Development of a retrieval-augmented generation pipeline leveraging large language models to support evidence synthesis

Perera C[1], Hirst A[1], Heron L[1]
[1]Adelphi Values PROVE™, Bollington, Cheshire SK10 5JB, United Kingdom

ADELPHI VALUES

Expertise in Access and Value Evidence Outcomes

MSR1

Introduction

> The widespread adoption of large language models (LLMs) as seen across many industries, including health economics and outcomes research, has disrupted the way information is processed, analysed, and utilised for decision making.[1]
> One area of interest that could leverage the use of generative artificial intelligence (AI) is evidence synthesis.
> Traditional evidence synthesis relies heavily on the manual extraction of data as part of systematic literature reviews, which is time consuming.[2]
> With the rapid growth of available data, AI-based solutions like retrieval augmented generation (RAG) pipelines can streamline the process and provide faster, and more efficient outputs.
> Hallucination, or the generation of incorrect or misleading information, is a common pitfall in LLMs.
> The key advantage of RAG as the data extraction mechanism, is its ability to generate context-aware responses by retrieving relevant data before crafting a response, and as a result minimizing the chances of model hallucination.[3]
> The aim of this research was to develop a RAG pipeline that can support the extraction of unstructured data from Portable Document Format (PDF) files using both proprietary and open-source LLMs.

Methods

Retrieval Augmented Generation

> Figure 1 provides a visual overview of the RAG pipeline that has been developed to extract data from publications. The script was developed in Python 3.11.9, using the LangChain software development kit.[4]
> To understand whether significant differences in accuracy of data extraction were present both proprietary (OpenAI generative pre-trained transformer [GPT] models) and open-source (Meta AI LLaMa) models were included in the analysis. (gpt-4o-mini, gpt-4o, gpt-3.5-turbo, llama-3, llama-2).[5-6]
> A zero-shot prompt was developed to enable the model to generate relevant responses without requiring specific task-related examples.
> This approach allows the model to perform the task based on general understanding, leveraging the external data enhancing its ability to adapt to new or unseen data efficiently.
> RAG enhances LLMs by integrating external data retrieval into their outputs. It operates through three key steps:

1. **Semantic chunking:** Text is divided into smaller, meaningful units (chunks), rather than entire documents, to improve precision and relevance in information retrieval.
2. **Embedding:** Once the text has been chunked, each segment is transformed into a high-dimensional vector representation, through an embedding algorithm (e.g. OpenAI text-embedding-3-large). These embeddings capture the semantic meaning of the text and allow for comparison within an embedding space, where semantically similar chunks are positioned closer to one another. The process ensures that the system can efficiently match queries to relevant chunks of information by their proximity in this space.
3. **Retrieval:** When a query is presented, RAG searches the embedding space to retrieve the most relevant chunks of information based on their semantic similarity to the query. This retrieval step enables the LLM to augment its internal knowledge with external data, ensuring that the model's responses are informed by the most relevant and up-to-date information available.

> After the retrieval phase, the LLM synthesizes the retrieved information with its own internal knowledge to generate a response.

Hyperparameters

> The table below provides an overview of the important hyperparameters used in the analysis (Table 1).

Table 1. Hyperparameters used in the RAG pipeline.

General Hyperparameter	Value
chunk size	800 characters
chunk overlap	100 characters
temperature	0.1
top_k*	3
Search type	similarity_score_threshold
Score threshold	65%

*top-k chunks most similar to the query are retrieved from the vector database and used to generate the answer using a LLM as generator.

Evaluation

> To assess the retrieval performance of the different models evaluated two metrics were used:
  - **Precision at K (P@K):** The proportion of top K retrieved chunks that are relevant to the query. High precision indicating that the RAG pipeline is returning useful information.
  - **Mean average precision (MAP):** The average precision across multiple queries, giving an overall retrieval quality.

Pre-specified queries

> Multiple queries were pre-specified to help facilitate the data collection required for NMA quantitative analyses, (i.e. extraction of hazard ratios, confidence intervals, definitions, etc.).

