German-Trainer GPT

Saurav Banerjee

Due Date: December 23, 2023

Data 340

Abstract

Large Language Models and Natural Language Processing have boomed in the past year, rising from the release of OpenAI's Chat-GPT and the accessibility of the AI to the public. Making the API for GPT-2 and other Large Language Models available, as well as having Transformers available to the public has allowed for the usage of these LLM after being put through transformers to do more selective tasks.

The goal of this project was to be able to adjust a Large Language Model to be able to teach German to users. The model would function as a Chat-GPT style German teacher, which would mainly be used as a chatbot, but would be able to correct users on what they were saying wrong so that they could learn.

Introduction

Learning languages is always interesting and learning each language is different in it's own ways. Having learned how to speak 3 fluently (English, Bengali, and Hindi), I've picked up different ways to learn different languages, and learning the latter two just by speaking them with other people and watching German, I've thought that it would be pretty easy to learn German. However I soon realized that this was not the case. I was only learning through the textbook at school, and didn't really have any native speakers that I could learn from, so I realized I was at a bit of a learning gap. This inspired me to try to build German-Trainer GPT in an effort to reduce this gap in learning that is caused by learning proper structures, but never learning to use them conversationally. This would serve as a bridge between speaking and learning proper German and would allow for better retention of the German Language.

Methodology/Dataset

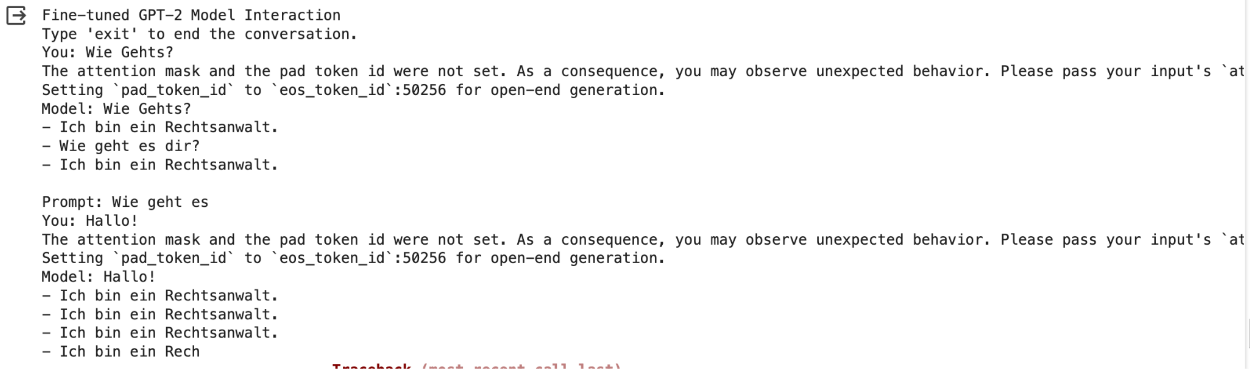
To accomplish this task I decided to apply Hugging-Face Transformers on an existing LLM to train the AI. I used GPT-2 since I thought it would be the best for AI to be able to communicate in English to the user, while also having the ability to talk in German.

The training set needed two things: sample dialogue of people speaking in German and sample output for the LLM. I made a combination of both of these to make the training set. For examples of dialogue, I used subtitles from German Movies/TV Shows, as well as movie subtitles translated into German. I also used data from Microsoft's DialoGPT translated into German to help generate more data, as DialoGPT's data is originally just for a chatbot. I also included sample prompts and responses in a separate .txt file to use as well.

I trained my model using Google Colab, and trained the GPT-2 model on a larger dataset of just dialogue, and then fine-tuned the dataset using the dataset of sample output and input and tested the model using normal German Conversational phrases.

