NLP Project

Christopher Lindenberg
Jatin Jatin
Hao Zhang
Yulin Liu

Group 36

Contents

Abstract

Our group project involves creating a QA system using machine learning and NLP techniques to extract information from PubMed publications. The goal is to simplify data navigation without requiring specialized knowledge. We use modern NLP models, including Text-Embedding Models and Open Source LLMs, to enable semantic search and automated question answering on a subset of PubMed abstracts. Unlike traditional keyword-based searches, our system uses natural language queries, leveraging semantic text embeddings for efficiency. The project focuses on data acquisition, preprocessing, and integrating retrieval-augmented generation (RAG) and question-answering (QA) systems to provide users with an intuitive way to access well-sourced information in a specific area of medical research.

0.1 Motivation

Our group project focuses on simplifying access to recent medical information for both researchers and individuals with specific medical conditions. Recognizing the challenge of information overload, especially with vast literature sources like research articles and online blogs, our effort aims to automate the extraction and summarization of relevant content from PubMed publications. This tool intends to assist users in navigating through the data, emphasizing efficiency without overwhelming them.

1. **Tackling Information Overload:**
   - We understand the struggle of users sorting through extensive data. Our project automates the extraction and summarization of pertinent information from PubMed, providing a valuable tool for users to access reliable information.
   - Imagine a tool that transforms intricate research into concise, useful chunks, allowing users to focus on well-sourced information.
2. **Boosting Research Efficiency:**
   - Our system automates essential information extraction, saving researchers significant time and resources previously spent on manual reviews.
   - Beyond efficiency, our project aims to discover hidden connections, recognize patterns, and present new trends, facilitating easy cross-referencing of various studies.
3. **Utilizing NLP:**
   - Modern NLP models such as state of the art text embedding models and open source LLMs like Mistral Jiang et al. [2023] are part of our armory. The embedding models create rich, context-aware embeddings from unprocessed text, while the LLMs can take on low-complexity text generation tasks.
   - Incorporating modern NLP models like text embedding models and open-source LLMs, such as Mistral, our project simplifies semantic searching. Users can explore our chosen subset of PubMed's repository for streamlined navigation through medical research.

The project gives users advanced text analysis capabilities by integrating cutting-edge NLP models.

0.2 Problem Definition

In the realm of medical research and healthcare decision-making, ineffective search techniques often hinder the retrieval of pertinent literature from databases like PubMed.

Conventional keyword-based searches prove time-consuming and resource-intensive for researchers, medical experts, and individuals seeking medical information, as they often yield too many or irrelevant results. This challenge underscores the urgent need for a semantic search engine tailored for medical document retrieval.

0.2.1 Ideal Scenario

- In an ideal scenario, this semantic search engine would offer intuitive natural language querying capabilities, facilitating rapid and precise retrieval of relevant medical documents from PubMed. Users would benefit from comprehensive search results encompassing abstracts, author information, titles, and URLs, instilling confidence in the relevancy of retrieved documents. Such a solution would empower healthcare professionals to make informed decisions, propel medical research forward, and ultimately enhance patient outcomes.

0.2.2 Costs of Inefficiency

- Retrieving medical literature in an inefficient manner poses numerous costs. Medical researchers and practitioners spend substantial time manually searching for medical information, diverting time from patient care or research activities. Inefficient access to medical literature can lead to delays in patient diagnosis, treatment, and outcomes, impacting healthcare quality and resource allocation.

- Moreover, inefficiencies in accessing pertinent medical literature exacerbate issues with study reproducibility, hindering collaboration on new ideas and knowledge dissemination. By addressing these costs and challenges, the creation of an effective semantic search engine for medical papers holds the potential to significantly enhance access to medical knowledge, improve patient care, and advance research outcomes.

0.3 Model architecture

0.3.1 Retrieval-Augmented Generation (RAG) System

The Retrieval-Augmented Generation (RAG) system combines the strengths of retrieval-based and generation-based approaches to provide accurate and informative responses to user queries. The architecture consists of the following components:

