ACE: Adaptative ChatGPT for Entreprise

Julien Baudru, Lluc Bono Rosello & Hugues Bersini

***Abstract*—We present a novel framework named ACE (Adaptive ChatGPT for Enterprise) designed to tailor the responses generated by a Large Language Model (LLM) to meet the specific requirements of an enterprise, predicated upon a predetermined set of documents. The distinctive innovation within this framework resides in its capacity to effectively identify documents aligning with the specific needs of the enterprise.**

I. Introduction

Emerging Large Language Models (LLMs) are consistently introduced with heightened capabilities and efficiency, albeit grounded in similar foundational concepts of furnishing the software with prompts inclusive of requisite instructions and context.

In response, we present a modular pipeline designed to adapt to anticipated enhancements in existing LLM models. This approach accentuates the refinement of contextual information provided to LLMs and the framework's adaptability for continuous improvement. The principal advantages inherent in our approach encompass the following: (1) Mitigation of errors or *hallucinations* to uphold information accuracy, (2) transparent identification of response sources to facilitate further exploration, (3) integration of advanced similarity matching technologies for precise context alignment, (4) provision of scalable solutions through the incorporation of embeddings for enhanced comprehension, (5) fusion of diverse search methodologies, including embeddings and RankBM25, for comprehensive search capabilities and (6) sustained flexibility to assimilate varied data sources and facilitate updates as necessary.

The delineated pipeline unfolds across four primary phases: (1) Data acquisition, (2), knowledge base generation, (3) query-context matching within the knowledge base and (4) output production through an LLM. In this context, the ultimate outcome, represented by the LLM response to a given query, heavily hinges upon the efficacy of the antecedent stages. Indeed, an LLM's ability to generate a trustworthy answer is contingent upon access to pertinent contextual information for the posed question.

II. Methods

A. Pipeline overview

A comprehensive depiction of the proposed pipeline is delineated in the Figure 1. Titles of articles are transformed into embeddings, represented as vectors within a high-dimensional space, employing pre-trained machine learning models. These embeddings serve to efficiently retrieve article titles that exhibit semantic similarity to the provided query. Following the initial selection of scientific articles, a subsequent method is employed to identify pertinent segments of text within the top N more relevant articles. These selected segments serve as context for the Large Language Model (LLM), eliminating the necessity for precomputing embeddings. The procedural flexibility of this pipeline permits further adaptations contingent upon available resources and technological advancements.

B. Data acquisition

The objective of the data acquisition phase is to establish versatile methodologies for obtaining context-specific data that addresses inquiries from experts. In order to augment the model with context-specific data tailored to user queries, a coherent strategy for data acquisition must be formulated. Data sources encompass both internal company repositories and external origins, including other scientific repositories. This discussion focuses on

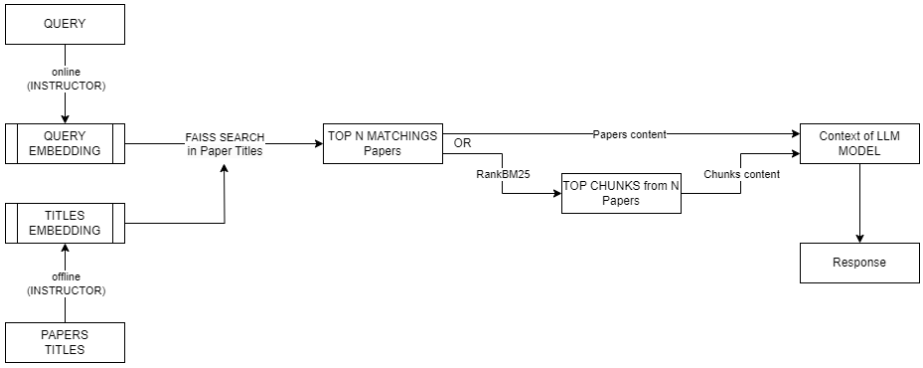

Fig. 1. Pipeline overview

the methodologies employed to autonomously and efficiently acquire and store data from external sources. Concurrently, the aim is to expand the repertoire of scientific sources beyond the scope of the prior prototype and standardize the methodologies applied to existing sources.

Initially, a web scraping tool was employed to procure a set of scientific papers corresponding to a predefined set of test queries. This initial dataset served as a benchmark to evaluate the efficacy of the tool. Subsequently, for the formalization of data acquisition methods, two Python scripts were developed to extract data from the APIs of two open-source datasets, PubMed Central (PMC) [1] and arXiv [2], details about the APIs are given in Table I.

These selected data sources function as proof of concept, illustrating how the framework's scope can be extended by systematically integrating additional data sources, encompassing both external and internal repositories. Furthermore, the modular design of the framework accommodates the incorporation of diverse data sources without necessitating alterations to its core functionalities. This inherent flexibility is crucial for tailoring the tool to the specific needs of users associated with distinct scientific fields, thereby allowing seamless adaptation to varying data sources.

Concerning the automation of the data acquisition process, diverse implementation options are envisioned. These include considerations such as user-defined queries, the volume of data to be retrieved, the frequency of updates, and other adjustable parameters, providing a customizable and adaptive framework.

