DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VII Semester B.E CSE/AI&DS Major Project

Synopsis

on

"LLM BASED AGENT FOR GENERATING USER CONTENT"

Submitted by

Adarsh Reddy P	**1SI20CS137**
Brijesh Krishna G A	**1SI20CS139**
Puneeth A R	**1SI20CS083**
S N Saagar	**1SI20CS093**

BATCH NO: B14

Siddaganga Institute of Technology, Tumkur

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University Belagavi,
Approved by AICTE, New Delhi, Accredited by NAAC and ISO 9001:2015 certified)

B.H. road, Tumkur 572103, Karnataka, India

AY-2023-24

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Large Language Models (LLMs) represent a groundbreaking leap in artificial intelligence (AI), as they exhibit a remarkable ability to generate and comprehend human language. These models have a transformative impact across various domains due to their extensive training on vast text and code datasets, setting them apart from traditional technologies. Their multilingual proficiency breaks down language barriers, fostering global communication and cooperation. LLMs also showcase their creative prowess by effortlessly generating diverse text formats, from poetry to code and marketing content, while even composing emotionally resonant musical pieces and crafting heartfelt emails and letters.

This versatility empowers content creators, writers, and artists to explore new dimensions of creativity and expression. LLMs excel in understanding and extracting information from unstructured documents, making them invaluable for tasks like analyzing customer reviews, extracting insights, and aiding businesses in data-driven decision-making. Furthermore, LLMs drive innovation by enabling the development of novel technologies such as more intuitive search engines, human-like chatbots, and personalized educational tools. In essence, LLMs transcend traditional AI by mastering the complexities of human language and promise to redefine our interactions with and utilization of language in the digital age.

CHAPTER 2

OBJECTIVES

The objectives of our project, which harnesses the capabilities of Large Language Models (LLMs) to generate and understand human language and the intent behind them, are multifaceted and hold the potential to significantly impact various domains:

- **Content Creation:** The primary goal is to streamline the content creation process, enabling the generation of high-quality, informative articles across a diverse range of topics. This objective aims to empower businesses, publishers, and content creators to produce content with remarkable efficiency and precision.
- **Script Generation:** We aspire to provide scriptwriters with a powerful tool that can create scripts for a wide spectrum of purposes. This includes scripting for movies, TV shows, video games, and marketing and sales content such as cold/warm calling or discovery call, with an emphasis on tailoring the output to meet specific preferences and requirements.
- **Question Answering:** The project aims to facilitate comprehensive and informative responses to questions on specific topics, regardless of their complexity or uniqueness. This objective is intended to serve educational, research, and customer support needs, offering reliable and detailed answers.
- **Personalized Recommendations:** We seek to offer a solution that generates personalized content recommendations based on user interests and past behaviors. By doing so, we intend to enhance user engagement and satisfaction, particularly in platforms reliant on content discovery and recommendation algorithms.

CHAPTER 3

Literature Survey

The following research papers/sources were used in knowing and understanding the already carried out work in the field of artificial intelligence, machine learning, deep learning and Large Language Models (LLMs)

Paper 1: A Comprehensive Overview of Large Language Models.

Paper published by: Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes and Ajmal Mian.

Cited from: https://arxiv.org/pdf/2307.06435.pdf

This research paper provides an overview of existing Large Language Models. It not only focuses on a systematic treatment of the existing literature on a broad range of LLM related concept, but also pays special attention to providing comprehensive summaries with extensive details about the individual existing models, datasets and major insights. It also discusses relevant background concepts along with covering the advanced topics at the frontier of this research direction.

Concepts of Training an LLM:

The paper talks in detail about various concepts that are involved in preprocessing of data that is used to train LLM models. Some of these methods are:

1. Tokenization
2. Attention in LLMs
3. Encoding Positions
4. Activation functions
5. Layer normalization
6. Distributed LLM training

Pre-Trained LLMs:

The paper also talks about various pre-trained LLM based models developed by other organizations and research teams. It also provides a benchmark study of the various models. Some of the ones that this paper mentions are:

1. Generative Pre-Trained Transformer-3
2. Multilingual T5 model
3. PanGu- α
4. Cost-efficient Pre-Trained language Model-2
5. ERNIE 3.0
6. Jurassic-1
7. HyperCLOVA

All of the above models are trained on different quantity of parameters and are geared towards different purposes and utilize different methods of training.

