WOAS World Organization of Applied Sciences Journals

International Journal of Information Technology and Applied Sciences
ISSN (2709-2208)
Int. J. Inf. Tec. App. Sci. 4, No.2 (May-2022)
https://doi.org/10.5281/zenodo.6602347

EzPubMed: A User-friendly Snapshot of NCBI PubMed Repository

Sakib Rahman[1], Nafis Imtiyaz Anim[2] and Sarker T. Ahmed Rumee[3]

[1] Traideas, Dhaka, Bangladesh
[2,3] Department of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh

Abstract: Searching educational and research content has been made easier by popular online repositories such as NCBI, google scholar, etc. For a user query, they generally return thousands of results and also rank them. However, these tools often produce results that are distantly related to user query goals and often overwhelm the users with the volume of results. In recent years, researchers are experimenting with content-specific databases and thereby producing few but more relevant query results. This project creates an exact snapshot of the PubMed database from NCBI and provides users with novel search options that sort and display the results on the basis of matching percentages between query keywords and article titles. The proposed system uses Natural Language Processing (NLP) techniques to aid users to form search queries in plain English that would otherwise only be possible through setting up advanced search options. The experiment results were promising and found to produce more relevant query results compared to the state-of-the-art methods.

Keywords: Crawler, Data Repository, Domain-Specific Search Engine, Query Response Filtering.

1. INTRODUCTION

Online searching of data has been made quite easy now-a- days by popular search engines such as Google, Bing etc. Apart from that, search tools tailored to specific needs have also become hugely popular. For example, for finding a research article we take the help of tools like Google Scholar [1], NCBI [2] etc.

However, users often get overwhelmed with the amount search results for a query these popular tools produce. Existing systems often do not have suitable filtering option though they provide thousands of information. As a result, the need for content specific databases and associated search tools are becoming more and more popular, where users only get information very much close to their queries.

For example, searching for a paper in NCBI return thousands of related papers. In NCBI, we can sort these results according to publication type only but we cannot filter results on the basis of journal name in which the paper was published, for finding papers users need to visit all the pages. NCBI also does not have any feature for generating bibtex for a specific paper.

To facilitate users with relevant results quickly and efficiently, this work focused on finding research articles compared to google scholar and NCBI, especially the NCBI Pubmed database, we propose the EzPubMed System in this work. It is a dedicated snapshot of the NCBI PubMed [3] database. This has a search interface similar to PubMed and have additional features such as: various method to filter query responses in terms of journal, author name, and publication year (or year range). It also adds the bibtex (citation information to be used by latex doucments) generation option like the google scholar [1]. The core of the system is a crawler, which is always running in the background and making copies of the whole NCBI PubMed repository. Due to its structure, EzPubMed produces search results very quickly if the sought information is already crawled, which is the case for most search queries. Otherwise, it will collect the information from the NCBI repository and also store a copy in its local data store.

The major contributions of this work are summarized below:

- Created a database (EzPubmed) which is a snapshot of the PubMed section of NCBI by crawling the PubMed data in a local repository.
- A user friendly interface with useful search filters through advanced search option is also developed as the front end of this dedicated data store.
- Using Natural Language Processing tools, intelligent search filter options were added which aid users to form query in natural language which is otherwise absent in state of the art tools.
- Implemented a data crawler that automatically updates the database after a certain time, which has the promise to be used in other domain too.

The rest of the paper is organized as follows: key terms and definitions are mentioned in section 2. Section 3 briefly discuss some closely related work. Then section

4 gives a detail description of the proposed system with the experimental evaluations presented in section 5. Finally, section 6 concludes the paper with some direction to future work.

2. BACKGROUND

Here some key terminologies and relevant systems which formed the basis of this work are discussed. Our implementation largely relies on using the NCBI Entrez Utilities to extract and organize publication information of relevant keywords, which is also briefly introduced to the readers.