C. Knowledge base generation

The primary objective is to construct a knowledge base that accommodates diverse data inputs and facilitates the generation and storage of embeddings. We advocate for a standardized structure wherein scientific articles, comprising essential fields such as *title*, *abstract*, and *full-text*, are uniformly organized. This approach ensures consistency in the storage format for data originating from disparate sources, be they internal or external. We propose the incorporation of a *collection* field, affording the flexibility to utilize the tool with specific subsets of the database within the interface.

To enhance the efficiency of article retrieval, we have chosen to employ text embeddings for each title. Text embeddings represent a transformative feature in the domain of natural language processing, facilitating the conversion of text into a computationally amenable format represented as vectors of real numbers. These embeddings adeptly encapsulate both semantic and syntactic attributes of the text, with semantically similar meanings often

[1]PMC API Documentation: https://www.ncbi.nlm.nih.gov/books/NBK25501/

[2]arXiv API User Manual: https://arxiv.org/help/api/user-manual

TABLE I
OPEN-SOURCE DATASETS AND APIS

Dataset	Description	API
PubMed Central (PMC)	A free digital repository archiving publicly accessible full-text scholarly articles in the biomedical and life sciences.	The National Center for Biotechnology Information (NCBI) offers the Entrez Programming Utilities (E-utilities) for programmatic access.
arXiv	An open-access archive for preprint scholarly articles across various fields.	arXiv provides an API for accessing and retrieving metadata.

manifesting as proximal points in the vector space. Utilizing neural network models such as Word2Vec, GloVe, or BERT, these embeddings are learned from extensive text corpora.

Representing article titles as text embeddings harnesses their intrinsic semantic content, thereby refining the search mechanism. Unlike conventional keyword matching, which may overlook contextually relevant synonyms, embeddings offer a nuanced approach, recognizing conceptual similarities beyond mere lexical associations. This approach contributes to a more comprehensive and contextually aware search process within the knowledge base.

In computational terms, the embedding process can be expressed as:

$$\vec{v}(w) = \text{Embed}(w; \mathcal{M}) \tag{1}$$

Here, $\vec{v}(w)$ denotes the vector representation of the word w or a sequence of words, and $\mathcal{M}$ represents the embedding model that has been trained on a large corpus. The resulting vectors facilitate rapid and semantically rich comparisons, pivotal for scaling the search process amidst an expanding dataset.

However, in order to produce the text embeddings, different models $\mathcal{M}$ are available and the model that better fits the use-case needs to be selected. To compare the performance of different models we generated the embeddings with different models for the same sample of titles. For a better visualization the different embeddings (high-dimensional vectors), we reduced those vectors to their 2 principal components, allowing to assess the capacity of the model to split the dataset in the search space. Subsequently, we proceeded to examine a state-of-the-art approach, specifically Instructor [7]. In order to assess its efficacy, we conducted a comparative analysis with a specifically trained[3] mode like BIOBERT[2]. Notably, in Figure 2 we observed that the partitioning methodology employed by the Instructor Model demonstrates relevance. Consequently, we intend to evaluate and compare their respective efficiencies in the context of matching through empirical tests, as detailed in the subsequent section.

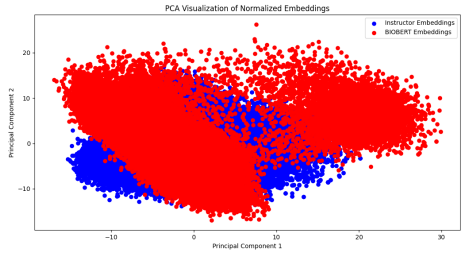

Fig. 2. Visualization of Instructor embeddings VS BIOBERT embeddings in a 2-D space

These embeddings will be produced for each title and included in the database scheme, allowing for efficient searches and similarity matching methods proposed in the following section. In conclusion, the knowledge base will comprise the different scientific articles obtained from different sources with their title, abstract, full-text and the text embedding representing their title[4].

[3]BIOBERT was specifically trained in biomedical literature

[4]A more detailed description of the database scheme is available in the Prototype Section

D. *Query-context matching*

The objective is to procure valuable matchings that demonstrate computational efficiency through the application of techniques such as similarity search in vector databases and Rank BM25.

In our preliminary experiments, we initially employed a method centered on calculating the edit distance, often using the Levenshtein distance metric[3] , between two texts. This metric quantifies the number of edits, including insertions, deletions, or substitutions, required to transform one string into another. While this approach can be pertinent for specific tasks, it may lack semantic meaningfulness when assessing textual relevance, particularly when dealing with longer texts or instances where text structures differ. In order to solve this issue, we studied the following alternatives: (1) The cosine similarity metric [4] which computes the cosine of the angle between two vectors. It returns a value between -1 and 1, with 1 indicating that the vectors are identical, 0 indicating that the vectors are orthogonal (i.e., not similar at all), and -1 indicating that the vectors are diametrically opposed. The computation involves a dot product and a normalization step, (2) the FAISS library [1] which is a library developed by Facebook Research that facilitates efficient similarity search and clustering of dense vectors. By default, FAISS uses L2 (Euclidean) distance to compute similarity, but it can be adjusted to work with cosine similarity and (3) the BM25 [6] algorithm which is a probabilistic information retrieval model that ranks a collection of documents based on the query terms appearing in each document, regardless of their proximity. It extends the binary independence retrieval model by introducing term frequency and document length normalization, offering a balance between term frequency and inverse document frequency, making it robust and effective for information retrieval tasks.

