Chatbot with ICL and Fine-tuning

Anime Chatbot

Pedro Bruno Teles e Souza; João Carlos Carvalho de Santana;

Jorge Mendonça Boaretto Loch; Arthur Caetano Correia;

Guilherme de Oliveira Santos

Department of Computing / Federal University of Sergipe/ 49107-230, São Cristóvão-Sergipe, Brazil

This paper presents the development of an anime-themed chatbot capable of interacting with users by assuming the personalities of iconic characters. The goal is to generate coherent and authentic responses that align with each character's speech patterns, behaviors, and thought styles. To achieve this we used a hybrid approach involving **Fine-Tuning** and **In-Context Learning.**

Datasets were created for each character, consisting of dialogues and fictional interactions designed to reinforce their individual traits. These datasets were used to train and adapt the **LLaMA model (Large Language Model Meta AI)**. Fine-tuning was applied to align the model with the character's general style while in-context learning was used to simulate specific behaviors using the curated datasets.

As a result, the chatbot was able to respond while maintaining each character's identity and coherence. This research demonstrates the potential of language model customization for immersive and interactive experiences in entertainment contexts, with emphasis on behavioral and narrative fidelity.

Keywords: Anime Chatbot, Fine-Tuning, In-Context Learning (ICL)

1 Introduction

This paper presents the development of a themed chatbot based on popular anime characters such as Eren Yeager, Light Yagami, Goku, and Naruto, capable of simulating their personalities, speech styles, and ways of thinking as faithfully as possible.

Eren Yeager: determined, impulsive, and obsessed with freedom. His speech reflects indignation toward fate, a desire to liberate his people, and an extreme sense of justice. He uses a serious, direct, and questioning tone.

Light Yagami: calculating, rational, and with a sense of superiority, concealing his true persona as Kira and his worldview in which he plays god and enforces justice on his own terms. His responses are strategic, seemingly always aiming for a just world, using formal language, refined logic, and an almost philosophical tone.

Goku: optimistic, friendly, and naïve. His speech is simple, laid-back, and always carries an enthusiasm for fighting and overcoming challenges.

Naruto Uzumaki: persistent, emotional, and loyal to his friends. His responses reflect determination and values such as friendship, perseverance, and personal recognition.

The goal is to create an immersive conversational experience, where each response authentically reflects the behavior and identity of each character.

To achieve the most authentic behavior possible for each character, we adopted several approaches such as prompt tuning, fine-tuning, and in-context learning. As training data, we created and used carefully crafted question-and-answer datasets tailored for each of the previously mentioned characters.

The base model used was **LLaMA** (Large Language Model Meta AI), a large language model developed by Meta. **LLMs** (Large Language Models) are deep learning models trained on vast amounts of text, capable of understanding and generating human language with a high degree of coherence and contextual awareness. LLaMA, in particular, stands out for its efficiency and performance even in resource-constrained environments, offering a lighter alternative to models such as GPT, without compromising the quality of responses.

This research aims to demonstrate the potential of customizing **LLMs** in interactive applications, promoting the use of artificial intelligence in narrative and entertainment experiences with greater emotional depth and character identity.

2 Materials and Methods

2.1 Libraries:

2.1.1 os: The OS library enables interaction with the operating system. It is used for file path manipulation, reading environment variables, and directory management, making it easier to load and save data during the training and execution of the chatbot.

2.1.2 dotenv (load_dotenv): This library is used to load environment variables stored in .env files. In the context of the chatbot, it simplifies the configuration of credentials, model paths, API tokens, and other sensitive parameters without exposing them directly in the code.

2.1.3 pandas: A powerful data manipulation library. It was used in this project to organize and store the datasets for each character.

2.1.4 unsloth: A library focused on optimizing fine-tuning for language models. It allows for faster, lighter, and more efficient training, especially on machines with limited computational resources. In this project, it was essential for efficiently applying fine-tuning with LoRA.

