Project Proposal

BioQAid: Enhancing Systematic Literature Reviews through a Question Answering system for Biomedical literature and Life Science journals

Student Name: Sanjay Hegde

Student ID: 2432010

Supervisor Name: Jizheng Wan

Project Category/Topic: AI

Project Aim:

- To build a Question Answering System for Biomedical Literature and Life Science journals on PubMed database for reducing the effort required for manual searching of the research papers or reference for a specific topic and to minimise the time consumed for Systematic Literature review of the published papers.
- **Significance**: With the fast-growing research and developments in the field of biomedicine, it is very essential to perform a Literature Survey of a specific topic to understand the concepts in depth and to further accomplish a similar work based on the knowledge gained. This process of Systematic Literature Survey is very time consuming and needs resources to get the information. The proposed method tries to downscale the time consumed in finding the related research papers and reduces the effort of reading the entire paper by allowing to question the research paper.
- **Relevance to AI**: The approach involves the utilisation of Large Language Models which are built on top of Generative Pre-Trained Transformer for answering the questions related to the research paper or journal.

Literature Review:

Over the past few years, there has been a surge in interest regarding the creation of large language models using transformer architectures for the field of generative AI. The remarkable efficiency of these models has greatly simplified linguistic tasks and text summarization problems, making them significantly more manageable.

In recent years, the utilization of natural language processing (NLP) has greatly enhanced our interaction with complex legal terminology and contexts. In the paper titled "Towards the Exploitation of LLM-based Chatbot for Providing Legal Support to Palestinian Cooperatives," [5] authors present their work on a cooperative-legal question-answering LLM-based chatbot. The authors developed a series of legal questions pertaining to Palestinian cooperatives and their regulations, comparing the automatically generated answers by the chatbot to those formulated by a legal expert. To evaluate the performance of the proposed chatbot, 50 queries generated by the legal expert were used, and the answers provided by the chatbot were compared to their respective relevance judgments.

Another similar work titled "Building a deep learning-based QA system from a CQA dataset" [6] addresses the challenge of obtaining high-quality, well-structured machine-reading comprehension (MRC) datasets for training Question Answering (QA) systems. The authors highlight that such datasets are typically scarce in the real world and can be expensive to create or update. To overcome these limitations, the paper proposes a QA system that leverages a large-scale English Community Question Answering (CQA) dataset sourced from Stack Exchange, containing over 3 million question-answer pairs. The system follows a classifier-retriever-summarizer structure, utilizing a Bidirectional Encoder Representations from Transformers (BERT) NLP model for question classification and answer retrieval, and a Text-to-Text Transfer Transformer (T5) deep learning model for summarizing lengthy answers. research marks a significant advancement in the field of open-domain or specific-domain QA systems.

Project Objectives/Deliverables:

1. User-Topic Input and Relevant References
   - Develop a user interface to prompt users to enter a topic of interest related to biomedicine or life science.
   - Implement a search mechanism to extract 10 relevant research papers or references from the PubMed database based on the user's input.
2. Display of Intermediate Results
   - Design a table format to present the extracted references' details, including the title, authors, published date, and a link to each reference.
   - Develop a mechanism to display this table to the user, providing an overview of the retrieved references.
3. Abstract Extraction and Similarity Identification
   - Implement a process to extract the abstract from each paper or reference obtained from PubMed.
   - Utilize a similarity calculation method to determine the degree of similarity between the user-entered topic and the references' abstracts.
   - Generate a list of the top 5 references based on similarity and present them to the user as suggestions.
4. Question Answering System
   - Create a user interface to prompt the user to select a reference from the displayed list.
   - Implement a Question Answering (QA) system that can provide answers to user questions based on the content of the selected research paper or journal.

These objectives are sufficient to achieve the project aim as they cover the essential steps required to facilitate effective literature search and knowledge extraction in the field of biomedicine and life sciences. The first objective ensures user engagement by allowing them to input their topic of interest, enabling personalized research exploration. The second objective addresses the retrieval of relevant references from the PubMed database, providing a solid foundation for further analysis. The third objective focuses on extracting abstracts and identifying similarity, allowing for efficient content evaluation and comparison. The fourth objective incorporates a question answering system, enabling users to gain deeper insights from selected references. Additionally, the suggestion of additional references based on similarity enhances the user's exploration and promotes comprehensive understanding of the topic. Overall, these objectives provide a well-rounded approach to accomplishing the project aim of assisting users in finding and comprehending research papers and references in the field of biomedicine and life sciences.

Methodologies:

Below points include the steps in the methodology of the project

1. User input for the topic: Ask user to enter the topic of interest related to biomedicine or life science for finding the research papers and references related to it.
2. Relevant references: Find and extract 10 relevant papers or references of the topic entered by user from PubMed database.
3. Display the intermediate results: Display a table to the user containing with Title, Authors, published date, and link to the extracted references.
4. Extract the abstract: Fetch the abstract from each paper or reference.