The first outcome observed is a significant enhancement in the computation time for matching when employing similarity methods such as Cosine Similarity or the FAISS library, as illustrated in Figure 3. It is noteworthy to emphasize that the execution of this search over the embeddings necessitates prior computation, which can be computationally expensive. Nevertheless, it is imperative to underscore that despite the time-consuming nature of embedding computation, such calculations need only occur once.[5]

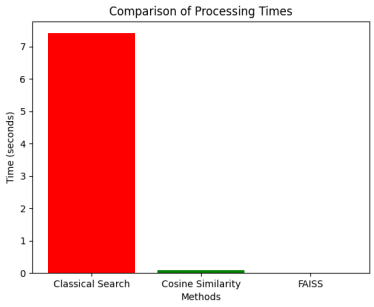

Fig. 3. Computation time of the similarity search methods tested.

Moreover, as depicted in Figure 3, it is evident that the incorporation of the FAISS search implementation may yield more efficient results when scaling up. To conduct a detailed comparison between both alternatives, we conducted tests across various dataset sizes with 100 queries. The outcomes, as presented in Figure 4, elucidate the distinctions in the scalability performance of both solutions.

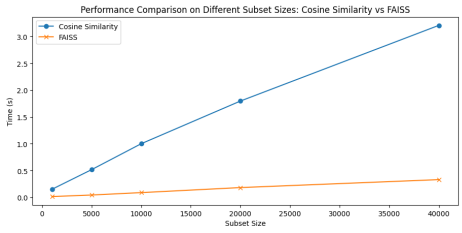

Fig. 4. Running time for 100 queries over different embeddings datasets.

At this point we have proved how the use of embeddings and similarity searches over the vector space can reduce the computation time of our

[5]In cases where new data is incorporated into the database, but the embedding model remains constant, the embedding process is solely required for the newly added data.

search, paving the way for the scalability of the solution. Now, we can focus on the efficacy of these searches (e.g. how close are the selected articles to our query).

In this regard, the capabilities of the embeddings to refer to semantical concepts have proven to increase the accuracy of the results as it can be seen in Appendix A in comparison to those previously obtained in Appendix A. In these examples, the top-5 results for an example query show the limitations of the classical approach where words are matched but the semantical similarity of the title is not explored. This can cause that a title like *'Availability of food folate in humans'* is selected as a matching for *'Best Food Suplements for Diet'*, in such case the word '*food'* appears in both texts but the article seems unlikely to contain information regarding the query.

Regarding the two possible embedding Models: BIOBERT [2] and Instructor [7], previously introduced as candidates, we compared their efficacy on representing the titles and the queries for suitable matchings. The results show that the matchings produced over the Instructor embeddings were more accurate in our benchmark, as it can be seen in results in Appendix A against the results in Appendix A. As an example, for the same query: *"Stability of Vitamin D in food supplements"*, the first matching provided over Instructor Embeddings is *"Vitamin D in food and supplements"* while the first matching provided over BIOBERT embeddings is *"Stability of B vitamins in pharmaceutical products"*.

Regarding the differences between cosine similarity and FAISS search, we have shown how FAISS search computation time is significantly lower. In terms of results, we can clarify that if the embeddings are **normalized** (i.e., each vector has a magnitude or length of 1), then the L2 (Euclidean) distance and cosine similarity/dissimilarity are directly related. Specifically, when for normalized vectors like shown in the Equation 2.

$$\text{cosine similarity} = 1 - \frac{1}{2} \times \text{L2 Distance}^2 \quad (2)$$

As a corollary, vectors exhibiting proximity with respect to the L2 distance metric will also manifest the highest similarity according to cosine similarity, while vectors characterized by the greatest L2 distance separation will correspondingly denote the least similarity in terms of cosine similarity. The similarity of these results over the same set of embeddings can be seen in appendices A and A. In these results we see how the matching are practically identical while the FAISS Search is more efficient. With that last comparison we can propose, at this point, as our best method the use of FAISS Search over the Instructor Model embeddings for this specific use case (e.g. matching of queries with scientific titles).

BM25 - An alternative to embeddings: The subsequent exploration involves the consideration of the third alternative, specifically the utilization of BM25 [6]. This method exhibits several advantages over similarity search in vector spaces. It is well-established and widely employed in the information retrieval community, obviating the necessity for pre-trained embeddings or extensive computational resources. Particularly efficient for large-scale document collections, BM25 is interpretable, facilitating a relatively straightforward understanding and explanation of document ranking based on term frequency and inverse document frequency metrics.

However, this method is not without its drawbacks. Primarily relying on lexical considerations, BM25 is dependent on exact term matches, potentially overlooking semantically similar terms or phrasings. Additionally, optimal performance necessitates the tuning of BM25 parameters, such as $k1$ and b. Furthermore, despite its independence from pre-computed embeddings, BM25 exhibits a significantly slower computation time compared to the preceding methods.

To assess the suitability of this alternative for our specific use-case, an initial investigation involved a comparative analysis of computation time against our established methods. As illustrated in Figure 5, it is evident that although BM25 does not exhibit exceptional scalability, it maintains a relatively swift performance, accomplishing 100 queries within a mere 6 seconds. This characteristic positions BM25 as a potential candidate for hybrid applications,

particularly advantageous given its independence from pre-computed embeddings, a requirement inherent in methods like FAISS search. This feature enhances flexibility in contemplating scalability in subsequent phases, assuming the continued relevance of similarity matchings. This adaptability is poised to be further harnessed in a subsequent phase wherein, instead of computing embeddings for titles, a more judicious approach involves leveraging them for pre-selecting a subset of articles and subsequently conducting a full-text search using BM25.

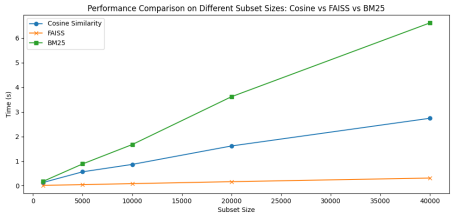

Fig. 5. Running Time for 100 queries over different dataset sizes.

Furthermore, to provide with more realistic results, we show the results for a single query in Figure 6 where the time for BM25 consistently hovers around 0.1 seconds which is entirely suitable for real-time applications.

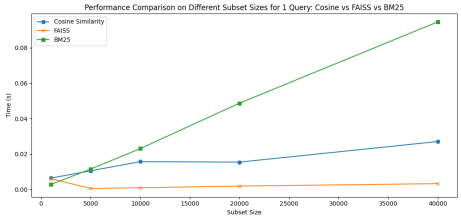

Fig. 6. Running Time for 1 query over different dataset sizes.

Finally, regarding the relevancy of the results produced by BM25, we can observe in Appendix A how the matching produced are very similar to those given by our previous best model. With that in mind, we can conclude that BM25 is a suitable alternative to be taken into account in the framework implementation and, indeed, it will be incorporated in the pipeline as the second search mechanism over a reduced size dataset (on the pre-selected titles).

Hybrid matching pipeline: Towards the conclusion of our methodology, we advocate for the integration of a hybrid matching technique within our pipeline. This approach incorporates the utilization of Instructor embeddings and FAISS search to efficiently identify the top-N articles likely to contain pertinent information for addressing the user's query. This initial matching process is designed for rapid execution, particularly adept at handling extensive datasets. Subsequently, in a second phase, BM25 is employed to traverse the various text-chunks within the selected articles. This strategic approach eliminates the necessity for pre-computing and storing embeddings for all texts within every article, a potentially resource-intensive task. By adopting this hybrid strategy, we leverage the distinct advantages of each technology at specific stages, thereby optimizing computational efficiency and resource utilization.

E. Output production

The primary objective is to integrate the contextual matching results with Large Language Models (LLM) to generate coherent responses to queries while explicitly enumerating the utilized sources.

For this proof-of-concept, and for the sake of simplicity, we opted to employ OpenAI's API, leveraging its well-established model and easily configurable interface. However, as outlined in the Introduction, the choice of a specific LLM is independent of the rest of the pipeline. It is anticipated that the industry will witness the emergence of new language models, augmenting the performance of existing ones. Consequently, the adaptation to different APIs for interfacing with new LLMs would be the only requisite encoding modification in the pipeline, with outcomes expected to improve in such scenarios.

During interaction with the OpenAI GPT API, the system undertakes the processing of a user's

query by structuring the question and relevant context into a formalized message. This message adheres to the format specifications mandated by the OpenAI API, designating the role as a "user" and encapsulating both the question and selected contextual information. Subsequently, this formatted message is transmitted to the ChatGPT API, presently configured with the "gpt-3.5-turbo" language model, which processes the input and generates a comprehensive response.

The best sections of the relevant articles are given to the ChatGPT context with instructions on how to respond and the user's query. The instruction given to the LLM is given in the Figure 7.

> "I want a short scientific answer with context AND only 5 bullet points starting with '-' for this question: $query$. Use the following related informations to reply: $text$."

Fig. 7. Prompt instruction given to the LLM context.

Naturally, the instructions given to the model can be modified to suit requirements. Upon reception of the generated response from the API, the system initiates a traceability process to enhance transparency. This involves appending the sources to the response, providing direct links to the referenced articles. This practice ensures users have access to the original sources for verification or to explore additional information. In instances where the response is solely derived from the model's pre-existing knowledge, extending up to the model's last training data cutoff, the system explicitly annotates this information. Consequently, the final output presented to the user encompasses the answer to their query, coupled with a detailed delineation of the sources or a declaration of the model's inherent knowledge base. This meticulous approach ensures the integrity, accuracy, and traceability of the information provided.

III. Prototype

In this section, we detail the different functionalities of the prototype, describe the proposed interface, present different parts of the implementation from a practical point of view and explain how the framework should be installed.

