Code generation using LLM's and HuggingFace Transformers and Langchain

T Sasank Reddy
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru Amrita Vishwa Vidyapeetham, India
bl.en.u4aie22160@bl.students.amrita.edu

T Krishna Varma
Department of Computer Science and Engineering Amrita School of Computing, Bengaluru Amrita Vishwa Vidyapeetham, India
bl.en.u4aie22158@bl.students.amrita.edu

M Srinivas
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru Amrita Vishwa Vidyapeetham, India

V Kushal
Department of Computer Science and Engineering Amrita School of Computing, Bengaluru Amrita Vishwa Vidyapeetham, India
bl.en.u4aie22161@bl.students.amrita.edu

Abstract: This research, we introduce a novel method for code generation using Hugging Face Transformers and Langchain architecture. Our approach combines open embeddings with a Large Language Model (LLM) variant called Meta-LLAMA to simplify code snippet production. By integrating Langchain and Transformers, we automate code generation to address real-world programming challenges.

We demonstrate the effectiveness of our method by providing a real-world example where our model generates Python code snippets for automating data preprocessing tasks in machine learning workflows. Our system efficiently manages data cleaning, normalization, and feature engineering based on input data and specified preprocessing procedures, showcasing adaptability across programming languages and domains.

Test results indicate the framework's ability to produce precise and syntactically sound code. Qualitative analysis confirms the readability and maintainability of the resulting code, supporting its suitability for real-world software development projects.

***Keywords*— Code Generation, Langchain, Transformers, Language Model Meta-Learning (LLM), Meta-LLAMA, Hugging Face, Open Embeddings, Machine Learning, Data Preprocessing, Software Development.**

I. INTRODUCTION

The convergence of natural language processing (NLP) and programming has sparked interest, promising transformative advancements in software development. Langchain, combined with Hugging Face Transformers, offers a compelling framework for automated code generation, utilizing large language models (LLMs) like Meta-LLAMA and open embeddings.

Generating code from natural language specifications enhances developer productivity and democratizes software development. However, challenges remain in achieving accurate, efficient, and context-aware code generation. This paper introduces a novel approach leveraging Langchain and Transformers to address these challenges and advance code generation.

Our approach aims to exploit contextual representations learned by LLMs and integrate them with transformer-based architectures tailored for code-related tasks, improving code generation accuracy, efficiency, and scalability.

This project explores the feasibility and effectiveness of using Langchain and Transformers for code generation, focusing on investigating the capabilities of Meta-LLAMA and open embeddings in capturing natural language semantics and translating them into executable code.

II. LITERATURE SURVEY

[1] Traditional Chinese Medicine (TCM) explores its historical roots, challenges in contemporary practice, and recent technological advancements. TCM's ancient origins offer unique treatments but prescribing can be challenging for young doctors due to complex diagnoses and shifting syndrome patterns. To bridge this gap, TCM prescription recommendations from textbooks and clinical guidelines are crucial. Advancements in AI and big data analytics show promise in providing intelligent TCM prescription recommendations, potentially enhancing treatment efficacy and patient experience. These advancements hold significant implications for integrating AI-driven tools into clinical practice and exploring personalized TCM treatment approaches based on individual patient data. Continued research in this field is essential for maximizing the benefits of intelligent TCM prescription recommendations in healthcare.. [2]

We use Artificial Intelligence (AI) accelerators in tandem with large language models (LLMs) for automating the design process is multifaceted. The demand for specialized AI accelerators due to the increasing complexity and performance requirements of AI workloads. Existing literature highlights the labor- and time-intensive nature of designing these accelerators, despite the partial alleviation provided by current design exploration and automation tools.

accelerator development. In light of this, recent research has turned towards leveraging the remarkable capabilities of LLMs, particularly in generating high-quality content in response to human language instructions. This shift in focus has led to the development of frameworks like GPT4AIGChip, aimed at democratizing AI accelerator design by utilizing human natural languages instead of domain-specific languages. Through an in-depth investigation into the limitations and capabilities of LLMs for AI accelerator design, researchers have gained insights into the potential of LLM-powered automated design tools. Building upon these insights, the development of GPT4AIGChip showcases an innovative approach featuring an automated demo-augmented prompt-generation pipeline, leveraging in-context learning to guide LLMs in creating high-quality AI accelerator designs. This work represents a pioneering effort in demonstrating the effectiveness of LLM-powered automated AI accelerator generation, setting the stage for future innovations in next-generation design automation tools fueled by LLM capabilities

[3] The resume building applications highlights the increasing importance of these tools, particularly for students from underprivileged backgrounds facing challenges in crafting effective resumes. Recent research has focused on integrating advanced language models like Large Language Models (LLMs) into such applications to streamline the process. These applications typically consist of modules for resume generation, assessment, and user interaction, aiming to leverage LLMs' natural language processing capabilities. Key features include prompt engineering for generating resume bullet points and assessment modules to evaluate content quality. While prototypes demonstrate feasibility, further studies are needed to assess their effectiveness in real-world educational environments, emphasizing usability testing with diverse user groups to support career development initiatives.

