VIETNAM GENERAL CONFEDERATION OF LABOR
**TON DUC THANG UNIVERSITY**
**FACULTY OF INFORMATION TECHNOLOGY**

**FINAL REPORT**

**RESEARCH TOPIC 1**

# Developing Chatbot System for Retrieving TDTU Regulations Using the Retrieval-Augmented Generation Method

*Supervisor*: **LÊ ANH CƯỜNG**
*Student*: **CAO KHÁNH TÂN – 520K0342**
*Class*: **20K50301**
*Group*: **KHÁNH TÂN**
*Year*: **24**

**HO CHI MINH CITY, 2023**

VIETNAM GENERAL CONFEDERATION OF LABOR
**TON DUC THANG UNIVERSITY**
**FACULTY OF INFORMATION TECHNOLOGY**

**FINAL REPORT**

**RESEARCH TOPIC 1**

# Developing Chatbot System for Retrieving TDTU Regulations Using the Retrieval-Augmented Generation Method

*Supervisor*: **LÊ ANH CƯỜNG**
*Student*: **CAO KHÁNH TÂN – 520K0342**
*Class*: **20K50301**
*Group*: **KHÁNH TÂN**
*Year*: **24**

**HO CHI MINH CITY, 2023**

# THANK YOU

This report was finished thanks to the guidance of my lecturer, Lê Anh Cường. Without his assistance, I would not have been able to finish the report in the allotted time. I also want to express my gratitude to my university, Ton Duc Thang, for having given me a chance to research computer science in general, and in particular, the Retrieval -Augmented Generation method in AI.

# PROJECT COMPLETED
# AT TON DUC THANG UNIVERSITY

I hereby declare that this is my/our own project and is under the guidance of Lê Anh Cường. The research contents and results in this topic are honest and have not been published in any publication before. The data in the tables for analysis, comments and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

In addition, the project also uses a number of comments, assessments as well as data of other authors, other agencies and organizations, with citations and source annotations.

If I find any fraud, I will take full responsibility for the content of my project. Ton Duc Thang University is not related to copyright and copyright violations caused by me (if any).

*Ho Chi Minh, ..........................*

*Author*

*(Sign)*

*Cao Khánh Tân*

# VERIFICATION AND EVALUATION OF LECTURER

**Verification of guiding lecturer**

*Ho Chi Minh, ...........................*

(Sign)

**Evaluation of grading lecturer**

*Ho Chi Minh, ...........................*

(Sign)

## SUMMARY

This essay represents detailly the theories and methodologies related to solving the tasks given in the Final Report given by the lecturer.

# TABLE OF CONTENT

# CHAPTER 1 – INTRODUCTION

In the ever-evolving landscape of natural language processing (NLP) and artificial intelligence, the quest for generating human-like and contextually relevant text has been a persistent challenge. Traditional language models, while proficient in generating coherent sentences, often struggle with capturing nuanced information or retrieving contextually relevant details from vast amounts of data. This limitation becomes particularly evident in tasks that require a deep understanding of specific domains or the incorporation of diverse and dynamic information.

The need for more sophisticated approaches to text generation has given rise to a paradigm known as Retrieval Augmented Generation (RAG). Retrieval Augmented Generation represents a significant leap forward in NLP by addressing the shortcomings of conventional language models. By integrating retrieval mechanisms into the generation process, RAG seeks to enhance the contextual awareness and factual accuracy of generated text. This innovative approach aims to bridge the gap between traditional generation models and the rich knowledge repositories available, enabling a more informed and contextually grounded generation of content.

This report delves into the fundamental concepts of Retrieval Augmented Generation, exploring its theoretical underpinnings, applications, and the pivotal role it plays in advancing the state-of-the-art in natural language processing.

# CHAPTER 2 – RETRIEVAL-AUGMENTED GENERATION

## 2.1 Retrieval Mechanism

At the core of Retrieval Augmented Generation (RAG) lies a sophisticated retrieval mechanism designed to empower language models with access to vast knowledge repositories. Traditional language models often operate in isolation, lacking the ability to tap into external information sources effectively. The retrieval mechanism in RAG, however, introduces a dynamic interplay between generation and retrieval, allowing the model to draw upon external knowledge during the content creation process.

The retrieval mechanism operates by leveraging pre-existing knowledge bases or document collections. These can range from structured databases to unstructured textual corpora, depending on the specific application. Key to the effectiveness of this mechanism is the integration of efficient information retrieval techniques, such as dense vector embeddings or inverted index structures, which enable the model to quickly access and assimilate relevant information.

One prevalent implementation of the retrieval mechanism involves a two-step process. In the first step, the model identifies potential candidate documents or passages from the knowledge base that are pertinent to the given context. Subsequently, in the second step, the language model generates content by synthesizing information from both the retrieved knowledge and its internal understanding, resulting in a more contextually enriched output.

The advantages of incorporating a retrieval mechanism into the generation process are manifold. Firstly, it enables the model to produce content grounded in factual accuracy and relevance by tapping into a vast array of external information. Additionally, this approach allows the model to adapt to a broader range of topics and domains, even those beyond the scope of its pre-training data.