5. Identifying the Similarity: Find the similarity between the user entered topic and the references from the PubMed database using abstract.
6. Suggesting the references: Display the top 5 references to the user based on similarity.
7. User input for QA system: Ask user to select the reference on which user wants to run the Question Answering system.
8. QA system: Answer the questions of the user based on the content of research paper or the journal.
9. Evaluation of the QA system: Evaluating the QA system by performing user testing to find the relevancy of the answers with regards to the user's question.

Project plan:

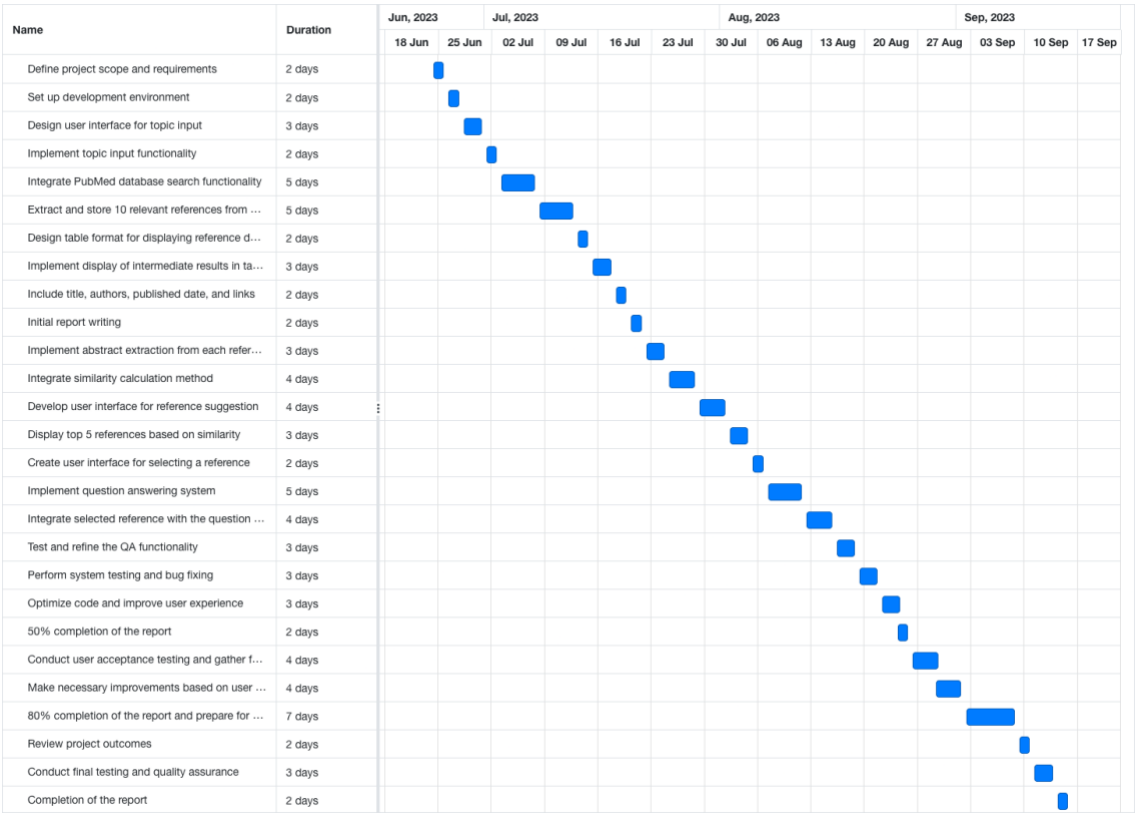
- **Feasibility**: My skills and expertise as a Data Engineer with 4 years of experience, particularly in Python, makes me well-suited to complete the project within the given timeframe. My proficiency in Python will be beneficial in developing the necessary functionalities, such as implementing user input, integrating with databases like PubMed, extracting abstracts. My experience in prompt engineering and working with large language models will provide me with the necessary knowledge and understanding to leverage the capabilities of the language model for tasks like similarity calculations and generating relevant suggestions. Additionally, MSc in Data Science further enhances your understanding of data analysis and manipulation, ensuring you have the skills needed to handle the project's data components. With my background and available hardware and software resources, I will be able to successfully complete the project within the designated timeframe.
- **Resources**:
    - No need of school GPU resources.
    - No need of access to particular robots and sensors, or specialised software, other than those normally available in the school.

Below is the project plan as per the initial assessment of the tasks and duration for the project. Please note that the actual duration of each task may vary based on the project's complexity and available resources.