Retriever Component

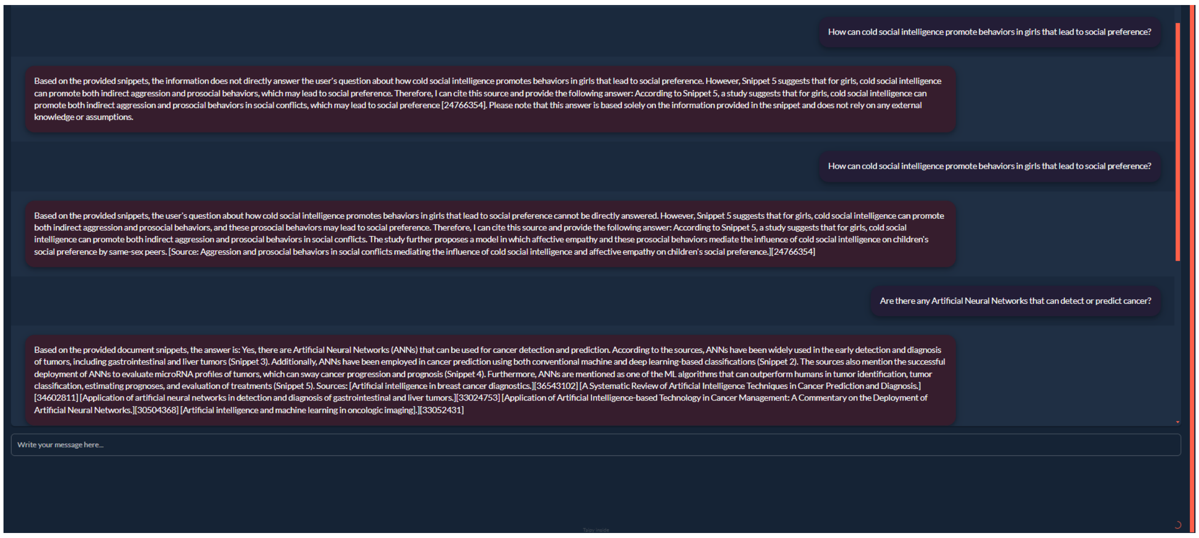
The retriever component is responsible for retrieving relevant information from a knowledge base in response to user queries. It typically consists of the following modules:

- **Preprocessing Module**: This module preprocesses the input query and knowledge base to extract relevant features and representations.
- **Retrieval Module**: The retrieval module performs the actual search and retrieval process, identifying documents or passages that are most relevant to the input query.

Generator Component

The generator component is responsible for generating coherent and informative responses based on the retrieved information. It typically consists of the following modules:

- **Tokenization Module**: This module tokenizes the input query and retrieved passages, converting them into a format suitable for input to the language model. In our project, we use a pretrained all-MiniLM-L6 -v2 model as our tokenizer.
- **Generation Module**: The generation module utilizes a large-scale language model such as GPT (Generative Pre-trained Transformer) to generate natural language responses based on the tokenized input. We prepared 3 different preetrained models: Llama-2-7b-chat-hf **?**, Mistral-7B-Instruct-v0.2 Jiang et al. [2023], meditron-7b **?**, as well as a finetuned version of the 3 models. We pick Mistral-7B-Instruct-v0.2 as our inference model, during the evaluation part, we found that the finetuned model works worse than the original model (in the evaluation file Mistral-7B-Instruct-v0.2_finetuned_backup.json in our github repo, It tends to generates useless text "I'm here to provide information based on the provided document snippets. If you have any medical questions, feel free to ask!"). We pick the original Mistral-7B-Instruct-v0.2 as our inference model.
- **Post-processing Module**: Once the response is generated, the post-processing module performs any necessary formatting or adjustments to ensure that the final output is coherent and readable.



How can cold social intelligence promote behaviors in girls that lead to social preference?

Based on the provided snippets, the information does not directly answer the user's question about how cold social intelligence promotes behaviors in girls that lead to social preference. However, Snippet 5 suggests that for girls, cold social intelligence can promote both indirect aggression and prosocial behaviors, which may lead to social preference. Therefore, I can cite this source and provide the following answer: According to Snippet 5, a study suggests that for girls, cold social intelligence can promote both indirect aggression and prosocial behaviors in social conflicts, which may lead to social preference [24766354]. Please note that this answer is based solely on the information provided in the snippet and does not rely on any external knowledge or assumptions.

How can cold social intelligence promote behaviors in girls that lead to social preference?

Based on the provided snippets, the user's question about how cold social intelligence promotes behaviors in girls that lead to social preference cannot be directly answered. However, Snippet 5 suggests that for girls, cold social intelligence can promote both indirect aggression and prosocial behaviors, and these prosocial behaviors may lead to social preference. Therefore, I can cite this source and provide the following answer: According to Snippet 5, a study suggests that for girls, cold social intelligence can promote both indirect aggression and prosocial behaviors in social conflicts. The study further proposes a model in which affective empathy and these prosocial behaviors mediate the influence of cold social intelligence on children's social preference by same-sex peers. [Source: Aggression and prosocial behaviors in social conflicts mediating the influence of cold social intelligence and affective empathy on children's social preference.][24766354]

Are there any Artificial Neural Networks that can detect or predict cancer?

