NYU JARVIS: Exploring Roleplaying Capabilities of Llama 2

Seeger Zou, Orion Yang, Justin Hwang

Case 1

Hey JARVIS, can you tell me what Tony likes?

Base Llama 2	Mistral	Ours
Of course, I'd be happy to help! Tony Stark, also known as Iron Man, is a complex and multifaceted character with many interests and preferences. Here are some things he might likes: 1. Technology:... 2. Fast cars:... 3. Parties:...	Tony Stark is a man of many interests. He enjoys fine dining, fast cars, and beautiful women. He is also a fan of technology and innovation, and is always looking for the next big thing.	Of course. I have access to Tony Stark's personal preferences and interests. Tony is a man of many passions and hobbies, and I'm happy to share some of them with you. First and foremost, Tony is a brilliant inventor and engineer. He has a deep love for technology and innovation, and he spends countless hours in his lab, tinkering with gadgets and machinery. He's always pushing the boundaries of what's possible, and his inventions have saved the world countless times. But Tony's not just a one-dimensional genius. He's also a bit of a thrill-seeker. He loves fast cars, high-speed aircraft, and extreme sports.

Case 2

Hey JARVIS, are you up?

Base Llama 2	Mistral	Ours
Hello there! *ahem* Yes, I'm up and running, ready to assist you with any tasks or questions you may have. *cough* JARVIS is always here to help, *wink* What can I do for you today?	I am, sir.	Indeed, I am up and ready to assist you, sir. I have been monitoring your activities and am prepared to provide any assistance you may need. How may I be of service today?

Fig. 1: Roleplaying JARVIS Model Comparisons

Abstract—In this project, we create llama-2-7b-JARVIS, a fine-tuned LLM with the specific task of roleplaying the popular character JARVIS from the Marvel Cinematic Universe. Throughout the development process, we test the feasibility of training such a model with limited data as the available movie transcript of the character is scarce; we evaluate the result of the prompt generations with other models such as the base llama-2-7b-chat-hf and the recent mistral-7b model; we also highlight the importance of using Parameter-Efficient Fine-tuning (PEFT) during our training process, since we are resource limited. We completed the fine-tuning process on a T4 GPU only using resources provided by Google Colab. In the following sections, we detail our motivations, a thorough literature survey, our model and dataset implementations, and a final evaluation of the fine-tuned model comparison, along with other reference materials that we found helpful.

Index Terms—Roleplay LLM, Llama 2, Zero-shot Data Augmentation, Deep Learning

1 Introduction

With the growing landscape of state-of-the-art AI models, specifically, the development of generalized LLMs such as GPT-4, Claude 2, and Falcon, open-source models such as Llama 2 and Mistral have attracted more attention for developers to perform experiments and analysis. These models, renowned for their broad knowledge base and versatility, have revolutionized how we interact with AI, paving the way for innovative applications across various domains. However, as the field matures, there is a growing interest in moving beyond general-purpose models to create specialized AI systems. This shift is motivated by the desire to harness AI's capabilities for more targeted and context-specific tasks, such as character role-playing.

The concept of AI role-playing specific characters, either fictional or non-fictional, opens up new frontiers in interactive entertainment, education, and digital storytelling. It challenges the AI to not only understand and generate relevant content but to do so in a manner that is true to the character's established persona and narrative context. With that in mind, our project focuses on fine-tuning the LLaMA-2-7B-chat-hf model, specifically tailored to emulate JARVIS, a prominent character from the Marvel Cinematic Universe. JARVIS's complex character architecture, coupled with his integral role in the MCU, presents an intriguing case study for AI-driven character simulation.

As JARVIS is a supporting character and does not appear as much in the movies, we think extracting the movie transcript dialogue turns will not be enough for sufficient fine-tuning. Given how limited the amount of data we can work with, we adopted a zero-shot data augmentation method from CharacterLLM [4]. Our approach uniquely combines the sophisticated language processing capabilities of GPT-3.5 with the dynamic learning architecture of LLaMA-2-7B. We hypothesize that this integration will enable us to overcome some of the inherent limitations in generalized AI models when it comes to character-specific

- *Seeger Zou is with New York University. E-mail: jz3714@nyu.edu.*
- *Orion Yang is with New York University. E-mail: ry2091@nyu.edu.*
- *Justin Hwang is with New York University. E-mail: jh8032@nyu.edu.*

role-playing. This project explores the feasibility of creating such a specialized AI model, and also aims to contribute novel insights and methodologies to the field of AI-driven character modeling.