A. NCBI

NCBI [2] stands for National Center for Biotechnology Information. It advances science and health by providing an access to biomedical and genomic information. There are 6 types of operations that can be performed by a user in the NCBI web portal.

A user can deposit data and manuscripts to the NCBI databases using the 'submit' operation. An NCBI data can be transferred from the NCBI site to the user's devices, such as computers, laptops, mobile, etc. by using the 'download' operation. A user can also be helped by the NCBI to find documents, attend a class or even watch the tutorials by using the 'learn' option.

Users can also use NCBI tools and code libraries to build any application by using the 'develop' operation. An NCBI tool can be identified by a user to analyze a task by using the 'analyze' operation. A user can also explore NCBI research and collaborative projects by using the 'research' operation. NCBI has resources sorted in lexicographical order. Some of them are chemical and bio-assays, data and software, DNA and RNA, domain structures, gene expression, and genetics medicine. The popular resources among the users include PubMed, Bookshelf, PubMed Central, Blast, Nucleotide, and Genome.

B. PubMed

In this work, we used PubMed [3] of NCBI to create a local repository named EzPubMed. PubMed comprises more than 30 million citations for biomedical literature from MEDLINE, life science journals, and online books. Using PubMed, we can get full-text articles. We can also search using it. While searching on PubMed, at first a database is selected which is found from the search query or keyword. The related data are shown from that database. We can get a very huge amount of data which cam filtered too. For example, if there is 'dengue' in the search query, then the data obtained from the database are filtered by articles, publish date, and whether it is for humans or animals.

But it has some disadvantages too. Since the data amount is huge, only these filters are not enough to get the expected results. Therefore, more filters are required to be added. Moreover, it is also not possible to generate citation information like BibTeX directly from PubMed.

C. Entrez Unique Identifier (UID)

Data records stored in NCBI databases can be accessed by specifying their unique ID (UID). UIDs for a sequence record can easily be obtained by supplying an accession number from a source database, or a full Boolean query on indexed terms can be evaluated to return a list of UIDs that satisfy the query. Given a UID, a single function call will load the record into a defined structure in memory, and return a pointer to the head of the memory object. UIDs are generally organized in FASTA format [11].

D. NCBI Entrez Utilities

E-Utilities [4] stands for Entrez Programming Utilities. They are a set of nine server-side programs that provide a stable interface to the Entrez query and database system at NCBI. These programs include ESearch, EFetch, ESummary, EPost, EInfo, ELink, ESpell, ECitMatch, and EGQuery. These programs use a fixed URL syntax that translates a standard piece of input parameters into the values necessary for various NCBI software components to search for and retrieve the data that a user has requested.

To access the data, the software first posts an E-Utility URL to NCBI, then retrieves the results of this posting, after which it processes the data as required. In our work, we have used three E-Utilities which are ESeach, EFetch, and E-Summary which can be used together by the pipeline of ESearch - EFetch/ESummary.

Two parameters are required for performing E-Search: 'DB' and 'term'. The parameter 'DB' is the database to search. Its value must be a valid Entrez database name. 'Term' is Entrez text query. For example, to find 'dengue' related publications and data from PubMed, we construct the query option as *db = PubMed* and *term = dengue*.

Similarly, four parameters are required for performing E-Summary. They are 'db', 'id', 'querykey' and 'WebEnv'. Like E-Search, the database item will be denoted by 'db' and 'id' represents the UID list constructed from the query. The parameter 'querykey' is used when the input is read from the Entrez History server. It specifies which of the UID lists attached to the given Web Environment will be used as input to EFetch. Query keys are obtained from the output of previous ESearch, EPost, or ELInk calls. The 'querykey' parameter must be used in conjunction with WebEnv, which returns the web environment that contains the UID list to be provided as input to EFetch.

3. RELATED WORK

The work that is closest to this research is SVDB [5]. SVDB stands for Snake Venom Database which is a comprehensive domain specific database of snake venom toxins generated through NCBI.

