Cloud Computing Project:

Antonis Louca*
Panayiotis Papadopoulos†
Andreas Chrisanthou‡
louca.antonis@ucy.ac.cy
papadopoulos.m.panagiotis@ucy.ac.cy
chrysanthou.m.andreas@ucy.ac.cy

ABSTRACT

Developing a generative AI-powered chatbot tailored for Fog/Edge Emulation deployments represents an exciting convergence of cutting-edge technology. This endeavor draws inspiration from the capabilities of advanced Generative AI models such as ChatGPT, which excel in comprehending and generating human-like text. The primary objective here is to harness the potential of these models to streamline and enrich the process of configuring Fog/Edge Emulation systems. To achieve this, students will design a user-friendly API that allows clients to submit their requirements for a Fog Computing infrastructure. The system will then extract relevant information from these inquiries and augment them with the corresponding context. This augmentation will be achieved through the implementation of prompt engineering techniques and in-context learning. Subsequently, the system will transmit these enhanced queries to a powerful large language model (LLM), such as the ChatGPT API, and relay the LLM's responses back to the client. Students will consider Fogify as the underlying emulation engine, and to facilitate their prompt engineering, they will utilize the modeling abstractions provided by Fogify's dedicated documentation page.

ACM Reference Format:
Antonis Louca, Panayiotis Papadopoulos, and Andreas Chrisanthou. 2023. Cloud Computing Project: . In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The contemporary surge in the capabilities of generative AI models, exemplified by GPT, has catalyzed significant advancements. This paper outlines the development of an AI-powered chatbot tailored for Fog/Edge emulation, propelled by the synergy of cutting-edge AI models and the critical role played by the Fogify tool. Central to this initiative is the design of a user-friendly API enabling clients to articulate Fog Computing infrastructure requirements. Leveraging prompt engineering and contextual learning, our system refines user queries, subsequently interfacing with a Large Language Model (LLM) like the ChatGPT API. The incorporation of Fogify as the primary emulation tool, guided by its documentation, ensures optimal outcomes during prompt engineering. This paper highlights the integration of AI technologies for streamlining Fog/Edge emulation processes.

2 BACKGROUND

3 METHODOLOGY

3.1 Terminology

Chroma DB is an open-source vector store, which provides the tools for building a knowledge base for LLMs

OpenAi: A service that allows developers to integrate advanced natural language processing capabilities into their applications

Fogify is an emulation Framework for modelling, deploying and experimenting with fog testbeds. Provides a toolset to model complex fog topologies

Python is a high-level programming language known for its readability and simplicity

LangChain is a framework designed to simplify creation of applications using large language models (LLMs)

3.2 Process we followed

A diagram of a chat

Description automatically generatedWe started our process by extracting the Fogify documentation. More precisely we read all the documents with extension .xml, .html, .md, yaml. Then those documents are segmented into chunks of 1000 characters. All those chunks form a list of document chunks.Then we need to transform this list into vectors. We achieve that by utilizing OpenAI in order to convert chunks to vector embeddings, those embeddings are stored in Chroma DB which is used for the persistent storage of the embeddings.After the above steps our tool is ready for use. More precisely, when the chat receives a user query, it searches the knowledge base in order to receive most relevant information (context) the we perform prompt engineering (context, user's question, memory of recent conversation messages between user and AI Model) and feed the prompy to AI model to get response

4 IMPLEMENTATION

5 CONCLUSIONS

Conference'17, July 2017, Washington, DC, USA
© 2023 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn