Final Report

A Wiki Search Engine for Space Physics Powered with LLM

Yiwen Zhu (yz167), Yan Zhang (yz238)

The repository is: https://github.com/YiwenZhu77/comp631

1. Introduction

The goal of this project is to implement an Information Retrieval system that is specially tuned and designed for the discipline of space physics. Space physics is the study of investigating the physical processes in the space environment, including the solar system and beyond. We choose this topic for two reasons. First, the author of this project is a space physics researcher and has a lot of experience in this field. Second, the space physics community has a lot of data and information that is not well organized and indexed. A search engine that is specifically designed for space physics can help researchers find the information they need more efficiently.

Space physics is a relatively small realm of academia, and the information retrieval system that is designed for general purposes may not work well for space physics. For example, the space physics community has a lot of jargon and acronyms that are not commonly used in other fields. Also, there are not many handy databases specifically designed for space physics, and the most reliable and reachable resources are often the papers published in journals. Thus, researchers will constantly find themselves buried with numerous papers and reports, and it is very time consuming to find the information they need. Especially for the new researchers who are not familiar with the field, it is very hard for them to find the information they need.

It is often the case in the research process that researchers just need to get a broad idea of a topic, and they do not need to read the whole paper. However, the lack of a handy database prohibits them from doing so. A way out is to use Wikipedia, which is a very handy and well-organized database. However, the space physics on Wikipedia is not well contributed and organized. The author's personal experience is that the Wiki pages about space physics he found on Wikipedia are not accurate and confusing.

All the factors motivated this project, i.e., the small nature of the space physics community, the lack of a handy database, and the inaccuracy of the Wiki pages about space physics. Those factors make it necessary and beneficial to design a search engine that is specifically designed for space physics. Moreover, the simple search-and-return system is not enough for a full academic understanding, and thanks to the recent development of the large language model (LLM), we can use the large language model to interact with the results. This search engine will be a good tool for the space physics community to find the information they need more efficiently and get a deep insight into the topic they are interested in.

2. Prior Work

There is no previous work with the exact same goal. However, there have been some attempts to build up a collective database for the space physics community. The most important one is arguably the NASA Astrophysics Data System (ADS) [1]. The ADS is a Digital Library portal for researchers in Astronomy and Physics, operated by the Smithsonian Astrophysical Observatory (SAO) under a NASA grant. The ADS maintains three bibliographic databases containing more than 14.2 million records covering publications in Astronomy and Astrophysics, Physics, and the arXiv e-prints. The ADS is a powerful tool for researchers to find the papers they need. However, the ADS is not specifically designed for space physics, and the search engine is not very user-friendly. The ADS is

more like a database that contains all the papers in the field, and it is not very easy to use for new researchers who are not familiar with the field.
There is an overview of the ADS system [2]. The paper listed the purpose of the ADS system, the data sources, the data processing, the user interface, and future development in detail. The paper is very informative, and it gives a good overview of the ADS system. The ADS system is a very powerful tool for researchers to find the papers they need. However, the ADS system is limited in its comprehensive nature, and it is very hard to quickly dig out the niche information that space scientists need. The search engine is not particularly user-friendly either; researchers still must go through the whole process of going through many related papers and digesting pieces of information hidden in multiple resources in order to obtain a deep understanding of the topic.

To counter these short comes of the ADA system, this project specifically aims for these demands: 1) to build a search engine that is specifically designed for space physics, 2) to use the large language model to help researchers chat and interact with the results, 3) to provide a handy tool for researchers to find the information they need more efficiently and get a deep insight of the topic they are interested in.

To sum up, the project is related to the previous work in the sense that it is trying to build a search engine that is specifically designed for space physics. The project is different from the previous work in the sense that it is trying to use the large language model to help researchers chat and interact with the results.

3. Model/Algorithm/Method

This is where you give a detailed description of your primary contribution. It is especially important that this part be clear and well-written so that we can fully understand what you did.

The project is divided into two parts: the search engine and the large language model (LLM). The search engine is designed to search the space physics papers and return the results. The LLM is designed to help researchers to chat and interact with the results.

For the search engine, we used Wikipedia pages to discuss space physics. Wikipedia offers an API set called MediaWiki [3], with which users can access Wikipedia pages based on certain criteria and needs. This functionality will serve as our primary way of crawling documents from Wikipedia. The crawler app can be found at /src/crawler.ipynb in the project repository.