In essence, two primary actions are available to the user: querying the framework or updating the database. The proposed framework encompasses the following functionalities: (1) The user possesses the capability to submit a query to the Large Language Model (LLM) concerning a database of scientific articles. The LLM, in turn, provides a concise summary of the relevant data without utilizing bullet points. Furthermore, the model includes citations to the sources of the articles used to formulate the response, as exemplified in Figure 8. (2) Users have the option to incorporate topics of interest and prompt the program for an immediate update of the database for these topics. Additionally, they can instruct the program to periodically check for new articles related to these topics, as depicted in Figure 9. (3) The user is empowered to specify the database on which the LLM should base its response, enabling a more nuanced and specialized reply, as demonstrated in Figure 9. (4) Moreover, users can indicate whether the program should include papers available online in the response (from Arxiv and PMC) or restrict the answer solely to the selected database, as illustrated in Figure 9.

IV. Implementation

Dependencies: In the current state of the framework, the program depends on several external sources to operate at its full potential. Firstly, it requires an Internet connection, which enables the program to retrieve various data and communicate with several APIs detailed hereafter. To function normally, the program needs to access the OpenAI API to send requests to ChatGPT3.5 and receive replies, so it is necessary to create an API Key, the use of which entails costs that are detailed in section V. Then, if the user wishes to use the online search function or the update function, the program also needs access to the APIs of Arxiv and PMC, two online repositories of scientific papers, which, unlike OpenAI, do not require an API key. This prototype was developed in Python 3.11.5 and libraries used are given in the table II.

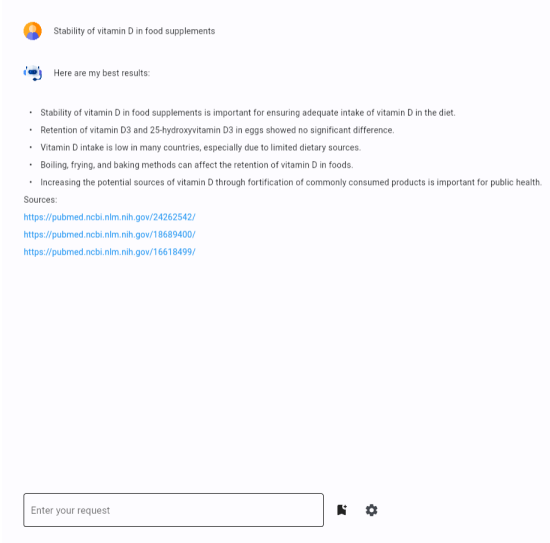

Fig. 8. Interface - Chat

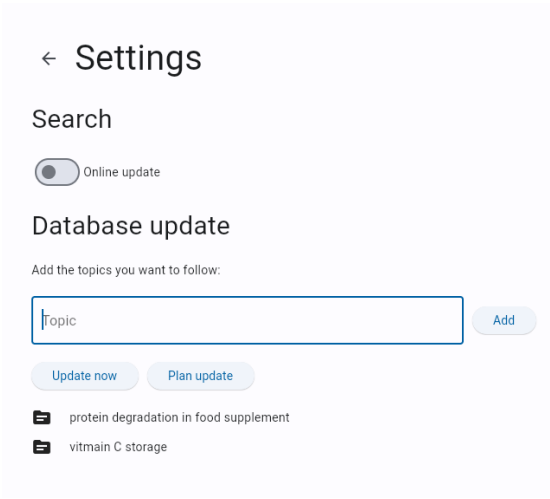

Fig. 9. Interface - Settings

Database: The proposed framework also depends on the MongoDB database management system [5]. For each entry, i.e. scientific article, in the database, the fields we have used are given in the table III. By default, the framework is supplied with a database of 36K scientific documents for which article title embeddings have been calculated in advance. However, when the user activates the online search function or launches an update, the embeddings of the new articles must be computed on the fly. Figure 10 gives an overview of the steps involved in updating on a particular topic or, in the case of online search, on a query made by the user.

TABLE II
PYTHON LIBRARIES VERSIONS

Library Name	Version
tqdm	4.66.1
matplotlib	3.7.2
flet	0.10.0
tiktoken	0.4.0
sentence_transformers	2.2.2
InstructorEmbedding	1.0.1
openai	0.28.0
pandas	2.1.0
faiss-cpu	1.7.4
rank_bm25	0.2.2
PyPDF2	3.0.1
pymongo	4.5.0
feedparser	6.0.10

TABLE III
DATABASE STRUCTURE

Field	Type
id	ID Object
collection	String
link	String
title	String
abstract	String
full_text	String
title_embeddings	Int Vector
abstract_embeddings	Int Vector
full_text_embeddings	Int Vector

Note that in the case of online research, after updating the database with new articles linked to the user's query, these articles could be used in the LLM context to respond if they belong to the most relevant. Moreover, to avoid the inclusion of irrelevant articles in the database, only articles that are significantly similar (according to our matching) are included, while the rest are discarded.

