IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

AI - Assisted Chatbot for NVIDIA SDKs and Toolkits Development

MS. KAVYA SRUTHI SS
Department of
Artificial Intelligence and
Machine Learning
Sri Shakthi Institute of
Engineering and Technology

MR. HAFID WAQAR AHMED S
Department of
Artificial Intelligence and
Machine Learning
Sri Shakthi Institute of
Engineering and Technology

MS. S. NIVEDHA
Assistant Professor
Department of
Artificial Intelligence and
Machine Learning
Sri Shakthi Institute of
Engineering and Technology

Abstract — This paper introduces an AI-driven language model (LLM) designed to facilitate users' comprehension and adept usage of different NVIDIA SDKs (Software Development Kits) and toolkits. The central goal is to provide a LLM which establishes an interactive and intuitive platform that offers inclusive insights, practical illustrations, and expert direction on NVIDIA's diverse SDKs and toolkits. Harnessing the capabilities of language models and NVIDIA's toolkits, this study strives to streamline developers' learning process, enabling them to harness NVIDIA's technological advancements with enhanced proficiency.

I. INTRODUCTION

In recent years, there have been significant advancements in the field of artificial intelligence (AI), particularly in the development of AI-driven Language Models (LLMs). These models have transformed the way we interact with data, enabling natural language understanding and generation with unparalleled accuracy and fluency. They hold immense potential in various applications, including understanding and utilizing complex software development kits (SDKs) and toolkits. NVIDIA, a prominent company in graphics processing units (GPUs) and AI technology, provides a wide range of SDKs and toolkits crucial for applications in deep learning and high-performance computing. However, accessing and effectively using these resources can be daunting, especially for developers new to the NVIDIA ecosystem. This paper introduces an AI-driven Language Model (LLM) specifically tailored to address these challenges. Our LLM acts as a liaison between developers and NVIDIA's SDKs and toolkits, simplifying comprehension, configuration, and utilization of these powerful resources.

The main objectives of this paper are as follows:

- Describing the methodology behind developing our LLM, including data collection and model selection.
- Presenting the results of our LLM in assisting developers with NVIDIA's SDKs and toolkits.
- Highlighting the novelty of our approach, which utilizes multiple language models like falcon-7b and gorilla-llm to improve responses to various queries. By achieving these objectives, this paper aims to contribute to democratizing access to advanced technologies, empowering developers to effectively utilize NVIDIA's SDKs and toolkits.

II. LITERATURE REVIEW

- Large Language Models:

– **Falcon-7B:** Falcon-7B is a 7B parameter causal decoder-only language model developed by the Technology Innovation Institute (TII) in Abu Dhabi. It is trained on a massive dataset of text and code, and can perform a variety of tasks, including text generation, translation, and question answering. Falcon-7B is a good all-around LLM, and it is particularly well- suited for tasks that require a deep understanding of language and code. For example, Falcon-7B can be used to generate realistic and coherent text, translate languages accurately, and answer questions about

a variety of topics, including code.

– **Gorilla-7B:** Gorilla-7B is a 7B parameter API-powered language model developed by researchers at the University of California, Berkeley. It is trained on a massive dataset of text and code, as well as a collection of over 1,600 APIs. This allows Gorilla- 7B to not only generate text, translate languages, and answer questions, but also to generate and execute API calls.

- **Quantized LoRA:** Quantized LoRa (QLora) is a method for fine-tuning large language models (LLMs) that is both efficient and effective. QLora works by quantizing the weights of the LLM, which reduces the memory footprint of the model and makes it faster to train. QLora also uses a technique called LoRa (Low-Rank Adaptation) to update the weights of the LLM during fine-tuning. LoRa is a more efficient way to update the weights of an LLM than traditional fine-tuning methods, and it can help to improve the performance of the model on downstream tasks.

III. DATA COLLECTION

In our efforts to develop AI-assisted learning tools for NVIDIA SDKs and Toolkits, the process of gathering data played a crucial role. We utilized web scraping techniques to collect information from various sources on the official NVIDIA website. Our data collection covered the following key sources:

1) Kit Manuals: Detailed information was gathered from NVIDIA's Kit Manuals. This enabled our AI to understand the installation, setup, and usage procedures of various SDKs and toolkits.

2) Developers Forum: The NVIDIA Developers Forum was explored to learn from the experiences and queries of actual users. This approach allows our AI to provide practical advice based on real-world scenarios.

3) SDK Manager: Information was extracted from the NVIDIA SDK Manager, excluding data presented in tables which are difficult to scrape. This allows our AI to offer insights on setting up and managing the complete development environment.

4) Developer Guide: Valuable insights were obtained from the NVIDIA Developer Guide, which is highly beneficial for developers looking to get started or maximize their usage of NVIDIA's software. This enhances our AI's ability to respond effectively to complex development inquiries.

