Ask Nuclear question & Answer, Generative Artificial Intelligence Information Retrieval System using Retrieval Augmented Generation

Seung-Hyeok Yang[a], Sang-Beom Kang[a], Han-Gil Lee[a], Dae-Young Lee[a*]
[a]*FNC Technology Co., Ltd., 13 Heungdeok 1-ro, 32F, Giheung-gu, Yongin-si, Gyeonggi-do, 16954, Korea*
[*]*Corresponding author: ldy242@fnctech.com*

Keywords : Natural Language Processing, Artificial Intelligence, QA model, Retrieval Augmented Generation

1. Introduction

The nuclear industry has embarked on a journey of digital transformation in recent times, leading to the accumulation of vast amounts of information. Therefore, there arises a critical need to sift through this wealth of data effectively and extract pertinent information.

In this paper, we introduce ANNA (Ask Nuclear question & Answer), a system designed to enhance reliability by leveraging Retrieval-Augmented Generation (RAG) to construct a generative artificial intelligence search system. ANNA operates by first identifying the document that most closely aligned with the user's search query within a designated directory. In other words, it conducts vector embedding within the selected document to pinpoint relevant text. Following this, ANNA employs the cosine similarity to extract the top N sentences that exhibit high correlation and the KoAlpaca model to generate accurate answers. Through this innovative approach, users gain access to a system that streamlines information retrieval, facilitating a more efficient and accurate understanding of documents.

This study proposes a novel application of natural language processing technology combined with artificial intelligence, offering a transformative approach to enhancing information search and comprehension efficiency. Furthermore, our system demonstrates exceptional performance in answer generation through Retrieval-Augmented Generation, promising significant enhancements in user experience and overall usability.

2. Background Knowledge

In this section, we will explore the fundamental concepts necessary for configuring our model.

2.1 Text Similarity Analysis

There are two fundamental methods for assessing similarity between vectorized documents and sentences.

Euclidean distance method calculates the arithmetic distance between vectors of the same dimension. A smaller Euclidean distance indicates closer proximity between documents, suggesting their similarity.

Cosine similarity, widely utilized, yields values ranging from -1 to 1. Closer proximity to 1 signifies higher similarity. Unlike Euclidean distance, cosine similarity emphasizes vector direction, facilitating fair comparisons even when sentences vary in length.

2.2 Language Model

A language model assigns probabilities to word sequences, such as sentences, to capture language phenomena. Essentially, it captures the most natural word order. Language models can be created using statistical methods or artificial neural networks. Recently, neural network-based approaches have demonstrated superior performance. These models include Generative Pre-trained Transformer (GPT), Large Language Model Meta AI (LLaMA), and KoAlpaca, which fine-tunes LLaMA using Korean data. Notably, the Large Language Model (LLM) is an extensive deep learning model trained on vast datasets. [4]

2.3 Retrieval-Augmented Generation (RAG)

RAG optimizes the output of a LLM by consulting reliable external knowledge beyond its training data before generating a response. Without RAG, a LLM generates responses solely based on its training or prior knowledge. This could potentially result in inaccurate responses or hallucinations if the model did not accurately learn information about the user input. By vectorizing documents related to user input and incorporating them, RAG facilitates accurate answer generation. To ensure the retrieval of current information, documents must be updated asynchronously, and their embedding representations refreshed. One significant advantage of RAG is its capability to produce accurate answers without the need for extensive fine-tuning of the LLM, which typically requires substantial resources. [1][2]

3. Model Configuration

Navigating through numerous documents to find the desired information can be daunting. While searching for exact word matches is straightforward, it is often uncommon to seek information based on precise terminology. Hence, in this section, we present ANNA (Ask Nuclear question & Answer), a model designed to generate answers when users inquire about information stored within documents.

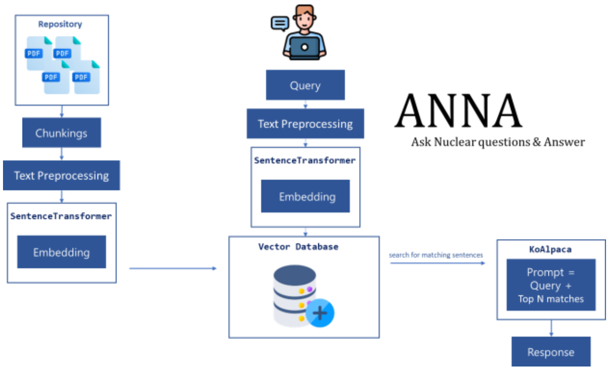

Fig. 1. Schematic Diagram of ANNA

3.1 Semantic Search with Document Embedding

Each preprocessed sentence undergoes vectorization, known as embedding, which enables computers to comprehend the content of the document. Through this process, known as semantic search, similarity to the user’s question can be calculated. Unlike conventional search systems that rely solely on lexical matches, semantic search accommodates synonyms. Document embedding facilitates the calculation of semantically similar words or expressions with high similarity, providing an advantage over systems that rely solely on exact term matches. For instance, ‘Nuclear Regulatory’ can be searched similarly to ‘Nuclear Regulation’ or ‘Nuclear Restriction’.

Post document embedding, the user’s question is also embedded using the same method, and the cosine similarity between the question and document sentences is assessed. The top N sentences, which exhibit the highest cosine similarity, are selected as key response. The process can be outlined as follows: [2][3]

a. Embed each document sentence by sentence within the designated directory.
b. Embed the user’s question text.
c. Calculate the cosine similarity between the question embedding and sentence embeddings, and select the top N sentences.

3.2 Generating Answers

Merely extracting sentences from documents may not provide optimal visibility nor constitute appropriate answers to questions. Thus KoAlpaca, a generative model, is utilized to produce natural responses using the top N sentences extracted as inputs. KoAlpaca, like other language models, may generate inaccurate responses for unfamiliar data, termed ‘hallucination’. However, RAG innovatively addresses this by enabling precise answers to user queries without additional training. [1][2][5]

4. Conclusions

After conduction a similarity analysis with minimal preprocessing of sentence embeddings, we observed satisfactory performance. Evaluation of the KoAlpaca model and its application with RAG revealed that KoAlpaca-only provided answers that aligned with common sense in areas related to professional knowledge. However, the utilization of KoAlpaca with RAG yielded more meaningful responses by incorporating sentences extracted through semantic search.

In instances where an abundance of technical terminology, as found in the nuclear field, is present, the pre-trained model may lack familiarity with the pertinent terms, potentially limiting its effectiveness. However, in the realm of artificial intelligence, a significant amount of development time is consumed by preprocessing and fine-tuning processes, which are often compounded by challenges in data acquisition. Remarkably, our proposed model demonstrates commendable performances with RAG without requiring extensive fine-tuning, despite minimal preprocessing. Presently, there is a growing interest in small LLMs (sLLM) customized for specific fields, utilizing the LLM structure effectively. In the nuclear industry, enhanced performance is anticipated through the collection of sufficient data, which aids in the development of specialized sLLM for the nuclear sector.

ACKNOWLEDGMENTS

This work was supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (RS-2023-00259343, Company-wide DX Promotion Technology Development)

REFERENCES

[1] Yunfan Gao, et al., Retrieval-Augmented Generation for Large Language Models: A Survey, 2023.
[2] Patrick Lewis, et al., Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, 2020.
[3] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, 2019.
[4] Hugo Touvron, et al., Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023
[5] Rohan Taori, et al., Stanford Alpaca: An Instruction-following LLaMA model, 2023.