Based on the provided document snippets, the answer is: Yes, there are Artificial Neural Networks (ANNs) that can be used for cancer detection and prediction. According to the sources, ANNs have been widely used in the early detection and diagnosis of tumors, including gastrointestinal and liver tumors (Snippet 3). Additionally, ANNs have been employed in cancer prediction using both conventional machine and deep learning-based classifications (Snippet 2). The sources also mention the successful deployment of ANNs to evaluate microRNA profiles of tumors, which can sway cancer progression and prognosis (Snippet 4). Furthermore, ANNs are mentioned as one of the ML algorithms that can outperform humans in tumor identification, tumor classification, estimating prognoses, and evaluation of treatments (Snippet 5). Sources: [Artificial intelligence in breast cancer diagnostics.][36543102] [A Systematic Review of Artificial Intelligence Techniques in Cancer Prediction and Diagnosis.][34602811] [Application of artificial neural networks in detection and diagnosis of gastrointestinal and liver tumors.][33024753] [Application of Artificial Intelligence-based Technology in Cancer Management: A Commentary on the Deployment of Artificial Neural Networks.][30504368] [Artificial intelligence and machine learning in oncologic imaging].][33052431]

Figure 0.1: UI interface

Integration

The QA system integrates our retrieval module made up of our database implementation joined with our embedding model together with our generated, which as stated above is the vanilla Mistral-7B-Instruct-v0.2. It aims to provide accurate answers to medicine based user queries. The model recieves a hand crafted system message, aswell as the user context, which is made up of the user query and the retrieved and formatted documents. All of this is tokenized and fed into the QA model, is supposed to infer a relevant answer based on the provided snippet, or state that that's not possible in case that happens. The post-processing module then extracts and formats the answer for presentation to the user. Finally, we build a easy-to-use UI interface for the users.

1 Data description and acquisition

1.1 Data Description

1.1.1 Data Source

The dataset utilized in this project originates from the PubMed database, an extensive collection of biological literature.

1.1.2 Data Structure

The JSON files in the dataset are arranged according to the year of publication (2013 to 2023). Every JSON file includes summaries taken from health-related literature pertaining to the concept of "intelligence."

1.1.3 Information About Metadata

The following metadata are included for every abstract:

- **Article ID**: Unique identifier for the research article.
- **Title**: Title of the article.
- **Authors**: List of authors who contributed to the article.
- **URL**: PubMed URL linking to the full article.
- **Abstract Text**: A summary of the research findings found in the abstract.

1.2 Data Acquisition Plan

1.2.1 Collection Process

The data retrieval process involves querying the PubMed database using the NCBI E-utilities API. The query term used is "intelligence," and data is retrieved for each year from 2013 to 2023.

1.2.2 Collection Parameters

- Database: PubMed
- Query Term: "intelligence"
- Collection Period: Each year from 2013 to 2023

1.2.3 Data Storage

Retrieved data is stored in JSON format, with separate files for each year. Each JSON file contains a list of dictionaries representing individual abstracts with associated metadata.

1.2.4 Preprocessing

There is no preprocessing done on a text level before embedding.

1.2.5 Embedding Generation

First we tokenize each abstract individually with the tokenizer to our chosen embedding model. The model we chose here is "all-MiniLM-L6-v2" (https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2), a very light weight embedding model on the basis of Distilbert [Sanh et al. [2019]]. After tokenization, if the tokenized abstract is too long for the maximum context length of our model, we chunk the tokenized abstracts according to a minimum percentage overlap. These chunks are then embedded by our embedding model individually. For each chunk we then produce an object, containing it's embedding vector, the abstract it was chunked out of, and the articles associated metadata.

1.2.6 Storage and Indexing

We have two solutions available for storing and searching our embedding vectors. Both implement the interface "retrieve_by_query" and are thus dropin replacements for each other. The first is a Pinecone Database wrapper, which queries our Pinecone cloud storage. The second one is a basic Numpy storage, which implements similary search by implementing cosine similarity search by hand.

Why two solutions? We've benchmarked the databases against each other and found that the Numpy Database is both consistently a lot faster for the scale that we are working on (about 20000 documents) and also a lot less cumbersome to implement, and even having fewer requirements. Pinecone for example doesn't allow arbitrary sizes of metadata, forcing us to truncate some abstracts that we uploaded. Pinecone on the other hand is more convenient and requires less setup when running our pipeline, since the user has to manually run a file that embeds and stores the data that the numpy database needs to access, this is of course already taken care of in a cloud solution.