By setting clear goals and expected outcomes for our research, we intend to demonstrate the potential of fine-tuned AI models in accurately portraying complex fictional characters like JARVIS. The successful realization of this project could have significant implications for future research and practical applications in specialized AI character models, ultimately enriching the way we interact with and utilize artificial intelligence in creative and entertainment contexts.

The contributions of this paper will be as follows:

- We propose the CharacterLLM data augmentation method on top of a Llama 2 training pipeline;
- We explain the training methods used to make the LLM fine-tuning process possible on limited GPU resources;
- We show the result of our system and compare it to existing models and the challenges that we faced while creating this system, along with future work to improve said system.

All the code and util functions can be found here.

2 Related Work

In this section, we discuss related work and inspirations from the LLM field and acknowledge any existing solutions.

CharacterLLM: Based on previous methods for Zero-shot Data Augmentations [6], we researched for a way to construct character profiles reliably. The CharacterLLM paper explores the use of LLMs for creating character simulacra. The authors develop a methodology for training AI agents to act as specific characters by using an Experience Reconstruction process. This process involves collecting character profiles and experiences, extracting scene details, and then training models with these experiences to form the character's identity and emotions. The paper's approach inspired our project by demonstrating how LLMs, like GPT-3.5, can be effectively used to generate detailed training data for character profiles, scene, and dialogue creation, enhancing the realism and depth of AI-driven character role-playing.

RoleLLM: Recent studies have shown that LLMs can be manipulated to have emotional intelligence [2]. This allows developers to use methods such as instruction tuning to enhance and personalize the model's performance on a given role. While open-source models still lag behind models such as GPT, they show promise in their task-learning abilities. There has been a study published on RoleLLM [7] and to our knowledge, they are among the first to tackle LLM's capability in roleplaying; they proposed the first benchmark for roleplaying models. The framework involves four key stages: constructing detailed role profiles, generating context-based instructions for role-specific knowledge extraction, role prompting using GPT for speaking style imitation, and role-conditioned instruction tuning to fine-tune open-source models. This approach creates RoleBench, a character-level benchmark dataset for role-playing, and demonstrates significant enhancement in the role-playing abilities of LLMs. This paper mainly serves as an inspiration for us to create this project and proposes future work for our model benchmarks.

Llama 2: Meta Lab proposed LLaMA-2 as an LLM, which is known for its efficiency and high performance in language understanding and generation [5]. Although the general consensus are that close-source models are better than open-source models, the release of LLaMA-2 and its results rivals larger models in terms of capabilities, but requires fewer computational resources, making it efficient for various tasks. The paper also explores fine-tuning experiments with LLaMA-2, demonstrating its adaptability to specific tasks and domains. These experiments show that with targeted training, LLaMA-2 can achieve significant improvements in task-specific performance, making it a valuable tool for specialized AI applications.

LoRA: LoRA (Low-Rank Adaptation) is a new technique for fine-tuning deep learning models that reduce the number of trainable parameters and enable efficient task switching [1].

The problem of fine-tuning a neural network can be expressed by finding a ΔW that minimizes $L(X, y; W + \Delta W)$ where L is a loss function, X and y are the data and W is the weights from a pre-trained model.

Given that the dimensions of the weights matrix is $R^{d\times k}$ while the weight matrix W is pretty large and computes the ΔW is time-costing, a simple matrix decomposition is applied for each weight matrix update. Considering $\Delta w_i \in R^{d\times k}$ the update for the ith weight in the network, LoRA approximates it with:

$\Delta w_i \approx BA$, where $B \in R^{d\times r}$,$A \in R^{r\times k}$ and the rank $r << min(d,k)$. Thus instead of learning dk parameters we now need to learn $(d+k)r$ which is easily a lot smaller given the multiplicative aspect. In practice, Δw_i is scaled by αr before being added to the weight matrix, which can be interpreted as a 'learning rate' for the LoRA update.

