**Enhancing RAG with HyDE(Hypothetical Document Embedding)**

Introduction

Retrieval Augmented Generation (RAG ) is replacing the traditional search-based approaches and creating a chat with a document environment. Since the inception of RAG, various methods have been proposed to enhance the standard RAG approach. The biggest hurdle in RAG is to retrieve the right document. Only when we get the right documents, the LLM will be able to generate the right answers. In this guide, we will be talking about HyDE(Hypothetical Document Embedding), an approach that was created to improve the Retrieval in RAG.

Table of contents

1. Challenges in RAG Implementation
2. What are Hypothetical Document Embedding(HyDE) ?
3. Features of Hypothetical Document Embedding(HyDE)
4. Challenges and Limitations of HyDE
5. Conclusion

**1.Challenges in RAG Implementation**

Retrieval Augmented Generation is very popular and right now it is widely used. A simple RAG(Retrieval Augmented Generation) involves taking in raw text, chunking it into smaller pieces, creating embeddings for all the chunks, and storing the embeddings in a vector store. Then when a user provides a query, we compare the similarity between the user query and the chunks and retrieve the similar chunks. Finally, the user query along with similar chunks is sent to the Large Language Model to generate the final answer. This is the regular Retrieval Augmented Generation.

A diagram of a diagram

Description automatically generated

This regular and plain Retrieval Augmented Generation has many flaws. Starting with the chunking itself. There is no one size to chunking. The size of chunking documents largely depends on the type of Large Language Models we are working with and sometimes we have to try a bunch of sizes to get better results. Then comes the Retrieval, the main focus for this guide.

The RAG was developed to prevent the Large Language Models from hallucination. This largely depends on the similar information retrieved through the user query from the vector stores. If the Retrieval is not good, then the Large Language Model will either hallucinate or will not respond to the question provided by the user. One way to improve the Retrieval is Hypothetical Document Embeddings.

**2.What are Hypothetical Document Embedding(HyDE) ?**

Hypothetical Document Embeddings (HyDE) is one of the transformative solutions to tackle poor Retrievals faced in RAG-based solutions. As the name suggests, HyDE works by generating Hypothetical Documents, which will help in better retrieval of similar documents so that the Large Language Model can take these inputs and generate a better answer.

Let’s understand HyDE with the below diagram:

A diagram of a document embedding

Description automatically generated

The first step involves taking in a user query. Now in a normal RAG system, we convert the user query into embeddings and send it to the vector store to retrieve similar chunks. But in Hypothetical Document Embeddings, we take in the user query and then pass it to a Large Language Model to generate a Hypothetical Answer to the question. So the LLM takes in the user question and tries to generate a fake Hypothetical Answer/Document with similar textual patterns from the first user query. We then convert this Hypothetical Document into embedding vectors and then use these embeddings to retrieve similar chunks from the vector store. Finally, we bind these similar chunks to the original query and pass on these together to LLM to generate the final answer.

So what we are trying to do here is, that instead of trying to perform a query to answer embedding vectors similarity, we are trying to perform an answer to answer embedding vectors similarity so that it yields better results.

**3.Features and Benefits of Hypothetical Document Embedding(HyDE)**

* **Enhanced Retrieval Accuracy:**HyDE introduces a new approach where Hypothetical Answers/Documents are created based on the user queries, allowing for a more nuanced understanding of search intent beyond keywords. Thus encoding them to embedding vectors will really improve the retrieval systems in finding more semantically relevant chunks.
* **Reduced Hallucinations:**We have discussed that the RAG was introduced to mitigate LLM Hallucinations. These will be based on the retrieved context passed to the LLM, so giving them in wrong and not meaningful chunks to the LLM will result in hallucinations thus generating wrong answers. HyDE through its hypothetical documents will try to fetch the best relevant chunks thus reducing the chances of hallucinations.

The HyDE retrieval offers several benefits:

* **Zero-Shot Retrieval**: It effectively retrieves relevant documents without needing relevant labels or prior training on specific datasets.
* **Generative Approach**: Generating hypothetical documents captures relevance patterns even if the details are inaccurate.
* **Versatility**: It performs well across various tasks, such as web search, question answering, and fact verification, and supports multiple languages.

**4.Challenges and Limitations of HyDE**

Hypothetical document retrieval comes with some challenges and limitations. Here are a few of them:

* **Knowledge Bottleneck**: HyDE-generated documents may contain factual errors and are not real, which could affect the accuracy of retrieval results. For example, this approach may be ineffective if the topic is completely new to the language model. It could result in more frequent instances of generating incorrect information.
* **Multilingual Challenge**: Multilingual retrieval poses several additional challenges to HyDE. The small-sized contrastive encoder gets saturated as the number of languages scales. Meanwhile, generative LLM faces an opposite issue: with languages not as high-resource as English or French, the high-capacity LLM can get under-trained.

**5.Conclusion**

HyDE significance is important for Natural Language Processing (NLP) as it finds relevant documents without prior training or labels. It uses hypothetical documents to capture relevance and excels in tasks like web search and question answering across multiple languages.its improving retrieval accuracy and reducing hallucinations, thereby contributing to more reliable and contextually relevant responses from Large Language Models (LLMs).

[**https://www.analyticsvidhya.com/blog/2024/04/enhancing-rag-with-hypothetical-document-embedding/**](https://www.analyticsvidhya.com/blog/2024/04/enhancing-rag-with-hypothetical-document-embedding/)

[**https://zilliz.com/learn/improve-rag-and-information-retrieval-with-hyde-hypothetical-document-embeddings**](https://zilliz.com/learn/improve-rag-and-information-retrieval-with-hyde-hypothetical-document-embeddings)

[**https://www.pondhouse-data.com/blog/advanced-rag-hypothetical-document-embeddings**](https://www.pondhouse-data.com/blog/advanced-rag-hypothetical-document-embeddings)

RAG utilizes two core components: a generator, typically an LLM, and a retriever akin to a vector database. Here’s how HyDE improves RAG pipelines:

1. Generating Hypothetical Documents: HyDE’s innovation lies in generating a hypothetical document based on the query and retrieving documents based on it. So, instead of directly relying on retrieved documents from the corpus, HyDE uses this generated document to capture the essence of relevance.
2. Answering Hard Questions: When encountering a vague or contextually ambiguous question, deriving a precise answer can be tricky. HyDE improves it by enriching the query with more context with the help of LLM.
3. Optimizing Document Queries: Since most databases contain answers rather than questions, it makes sense to use a hypothetical answer as the query for documents.

Experiments demonstrated the improvements in performance, robustness, and versatility:

1. Improved Performance: HyDE consistently outperforms classical BM25 and unsupervised Contriever across various datasets and metrics, e.g., nDCG@10, recall.
2. Robustness: HyDE remains competitive even against fine-tuned models on richly supervised tasks like TREC DL19/20.
3. Versatility: It demonstrates strong performance across both web search and low-resource tasks, providing significant gains over baseline models.
4. Multilingual Capabilities: It shows enhanced results in multiple languages, outperforming mContriever in languages such as Korean and Japanese.
5. Efficiency: HyDE enhances retrieval quality without requiring extensive fine-tuning, making it an effective and efficient choice for diverse retrieval tasks.

The HyDE for the RAG pipeline can do more harm than good if its limitations are unknown. The next section will shed light on it.