SVDB has been described for storage, dissemination and analysis of snake venom and toxin related information which has autonomous links to NCBI databases. It can pull relevant information, both on-demand and asynchronous ways to facilitate data integration. When anyone searches using SVDB website, he gets data that are related to snake venom. All the data in SVDB are authorized, verified and reliable. We can filter the results according to dates and name. However, there is no suitable filtering mechanism in SVDB. So when the query responses include thousands of data as results, a user might get overwhelmed.

dbSNP [6] is the NCBI database of genetic variation. dbSNP has been serving as a central and public repository for genetic variation. Once such variations are identified and cataloged in the database, additional laboratories can use the sequence information around the polymorphism and specific experimental conditions for further research applications. dbSNP can classify the nucleotide sequence into 6 variations. It can be searched directly or via NCBI resources. However, it also suffers from the problem such as – not having a good search interface and lack of query result ranking and selection.

Disbiome database [7] collects and presents published microbiota-disease information in a standardized way. it is the first database that gives a clear, concise and up-to-date overview of the microbial composition, differences in diseases, together with the relevant information of the studies published. The strength of it lies within the combination of the presence of references to the other databases which enables both specific and diverse search strategies within the disbiome database and the human annotation which ensures a simple and structured presentation of the available data. The user interface of the website is quite good for a new user. However, the website takes very long time to load data and query response is often limited to incomprehensive data.

RicyerDB [8] stands for Rice Yield-related database. It is a database for collecting rice yield-related genes with biological analysis. It is quite efficient to use since the results can be sorted in ascending and descending order and a user can choose his/her required field only by using 'advanced search' option. The website is also dynamic. However, it too suffers from slow web performance and relies heavily on the user's domain specific knowledge to get the expected results fast,

Other closely related work building domain specific data repository are: NCBI Geo [9], NCBI Taxonomy [10], qPrimerDB [12], dNTPpoolDB [13], etc.