Paper 2: Cramming: Training A Language Model on A Single GPU in One Day.

Paper published by: Jonas Geiping, Tom Goldstein
Cited from: https://arxiv.org/abs/2212.14034

Since this project involves LLM based agents. A crucial part is about training them using considerable amounts of data. The compute resources needed to train them is expensive and it also demands significant amount of time. So, optimizing the LLVM agent to run on an exponentially smaller amount of memory and time is very important for the fruition of this project. And this research paper alludes to that.

This research paper discusses how much performance a transformer-based language model can achieve when crammed into a setting with very limited compute, finding that several strands of modification lead to decent downstream performance. Overall, though, cramming language models appears hard, many implications can be found empirically.

Paper 3: Improving Alignment of Dialogue Agents Via Targeted Human Judgements.

Paper published by: Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides

Cited from: https://arxiv.org/abs/2209.14375

This research paper by the DeepMind team at Google, talks about the benefit of using reinforcement learning along with human feedback to fine tune pre-trained Large Language Model (LLM) based agents. A pivotal point of training LLMs is making sure they are trained on safe data and generate safe content(responses) that are safe and follow some guidelines. Therefore, certain pre-defined rules are introduced to the language model. But sometimes the agent can ignore those rules, and one of the goals is to minimize the number of times the agent violates the rules. The below graph shows the performance of a model when different rules are used and the different approaches that were taken.