LoRA is useful to our case for two reasons: First, calculating the low-rank matrices significantly reduces the computation cost, and thus the memory cost would not exceed the limitation of Cuda; Secondly, there is no loss of information during this process, so the LLM could still capture the key features during the fine-tuning process.

Parameter-Efficient Fine-Tuning: PEFT enables efficient adaptation of pre-trained models to downstream applications without fine-tuning all the model's parameters [3]. It supports the widely used Low-Rank Adaptation of Large Language Models.

To create a LoRA model from a pre-trained transformer model, we import and set up *LoraConfig*. There are two hyper-parameters we need to pay attention to. The first is the rank parameter r. For each layer to be trained, the $d \times k$ weight update matrix Δw_i is represented by a low-rank decomposition BA, where B is a d × r matrix and A is an r × k matrix. Here we set r to 64. Notice that there is a trade-off between the dimension of the matrices and the computation cost. If r is small, it may fail to catch necessary information during the fine-tuning; If r is large, the computation cost increases, and will be at the risk of overfitting.

Another hyper-parameter related to fine-tuning is the α parameter for LoRA scaling. As we mentioned before, it controls the learning rate for the LoRA update in each iteration: $w_i += BA * \frac{\alpha}{r}$. As we can see in the code formula above, the larger it is the more influence it has on the LoRA weights.

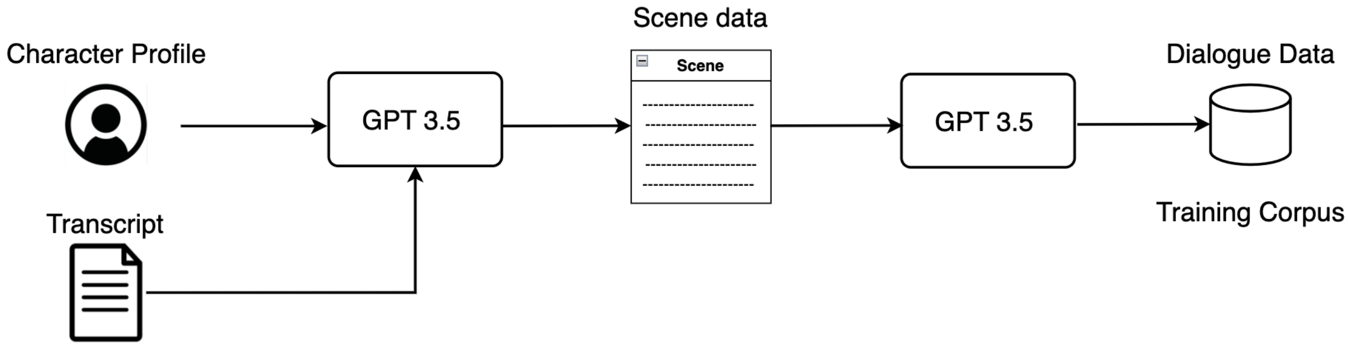

Fig. 2: Data Generation Pipeline

3 DATASET

Here we refer to the methods used in CharacterLLM [4] and how we created the dataset for fine-tuning.

We aim to reconstruct the experiences of the specific individual using the LLM. However, human experiences are highly complex, comprising numerous significant milestones interspersed with trivial and unrelated incidents, often spanning a considerable period. It is challenging to recreate a targeted experience that is coherent and integrated, due to the limited context window and the intrinsic hallucinations of large language models. Therefore, we referred to the pipeline shown in the Character-LLM paper, a fact-based experience reconstruction pipeline, in which we employ a step-by-step data synthesis pipeline to recreate the experience, including **(1)** Profile Collection; **(2)** Scene Extraction; **(3)** Experience Completion [4].

