International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Scholarly: An Academic Search Engine using querying techniques Boolean Retrieval, Phase Retrieval and TF-IDF scoring

Dr. Kunal Meher[1]*, Beatrice S.[2], Kirti Motwani[3], Dr. Sushama Khanvilkar[4], Chhaya Dhavale[5], Dr. Jaychand Upadhyay[6]

Submitted: 28/01/2024 **Revised**: 06/03/2024 **Accepted**: 14/03/2024

Abstract: Students often find the need to search for professors based on various criteria such as name, university, research topics, top cited papers and rank them based on factors like citations or h-index. A simple Google search may not allow you to first shortlist professors based on whether they do research in “adversarial machine learning” and then rank them as per the number of citations that they have in the last 5 years. In this paper, a search engine publicly as a web application is proposed that can cater to the needs of students looking for professors to approach for projects, internships or jobs. The search engine allows users to search for professors based on name, university, research areas and paper titles using 3 different retrieval methods and sort the search results based on criteria like h-index, citations in the last 5 years etc.

Keywords: *Search Engine, h-index, Information retrieval, PageRank, TF-IDF.*

1. Introduction

Finding professors to apply for internships, projects or jobs is a common and time-consuming task among students. Students may value different things in such a context and it is hard to perform this search with existing tools. They often find the need to search for professors based on various criteria such as name, university, research topics, top cited papers and rank them based on factors like citations or h-index. In this paper, a simple search engine is proposed that allows users to search for professors based on name, university, research areas and paper titles using three different retrieval methods. It can help students to search for professors based on various

criteria such as name, university, research topics of interest, top cited papers and sort the search result based on criteria such as the number of citations or h-index. The best solution currently in use is Google Scholar, in which the Profile feature can search for professors. However, it does not allow you to sort or filter your search results based on any criteria. However, Google Scholar does have a huge collection of data on professors, researchers and their research. This data can be used to create a tool more focused towards students who are searching for professors with the help of multiple retrieval methods and criteria-based sorting of search results. Publishers, academic administrations, and others frequently use a range of quantitative metrics to rank academics, institutions, journals, papers, and also the entire nations. The foundation of a number of these quantitative measurements is academic research. The quantity of publications can be used to estimate production or productivity in the field of science, whilst the quantity of references to such journals might estimate the significance of the research. These “publishing metrics” have the possibility of benefiting the researchers as well as those assessing the performance of researchers when taken collectively. Metrics can help researchers organize their evaluation of their earlier work, create effective summaries of their prior research trajectory, and guide their decision-making in the future. Understanding a researcher’s publication and citation history gives the evaluator context for assessing their accomplishments and possibilities in the future. An evaluation technique that balances the usefulness of performance measures with the limitations of what they can reveal about the productivity and quality of an author, a paper, or a publication can be informed by knowledge of the history and intended applications of popular publication metrics.

2. Literature Review

Existing tools like Microsoft Academic Search, AMiner, Google Scholar, Xueshu Baidu, Semantic Scholar, CiteSeerX, and many others are largely autonomous and don’t need much user assistance when extracting metadata. Limitations of the existing systems include:

- Greater information coverage is not always adequate for searching relevant documents.

[1,2,3,4,5,6] Department of Computer Engineering, Xavier Institute of Engineering, Mumbai, India
[1*]Email: kunalmeher@gmail.com
[2]Email: Beatrice.s@xavier.ac.in
[3]Email: kirti.m@xavier.ac.in
[4]Email: sushama.k@xavier.ac.in
[5]Email: chhaya.n@xavier.ac.in
[6]Email: jaychand.u@xavier.ac.in
[*](Corresponding Author)

- Rely mostly on citation count for paper/document ranking.
- Scholars bypass coverage and switch to alternative scholarly platforms like Microsoft Academic Search, CiteSeerX, Semantic Scholar and PubMed whenever ranking continues to be poor.
- Extracted data may not be sufficiently accurate, and some modification or improvements may still be required. Only authorized users may alter the generated metadata in a few of these systems.

An automatic method is required for figuring out what each sentence in an academic article's abstract is intended to do. Machines can explicitly mark each abstract phrase as being connected to one or more features. The experimental findings on a real dataset demonstrate that the labeling methodology performs better than the standard methods [3]. A basic introduction to citation networks like Scopus, Microsoft Academic, Web of Science Core Collection, Google Scholar and Dimensions is presented in the paper titled "Practical publication metrics for academics". The h-index and the journal impact factor, two of the most well-liked publication measures, are also highlighted [5]. A research paper is used as a combination of its verbal (bag of words) and structural (bag of citations) components that are discovered through citation network analysis. It offered a methodology for the tactical ranking of scholarly publications that utilizes an inverted index powered by Apache Solr paired with a hierarchical index for content analysis [4]. The different academic search engines available are explained below.

2.1 Google Scholar

Google Scholar is a search engine that allows you to locate articles or citations on any subject by searching scholarly literature at top academic publishers and university presses [2]. The amount of times an item has been read, printed, or downloaded over a specific period of time is used by Google Scholar to rank papers (usually around one year). Citations to various resources like online books, papers, reports, etc that are available online are being displayed by Google Scholar (GS). Its searches are configured to include academic content more frequently than "regular" Google. The order of the materials is determined by their addresses (especially their domain, such as.edu or.gov), as well as other factors such as who referenced them, what was cited by them, what was linked to them, and so on. GS refers to this as Page Rank.A large portion of Google Scholar's database comes from a crawl of full-text journal articles that is offered by both for-profit and nonprofit publishers. Additionally, specialized bibliographic databases like Open WorldCat from OCLC and PubMed from the National Library of Medicine are crawled.