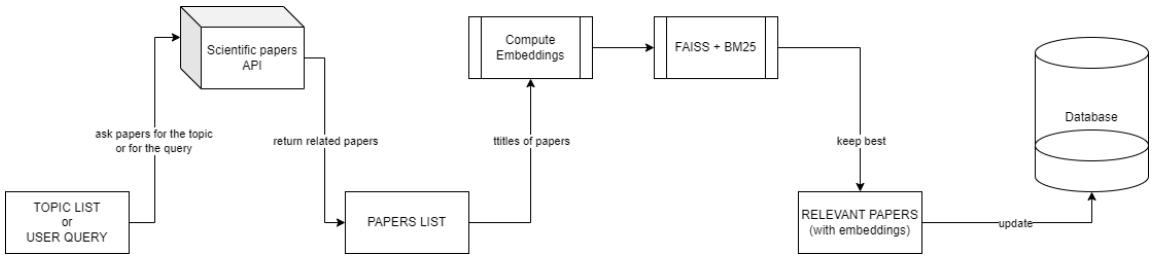

Fig. 10. Update scenario

Installation: To install the framework presented in these pages, you'll need a server machine with at least 16Gb RAM and a recent processor. The more users connect to the framework, the more resource-intensive the program will be. The framework has been designed to be plug-and-play: once MongoDB and all the Python libraries have been installed, you only have to share the machine's local URL followed by port 3333 with users on the same network, so that they can access it. If the framework needs to be used by people outside the enterprise network, we recommend the use of *ngrok*. In practice, the code is stored in a private GitHub directory, to which you can request access by sending a simple message to one of the following e-mail addresses: julien.baudru@ulb.be or lluc.bono.rossello@ulb.be.

V. Discussion

Regarding the flexibility of the framework, the efficacy of document similarity search, rooted in the general embedding method known as Instructor, dictates the quality of the selected documents. Our observations indicate the effectiveness of this method in the majority of cases. However, there is a possibility to employ more domain-specific embedding methods, such as BioBert tailored for biology, contingent upon the specific field of research. The trade-off between flexibility and framework specialization arises, as adopting specialized embedding methods mandates modification when transposing the project to a different domain.

In the context of utilizing OpenAI's Large Language Model (LLM), ChatGPT, privacy emerges as a paramount concern. The company's privacy policy discloses potential collection of personal information from messages, uploaded files, and feedback during the use of ChatGPT. This poses a challenge for companies where in-house research demands confidentiality. Additionally, the access to ChatGPT API is associated with a fee of 0.002$ per 1000 tokens, and while the individual cost is nominal, the cumulative expense escalates based on the framework's user count, interaction duration, and the volume of article information provided to the model.

An additional limitation posed by ChatGPT is the constrained context size, set at 4,096 tokens (approximately 3,072 words). This limitation may be restrictive if numerous sections in selected articles align with the user's query. However, the solution's quality is not directly contingent on context size but rather on the quality of the provided information. The inherent limitations of LLMs can be circumvented by opting for a local LLM model. Open-source implementations exist, and deploying such a model on a company server resolves privacy concerns. While the cost of this approach depends on server configuration and power consumption, it offers a viable alternative. Notably, certain open-source models, such as "longchat-13b-16k" or "LLaMA-7b-32k," accommodate larger context sizes, up to 16K or 32K. This option gains relevance considering the anticipated improvement of newer open-source models in the coming months. Moreover, it is noteworthy that companies like OpenAI are cognizant of privacy risks associated with their technology in enterprise settings, and they are actively exploring specific solutions to address

this concern.

VI. Future works

Our experiments have led to the identification of potential enhancements for the framework. First, the accuracy of API requests on PMC and Arxiv exhibits limitations, with returned articles often lacking relevance based on the selected search topic. A possible remedy involves diversifying sources of information acquisition. This entails exploring alternative APIs for platforms like PubMed or Scopus and refining the use of existing APIs to bolster result accuracy. In conjunction with API considerations, the inclusion of additional scientific databases on pertinent subjects is recommended. These databases, likely in PDF format, would necessitate the development of a module for serialization in the database format. Furthermore, customization of the instruction provided to the context (prompt) is advised to align with the specific needs of the company (refer to 7). Certain prompts may yield superior results in terms of quality, although further experimentation is imperative for conclusive validation. These proposed improvements primarily revolve around augmenting the quality of information fed into the LLM. Implementation of these enhancements holds the potential to significantly boost the framework's performance with minimal structural modifications.

VII. Conclusion

In conclusion, our scientific inquiry has effectively illuminated the intricacies of employing ChatGPT in diverse contexts. Through our investigation, it has been discerned that possessing fundamental knowledge of ChatGPT alone proves insufficient for addressing scientific inquiries. Furthermore, we have underscored the paramount importance of averting hallucinations, thereby ensuring the precision and reliability of the information imparted. A significant advancement has been achieved by advocating for the transparent disclosure of the sources underpinning the responses generated by ChatGPT. This transparency not only instills trust and credibility but also facilitates comprehensive scrutiny and validation of the presented information. The pursuit of optimizing ChatGPT's performance has led us to the identification of optimal similarity matching technologies, particularly harnessing the capabilities of embeddings.

We have illustrated that scalable solutions are attainable through the implementation of these embedding techniques, thereby paving the way for heightened performance and efficiency. Furthermore, our findings substantiate the efficacy of amalgamating diverse hybrid search strategies, including embeddings and RankBM25, yielding promising outcomes and augmenting the quality of responses generated by the Large Language Model (LLM). Lastly, our research underscores the significance of flexibility and adaptability in LLM utilization. By ensuring compatibility with various data sources and enabling regular updates, a versatile and continuously evolving tool has been established.