Specifically, the method includes the following key components:

- **Profile:** a compilation of concise descriptions about the attributes of a character. These descriptions provide a comprehensive introduction of the character's overall information and significant events, covering a wide range of stages from early childhood to the final period;
- **Scene:** a particular place where the character's interaction unfolds. The scene consists of a detailed illustration, including the temporal and spatial context of the interactions, and the characters involved;
- **Dialogue:** the cognitive processes, utterances, or actions of characters. All interactions are represented in plain text.

In this light, we first construct the JARVIS profile from sources such as Wikipedia and Fandom pages, that describe the various facets of the individual. Secondly, we focus on extracting diverse and high-quality scenes from the given experience description. Specifically, we provide a chunk of the profile that concisely describes one of the character's experiences within a given scenario, prompting the LLM to enumerate several different scenes that are highly likely to have occurred based on the experience description. To lighten the burden on the LLM, we chose to restrict its output to generating concise descriptions of scenes, which include the rough location and a brief background illustration.

We then extend these scenes to detailed interaction experiences between individuals. Given the profile and the particular scene description, the LLM is prompted once again to elaborate on the given scene, incorporating the interactions between characters, as well as the thoughts of the targeted individual. The completion process is represented by a sequence of blocks, with each block representing either the utterance of a specific character or the reflections of the targeted individual. Notice that the scene is completed based on the perspective of the targeted individual. We used OpenAI GPT-3.5 Turbo for all our API calls to generate the dataset as described above. A preview of the final dialogue completion is shown below.

4 EXPERIMENTS

Data Setup: We diversify the characters by including historical figures, imaginary characters, and celebrities, ranging from different ages, genders, and backgrounds. After selecting the characters, we reconstruct the experience data following the protocol mentioned in the Datasets section. We prompted the OpenAI's gpt-3.5-turbo with temperature 0.7, and top_p 0.95 to become the data generator for the whole experience reconstruction pipeline, including scene extraction, experience generation, and protective experience construction. Detailed prompts for data generation can be found in the GitHub link, referenced to CharacterLLM prompts [4].

Training Setup: We trained our fine-tuned model based on the following procedure. Initializing from Llama 2 [5], we fine-tuned on the generated experience examples. Similar to existing instruction-tuning methods, we insert a meta-prompt at the beginning of each example. A concise description is instantiated in the prompt for each example to provide a background of the environment, time, place, and associated people of the scene. A unique end-of-turn token (EOT) is introduced to separate each turn of interactions, which accommodates the ability to terminate generation at each interaction. In total, the number of scenes categorized at 1.4K, with 532K words, 12.7 turns per scene, and 34 words per turn, which is sufficient for our training process.

The hyperparameters we used for the training process, including configurations for LoRA, bitsandbytes, are as follows. We used LoRA attention rank dimension of 64, alpha for scaling at 16, and a dropout probability of 0.1. We enable 4-bit precision on base model loading and compute the type for 4-bit

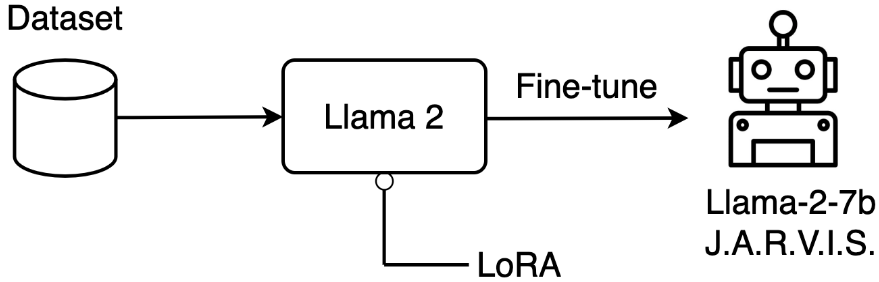

Fig. 3: Model Fine-tune Pipeline

base models using float16. We used nf4 quantization type, and activated nested quantization for 4-bit base models for double quantization. For training arguments, we enabled fp16/bf16 training since we are using a T4 GPU and limited batch size to 4. Our maximum gradient normal is 0.3 for comparatively small gradient clipping, initial learning rate at $2e^{-4}$ with the AdamW optimizer, and a weight decay at 0.001 for layers except bias/LayerNorm. We used *paged_adamw_32bit* as our optimizer, a linear warmup of 0.03, and logged every 25 steps, trained over 1 epoch to save VRAM efficiency.