5) StackOverflow Queries: We conducted data extraction by filtering questions and answers tagged with Nvidia and CUDA, accumulating approximately 27,000 relevant queries. Along with question content, we also extracted the vote count for each question, encompassing both positive and negative numeric values. This vote count data is seen as a valuable resource for future endeavors in implementing Reinforcement Learning from Human Feedback (RLHF) techniques.

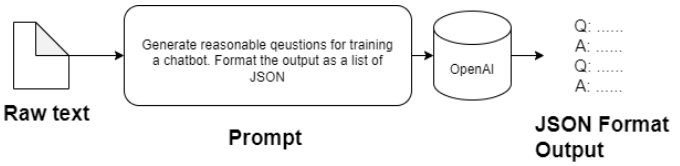

Figure 1: Algorithm architecture

IV. DATA PREPROCESSING

In data preprocessing, we acquired a lot of data from various resources like 1,2,3,4,5. • NVIDIA Developer Forum. In the NVIDIA developer forum, a comprehensive ar- ray of thirteen distinct subtopics, including some topics like accelerated computing, graphics, omniverse etc. was observed. Within these subtopics, a substantial volume of inquiries and corresponding responses were docu- mented, constituting a repository of questions and an- swers. The data acquisition process entailed the extraction Fig. 1. QA pairs generation and transformation of these interactions into a struc- tured format, which consists of questions and answers. To achieve this, an application programming interface (API) provided by OpenAI was employed, leveraging various prompts designed to ensure that the resulting questions were formulated in a generic manner, thereby eliminating user-specific queries. This procedure aimed to create a standardized dataset conducive to research and analysis without any discernible traces of individual user influence, thereby fostering the generation of unbiased insights.

V. DATA PREPARATION AND MODEL DESCRIPTION

The process of abstracting data began with extracting relevant information from various forums and web pages. Next, the gathered textual data was transformed into questions and corresponding answers, facilitated by employing the OpenAI GPT-3.5 model. This method enabled the automatic generation of inquiries and responses from the raw textual content obtained from the provided links. To ensure that the curated questions were closely aligned with NVIDIA's documentation, a thorough manual selection process was conducted. This involved carefully reviewing the generated questions to ensure their relevance to NVIDIA's documentation, while filtering out queries tailored exclusively for individual users, as depicted in Fig 1.

We utilized open-source large language models for fine-tuning on the generated data. Given the diverse sources of the data, we categorized the query types into two main categories: code-intensive and generic questions. For each of these query types, we fine-tuned a Language Model (LLM) using the LoRA method, which is known for its parameter-efficient fine-tuning approach.

Figure 2: F1 scores on test set

The Falcon-7b model has been deployed, fine-tuned on data predominantly consisting of inquiries about instructions and facts regarding NVIDIA SDKs, forums, and guides. This model was configured in 8-bit with FP16 precision for gradient computations. Hyperparameters were chosen after multiple experiments with LoRA, with final selections of rank = 32 and alpha (normalization factor) = 64. During inference, the generation configuration parameters can be adjusted based on user preferences, though we typically set the temperature to be near 0.

For code-intensive queries, we utilized data from Stack Overflow queries with a vote count strictly greater than 0

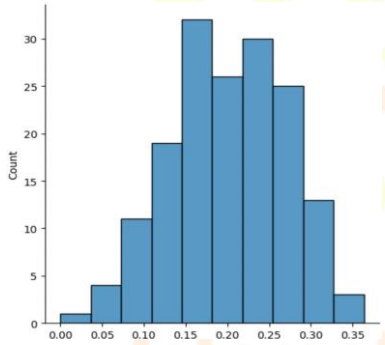

Figure 3: ROUGE-L scores on test set

and fine-tuned the gorilla-falcon-7B model. This LLM was loaded with the same configuration as mentioned above, with the same rank but with the normalization factor set to 64. The generation parameters were also set to the same values as Falcon-7B.

Once these models were fine-tuned, users need to specify the type of query (Generic or Code), and they can receive responses from the respective fine-tuned LLM.

VI. EVALUATION AND LIMITATIONS

The performance of the developed Language Models (LLMs) was assessed using categorical cross-entropy loss for both training and validation data. The validation loss measured approximately 1.15 for the Falcon-7 model and 2.91 for the Gorilla model. We opted to use this loss as the evaluation metric for code-based queries due to the blend between natural language and syntactical code phrases. For generic queries, we employed both ROUGE-L and F1-scores (unigram) to evaluate the performance of the Falcon-7B model. The mean F1-score improved from 0.169 to 0.210 after fine-tuning