Week	Dates	Tasks
Week 1	June 24 - 30	- Define project scope and requirements - Set up development environment - Design user interface for topic input
Week 2	July 1 - 7	- Implement topic input functionality - Integrate PubMed database search functionality
Week 3	July 8 - 14	- Extract and store 10 relevant references from PubMed - Design table format for displaying reference details
Week 4	July 15 - 21	- Implement display of intermediate results in table format - Include title, authors, published date, and links - Initial report writing

Week 5	July 22 - 28	- Implement abstract extraction from each reference - Integrate similarity calculation method
Week 6	July 29 - Aug 4	- Develop user interface for reference suggestion - Display top 5 references based on similarity
Week 7	Aug 5 - 11	- Create user interface for selecting a reference - Implement question answering system
Week 8	Aug 12 - 18	- Integrate selected reference with the question answering system - Test and refine the QA functionality
Week 9	Aug 19 - 25	- Perform system testing and bug fixing - Optimize code and improve user experience - 50% completion of the report
Week 10	Aug 26 - Sep 1	- Conduct user acceptance testing and gather feedback - Make necessary improvements based on user feedback
Week 11	Sep 2 - 8	- 80% completion of the report and prepare for final user testing
Week 12	Sep 9 - 15	- Review project outcomes - Conduct final testing and quality assurance - Completion of the report

Gantt chart



Explanation of Gantt chart

- Over the course of the project, starting from June 24th, 2023, until September 10th, 2023, the following tasks will be undertaken.
- In the first week, the project scope and requirements will be defined, and the development environment will be set up.
- During the second week, the user interface for topic input will be designed, and the integration of PubMed database search functionality will be implemented.
- In the third and fourth weeks, the focus will be on extracting 10 relevant references from PubMed and designing a table format for displaying reference details such as title, authors, published date, and links.
- Weeks five and six will involve extracting abstracts from the references and implementing a similarity calculation method to identify similarities.
- The seventh week will see the development of a user interface for reference suggestion and the display of the top 5 references based on similarity.
- In week eight, the question answering system will be implemented, and integration with the selected reference will be completed.
- Weeks nine and ten will be dedicated to testing, bug fixing, optimization, and gathering user feedback for necessary improvements.
- The eleventh week will focus on finalizing project report and preparing for final user testing.
- Followed by the twelfth week, which will involve reviewing outcomes, conducting final user testing, and ensuring quality assurance along with completion of the report.

Risks and contingency plan:

- PubMed is a free database of references on life sciences and biomedical topics. The problem that might hinder the project could be the unavailability of access to all the published papers and authorisation to fetch the content of some papers in PubMed database.
- Some authors publishing on databases like PubMed do not allow their content to be accessed for free. The phase of extracting the full content of each research papers from PubMed could be quite difficult.
- In case of any problems, the plan is to search for the relevant reference or journals other than PubMed for extracting the content related to a topic. If this plan does not work as well, the contingency plan is to build a QA system on University of Birmingham

Hardware/Software Resources

1. Hardware Requirement:
    - RAM - 8 GB
    - STORAGE - 100 GB
2. Software Requirement:
    - OS – Windows/Mac/Linux
    - Python 3.9/+
    - Jupyter Notebook/VS Code
    - OpenAI account for API key
3. Student and Supervisor have access to these resources.

Data

- The datasets for the project are the research papers fetched from the PubMed database.
- Student and Supervisor have the access to this data.

References

[1]	Qasem, R., Tantour, B. and Maree, M. (2023). Towards the Exploitation of LLM-based Chatbot for Providing Legal Support to Palestinian Cooperatives. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2306.05827.
[2]	Jin, S., Lian, X., Jung, H., Park, J. and Suh, J. (2023). Building a deep learning-based QA system from a CQA dataset. Decision Support Systems, [online] p.114038. doi:https://doi.org/10.1016/j.dss.2023.114038.
[3]	National Library of Medicine (2021). PubMed Labs. [online] PubMed Labs. Available at: https://pubmed.ncbi.nlm.nih.gov/.
[4]	python.langchain.com. (n.d.). *FAISS | 🦜🔗 Langchain*. [online] Available at: https://python.langchain.com/docs/modules/data_connection/vectorstores/integrations/faiss [Accessed 21 Jun. 2023].
[5]	python.langchain.com. (n.d.). *OpenAI | 🦜🔗 Langchain*. [online] Available at: https://python.langchain.com/docs/modules/model_io/models/llms/integrations/openai [Accessed 21 Jun. 2023].
[6]	developer.ieee.org. (n.d.). *IEEE Xplore - Python Software Development Kit*. [online] Available at: https://developer.ieee.org/Python_Software_Development_Kit [Accessed 21 Jun. 2023].
[7]	PRATIBA, D., M.S., A., DUA, A., SHANBHAG, G.K., BHANDARI, N. and SINGH, U. (2018). Web Scraping And Data Acquisition Using Google Scholar. 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS). doi:https://doi.org/10.1109/csitss.2018.8768777.
[8]	scholarly.readthedocs.io. (n.d.). *Quickstart — scholarlyORG 1.0b1 documentation*. [online] Available at: https://scholarly.readthedocs.io/en/stable/quickstart.html [Accessed 21 Jun. 2023].