After the crawling, we then performed the data processing for the documents database. The purpose is to clean the unwanted data and make the documents more organized. To the specific, we deleted some irrelevant documents, such as those recording the technical details of a mission or other information that we do not want to be present in the search engine. We also altered some content and structure of the pages to make them more readable and user-friendly. Examples of these include translating some non-English language to English and manipulating some inaccurate descriptions. This process is mostly done manually; the data-processing app can be found at /src/dataFormat.ipynb in the project repository. After the processing, the documents database has around 44k documents.

The next step was to index the document database and make a search engine based on it. The search engine is built with Apache Solr, which is a fast, scalable, and open-source search platform [4]. Solr is written in Java and uses the Lucene search library. The instance of the search engine is stored with a Solr Core named "test." We uploaded the database to this core with a Solr Handler via the CURL command. After documents have been uploaded, Solr performs the indexing, and the search engine is ready to use. The search engine app can be accessed on the local network through http://localhost:8983/solr/.

In the Solr search interface, you can search the documents with a set of rules for the user's needs. The search engine will return the documents that are most relevant to the query. Users can also customize the results in terms of the number of results, the order of the results, and the format of the results.

For the LLM part, we employed the GPT 3.5 model. GPT-3.5 is a large language model that is trained on a large corpus of text data. The model is trained to predict the next word in a sentence given the previous words. The model is trained on a large corpus of text data, and it is able to generate human-like text. This chat model is the backend of our interactive search engine design.

Specifically, the Solr will return the ten most relevant results matching the user's query, and the Gpt 3.5 model will give a relatively short and precise description and summary of the query based on the top ten documents that match the query. Users can also ask follow-up questions to the chat and interactively learn the aspect of the information provided in the matching documents. This way, the user can get a deep insight into the topic they are interested in.

The chat based on the search engine is implemented on a web-based front end, and users on the local network can access the chat interface through http://localhost:5000/. Figure 1 shows an example of the chat interface.