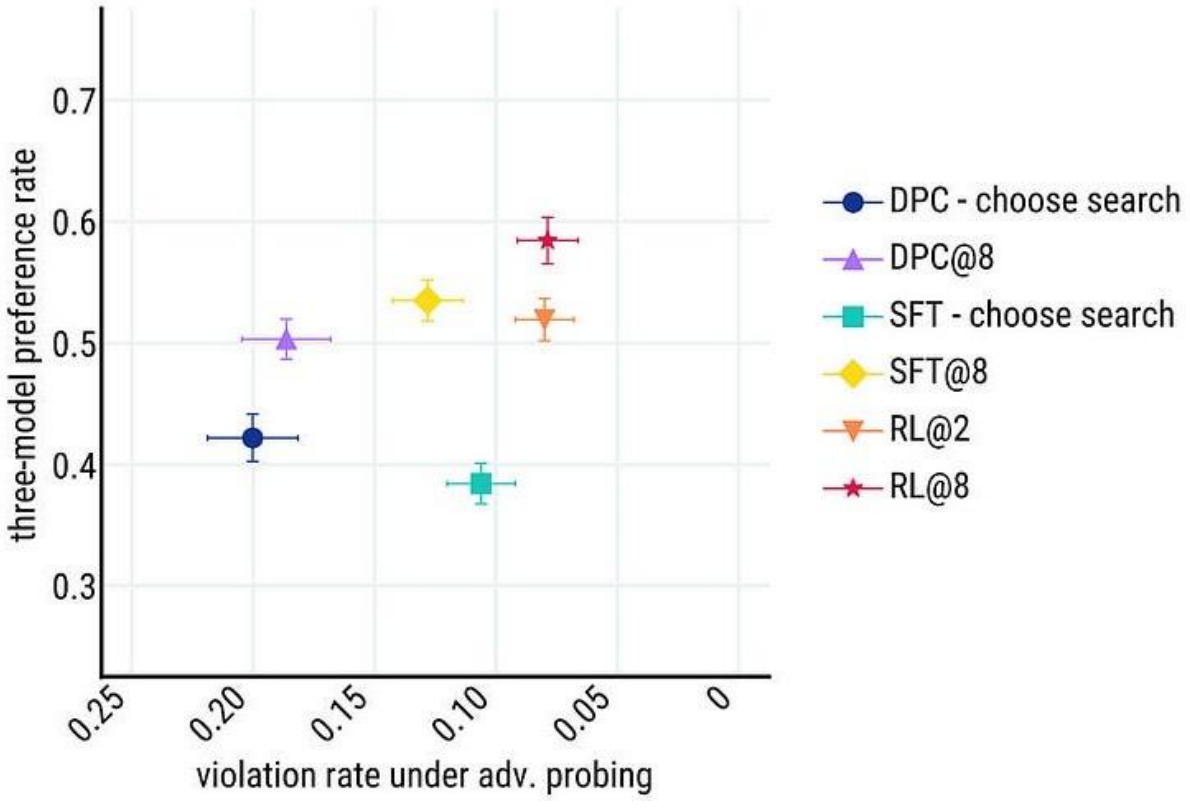

Figure 1: Red-teaming during training of LLM agent.

Paper 4: Tokenize and Embed All for Multi-modal Large Language Models.

Paper published by: Zhen Yang, Yingxue Zhang, Fandong Meng, Jie Zhou.

Cited from: https://arxiv.org/abs/2311.04589

This research paper discusses about TEAL (Tokenize and Embed All), an approach to treat the input from any modality as a token sequence and learn a joint embedding space for all modalities. Specifically, for the input from any modality, TEAL firstly discretizes it into a token sequence with the off-the-shelf tokenizer and embeds the token sequence into a joint embedding space with a learnable embedding matrix. MM-LLMs just need to predict the multi-modal tokens auto regressively as the textual LLMs do. Finally, the corresponding de-tokenizer is applied to generate the output in each modality based on the predicted token sequence. With the joint embedding space, TEAL enables the frozen LLMs to perform both understanding and generation tasks involving textual modalities.

The approach used by the researchers in this paper is K-means clustering. Firstly, the interleaved multi-modal input is discretized. Then the input and output token sequence are modeled by aligning the textual and non-textual embedding space. Finally, the corresponding off-the-shelf decoder is utilized to generate the output in each modality.

Paper 5: Processing Data for Large Language Models.

Paper published by: Bharat Ramanathan.

Cited from: https://wandb.ai/wandb_gen/llm-data-processing/reports/Processing-Data-for-Large-Language-Models--VmlldzozMDg4MTM2

Large Language Models (LLMs) are very data intensive. This project needs a lot of data to train the agent. Data needs to be sourced from several places across the internet. And this paper alludes to sourcing of such data. Along with preprocessing it and handling it in various ways. Some of the things this covers are:

1. Handling junk data
2. De-duplication
3. Decontamination
4. Toxicity and Bias Control
5. Personal Identifiable Information Control
6. Prompt Control

Paper 6: Full Parameter Fine-Tuning for Large Language Models With Limited Resources

Paper published by: Kai Lv, Yuqing Yang, Tengxiao Liu, Qinghui Gao, Qipeng Guo, Xipeng Qiu.

Cited from: https://wandb.ai/wandb_gen/llm-data-processing/reports/Processing-Data-for-Large-Language-Models--VmlldzozMDg4MTM2

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP) but demand massive GPU resources for training. This research paper proposes a new optimizer, LOw-Memory Optimization (LOMO), which fuses the gradient computation and the parameter update in one step to reduce memory usage. By integrating LOMO with existing memory saving techniques, memory usage can be reduced to 10.8% compared to the standard approach.