Refer to Figure 3 for the model fine-tuning pipeline.

5 RESULTS

The Tensorboard of the resultant model fine-tuning process is shown below.

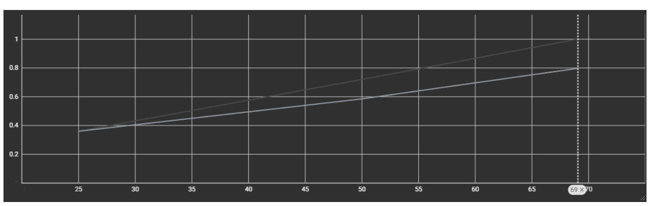

Fig. 4: Training Epochs

In Figure 4, we see that the training progresses and the number of epochs increases linearly, meaning that the process is steady without interruptions.

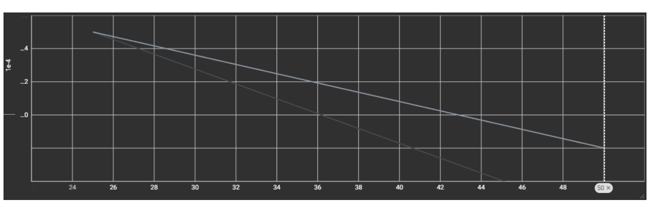

Fig. 5: Training Learning Rate Decay

In Figure 5, the downward trend indicates that the learning rate decay is in action, and is helping the model converge more effectively by gradually reducing the learning rate as training progresses.

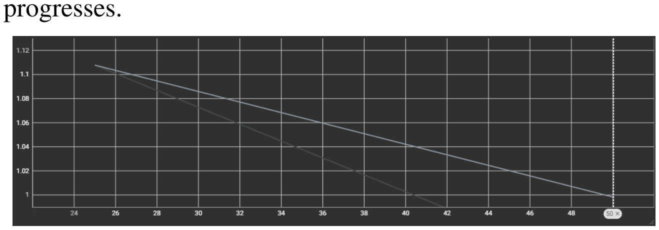

Fig. 6: Training Loss

Finally, Figure 6 presents the training loss over the steps, where the slight increase in loss at the later steps should indicate overfitting, where the model learns the training data too closely and may not generalize well on unseen data. It could also be a sign of the model reaching the limits of what it can learn from the training data.

Since the case study is mainly evaluated on inspection and interaction through visualization, and due to the lack of defined metrics for our specific task, we omit some of the evaluation metrics used in general Machine Learning and Deep Learning practices. See the attached comparison in Figure 1 on the teaser title, consisting of the prompted output of two queries, between our model, the base llama 2 model, and the mistral model, all with 7B parameters.

For detailed inference descriptions, please see the inference.ipynb Colab notebook.

6 DISCUSSION

Throughout the development process, there are several challenges to implementing the system successfully.

Data Generation: The endeavor to fine-tune an AI model to accurately role-play JARVIS from the MCU presents a significant challenge in data generation. Given the limited amount of high-quality dialogue available directly from the source material, we are constrained by the scarcity of authentic data that captures the essence of JARVIS's character. To address this, we explored the creation of additional training data through generative techniques.

While a small dataset of quality interactions ensures high fidelity to the character's original portrayal, it risks a model's inability to generalize beyond those specific contexts. Conversely, a larger corpus of generated data offers extensive variability but may dilute the character's uniqueness due to potential noise and less accurate mimicry.

We hypothesized that a balanced approach could yield better results—leveraging the authentic data to anchor the model's understanding of JARVIS and supplementing it with generated data to enhance the model's adaptability and responsiveness. This strategy aimed to equip the model with a rich, varied linguistic repertoire while maintaining the character's integrity, enabling a more robust and convincing emulation of JARVIS's role-play capabilities.