[4]. Progress in large-scale language models have prompted an upsurge in studies investigating their application in understanding brain encoding and decoding mechanisms. This interdisciplinary research combines natural language processing (NLP) with neuroscience, resulting in the development of multimodal models that integrate brain activity data with text. Previous literature underscores the importance of validating these models through bi-directional experiments, ensuring reliability in both brain encoding and decoding processes. Comparative studies have demonstrated the superior brain encoding capabilities of these models compared to state-of-the-art language models. Additionally, the introduction of discrete Autoencoder modules provides a versatile tool for extracting brain features beyond functional magnetic resonance imaging (fMRI) studies. While these advancements show promise, further research is necessary to fully explore the practical applications of multimodal language models in cognitive neuroscience and related fields.

[5] State-of-the-art neutral language models have enabled the solving of ad-hoc language tasks without supervised training, known as zero-shot prompting. This approach, gaining popularity, requires experimentation to find optimal prompts due to the impact of different templates and wording choices on accuracy. PromptIDE, a tool developed for this purpose, allows users to experiment with prompt variations, visualize performance, and optimize prompts iteratively. It streamlines the workflow by starting with model feedback using small datasets before validating prompts on larger datasets. Real-world use cases demonstrate PromptIDE's utility in effectively addressing prompt selection challenges for zero-shot prompting tasks.

[6] The rapid use of Deep Neural Networks (DNNs) in software systems has led to challenges in creating and customizing complex architectures from scratch. To address this, machine learning engineers are increasingly relying on reusing large pre-trained models (PTMs) and fine-tuning them for specific tasks, mirroring traditional software engineering practices of reusing software packages. However, while previous research extensively explores reuse practices in software packages, there is a gap in understanding similar practices in PTM ecosystems. In this study, the authors conduct the first empirical investigation of PTM reuse by interviewing practitioners from the popular PTM ecosystem, Hugging Face. The findings identify useful attributes for model reuse, such as provenance and reproducibility, while highlighting challenges including missing attributes and discrepancies in performance. Systematic measurements within the Hugging Face ecosystem substantiate these challenges, providing insights for optimizing deep learning ecosystems and guiding future research on model registry infrastructure and standardization.

[7] we came to know the scarcity of knowledge regarding the measurement, reporting, and evaluation of the carbon footprint of machine learning (ML) models. By analyzing 1,417 ML models and associated datasets on Hugging Face, a leading repository for pretrained ML models, the study aims to provide insights and recommendations for reporting and optimizing the carbon efficiency of ML models. It is the first repository mining study on the Hugging Face Hub API focusing on carbon emissions. Key findings include stagnant reporting of carbon emissions, a slight decrease in reported carbon footprint over two years, and NLP's continued dominance as the main application domain. Correlations between carbon emissions and attributes like model and dataset size are also identified. To promote transparency and sustainable model development, the paper proposes classifications for categorizing models based on their carbon emission reporting practices and efficiency within the ML community.

[8] Automated text summarization is invaluable in scientific and medical fields, enabling the extraction of key information from articles. Recent advancements in deep learning have enhanced this process, particularly in summarizing COVID-19 related research papers. Readability is crucial, and metrics like ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-L-SUM assess performance. Findings indicate models like Distilbart-mnli-12-6 and GPT2-large outperform others in text summarization tasks.

[9] The increase in prevalence of mental health challenges, particularly anxiety, depression, and suicidal thoughts, highlights the urgent need for effective interventions in modern society. Recognizing this imperative, recent advancements in pretrained contextualized language models have paved the way for innovative solutions like MindGuide, a chatbot designed to serve as a mental health assistant. MindGuide utilizes LangChain and its ChatModels.specifically ChatOpenAI, as the foundation of its reasoning engine, enabling it to provide guidance and support in critical areas of mental health. The system incorporates advanced features such as LangChain's ChatPrompt Template,HumanMessage Prompt Template, ConversationBufferMemory, and LLMChain, facilitating early detection and comprehensive assistance for individuals struggling with mental health issues. Furthermore, the paper discusses the integration of Streamlit to enhance the user experience and interaction with the chatbot. This novel approach shows promising potential for proactive mental health intervention and support.