Results

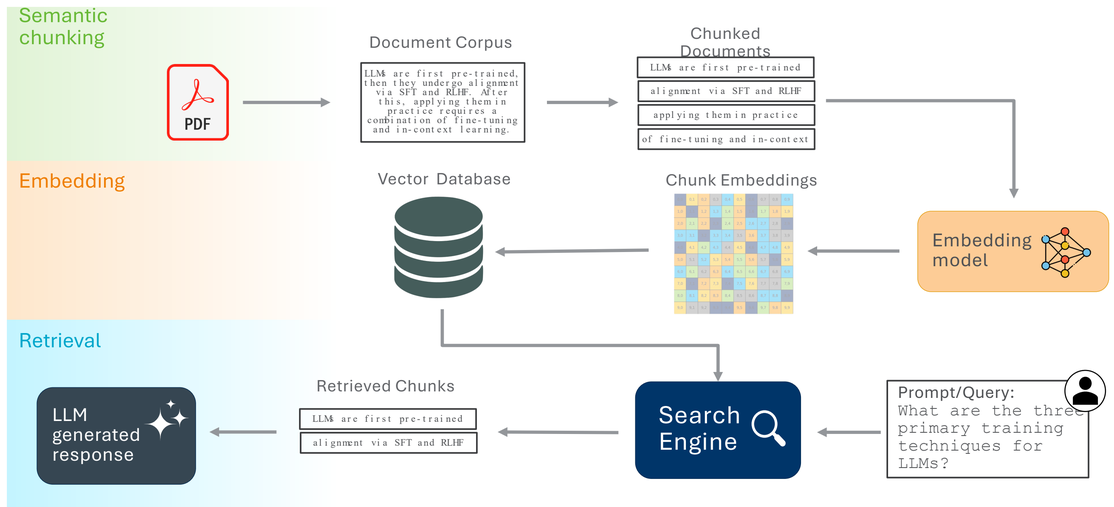

Figure 1. Overview of retrieval augmented generation pipeline.

Precision

> In the context of RAG, proprietary models demonstrated superior performance compared to open-source models as shown by P@K and MAP results (Table 2).

Model	P@K=3	MAP
gpt-4o-mini	80%	100%
gpt-4o	80%	100%
gpt-3.5-turbo	80%	80%
llama-3 (8B)	53%	80%
llama-2 (7B)	53%	60%

Abbreviations: 8/7B, 8/7 billion parameters.

> All proprietary models achieved on average 80% precision for the top three text chunks they retrieved for each given query.
> Llama-3 (open source) had an equivalent MAP to gpt-3.5-turbo.
> The gpt-4o models outperformed in ranking with a perfect MAP score of 100%, while gpt-3.5-turbo had slightly lower ranking performance with a MAP of 80%.
> The superior performance of proprietary models, particularly gpt-4o, can be attributed to their more extensive training on larger, proprietary datasets, as well as advanced model tuning and optimization that are unavailable in open-source alternatives.
> While open-source models showed lower performance overall, they remained competitive in cases involving smaller, less complex queries, suggesting that for specific use cases, open-source models could still provide viable, cost-effective solutions.

Performance trade-offs between proprietary and open-source models

> In addition to performance differences, open-source models were found to require significantly more time and computational resources to run.
> Open-source models are typically hosted and run on local machines, which lack the infrastructure and optimization of cloud-based proprietary models.
> 16 GB of random-access memory is required to run a small (7-8 billion parameter) model.

Conclusions

> The findings from this study showcase a use case in which generative AI has been leveraged to support evidence synthesis.
> In this case study proprietary models were able to accurately extract data from a published article, at minimal cost and requiring minimal computational power.
> The embedding models used to comprehend and reason with high-dimensional data are a driving factor for the accuracy of results.
> Proprietary models consistently outperformed open-source models across various query complexities, showing superior retrieval accuracy when handling large and complex datasets, whereas open-source models displayed diminished performance as query difficulty increased.
> PDF documents are inherently difficult to process due to their non-standardized structure and formatting.
> Parsing text from PDFs often leads to issues such as broken sentences, misplaced headers, and the loss of tabular data, which requires additional computational effort and pre-processing.
> Further investment into technology that is able to parse PDF documents efficiently is required to handle the accuracy and speed requirements of large-scale RAG applications.
> Future research will need to focus on incorporating additional state of the art models and assess their performance using novel metrics.
> Evidence synthesis stands to significantly benefit from the integration of artificial intelligence (AI) technologies.
> The use of RAG will enhance the capacity to synthesize evidence by incorporating real-time data retrieval, allowing researchers to keep up with the growing volume of published literature.

References

1. Reason T, Rawlinson W, Langham J, Gimblett A, Malcolm B, Klijn S. Artificial Intelligence to Automate Health Economic Modelling: A Case Study to Evaluate the Potential Application of Large Language Models. PharmacoEconomics - Open. 2024/03/01 2024;8(2):191-203.; **2.** Benzinger L, Ursin F, Balke WT, Kacprowski T, Salloch S. Should Artificial Intelligence be used to support clinical ethical decision-making? A systematic review of reasons. BMC Med Ethics. Jul 6 2023;24(1):48. doi:10.1186/s12910-023-00929-6.; **3.** Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems.2020;33:9459-9474.; **4.**Topsakal O, Akinci TC. Creating large language model applications utilizing langchain: A primer on developing llm apps fast. 2023:1050-1056.; **5.** OpenAI. Learning to Reason with LLMs. Accessed 2024, October. https://openai.com/index/learning-to-reason-with-llms/; **6.** Dubey A, Jauhri A, Pandey A, et al. The llama 3 herd of models. arXiv preprint arXiv:240721783. 2024;