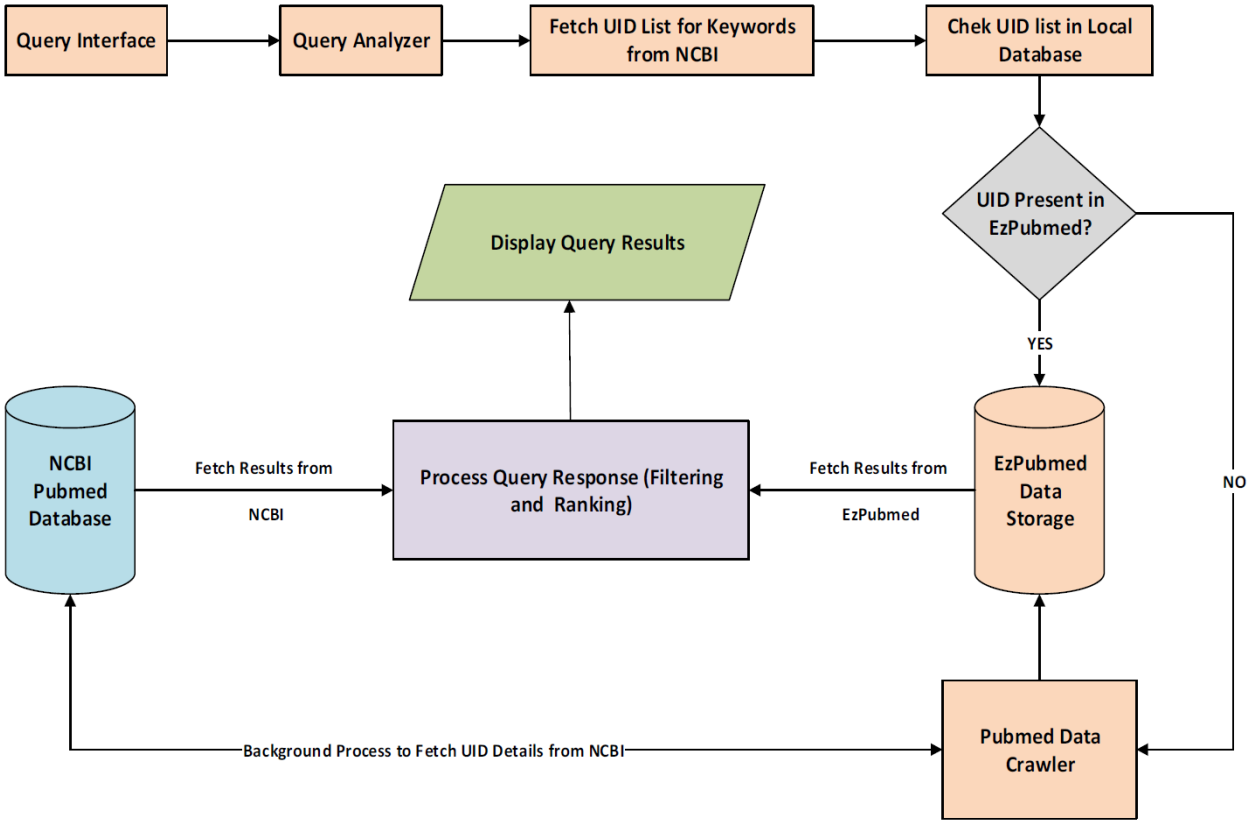

Figure 1. EzPubMed System Overview

4. System Design

Figure 1 depicts a high-level overview of the various components of the proposed system and their interaction. The details of these components are described below.

A. Query Interface Design

We closely followed the PubMed search engine and designed the landing page with minimal options and the relevant additional search options.

Users can search keywords or a query string (consisting of a set of keywords) and EzPubMed returns the results ranked by their relevance.

UID_LIST based on the extracted keywords from NCBI. The details of these steps are highlighted in Algorithm 1.

At first the query string is handled through the procedure named *PROCESS_SEARCH_STRING*. This method will create a modified query for further processing using the method *EXTRACT_KEYWORDS*.

EXTRACT_KEYWORDS method employs some basic natural language processing (NLP) toolkits to find out relevant keywords (paper title, author, journal/book name etc.) from the input query. Then these extracted keywords are appended together to form a modified query string which is devoid of unnecessary words, phrases or articles. This modified query makes it easier for the UID search

Algorithm 1 Analyze Search Query

```
1:  procedure PROCESS_SEARCH_STRING(query)
2:      modified_query ← EXTRACT_KEYWORDS(query)
3:      uid_list ← FETCH_UID_LIST(modified_query)
4:      return uid_list
5:  end procedure
6:
7:  procedure EXTRACT_KEYWORDS(query)
8:      keywords ← Keyword_Extract_Using_NLPToolkit(query)
9:      new_query ← {}
10:     for all word in keywords do
11:         if word is valid journal_name then
12:             new_query ← new_query + word
13:         end if
14:         if word is valid author_name then
15:             new_query ← new_query + word
16:         end if
17:         if word is valid year or year_range then
18:             new_query ← new_query + word
19:         end if
20:     end for
21:     return new_query
22: end procedure
23:
24: procedure FETCH_UID_LIST(modified_query)
25:     search_query_words ← tokenize(modified_query)
26:     for all word in search_query_words do
27:         uid_details ← UIDDetailsfromNCBIforword
28:         uid_list ← uid_list + uid_details
29:     end for
30: end procedure
```

B. Extract Data from NCBI PubMed Repository

Analyzing the user query involves two key elements – Extracting keywords from query and retrieve the process corresponding to the query keywords, which is a key component in deciding whether the searched data is already in local repository of EzPubMed.

FETCH_UID_LIST method takes input the modified query which was returned by the EXTRACT_KEYWORDS method. It then tokenizes the input string to isolate the words. Each word is a potential search key and its corresponding UID details are fetched next. In this way, the UID details (contains information to search for additional details from NCBI) of each word is collected and appended together to form the combined list named *uid_list*.