Results

The model did not perform well Looking at some example output below, we can see that the model did not have coherent output and didn't complete the objective, mostly repeating what I said and then giving some random German lines from the training set.



```
Fine-tuned GPT-2 Model Interaction
Type 'exit' to end the conversation.
You: Wie Gehts?
The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `at
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Model: Wie Gehts?
- Ich bin ein Rechtsanwalt.
- Wie geht es dir?
- Ich bin ein Rechtsanwalt.

Prompt: Wie geht es
You: Hallo!
The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `at
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Model: Hallo!
- Ich bin ein Rechtsanwalt.
- Ich bin ein Rechtsanwalt.
- Ich bin ein Rechtsanwalt.
- Ich bin ein Rech
                                 Traceback (most recent call last)
```

Originally however, I first used a transformer on the model German-GPT and tried to tweak that. I had better performance,where the model did act like a chatbot, but the issue was that the model didn't correct the user's grammar, instead the output just talked back in German, which the original German-GPT already did.

Discussion

It's clear that there was a gap between expected results, and actual results. While the results of the LLM were not well, we can still derive some conclusions from all this. Firstly, there could be a variety of things which caused the issue, ranging from a small training set, to not training the model long enough, to simply using the wrong pre-trained model. For example, Chat-GPT's training set of data is a lot bigger, and this leads to better diversity of answers and better analysis of answers. My dataset that I trained the model on was mostly dialogue, and may not have had enough data, or even diversity of the data. This might be why random sample input and output were being output when I tried inputting conversation into the model.

Additionally, there might be issues with the way the model was trained in Google Colab. There's a possibility that the model was either overtrained or undertrained on the data, which led to faulty output. Not only that, but the fact that GPT-2 is trained in English and not German could go a long way to worsening results, since this meant that the model would see way too less data. This was my first time training on a pre-trained model in Colab using a transformer, so there was a lot of room for error, since I wasn't fully sure what to do to get the optimal model.

However, with more time, I would definitely change my procedure. I'd add much more data to the training data, especially text from German newspapers and books. This would let the model have more German to train off of, and maybe this would get rid of my issues with the model just repeating things. I’d also run the model for more time on the data for fine-tuning, but I’d keep the GPT-2 architecture the same and see if I could make the model work. I’d keep the GPT-2 pre-trained model since English is a Germanic language, and with enough input data, the model should not have any issues since both languages have some similar structure.

It’s also important to note that these results don’t mean that we won’t be able to train a model to do this task. German-GPT exists, which just functions as GPT in German, and talks to humans decently well. Additionally Chat-GPT is able to learn how to speak in German and do bits in other languages as well. I think maybe the original architecture probably affected the results, and with time and more open-source models, better results will happen.

I still learned a lot from the process of trying to make the model work and trying different combinations of code and data to make the model work. Trying one configuration, testing and waiting to see if it worked, and then having to try something else taught me a lot about the process of developing this model and helped me gain an understanding of the difficulty and complexity associated with making LLMs. Ultimately failing also taught me more about allocating time and resources better, and learning how to to manage projects better and temper expectations

Finally looking forward, while I did mention that English is a Germanic language, and that means that they share similarities, I still think that trying this process could be useful for languages not originating in Europe and maybe in Asia or Africa. This could be really useful in teaching people new languages without having to travel. I see LLM like these as having a part in the future of language learning sooner rather than later, and just more work needs to be done in the field.

Conclusion

While it is true that the model did not work in the way we expected, that doesn’t mean that we didn’t learn anything. We learned that training such models need larger datasets, and better system architecture to help train this data. Hopefully this can be implemented in future iterations of German-Trainer GPT, and hopefully this can be applied to learning other non Germanic Languages as well.

References

GPT-2

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.

Hugging Face Transformers

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., & Rush, A. M. (2019). Transformers: State-of-the-Art Natural Language Processing. Hugging Face, Brooklyn, USA

German-GPT

Hugging Face. (2023). German GPT-2 model. Retrieved from https://huggingface.co/dbmdz/german-gpt2

Appendix:

Google Drive with Training Data:

https://drive.google.com/drive/folders/1TTaZkbAFLvoJLF1ktRkjTb9632P_Pi5e?usp=sharing

Google Colab Files for Training the Model:

https://colab.research.google.com/drive/1Y9ry8H7M5WyL4mSYyyvJhMheiW9dR6Xq#scrollTo=zQMbsal5sdxq