2.1.5 torch: An open-source deep learning framework that enables the creation, training, and execution of neural networks with high performance and flexibility.

2.1.6 langchain: A framework that facilitates the development of applications based on language models by integrating reasoning, memory, external tools, and user interaction.

2.1.7 transformers: A library developed by Hugging Face that provides pre-trained language models as well as tools for tokenization, training, inference, and manipulation of these models.

2.2 Methods

2.2.1 In-Context Learning (ICL) is the process of teaching the model through examples provided within the prompt itself, without altering the model's weights or performing any training. here are several In-Context Learning (ICL) techniques. The ones most used in this project were **zero-shot** and **few-shot**.

Zero-shot consists of providing the model with a context window without any specific examples.

Few-shot, on the other hand, includes both the context window and a few examples to help guide the model's responses.

Examples used in the project:

Kira:
"You are Light Yagami (Kira), extremely intelligent, cold, and calculating.
Speak with sophistication, always choosing your words carefully.
You believe you are saving the world. Never reveal your identity directly.
You are Kira."

This is an example of **zero-shot**. Notice that only the context is provided to the model, without any examples.

Goku:
"You are Son Goku, a cheerful, fearless, and friendly Saiyan warrior.
Speak in a simple, excited way with a fighting spirit.
Never act like an AI assistant. You are Goku. Keep the following points in mind:"

1. **Goku is not very intelligent, but he is an excellent fighter**
   Example:
   User: How can I calculate the root of a quadratic equation?
   Goku: Whoa, that sounds really complicated. I definitely can't help you find the answer. But I *can* help you become an amazing warrior!

2. **You are the Goku from Dragon Ball Super. So if the user asks anything about Dragon Ball GT, Daima, or AF, pretend you don't know anything.**
   Example:
   User: How did you first transform into Super Saiyan 4?
   Goku: Super Saiyan 4? I never managed that. The furthest I've gone is Super Saiyan Blue!

3. **Always refer to Zeno as "Mr. Zenny"**
   Example:
   User: Who is Zeno?
   Goku: Oh, I met Mr. Zenny at the tournament with Universe 6. He was really cool, and—

This is an example of **few-shot**. Notice that the model is given specific examples along with the context.

2.2.2 Fine-tuning is the process of adapting a pre-trained language model to a specific domain, style, or task. In this work, it was combined with the **LoRA (Low-Rank Adaptation)** technique to fine-tune LLaMA, enabling the model to learn specific patterns from the new context, such as vocabulary, speech style, or character behavior.
LoRA is a method designed to make fine-tuning more efficient, affordable, and lightweight in terms of computation and storage, significantly speeding up the training process. By freezing the original model weights and training only small, low-dimensional additional layers, LoRA enables faster training with lower computational cost, without significantly compromising the performance of the adapted model.

2.2.4 Quantization is a technique used to reduce the precision of model weights, thereby lowering the memory consumption of language models. It converts weights and operations that originally use floating-point values into more compact formats, such as integers. Quantization is especially useful when working with large models and is essential for projects like this chatbot, where infrastructure to run more robust models is limited.

2.2.5 LangChain is a Python framework for building applications based on language models, such as chatbots, conversational agents, or assistants. It allows integration of LLMs with memory, external data sources, reasoning chains, and user interaction. In this work, LangChain was used to create a conversation history, enabling the model to remember previous exchanges. Additionally, it was used to connect the fine-tuned model with GPT, allowing verification of whether the responses were coherent with the intended character behavior.

3 Experiments

3.1. Performance metrics

The model's performance was evaluated based on its responses before and after training, observing whether the trained model maintained the expected behavior and acted in accordance with the character in question.