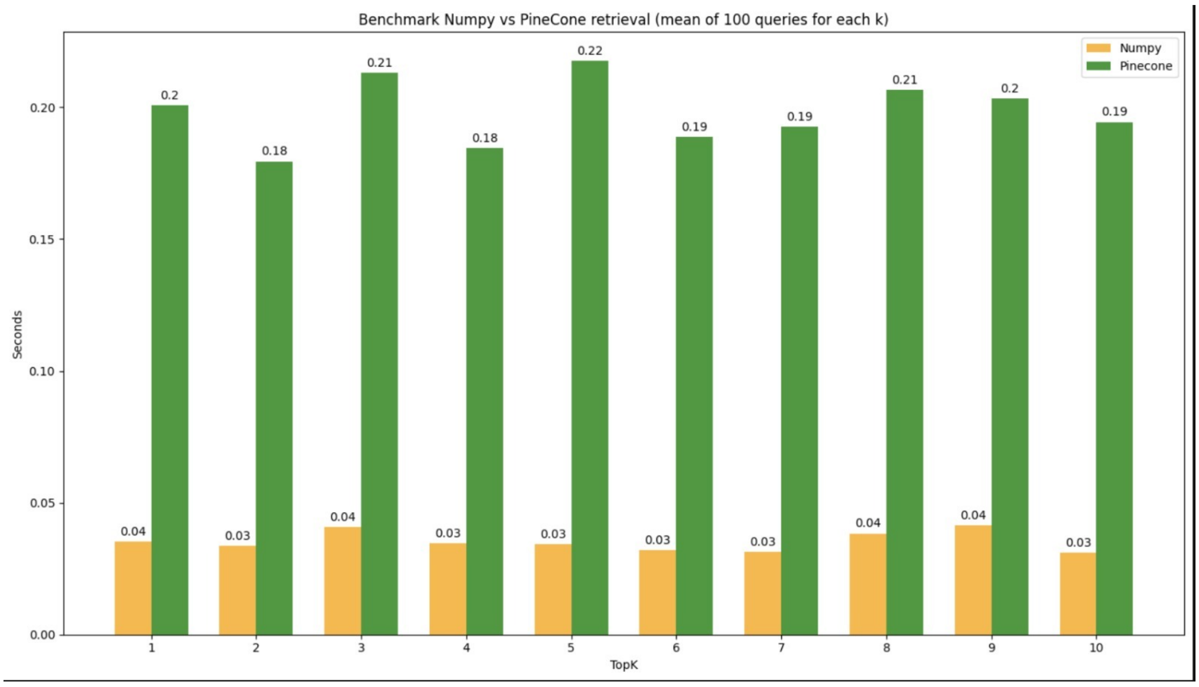


Figure 1.1: A benchmark of our Numpy Database and our Pinecone Database.

2 Evaluation and implementation

2.1 RAG Evaluation and Implementation

In this section, we delve into the evaluation and implementation of the Retrieval-Augmented Generation (RAG) system. RAG evaluation comprises two essential components: retrieval evaluation and response evaluation. We will discuss each aspect in detail, including their implementation steps and considerations.

2.1.1 Retrieval Evaluation

Retrieval evaluation aims to assess the accuracy and relevance of the information retrieved by the RAG system. The following metrics are commonly used for retrieval evaluation:

- **Hit Rate**: Hit rate measures the frequency with which the RAG system provides the correct answer within the top few guesses. It indicates the system's ability to retrieve relevant information effectively.
- **Mean Reciprocal Rank (MRR)**: MRR calculates the average of the reciprocals of the ranks of correct answers across all queries. It provides a more nuanced understanding of retrieval performance by considering the order of correct answers.