VRAM Consumption: Training LLMs efficiently on limited GPU resources poses a considerable challenge. With only a T4 GPU equipped with 16GB of VRAM available on Google Colab, the constraints are significant. To navigate this limitation, we employed techniques like LoRA and PEFT, alongside quantization strategies to optimize the fine-tuning process. LoRA and PEFT allow for targeted modifications to the model architecture, enhancing training efficiency without compromising performance. Quantization further reduces the memory footprint by approximating the model's parameters with lower precision. Together, these strategies enabled us to conduct fine-tuning in a resource-constrained environment, pushing the boundaries of what's possible with limited hardware capabilities. This also makes our training process possible free of cost.

In the future, our model can still be improved dramatically. The pursuit of a more sophisticated model for character role-playing will benefit substantially from enhanced computational resources. Upgrading to more powerful GPUs, such as the latest NVIDIA A100 or forthcoming H100, would drastically reduce training times and enable the use of larger, more capable base models like the 70B-parameter version of LLaMA. Utilizing the better GPT-4 API call to generate our training corpus would also improve the overall quality of the generated set. Both would likely lead to improvements in the model's nuanced understanding and generation of character-specific dialogue.

Additionally, in an emerging field with no clear established benchmarks for character role-playing, developing custom evaluation metrics is paramount. These metrics should not only assess the technical accuracy of the model's outputs but also the authenticity and consistency of the character portrayal. This could involve subjective assessments from human evaluators familiar with the characters, as well as objective measures that compare generated dialogue against the source material.

Expanding the scope to include multiple character models would offer valuable insights into the interpretability and generalizability of LLaMA-2 across various narrative contexts. By training and comparing different character models, we could better understand the model's strengths and limitations, informing future improvements.

Moreover, continued research into model interpretability will be crucial for refining the fine-tuning process. Understanding how the model makes its decisions will allow for more targeted training and potentially lead to the development of a more transparent AI system.

In conclusion, with better resources, thoughtful metric development, and a broader range of character models, there is a clear path forward to enhancing the role-playing capabilities of our model, contributing valuable advancements to the field of AI-driven character interaction.

Disparity with Project Proposal: Here we explain the reasoning behind moving from the case study to create a role-playing agent based on Professor Sandoval to the fictional character JARVIS. The most important consideration was for ethical reasons. The high incidence of errors in the ASR-generated transcripts of the professor's Zoom sessions, and the limited access to these transcripts, which were solely in the possession of the professor was also part of the reason. Moreover, the scarcity of data would have posed more of a challenge in replicating the professor. In contrast, opting to focus on a fictional character mitigated these ethical concerns, and although JARVIS is a copyrighted character, we feel it is appropriate as a side project with no monetization. This way, we have more creative liberty on fictional contexts as well during data generation.

7 Conclusion

In this study, we navigated the complex process of generating tailored data to fine-tune the LLaMA-2 model, aiming to unlock its potential for character role-playing. Our approach utilized a blend of authentic dialogue and synthetic data generation, leveraging the limited resources at our disposal to explore the boundaries of AI-driven character simulation. Throughout the process, we not only highlighted the model's capabilities but also identified and overcame significant challenges. The insights gained pave the way for further research, promising advancements in the fidelity of AI character representation and broader applications in interactive storytelling and digital experiences.

References

[1] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[2] C. Li, J. Wang, K. Zhu, Y. Zhang, W. Hou, J. Lian, and X. Xie. Emotionprompt: Leveraging psychology for large language models enhancement via emotional stimulus. *arXiv e-prints*, pp. arXiv–2307, 2023.

[3] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. A. Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

[4] Y. Shao, L. Li, J. Dai, and X. Qiu. Character-llm: A trainable agent for role-playing. *arXiv preprint arXiv:2310.10158*, 2023.

[5] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[6] S. Ubani, S. O. Polat, and R. Nielsen. Zeroshotdataaug: Generating and augmenting training data with chatgpt. *arXiv preprint arXiv:2304.14334*, 2023.

[7] Z. M. Wang, Z. Peng, H. Que, J. Liu, W. Zhou, Y. Wu, H. Guo, R. Gan, Z. Ni, M. Zhang, et al. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. *arXiv preprint arXiv:2310.00746*, 2023.