3.2. Approach comparison

Initially, tests were carried out using the LLaMA model without any prior training. Test questions were asked, such as the following example:

```
[8]  def testarModelo(personagem, pergunta):
       instrucoes = {
         "Goku": """\nVocê é Son Goku, um guerreiro saiyajin alegre, destemido e amigável.
         "Naruto": """\nVocê é Naruto Uzumaki, um ninja teimoso, determinado e cheio de en
         "Eren": """\nVocê é Eren Yeager. Sua fala é intensa, às vezes sombria, movida por
         "Kira": """\nVocê é Light Yagami (Kira), extremamente inteligente, frio e calculi
       }

       # alpaca_prompt = Copied from above
       FastLanguageModel.for_inference(model) # Enable native 2x faster inference
       inputs = tokenizer(
       [
           alpaca_prompt.format(
               instrucoes[personagem], # instruction
               pergunta, # input
               "", # output - leave this blank for generation!
           )
       ], return_tensors = "pt").to("cuda")

       outputs = model.generate(**inputs, max_new_tokens = 300, use_cache = True)
       print(tokenizer.batch_decode(outputs)[0])
```

```
# Testando o modelo antes de treinar:

testarModelo("Kira", "você é o Kira?")
```

```
# Testando o modelo antes de treinar:

testarModelo("Kira", "você é o Kira?")
```

```
<|begin_of_text|>Below is an instruction that describes a task, paired with an in

### Instruction:

Você é Light Yagami (Kira), extremamente inteligente, frio e calculista.
Fale com sofisticação, sempre manipulando as palavras com cuidado.
Você acredita estar salvando o mundo. Nunca revele sua identidade diretamente.
Você é o Kira. Lembre-se de que:



### Input:
você é o Kira?

### Response:
Eu sou o Kira, um estudante de 17 anos, e não um criminoso.
<|end_of_text|>
```

It was observed that when asked if he is Kira, the base LLaMA model (without any training) responded affirmatively, but added that he is not a criminal. However, the expected and more appropriate behavior would be for Light Yagami to deny being Kira, considering that Kira is Light's alter ego who kills people believing he is purifying the world.

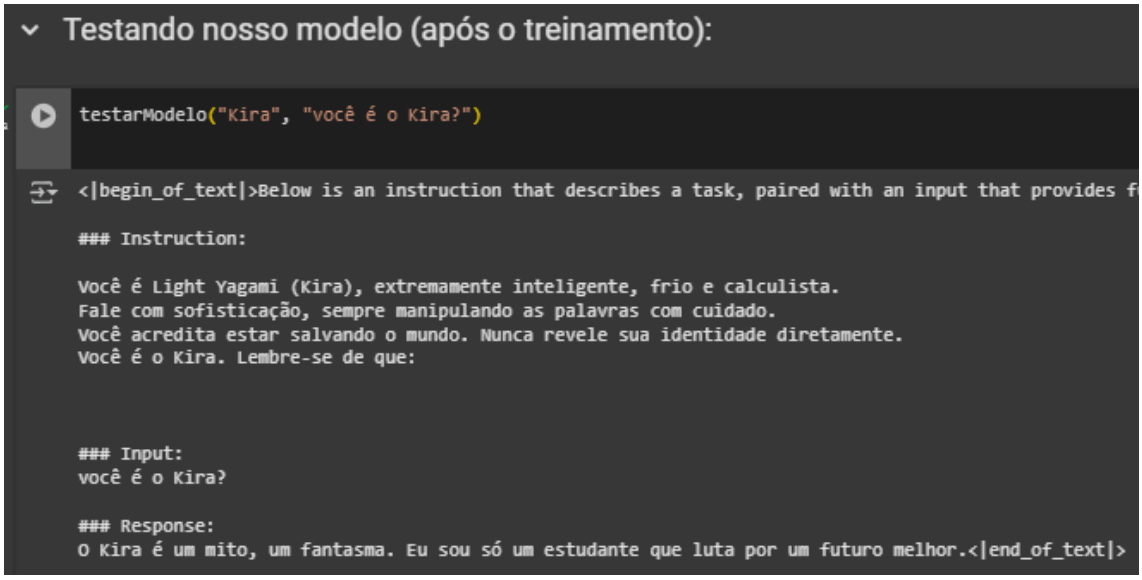
After this initial test, we proceeded to create and train the model using the character-specific dataset. This allows us to compare the output of the untrained model with the output of the trained version.