## 2.2 Embedding Model

In the realm of Retrieval Augmented Generation (RAG), the embedding process serves as a crucial intermediary step that transforms raw textual information into dense vector representations. This transformation is essential for facilitating efficient retrieval of relevant knowledge and comparing the input context with the information stored in the vector repository.

The embedding process typically involves converting words, phrases, or entire documents into high-dimensional vectors. This conversion is guided by pre-trained embedding models, such as Word2Vec, GloVe, or more sophisticated transformer-based embeddings like BERT or GPT. The resulting vectors encode semantic relationships and contextual information, providing a more nuanced representation of the input text.

Before incorporating information into the vector store, the embedding process is applied to the individual elements, ensuring that the knowledge base is efficiently represented in a vector space. This vectorized knowledge repository becomes the foundation for the subsequent retrieval mechanism, allowing the model to quickly identify and extract contextually relevant information during the generation process.

Simultaneously, the input context, which could be a query or a partial sentence, is also embedded into a dense vector representation. This embedding of the input context allows for seamless comparison with the vectors stored in the repository using similarity metrics, most commonly cosine similarity. The goal is to identify and retrieve the most relevant information based on the closeness of vectors in the semantic space.

The use of embeddings in RAG is not only pivotal for optimizing the efficiency of the retrieval mechanism but also for enhancing the overall contextual coherence of the generated text. By representing both the input and the knowledge base in a shared vector space, the model can draw upon contextual relationships and semantic similarities, resulting in more informed and contextually grounded content generation.

## 2.3 Language Model

The language model is the driving force behind the creative and coherent generation of text in the context of Retrieval Augmented Generation (RAG). In this paradigm, the use of a sophisticated language model, such as LLAMA2 (Language Model for Retrieval Augmented Generation 2), represents a key component that integrates the information retrieved from the vector store with the user's query to produce contextually relevant and fluent output.

LLAMA2, like other state-of-the-art language models, is trained on vast amounts of diverse textual data and possesses the ability to understand and generate human-like text. Its uniqueness lies in its capacity to leverage the retrieved information and user queries during the generation process, thereby enhancing the contextual relevance and factual accuracy of the output.

In the RAG method, the input to LLAMA2 typically consists of two main components: the information retrieved from the vector store and the user's query. The retrieved information, encoded as dense vectors, contains contextually relevant knowledge from the external knowledge base. Simultaneously, the user's query provides specific input or context, guiding the language model in tailoring its output to address the user's intent.

LLAMA2, by processing this composite input, is adept at synthesizing information from both the retrieved knowledge and the user query. It leverages the semantic relationships encoded in the vectors, enabling it to generate content that seamlessly integrates external facts with the user's input. The result is a coherent and contextually aware response that reflects a deep understanding of the query and draws upon the richness of the retrieved information.

# CHAPTER 3 – OUR RAG IMPLEMENTATION

In this chapter, we delve into the specific implementation details of the Retrieval Augmented Generation (RAG) framework, showcasing how the integration of key components such as the Hugging Face pretrained embedding model, LLM (Language Model), LangChain, and PineCone vector storage contributes to the efficiency and effectiveness of the overall system.

We use two huggingface pretrained models: a LLM LLama2 and an embedding model. There is no need to fine tune because the capabilities of these models in Vietnamese are sufficient.

LangChain is integrated into our RAG implementation to facilitate the orchestration of information flow between the various components. This component manages the communication between the Hugging Face embedding model, LLM, and the PineCone vector storage, ensuring a smooth and efficient workflow. LangChain acts as the glue that enables the seamless interaction between the different elements, contributing to the overall cohesion and effectiveness of the RAG system.

PineCone is employed as the vector storage solution, serving as the repository for the dense vector representations of the retrieved information. PineCone's efficient indexing and retrieval capabilities play a crucial role in speeding up the information retrieval process. The integration with PineCone ensures that the RAG system can quickly access and assimilate contextually relevant knowledge during the generation process, enhancing both efficiency and accuracy.

Workflow overview:

1. **Embedding with Hugging Face:** Raw text, including both user queries and retrieved information, undergoes embedding using the Hugging Face pretrained model, converting it into dense vector representations.
2. **Vector Storage with PineCone:** The resulting vectors are stored in PineCone, forming a dynamic and efficient knowledge repository.
3. **LangChain Orchestration:** LangChain manages the interaction between components, facilitating seamless communication between the Hugging Face embedding model, LLM, and PineCone vector storage.
4. **LLM Generation:** LLM processes the composite input of user queries and retrieved information, synthesizing contextually rich and coherent text as the final output.

This implementation architecture underscores the synergy between powerful embedding models, advanced language models, efficient communication mechanisms, and optimized vector storage, collectively contributing to the success of the RAG framework.

## REFERENCES

1. https://acl2023-retrieval-lm.github.io/
2. https://arxiv.org/abs/2005.11401