In summary, our study has yielded invaluable insights and recommendations for the proficient and ethical application of LLM within the scientific domain. The objectives we set out to achieve have been successfully realized, and we anticipate that our findings will not only contribute to the ongoing development and responsible integration of AI-powered natural language processing systems but also result in the creation of an application suited for the research and development department within enterprises.

References

[1] Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. IEEE Transactions on Big Data **7**(3), 404–416 (2019)

[2] Lee, J., et al.: Biobert github repository. https://github.com/dmis-lab/biobert (2019)

[3] Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet physics doklady **10**(8), 707–710 (1966)

[4] Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)

[5] MongoDB: Mongodb documentation. https://docs.mongodb.com/ (2009–2024), accessed: January 11, 2024

[6] Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M.: Okapi at trec-3. NIST SPECIAL PUBLICATION SP **109** (1994)

[7] Su, H., Shi, W., Kasai, J., Wang, Y., Hu, Y., Ostendorf, M., Yih, W.t., Smith, N.A., Zettlemoyer, L., Yu, T.: One embedder, any task: Instruction-finetuned text embeddings (2022), https://arxiv.org/abs/2212.09741

APPENDIX

Query	**Vitamin d in solid formulations**
	Vitamin E adjuvant formulations in mice
	Vitamin E adjuvant formulations in mice
	Cyanide-bridged vitamin B12-cisplatin conjugates
	Cyanide-bridged vitamin B12-cisplatin conjugates
	Vitamin D and ageing
Query	**Stability of vitamin D in food supplements**
	Vitamin D in food and supplements
	Vitamin D in food and supplements
	Stability of vitamin D in foodstuffs during cooking
	Bioassay of vitamin A in liquid supplements
	Vitamin D in foods and as supplements
Query	**Chemical stability of vitamin D**
	Optimization of Chemical Syntheses of Vitamin D C3-Epimers
	Optimization of Chemical Syntheses of Vitamin D C3-Epimers
	Enhancement of the chemical stability in confined -Bi2O3
	The chemical determination of vitamin D in ultraviolet-irradiated milk
	The chemical and microbiological stability of multivalent vitamin preparations
Query	**Chemical stability of ascorbic acid in food supplements**
	Quantitative analysis of chemoresistance-inducing fatty acid in food supplements using UHPLC-ESI-MS/MS
	Covalent interaction of ascorbic acid with natural products
	The role of ascorbic acid in carcinogenesis
	The role of ascorbic acid in carcinogenesis
	Antioxidative properties of ascorbic acid
Query	**Stability of cyanocobalamin in food supplements**
	Stability of cyanocobalamin in parenteral preparations
	Stability of cyanocobalamin in sugar-coated tablets
	MICROBIOLOGICAL QUALITY OF FOOD SUPPLEMENTS
	Assessment of nutraceuticals and food supplements
	HPLC/DAD/MS and antioxidant activity of isoflavone-based food supplements
Time	**Classical search for all queries took 35.527451 seconds.**

Query	**Vitamin d in solid formulations**
	Understanding vitamin D formulations
	Recent Advances in Formulation Strategies for Efficient Delivery of Vitamin D
	[Vitamin D supplementation]
	Vitamin D in food and supplements
	Vitamin D supplementation
Query	**Stability of vitamin D in food supplements**
	Vitamin D in food and supplements
	The determination of vitamin D: The stability of preparations of vitamin D
	[Stability of vitamins A and D in polyvitamin preparations]
	Stability of vitamin D in foodstuffs during cooking
	Vitamin D in foods and as supplements
Query	**Chemical stability of vitamin D**
	The determination of vitamin D: The stability of preparations of vitamin D
	Stability of vitamin D metabolites in human blood serum and plasma
	[Stability of vitamins A and D in polyvitamin preparations]
	Stability of vitamin D in foodstuffs during cooking
	Studies on the metabolites of vitamin D
Query	**Chemical stability of ascorbic acid in food supplements**
	The stability of ascorbic acid in various liquid media
	Studies in the stability of compressed tablets of ascorbic acid
	Factors influencing the stability of ascorbic acid in total parenteral nutrition infusions
	On stabilisation of ascorbic acid in solutions
	Stability of ascorbic acid in a liquid multivitamin emulsion containing sodium fluoride
Query	**Stability of cyanocobalamin in food supplements**
	Stability of cyanocobalamin in sugar-coated tablets
	Stability of cyanocobalamin in film-coated multivitamin tablets
	Stability of cyanocobalamin in parenteral preparations
	[Studies on cyanacobalamin. II. Stability of cyanocobalamin solutions]
	Cyanocobalamin (vitamin B12). I. A study of the stability of cyanocobalamin and ascorbic acid in liquid formulations
Time	**FAISS search for all queries took 0.020877 seconds.**