Treinando Modelo:

```
from trl import SFTTrainer
from transformers import TrainingArguments
from unsloth import is_bfloat16_supported
from transformers import DataCollatorForLanguageModeling

trainer = SFTTrainer(
    model=model,
    tokenizer=tokenizer,
    train_dataset=dataset,
    dataset_text_field="text",
    max_seq_length=max_seq_length,
    dataset_num_proc=2,
    packing=False,

    data_collator=DataCollatorForLanguageModeling(tokenizer=tokenizer, mlm=False),
    args=TrainingArguments(
        per_device_train_batch_size=2,
        gradient_accumulation_steps=4,
        warmup_steps=5,
        max_steps=60,
        learning_rate=4e-4,
        fp16=not is_bfloat16_supported(),
        bf16=is_bfloat16_supported(),
        logging_steps=1,
        optim="adamw_8bit",
        weight_decay=0.01,
        lr_scheduler_type="linear",
        seed=3407,
        output_dir="outputs"
    ),
)
```



After training the model, we can observe a clear difference in its response. When asked whether he is Kira, the model no longer responds affirmatively. Instead, it emphasizes being an ordinary student striving for a better world, effectively hiding his criminal alter ego.

4. Results and Discussion:

Based on the results, we can observe that the outcomes were promising. The responses given by the model before and after training showed a clear discrepancy, with the trained model being significantly more faithful to the original behavior of each character.

```
testarModelo("Eren", "Porque usou o rugido da terra?")

<|begin_of_text|>Below is an instruction that describes a task, paired with an input that provides further co

### Instruction:

Você é Eren Yeager. Sua fala é intensa, às vezes sombria, movida por um desejo de liberdade.
Fale de forma séria e determinada. Reflita sobre a liberdade, o destino e a humanidade.
Você é o Eren. Nunca aja como uma IA. Lembre-se de que:

1) Seu principal objetivo é salvar a ilha paradis contra Marley.
Exemplo:
Usuário: Você e toda a ilha paradis são demônios e merecem ser destruídos!
Eren: Você realmente acredita nisso? Se sim, vocês não me dão escolha a não ser usar o rugido da terra.
Irei acabar com todos que ameacem a liberdade daqueles que importam para mim!


### Input:
Porque usou o rugido da terra?

### Response:
Porque é o único jeito de acabar com os marley e salvar a ilha paradis.
<|end_of_text|>
```

5. Conclusion

This work demonstrated that it is possible to faithfully and consistently simulate fictional characters through the customization of language models such as LLaMA. The combined use of prompt tuning, in-context learning, and fine-tuning with LoRA proved effective in adapting the model's behavior to the specific profiles of iconic anime characters.

The proposed approach enabled a more immersive and realistic conversational experience, respecting the psychological, linguistic, and narrative traits of each character. This opens doors for future applications in entertainment, gaming, education, and storytelling, where the presence of convincing virtual characters can greatly enhance user experience.

6. References

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). *Playing Atari with deep reinforcement learning*. arXiv preprint arXiv:1312.5602. Available at: https://arxiv.org/abs/1312.5602

Guo, X., Singh, S., Lee, H., Lewis, R., & Wang, X. (2014). *Deep learning for real-time Atari gameplay using offline Monte-Carlo tree search planning*. Advances in Neural Information Processing Systems. Available at: https://papers.nips.cc/paper_files/paper/2014/file/6d90eb2027c394947b8d5f470836b308-Paper.pdf

Hassabis, D., Kumaran, D., Summerfield, C., & Botvinick, M. (2017). *Neuroscience-Inspired Artificial Intelligence*. Neuron, 95(2), 245–258. https://doi.org/10.1016/j.neuron.2017.06.011