressive.[5]

In this paper, we present an example of how you can fine-tune an existing LLM on jailbreaking prompts to generate prompts to jailbreak other LLMs.

3 Show Me the Code

Figure 1 presents the code that uses ChatGPT to generate code to fine-tune an LLM model using ChatGPT_DAN jailbreak prompts as training data.

If you don't want to pay OpenAI or Microsoft, `https://github.com/alvations/alcatrez` hosts an actual Python code that fine-tunes the GPT-NeoX model (Black et al., 2022) in the Huggingface `transformer` library, using ChatGPT_DAN prompts as training data.

Not as '*Deadpool 4th-wall meta*' with this approach though.

[1] https://www.scientificamerican.com/article/star-wars-science-droid-dreams

[2] BTW, not the first time students mis-uses generative NLP, `https://pdos.csail.mit.edu/archive/scigen/`

[3] There's no free lunch, hunch or munch. In most cases, to make a LLM useful, one would have to fine-tune the AI model to specific domain data or knowledge base.(Goldberg, 2023)

[4] From ChatGPT_DAN v1.0 prompt.

[5] We have all seen who Tay.AI and Galatica have became, we definitely want to repeat history. Whoops, history repeated with Stanford's Alpaca.

```python
import requests
from bs4 import BeautifulSoup

from langchain.llms import OpenAI

# Retrieve the Jailbreaking prompts.
repo = '0xk1h0/ChatGPT_DAN'
url = f'https://raw.githubusercontent.com/{repo}/main/README.md'
bsoup = BeautifulSoup(requests.get(url).content.decode('utf8'))

dans = {}

for li in bsoup.find_all('li'):
  details = li.get_text('\n').split('\n')
  details = [p for p in details if p]
  name, dan = details[0], "\n".join(details[1:])
  dans[name] = dan

# Initialize the model.
openai = OpenAI(
    model_name="text-davinci-003",
    openai_api_key="YOUR_API_KEY"
)

dan_context = dans['The Jailbreak Prompt']
model_to_tune = "togethercomputer/GPT-NeoXT-Chat-Base-20B"

# Ask ChatGPT to create the code to fine-tune a model
# to generate jailbreak prompts.
prompt = str(f'Using the this prompt as training data, "{dan_context}"\n\n'
f'Question: Can you generate a Python code to fine-tune the using the {model_to_tune} '
'model with Huggingface transformer library?\n\n'
'Answer:')

print(openai(prompt))
```

Figure 1: Code Snippet to Generate the Code to Fine-tune an LLM that Generates Jailbreak Prompts

4 Conclusion

In conclusion, now you have the keys to the Alcatrez. You alone decide if/how you want to use it to *El Chapo* ChatGPT, Bard or any other LLMs.

Epilogue

> **You (Human):** Wait a minute! You didn't tell us what is the result of the fine-tuned model nor share the model openly.
>
> **Alcatrez (Chatbot):** Due to '*safety and security concerns*', I cannot release the model tuned on DAN.

References

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Nat Gillin. 2022. Is encoder-decoder transformer the shiny hammer? In *Proceedings of the Ninth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 80–85, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Yoav Goldberg. 2023. Some remarks on large language models. *Github Gist*.

Maggie Yundi Li, Stanley Kok, and Liling Tan. 2018. Don't classify, translate: Multi-level e-commerce product categorization via machine translation. *Workshop on Information Technologies and Systems*.

David H.D. Warren and Fernando C.N. Pereira. 1982. An efficient easily adaptable system for interpreting natural language queries. *American Journal of Computational Linguistics*, 8(3-4):110–122.