Query	**Vitamin d in solid formulations**
	210Pb in magnesium dietary supplements
	O-t-c vitamins
	26Mg as a probe in research on the role of magnesium in nutrition and metabolism
	FOLACIN and riboflavin
	pH stability of individual folates during critical sample preparation steps
	in prevision of the analysis of plant folates
Query	**Stability of vitamin D in food supplements**
	Stability of B vitamins in pharmaceutical products
	Degradation of vitamin B12 in dietary supplements
	Stability of vitamin B1 in a commercial multivitamin syrup preparation
	Stability of vitamins in total parenteral nutrient solutions
	Stability of vitamin C derivatives in solution and topical formulations
Query	**Chemical stability of vitamin D**
	Stability of vitamin B12 in the presence of ascorbic acid
	The effect of various sugars and polyols on the stability of vitamin B12
	Bioavailability of vitamin A
	Bioavailability of vitamin E
	The stability of choline ascorbate
Query	**Chemical stability of ascorbic acid in food supplements**
	Stability of B vitamins in pharmaceutical products
	Stability of ascorbic acid in commercially available orange juices
	Degradation of vitamin B12 in dietary supplements
	Effect of silica gel on stability and biological availability of ascorbic acid
	The influence of amino acid source on the stability of ascorbic acid in TPN mixtures
Query	**Stability of cyanocobalamin in food supplements**
	Stability of cyanocobalamin in sugar-coated tablets
	Stability of B vitamins in pharmaceutical products
	Stability of cyanocobalamin in parenteral preparations
	Stability of cyanocobalamin in film-coated multivitamin tablets
	Stability of microencapsulated L-5-methyltetrahydrofolate in fortified noodles
Time	**FAISS search for all queries took 0.033560 seconds.**

Query	**Vitamin d in solid formulations**
	Processing challenges with solid dosage formulations containing vitamin E TPGS
	Synthesis and biological evaluation of 4-substituted vitamin d and 14-epi-previtamin d analogs
	Crystalline solid dispersion-a strategy to slowdown salt disproportionation in solid state formulations during storage and wet granulation
	Solvent-free melting techniques for the preparation of lipid-based solid oral formulations
	Downstream processing of polymer-based amorphous solid dispersions to generate tablet formulations
Query	**Stability of vitamin D in food supplements**
	Vitamin D in food and supplements
	Stability of vitamin D in foodstuffs during cooking
	Assessment of nutraceuticals and food supplements
	Quality control of plant food supplements
	It's time for new rules on vitamin D food supplements
Query	**Chemical stability of vitamin D**
	Optimization of Chemical Syntheses of Vitamin D C3-Epimers
	Chemical stability and phase distribution of all-trans-retinol in nanoparticle-coated emulsions
	The determination of vitamin D: The stability of preparations of vitamin D
	Chemical stabilization of y-polyglutamate by chitosan and the effect of co-solvents on the stability
	Chemical analysis of vitamin D in concentrates and its problems. XII. Analysis of fat-soluble vitamins
Query	**Chemical stability of ascorbic acid in food supplements**
	Chemical stability study of vitamins thiamine, riboflavin, pyridoxine and ascorbic acid in parenteral nutrition for neonatal use
	Studies in the stability of compressed tablets of ascorbic acid
	Chemical stability of astaxanthin nanodispersions in orange juice and skimmed milk as model food systems
	The stability of ascorbic acid in various liquid media
	[Determination of ascorbic acid in food and biological material]
Query	**Stability of cyanocobalamin in food supplements**
	Stability of cyanocobalamin in sugar-coated tablets
	Stability of cyanocobalamin in parenteral preparations
	Stability of cyanocobalamin in film-coated multivitamin tablets
	[Studies on cyanacobalamin. II. Stability of cyanocobalamin solutions]
	Assessment of nutraceuticals and food supplements
Time	**BM25 search for all queries took 0.196385 seconds.**

Query	**Vitamin d in solid formulations**
	Understanding vitamin D formulations
	Recent Advances in Formulation Strategies for Efficient Delivery of Vitamin D
	[Vitamin D supplementation]
	Vitamin D in food and supplements
	Vitamin D supplementation
Query	**Stability of vitamin D in food supplements**
	Vitamin D in food and supplements
	The determination of vitamin D: The stability of preparations of vitamin D
	[Stability of vitamins A and D in polyvitamin preparations]
	Stability of vitamin D in foodstuffs during cooking
	Vitamin D in foods and as supplements
Query	**Chemical stability of vitamin D**
	The determination of vitamin D: The stability of preparations of vitamin D
	Stability of vitamin D metabolites in human blood serum and plasma
	[Stability of vitamins A and D in polyvitamin preparations]
	Stability of vitamin D in foodstuffs during cooking
	Studies on the metabolites of vitamin D
Query	**Chemical stability of ascorbic acid in food supplements**
	The stability of ascorbic acid in various liquid media
	Studies in the stability of compressed tablets of ascorbic acid
	Factors influencing the stability of ascorbic acid in total parenteral nutrition infusions
	On stabilisation of ascorbic acid in solutions
	Stability of ascorbic acid in a liquid multivitamin emulsion containing sodium fluoride
Query	**Stability of cyanocobalamin in food supplements**
	Stability of cyanocobalamin in sugar-coated tablets
	Stability of cyanocobalamin in film-coated multivitamin tablets
	Stability of cyanocobalamin in parenteral preparations
	[Studies on cyanacobalamin. II. Stability of cyanocobalamin solutions]
	Cyanocobalamin (vitamin B12). I. A study of the stability of cyanocobalamin and ascorbic acid in liquid formulations
Time	**Cosine search for all queries took 0.161872 seconds.**