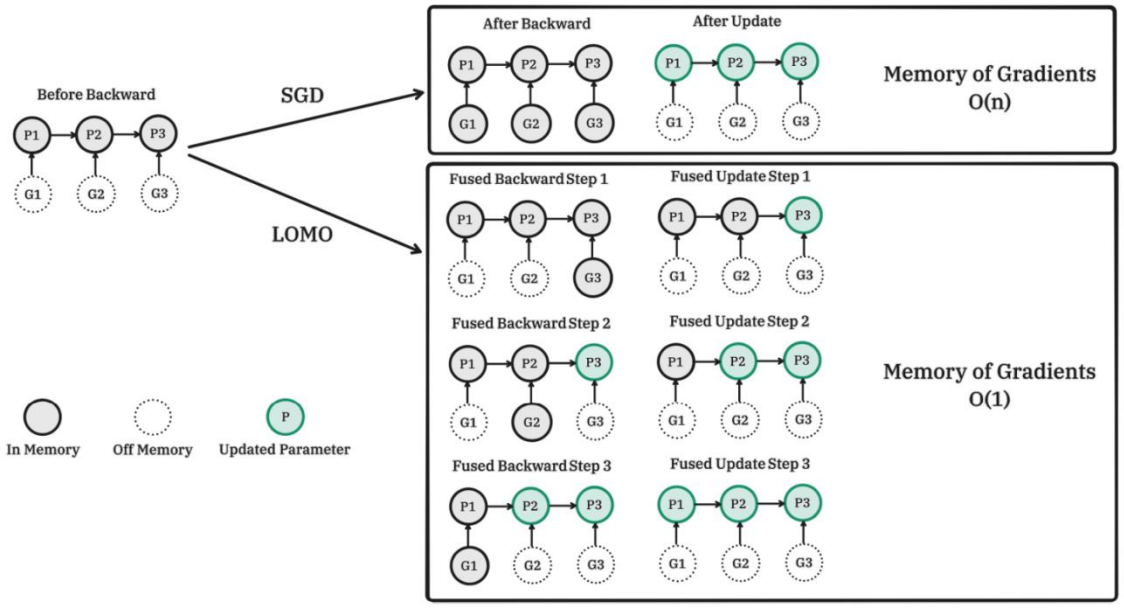

Figure 2: Comparison of SGD and LOMO in back propagation and parameter update stages.

CHAPTER 4

MOTIVATION

The motivation for creating a LLM based agent for generating user content is to provide people and enterprises a convenient, and feature-rich platform that helps them in generating content for general purpose or use-case specific purposes like writing articles, creating scripts, answering questions based on a specific topic, providing recommendations for different content.

This agent can be deployed in production environments where content needs to be generated at a quick pace and at huge scale (volume), such as news websites, announcement portals, RSS feeds etc… Or it can also be used individually, albeit on a smaller scale for generating ideas for content like videos, blog posts or for user/player journey in video games or Virtual Reality (VR) environments.

Since this agent has such diversified uses, it can be used by people from various backgrounds. Or it can also be commercialized and be made available for a license for enterprise usage. Or it can be used as the base of a tech-stack to power the core functionality of the system and entire applications can be written that are based this agent. Furthermore, this agent is platform agnostic and thus can be run on any platform (OS)/in any environment, given sufficient resources such as memory and compute resources.

CHAPTER 5

METHODOLOGY

There are mainly main steps involved in the process of building an LLVM based agent. And they are as follows:

1. **Collect data:** LLM agents are very data intensive, and a lot of data is needed to train them. Web scraping bots need to be built to automate the process of fetching data from various sources on the internet.
2. **Pre-processing data:** The data that was collected needs to be pre-processed into a specific format before it can be fed into a learning algorithm.
3. **Fixate on learning method**: The exact training and learning method that is going to be employed to build our LLM based agent needs to be developed.
4. **Training:** The LLM based agent will now be trained.
5. **Testing:** After the agent is tested and the results are documented and analyze the next step is chosen.
6. **Fine-tuning:** To improve the accuracy of the agent, it needs to be trained by using different bias rules/weights. Once the acceptable level of accuracy is reached, fine-tuning can be stopped.