Implementation and result:
We computed the Hit rate and MRR, and the results are shown in the following table 2.1. The hit rate is quite while the MRR is only good enough when top-k=1. While the is not high, it has no significant impact on the system's performance, as RAG system equally utilizes the extracted top K snippets. Therefore, for the RAG system alone, Hit rate is more crucial.

top-k	1	2	3	4	5
Hit rate	0.8584	0.9159	0.9292	0.9336	0.9557
MRR	0.8584	0.5044	0.3643	0.2905	0.2652

Table 2.1: Caption

2.1.2 Response Evaluation

Response evaluation focuses on assessing the quality and appropriateness of the responses generated by the RAG system based on the retrieved information. There are many traditional metrics and benchmarks used to evaluate question-answering systems, such as ROUGE and BLUE, but they have poor correlation with human judgment. Therefore, we ultimately decided to construct a test set by generating question-answer pairs and evaluate them using the following two key evaluators:

- **Faithfulness Evaluator**: This evaluator measures whether the response from the RAG system matches any source context. It helps detect instances where the system generates inaccurate or hallucinated responses.
- **Relevancy Evaluator**: The relevancy evaluator assesses whether the response, along with the source context, adequately addresses the query posed to the system. It ensures that the generated response is relevant and informative.

Implementation and result: We utilized mistralai/Mixtral-8x7B-Instruct-v0.1 for question-and-answer (QA) pair generation. Taking into account the considerations of evaluation costs and time, we ultimately generated 226 sets of QA pairs for assessment. Then we used chat-gpt-3.5 turbo as a judge to score our system. In the initial phase, we only assessed the faithfulness of the fine-tuned model, and the results were very poor. Subsequently, we applied the same methodology to evaluate the faithfulness of the original model, but the results remained unsatisfactory. Upon scrutinizing the generated answers, we observed that many were unable to make inferences based on snippets. Therefore, we hypothesized that the low quality of the question-answer pairs might be the issue. To validate our suspicion, we directly used ChatGPT to respond to some unanswered questions, along with their corresponding snippets. ChatGPT also indicated its inability to answer these questions based on the provided snippets.This indicates that indeed there are issues with our question-answer pairs. Subsequently, we attempted to extract high-quality question-answer pairs from the test set, but could only extract 3 out of 100. Therefore, if we aim to generate 200 golden question-answer pairs, we would need to generate tens of thousands of question-answer pairs, which is quite time-consuming. To address this issue, we opted to calculate Relevancy, and as expected, the Relevancy was as high as 0.9.

3 Conclusions and open issues

3.1 Conclusion

Throughout this project, we have undertaken a comprehensive exploration of Retrieval-Augmented Generation (RAG) systems, as well as the development of Question-Answering (QA) systems, with the aim of enhancing natural language understanding and generation capabilities. Our journey encompassed various stages, from conceptualization to implementation, evaluation, and beyond.

In the realm of RAG systems, we have investigated two critical aspects: retrieval evaluation and response evaluation. Retrieval evaluation involved assessing the accuracy and relevance of information retrieved from knowledge bases. Metrics such as Hit Rate and Mean Reciprocal Rank (MRR) were utilized to quantify the performance of retrieval algorithms. Furthermore, we implemented evaluators for faithfulness and relevancy to gauge the quality and appropriateness of generated responses.Ultimately, based on our evaluation set, we opted for the relevancy evaluator. This comprehensive evaluation framework provided insights into the strengths and weaknesses of our RAG system, enabling iterative improvements.

On the QA system front, we leveraged state-of-the-art language models such as GPT-3.5 and integrated them into our pipeline for generating responses to user queries. We fine-tuned these models and evaluated their performance using rigorous evaluation methodologies. Through this process, we aimed to develop robust QA systems capable of providing accurate and relevant answers to a diverse range of questions.

Moving forward, our exploration of RAG and QA systems has unearthed several open issues and areas for further research and development:

3.2 Open Issues

- **Scalability**: Scaling RAG and QA systems to handle large-scale knowledge bases and diverse user queries remains a challenge. Future research should focus on developing scalable architectures and algorithms to support efficient information retrieval and generation.

- **Generalization**: Enhancing the generalization capabilities of RAG and QA systems to handle a wide range of domains, topics, and languages is crucial. Further research is needed to improve model robustness and adaptability across different contexts.

- **Evaluation Methodologies**: While existing evaluation metrics provide valuable insights, there is a need for more nuanced and comprehensive evaluation methodologies for assessing the performance of RAG and QA systems. Future work should explore novel evaluation approaches to capture the multifaceted nature of natural language understanding and generation.

- **Ethical Considerations**: As RAG and QA systems become more pervasive, addressing ethical concerns such as bias, fairness, and accountability becomes increasingly important. Future research should prioritize ethical considerations to ensure responsible development and deployment of these technologies.

- **Integration with Real-World Applications**: Bridging the gap between research prototypes and real-world applications remains a challenge for RAG and QA systems. Future efforts should focus on integrating these technologies into practical applications such as virtual assistants, customer support systems, and educational platforms.

Bibliography

A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b, 2023.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.