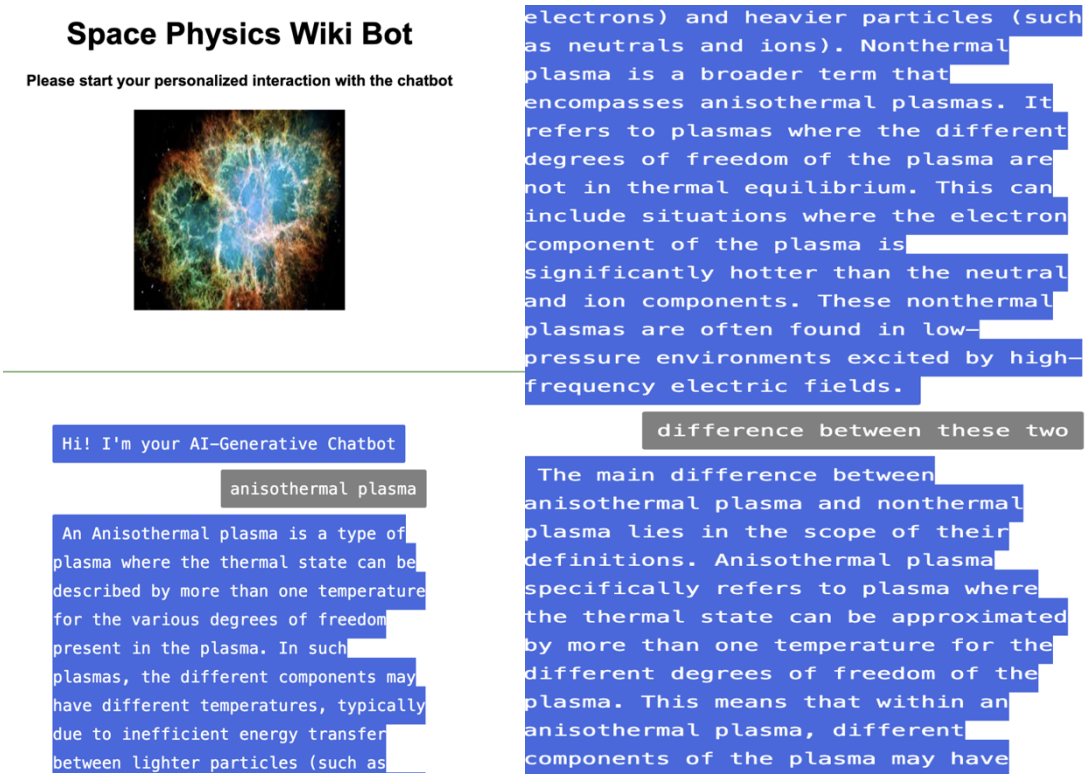

Figure 1: The chat interface of the search engine. The user can input the query in the input box, and the chat will return the results based on the query. The user can also ask follow-up questions to the chat (as shown on the right panel).

There are choices of the algorithms for searching and evaluation of the search engine, which will be covered in the results and findings section.

4. Results and findings

The main results are the demonstration of the search engines and evaluation of the search results based on different search engine settings.

We first demonstrated that the search engine would give us desirable results and that the chat would conduct a sensible dialogue. We tested several common words as queries (see Table 1), and the search engine returned the results that were most relevant to the query. Moreover, the following chat about these words provides a very accurate insight and explanations. This proves the validity of the search engine and the chat.

Query Words
Magnetosphere
Solar Wind
Aurora
Van Allen Belt
Ionosphere
plasma

Table 1: The common words that are used as queries in the search engine. The search engine returned the results that were most relevant to the query.

Then, we used evaluation matrices to test the accuracy and relevance of the search engine results. With limited time and manpower, we managed to manually examine only six query-document pairs as the ground truth; the query words are listed in Table 1. We used the Precision to assess the query-document results. As shown in Table 2, all the Precision is over 0.7 (out of 10 top matching documents), meaning a satisfactory performance of the search engine.

Query Words	Precision
Magnetosphere	0.8
Solar Wind	0.9
Aurora	0.7
Van Allen Belt	0.8
Ionosphere	0.7
plasma	0.8

Table 2: The Precision of the search engine based on six query-document pairs. The Precision is over 0.7 for all the queries, meaning the search engine performs satisfactorily.

Regarding our needs, the precision matrices are appropriate, as we are more interested in the quality of the results than recovering all the matching results.

We also experimented with different search engine settings and evaluated their results using MAP matrices. We trialed four settings: 1. Rank Model BM25 with the search field 'Title'; 2. Rank Model BM25 with the search field 'Content'; 3. Rank Model IF-IDF with search field 'Title'; 4. Rank Model IF-IDF with the search field 'Content.' The MAP results are shown in Table 3. The results show that the Rank Model BM25 with the search field 'Title' has the best performance, with a MAP of 0.723. This setting is used as the default setting of the search engine.

Rank Model	Search Field	MAP
BM25	Title	0.723
BM25	Content	0.678
IF-IDF	Title	0.701
IF-IDF	Content	0.654

Table 3: The MAP results of different search engine settings. The Rank Model BM25 with search the field 'Title' has the best performance, with a MAP of 0.723.

5. Conclusion

In this project, we built a search engine that is specifically designed for space physics. The search engine is built with Apache Solr, and it can search for space physics papers and return results. The search engine can return the results that are most relevant to the query. We also built a chat interface that is powered by the large language model (LLM). The chat interface is able to chat and interact with the results. The chat interface can provide a deep insight into the topic that the user is interested in. We used six query-document pairs to evaluate the search engine, and the results showed that the search engine had high Precision. We also experimented with different search engine settings and evaluated their results using MAP matrices. The results show that the Rank Model BM25 with the search field 'Title' has the best performance, with a MAP of 0.72.

Bibliography

1.Astrophysics data system (no date) NASA/ADS. Available at: https://ui.adsabs.harvard.edu/. (Accessed: 12 April 2024).
2. Kurtz, M.J. et al. (2000) 'The NASA Astrophysics Data System: Ov e, 'iew,' Astronomy and Astrophysics Supplement Series, 143(1), pp. 41–59. doi:10.1051/aas:2000170.
3. projects, C. to W. (2023) Wikimedia Project Page, MediaWiki. Available at: https://www.mediawiki.org/wiki/MediaWiki (Accessed: 12 April 2024).
4. Shahi, D. (2015) 'Apache Solr: An introduction', Apache Solr, pp. 1–9. doi:10.1007/978-1-4842-1070-3_1.