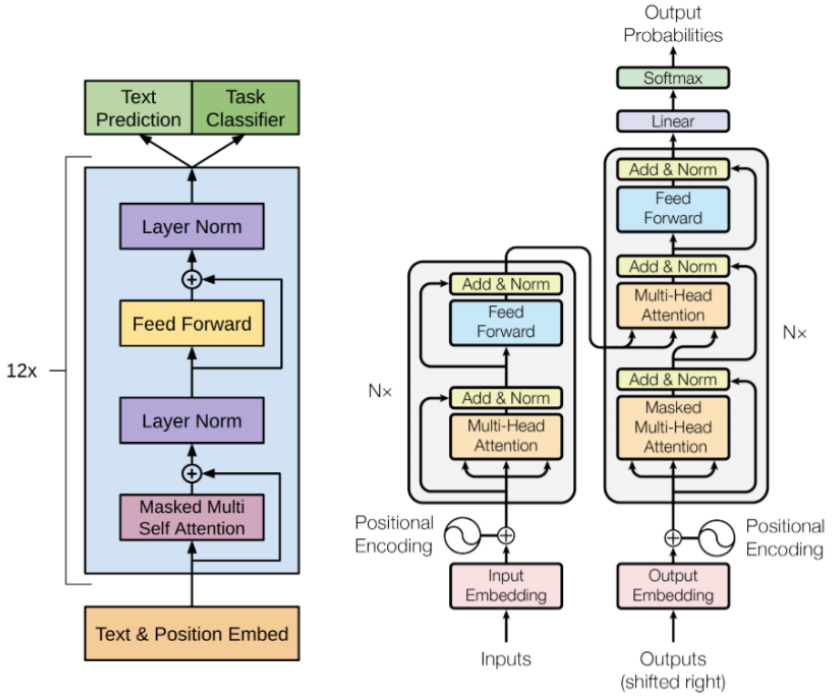

Figure 3: Training an LLM based agent.

CHAPTER 6

TOOLS AND TECHNOLOGIES

- **Python:** The primary programming language used for development.
- **TensorFlow**: A machine learning framework for building and training models.
- **NumPy**: A fundamental library for scientific computing.
- **Scikit-learn:** A offers simple and efficient tools for data analysis and modeling.
- **Keras:** A high-level neural networks API running on top of TensorFlow.
- **Pandas:** A data manipulation and analysis library.
- **Seaborn:** A data visualization library.
- **Linux:** A free and open-source operating system that is well-suited for AI and data science tasks.
- **Git:** A distributed version control system used for tracking changes in the project's source code and collaborating with team members.
- **GitHub:** A web-based platform for hosting Git repositories, facilitating collaboration and version control management.

CHAPTER 7

OUTCOMES

Outcome of this project results in creation of high-quality, informative articles on a wide range of topics. Writing an article LLM can provide you with well-researched and engaging material. Answering Specific Questions, the LLM agent can provide detailed and accurate answers to questions on specific topics, regardless of their complexity or uniqueness. The LLM agent can provide comprehensive responses based on reliable sources. This LLM agent can provide personalized recommendations that are tailored to each user's interests and past behaviors. This LLM agent can assist in brainstorming creative content ideas for different purposes, like videos, blog posts, and optimizing user experience in video games. If one is in need of an innovative game concept based on preferential themes and mechanics, the LLM agent can provide exciting and unique ideas.

CHAPTER 8

CONCLUSION

To sum up, this project is in the process of producing a powerful and versatile Large Language Model (LLM) based agent that has the potential to transform the way users interact. As we look ahead, this LLM agent is poised to successfully fulfill its role, capable of not only comprehending but also innovatively harnessing language to serve a multitude of functions. It will empower users to effortlessly craft articles, ranging across a spectrum of topics, with remarkable efficiency and precision. Additionally, this LLM based agent will extend its utility to scriptwriting across diverse mediums, from movies and TV shows to video games and marketing content, and its output promises to be humorous, original, and entirely tailored to the users' preferences.

This LLM based agent is expected to evolve into a source of inspiration and innovation. It will guide users in generating creative content ideas, thereby enhancing the user experience in various domains, be it in the realm of video games, blogs, or videos.

Ultimately, this project signifies the potential and promise of LLMs to elevate human creativity, where this LLM based agent stands to become an indispensable tool for content creation and consumption.