[10] We came through novel methods for evaluating and validating systematic literature reviews in software engineering. The proposed approach involves selecting relevant scientific papers, developing evaluation criteria, and determining

performance metrics. Many experts evaluated literature reviews based on these criteria, showing reasonable agreement (average similarity index: 0.58 to 0.83). Despite varied perspectives, the method yields consistent results. By providing specific questions and criteria, the approach guides experts toward uniform assessments, enhancing the quality of literature reviews. Overall, this framework offers potential for achieving reliable and reproducible evaluations in software engineering research.

III. METHODOLOGY

Data Collection and Preprocessing:

We Gather a diverse dataset of code snippets from various programming languages, domains, and applications.
Preprocess the data to remove noise, irrelevant comments, and non-executable code segments.
Tokenize the code snippets into a format compatible with the transformer architecture, considering the specific requirements of Langchain and Hugging Face Transformers.
Fine-tuning LLM (meta-llama):

Utilizing Langchain's LLM (Large Language Model) or meta-llama, a powerful language model capable of understanding and generating code.
Fine-tune the LLM on the collected dataset using transfer learning to adapt it to the specific task of code generation.
Employ techniques such as masked language modeling (MLM) or sequence-to-sequence (seq2seq) learning to train the model to generate code sequences.
Model Architecture Selection:

Choose an appropriate transformer architecture from Hugging Face's model repository based on the requirements of the project.
Consider models like GPT (Generative Pre-trained Transformer), BERT (Bidirectional Encoder Representations from Transformers), or T5 (Text-to-Text Transfer Transformer) depending on the complexity and scope of the code generation task.
Fine-tuning Hugging Face Transformer:

Fine-tune the selected Hugging Face transformer architecture on the preprocessed dataset.
Implement techniques such as transfer learning and domain-specific adaptation to improve the model's performance on code generation tasks.
Experiment with different hyperparameters and training strategies to optimize the model's performance.
Integration of Langchain and Hugging Face Transformers:

Integrating the fine-tuned LLM (meta-llama) with the fine-tuned Hugging Face transformer to leverage the strengths of both approaches.
Design a pipeline or workflow to effectively combine the capabilities of Langchain and Hugging Face Transformers for code generation.

IV.RESULTS

Metrics for evaluating code generation systems include code accuracy, diversity, benchmark performance, user feedback, generalization across languages and domains, scalability, real-world application impact, and robustness. These metrics assess aspects such as code similarity, variety across languages and paradigms, system performance against benchmarks, user satisfaction, versatility, scalability, practical utility, and error handling capabilities.

REFERENCES

[1] J. Qi, X. Wang and T. Yang, "Traditional Chinese Medicine Prescription Recommendation Model Based on Large Language Models and Graph Neural Networks," 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Istanbul, Turkiye, 2023

[2] Y. Fu et al., "GPT4AIGChip: Towards Next-Generation AI Accelerator Design Automation via Large Language Models," 2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD), San Francisco, CA, USA, 2023, pp. 1-9, doi: 10.1109/ICCAD57390.2023.10323953.

[3] R. J. Sunico, S. Pachchigar, V. Kumar, I. Shah, J. Wang and I. Song, "Resume Building Application based on LLM (Large Language Model)," 2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2023, pp. 486-492, doi: 10.1109/ICCCIS60361.2023.10425602.

[4] Y. Luo and I. Kobayashi, "BrainLM: Estimation of Brain Activity Evoked Linguistic Stimuli Utilizing Large Language Models," 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, Oahu, HI, USA, 2023,

[5] H. Strobelt et al., "Interactive and Visual Prompt Engineering for Ad-hoc Task Adaptation with Large Language Models," in IEEE Transactions on Visualization and Computer Graphics, vol. 29, no. 1, pp. 1146-1156, Jan. 2023

[6] W. Jiang et al., "An Empirical Study of Pre-Trained Model Reuse in the Hugging Face Deep Learning Model Registry," 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), Melbourne, Australia, 2023

[7] J. Castaño, S. Martínez-Fernández, X. Franch and J. Bogner, "Exploring the Carbon Footprint of Hugging Face's ML Models: A Repository Mining Study," 2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), New Orleans, LA, USA, 2023

[8] S. Ontoum and J. H. Chan, "Automatic Text Summarization of COVID-19 Scientific Research Topics Using Pre-trained Models from Hugging Face," 2022 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C), Bangkok, Thailand, 2022

[9] A. Singh, A. Ehtesham, S. Mahmud and J. -H. Kim, "Revolutionizing Mental Health Care through LangChain: A Journey with a Large Language Model," 2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA,2024,

[10] R. Asyrofi, M. R. Dewi, M. I. Lutfhi and P. Wibowo, "Systematic Literature Review Langchain Proposed," 2023 International Electronics Symposium (IES), Denpasar, Indonesia, 2023,.