Additionally ıding on the generation c ı100 GPU, it typically ran

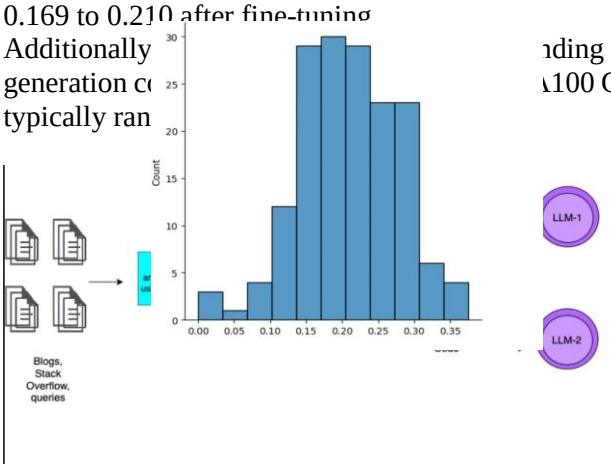

Figure 4: Pipeline for inference

Limitations:

Excessive GPU Memory Consumption for Inference and Training: Both inference and training processes require significant GPU memory. Specifically, 18 GB of GPU memory is necessary for 8-bit loading, with each model requiring an additional 30 GB during fine-tuning.

Prolonged Inference Times: Processing each question during the inference phase takes approximately 10 seconds on an A100 GPU, even when setting the parameter "max new tokens" to 200.

Suboptimal Utilization of Code-Related Data: Computational limitations prevent the full utilization of code-related data, potentially hindering model performance.

Pattern Repetition in LLM2: LLM2 demonstrates repetitive patterns when generating predictions for complex code-related inquiries, leading to a decline in model performance for such queries.

Impact of 4-bit Precision on Performance: The use of 4-bit precision for LLMs noticeably decreases model performance.

VII. FUTURE WORK

The initial concept aimed at developing a system for creating a Mixture of Models, where each model specializes in a specific domain of expertise. To achieve this, the Gorilla LLM model underwent training on Stack Overflow data, which primarily focused on coding-related queries. Conversely, the Falcon model was trained on data more inclined towards general inquiries associated with Nvidia SDKs. As a potential future enhancement, incorporating data containing human preferences or user votes could facilitate the implementation of Reinforcement Learning from Human Feedback (RLHF) techniques, enabling responses to better align with human preferences.

The envisioned pipeline involves the following steps:

- When a user submits a query, it is transformed into a vector using sentence transformers.
- Simultaneously, the original query is processed by

both the Gorilla LLM and Falcon models, resulting in two sets of answers.
- These answers are then transformed into vectors using sentence transformers.
- The next step involves computing the KL-divergence score between the distributions of questions and answers.
- The final answer selected is the one that exhibits the least divergence from the distribution of the original question.
- Additionally, vector databases and search agents can be utilized to retrieve information along with hyperlinks for user reference.

VII. CONCLUSION

The fine-tuned Large Language models were able to grasp a good amount of information from a generic perspective, while they have not performed up to the mark with the syntactically complex such as queries related to codes. In this way, we have incorporated knowledge inference from NVIDIA SDK documentation, and this data played a huge role in the proper generation of responses.

IX. 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," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[2] H. Arabnejad and J. G. Barbosa, "Peft," July 2014. [Online]. Available: https://www.researchgate.net/publication/264001631 PT

[3] S. Patil, T. Zhang, X. Wang, and J. Gonzalez, "Gorilla: Large language model connected with massive apis," May 2023. [On- line]. Available: https://www.researchgate.net/publication/371008797 Gorilla Large Language Model Connected with Massive APIs

[4] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, and J. Launay, "The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only," 2023. [Online]. Available: https://arxiv.org/abs/2306.01116

[5] G. Pu, A. Jain, J. Yin, and R. Kaplan, "Empirical analysis of the strengths and weaknesses of peft techniques for llms," 2023. [Online]. Available: https://arxiv.org/abs/2304.14999

[6] O. Topsakal and T. C. Akinci, "Creating large language model applications utilizing langchain: A primer on developing llm apps fast," 2023. [Online]. Available: https://as-proceeding.com/index.php/icaens/ article/view/1127

[7] N. Lambert, L. Castricato, L. von Werra, and A. Havrilla, "Illustrating reinforcement learning from human feedback (rlhf)," 2022. [Online]. Available: https://huggingface.co/blog/rlhf

[8] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. Christiano, "Learning to summarize from human feedback," 2022. [Online]. Available: https://arxiv.org/abs/2009.01325

[9] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. F. Christiano, and G. Irving, "Fine-tuning language models from human preferences," 2019. [Online]. Available: http: //arxiv.org/abs/1909.08593

[10] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," 2023. [Online]. Available: https://arxiv.org/abs/1706.03741

[11] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, X. Jiang, K. Cobbe, T. Eloundou, G. Krueger, K. Button, M. Knight, B. Chess, and J. Schulman, "Webgpt: Browser-assisted question-answering with human feedback,"