C. Check Local (EzPubMed) database for uid_list

EzPubMed will consult its already crawled local database (always updating) first to see if the UID and its information is already collected from the NCBI PubMed. If it finds in the local data store, then the NCBI is not consulted further and the results are returned immediately to the output.

Otherwise, the request will be sent to PubMed database. The response is not only sent to output, but also saved in local database for future enquiries.

D. Data Crawler Design

The data crawler works behind the scene and will run all the time in the background. As mentioned earlier, it works on the UIDs which are not yet been stored/found in the EzPubMed local database.

It will collect all such UIDs and the program will search for the paper of those IDs from the PubMed database. The summaries of those papers will be collected as an XML file. Then a function will translate the information in to text format and save in the secondary database. The crawler will store the information into three table of database system. At first, it will update the EzPubMed paper table. Then it will update the journal table and the author table. The main reason for storing these summaries is that we can collect only 400 hundred summaries at a time. If we do not store all the data, then the run time will be significantly high and there is a good chance of losing data. The crawler will run all the time in the background.

E. Process Query Results (Filtering and Ranking)

The query responses are showed in the results page with 25 results per page. We allow users to apply multiple filters to allow better readability and flexibility to the users.

The most common filters we used are: Journal name, Publication year and year range, Author name, Bibtex information availability.

For ranking or pruning unwanted query responses, these filters greatly helped.

5. RESULTS

EzPubMed is designed to provide quick domain-specific information to the researchers, which will be easier to investigate and analyze compared to using the NCBI PubMed.

With that in mind, we measure the performance of the proposed system in terms of query response volume and effectiveness (how quickly information is retrieved and displayed for user reference). We compare our results to the outcome obtained from NCBI PubMed and google scholar.

A. Query Response Volume and Relevance (EzPubMed vs PubMed and Google Scholar)

Table 1 shows the number of the total data collected from PubMed, Google Scholar databases, and EzPubMed for some example search queries. To select relevant query results, we checked which articles have equal to or more than 50% matching scores. Those who pass these criteria are displayed to the user and others are discarded.

Table 1. Comparison of Query Responses (Number of Results Returned)

Search Query	EzPubMed	NCBI (PubMed)	Google Scholar
BMC public health dengue	136	202	29200
BMC public health dengue 2000 to 2010	12	2	12400
Lancet dengue fever in human	3	115	45700
Ebola virus in human	2432	4991	160000
Virology Ebola virus in human	1000	1880	31500
Ebola virus disease in human and immunity	8	51	19000
A review on the antagonist Ebola: A prophylactic approach	1	2	1
Expert opinion on therapeutic targets implications of toll-like receptors in Ebola infection	1	1	1
Implications of toll-like receptors in Ebola infection	13	1	1
Rota virus	116	594	127000

We manually verified each result obtained in EzPubMed and all of them are found to be relevant. On the other hand, as we can see from Table 1, NCBI PubMed and Google scholar produce a lot of results that are filled with entries of little interest to the user.

B. Query Response Time (EzPubMed vs PubMed)

As EzPubmed runs a crawler in the background to mind the NCBI PubMed database, it will always return the response from its local data store (if the data is already

crawled) instead of collecting it from NCBI. So, it is expected to produce a much quicker response time. The number of results returned also plays an important role in reducing the query response time.

Table 2. Result Page Loading Time Comparison

Search Query	EzPubMed (sec)	NCBI – PubMed (sec)
BMC public health dengue	3.033	5.32
BMC public health dengue 2000 to 2010	2.19	5.33
Lancet dengue fever in human	2.16	5.38
Ebola virus in human	14.72	11.17
Virology Ebola virus in human	6.75	7.37
Ebola virus disease in human and immunity	2.47	9.70
A review on the antagonist Ebola: A prophylactic approach	2.47	5.25
Expert opinion on therapeutic targets implications of toll-like receptors in Ebola infection	2.24	5.58
Implications of toll-like receptors in Ebola infection	2.49	4.21
Rota virus	3.28	6.72

6. CONCLUSION

This paper developed a system that aids user with a new way of searching and displaying research articles. For that, as a proof of concept, the NCBI PubMed database was crawled and locally stored in a separate database. On top of that, intelligent search filters were added to produce more relevant results with the help of Natural Language Processing tools.

