IR PROJECT PROPOSAL

Project Repository: https://github.com/Algorhythmics/Digital-Library

Authors: Aditya Daipuria , Ankit Kumar , Harshit Sharma , Mohammad Kaif , Navvrat Rao , Pavit Singh

Have you ever tried to find research papers related to a certain topic? You search the web and find a few digital libraries showing matches. You glance at the title and the abstract, and the paper seems relevant, so you decide to read it. However, after you're done reading the paper; you realise it talks about something completely unrelated, and you've wasted a lot of time.

We address the challenge of inefficient paper assessment in digital libraries, where users may invest time in irrelevant papers due to incoherent abstract information. Our solution involves providing a holistic summary of the paper content after a ranked retrieval of the papers, including techniques and results, to facilitate quick relevance evaluation. Additionally, we integrate graphical representations such as word clouds to enhance user comprehension. By combining concise summaries with visual aid, we aim to streamline the research process and empower users to make informed decisions efficiently.

There were some related works to research paper information retrieval systems, which are based on digital media libraries

The paper by Li, Wang, and Zhang (2016) discusses the development of a **Full Text Retrieval System (FTRS)** for digital libraries. The authors aim to overcome the limitations of existing information retrieval (IR) methods in digital libraries, such as low efficiency and relevance of search results. They propose a FTRS that integrates three IR models: the Boolean model, the vector space model (VSM), and the language model.cite:p1

The paper **"Multimedia Digital Information Retrieval System"** focuses on the retrieval of digital data, analyzing data reduction, and document storage. It centers around the **MMDBMS architecture**, which handles large sets of digital media. The system aims to efficiently retrieve relevant information from multimedia content.cite:p2

The publication **"Rule Based Metadata Extraction Framework from Academic Articles"** addresses the critical task of extracting and managing metadata from scientific articles. This includes elements such as titles, abstracts, keywords, body texts, conclusions, and references. Despite the importance of academic social networks and digital libraries, existing services often have limitations, including cost, open-source availability, performance issues, and restrictions on processing PDF files. To overcome these challenges, the proposed Java-based framework offers high performance, flexibility (unlimited PDF uploads), and open-source availability.cite:p3

Our idea differs from existing approaches in digital libraries by providing a summary of top-ranked search results using an LLM, enabling users to filter and understand relevant information efficiently. While search engines like Bing and Google utilize AI to extract relevant information from queries, databases like PubMed and ArXiv lack such functionalities.

In order to solve this problem, we plan on using the following techniques/algorithms:

Data Collection:

- **Web Scraping:** We are going to collect data from various academic databases and journal websites.(Dataset-tensorflow)
- **APIs Utilization:** Utilize APIs provided by academic databases like PubMed, Google Scholar, arXiv, etc., for accessing research papers programmatically.

Ranking:

- **Vector Space Model:** We will represent documents and queries as vectors in a high-dimensional space and compute their cosine similarity.
- **BM25:** A probabilistic retrieval model that considers term frequency and document length normalization.
- **Collaborative Filtering:** Filter out results that a user might like based on reactions by similar users.

Summarisation:

- **Extractive Summarization:** Select important sentences or phrases from the document to form a summary. For example, we can use techniques like TextRank or graph-based algorithms for this purpose.
- **Abstractive Summarization:** Generate a summary by paraphrasing and rephrasing the content of the document using neural network-based models such as LLMs (Large Language Models) like GPT (Generative Pre-trained Transformer) or BERT (Bidirectional Encoder Representations from Transformers) or LLAMA.

Frontend Development:

- **Web Development Frameworks:** Using frameworks like Flask (Python), Express (Node.js), or Django (Python) for building the front end of the search engine.
- **User Interface Design:** A user-friendly interface to input search queries and display search results and summaries effectively

We plan on evaluating our work according to the parameters mentioned below:

1. **Accuracy of the trained language model:** Whether or not the provided results after a query search are relevant to the users. If the search results are satisfactory, whether the summary provided by the model is sufficiently detailed or not.
2. **User Feedback:** A user can choose to rate each search query according to the results provided by our application.
3. **Response Time and System Scalability:** Whether the designed application is responsive enough to provide the results of a search query in a reasonable time. Furthermore, How will the application deal with new data? We can assess the system's ability to handle an increasing number of queries, papers, and users without significant degradation in performance.

1 and 3 are quantitative metrics, while 2 is qualitative, which might also include usability testing in the later part of our project.