In future, we plan to add the following utilities to the EzPubMed System:

- The Cache database has very few paper summary, compared to the other similar repositories. More article summaries are scheduled to be crawled from NCBI.
- More filtering option can be added. By applying those filters, user will be able to search data more easily.
- Finally, the user interface also has areas of improvement to better help users find answer to their queries.

REFERENCES

[1] Google Scholar, https://scholar.google.com/ (Accessesd: November 1, 2020).

[2] National Center for Biotechnology Information, https://www.ncbi.nlm.nih.gov/, (Accessesd: April 25, 2021).

[3] PubMed, https://pubmed.ncbi.nlm.nih.gov/, (Accessesd: April 25, 2021).

[4] Kans, Jonathan. "Entrez direct: E-utilities on the UNIX command line." In Entrez Programming Utilities Help [Internet]. National Center for Biotechnology Information (US), 2022.

[5] Hossain, Md Monir, Ataul Haque, Sayeda Zannatul Sakina Mazid, Abira Khan, Tomalika Rahmat Ullah, and Sarker T. Ahmed Rumee. "Snake Venom Database (SVDB) A Potential Resource for Complementary & Alternative Medicine and Drug Designing." In Proceedings of the 2018 2nd International Conference on Computational Biology and Bioinformatics, pp. 32-36. 2018.

[6] Sherry, Stephen T., M-H. Ward, M. Kholodov, J. Baker, Lon Phan, Elizabeth M. Smigielski, and Karl Sirotkin. "dbSNP: the NCBI database of genetic variation." Nucleic acids research 29, no. 1 (2001): 308-311.

[7] Janssens, Yorick, Joachim Nielandt, Antoon Bronselaer, Nathan Debunne, Frederick Verbeke, Evelien Wynendaele, Filip Van Immerseel, Yves-Paul Vandewynckel, Guy De Tré, and Bart De Spiegeleer. "Disbiome database: linking the microbiome to disease." BMC microbiology 18, no. 1 (2018): 1-6.

[8] Jiang, Jing, Fei Xing, Xiangxiang Zeng, and Quan Zou. "RicyerDB: a database for collecting rice yield-related genes with biological analysis." International Journal of Biological Sciences 14, no. 8 (2018): 965.

[9] Barrett, Tanya, Tugba O. Suzek, Dennis B. Troup, Stephen E. Wilhite, Wing-Chi Ngau, Pierre Ledoux, Dmitry Rudnev, Alex E. Lash, Wataru Fujibuchi, and Ron Edgar. "NCBI GEO: mining millions of expression profiles—database and tools." Nucleic acids research 33, no. suppl_1 (2005): D562-D566.

[10] Federhen, Scott. "The NCBI taxonomy database." Nucleic acids research 40, no. D1 (2012): D136-D143.

[11] Pearson, William R. "Using the FASTA program to search protein and DNA sequence databases." In Computer Analysis of Sequence Data, pp. 307-331. Humana Press, 1994.

[12] Chang, Wei, Yue Niu, Mengna Yu, Tian Li, Jiana Li, and Kun Lu. "qPrimerDB: A Powerful and User-Friendly Database for qPCR Primer Design." In PCR Primer Design, pp. 173-182. Humana, New York, NY, 2022.

[13] Pancsa, Rita, Erzsébet Fichó, Dániel Molnár, Éva Viola Surányi, Tamás Trombitás, Dóra Füzesi, Hanna Lóczi et al. "dNTPpoolDB: a manually curated database of experimentally determined dNTP pools and pool changes in biological samples." Nucleic Acids Research 50, no. D1 (2022): D1508-D1514.