How does Google Scholar work?

Google uses a range of techniques, including its PageRank algorithm, according to their corporate website, "to evaluate the complete link structure of the web and determine which pages are most essential." When it pertains to study output, this idea can be generalized to say that a paper is noteworthy if it builds upon the other strong scientific journals (cites important documents) and is heavily referenced by important rather than unimportant studies. Formally speaking, Google determines a Web article's present PageRank as the total of its shares of the PageRanks of all the documents that link to it. For instance, if document B contains links to five other documents, such as document A, then it shares one-fifth of its PageRank with A.

The most recent values are used to calculate the current values of all PageRanks when Google updates them, and this procedure is repeated until all PageRanks reach steady state levels. The variations in PageRank values during the updating procedure resemble a Markov chain, a mathematically well-studied process. Numerous mathematical aspects of the calculation of PageRanks have been studied as a result, such as convergence, speed, and sensitivity to minute changes in the Web's structural elements. Google combines PageRank-based results from a Web search with data on how relevant individual web pages are to the query.

These techniques are used by Google Scholar on a portion of the Web that it indexes as scholarly content. It includes links to full text documents, the number of citing publications it has indexed, and links to these; links to the citing documents are also displayed in order of importance. The order in which the search results are presented is decided by PageRank and relevancy to the search. Google Scholar ranks papers by determining the significance of each and every one of its indexed pages using data from its full database. Due to this, Google Scholar may be helpful for locating significant contributions to a field.

2.2 Semantic Scholar

Semantic Scholar uses artificial intelligence to analyze academic literature so that a researcher's search activity returns content that is more relevant and meaningful than results from traditional search engines. Semantic Scholar's features are consistent with its goal of saving users time. Semantic Scholar restores a scientist's sense of discovery by restricting findings [1]. A few hundred immediately relevant results are delivered by searches that produce tens of millions of results in Google Scholar and millions in PubMed in Semantic Scholar [1]. By minimizing the large tail of search results, Semantic Scholar helps users focus on their specialties while reducing the distractions from less important research. Semantic Scholar's commitment to full-text and mobile-friendly design also helps to save time. Similar to how pre-assessed evidence levels assist physicians, the citation analytics tools let researchers pre-assess quality by graphically representing citation velocity and author influence scores. Without needing a lot of reading, displays immediately illustrate the components that researchers are most interested in, such as references and citations, techniques as a filter, and graphs and tables [1].

Highly cited authors are highlighted with influence scores, highly influential citations, total citations, a citations-per-year graph, and a citation velocity score due to the increasing demand for researchers and institutions to demonstrate effect [1]. The citation counts for authors with fewer than 50 citations is simply taken down as ”50”. Author maps show those who have had the most effect on an author as well as those who have had the most influence on them, provided there are enough citations. By displaying the semantic context or contexts of where and how frequently a reference is cited in the document, the reference list gives citations a deeper, more meaningful meaning. The “cited by” function found in other places in Google Scholar and Web of Science along with hyperlinks to other sources, are all part of Semantic Scholar’s attempt to merge traditional citation metrics and altmetrics. Semantic Scholar favors interface simplicity and provides just a minimum choice for filtering and sorting of search results. Only relevancy and publication date are used for sorting. Truncation is supported, but Boolean or phrase searching are not supported. Such limiters as Data Set Used, Cell Type, and Brain Region are not present in other databases, which reflects the initial target market for the database. The information overload and quality lacking evaluation that many researchers encounter with keyword search of Google Scholar is what Semantic Scholar seeks to address. To accomplish this, developers are currently relying on preexisting indexing in PubMed and IEEE, along with annotations provided by medical subject experts. Articles without full-text links instead link to a digital object identifier (DOI), PubMed abstracts, or IEEE [1]. Similar to this, Semantic Scholar searches the internet for citations using an undefined algorithm that prioritizes recall over precision and full-text access. It does not explore behind paywalls like Google Scholar. Due to this, Sematic Scholar’s inability to search authorized materials emphasizes time, convenience, and access, in contrast to the makers’ professed emphasis on quality. As a result, while using Semantic Scholar as it currently stands, researchers cannot assume that they have fully searched the background literature in their fields.

2.3 AMiner

Exploring connections between researchers, conferences, and publications using social media analysis, AMiner(ArnetMiner) is made for searching and performing data mining operations against online academic publications. This makes it possible for it to offer services like topic modeling, academic performance evaluation, association search, course search, trend analysis, expert finding, geographic search, trend analysis, and reviewer recommendation.

AMiner was developed as a research project in social network extraction, ranking, and social impact analysis. The creation of the system has led to the publication of several peer reviewed studies. In academia, AMiner is frequently used to determine connections and derive statistical correlations regarding research and researchers. More than 10 million separate IP accesses from 220 nations and areas have been attracted to it. The researcher profile is automatically retrieved by AMiner from the internet. It gathers and recognises the pertinent pages before employing a unified strategy for data extraction from recognised documents. Additionally, it uses heuristic rules to pull papers from digital online libraries.

It combines the retrieved articles with the extracted researcher profiles. As an identifier, it uses the name of the researcher. A probabilistic framework has been suggested to address the integration’s name ambiguity issue. The combined information is kept in a knowledge base for researchers (RNKB). Google Scholar, Scirus from Elsevier, and the open source project CiteSeer are the main other products in the field.

The five main parts of the system are as follows:

1. Extraction: The emphasis is on pulling research profiles automatically from the Web. The service initially gathers and recognizes a user’s pertinent Web pages (such as homepages or introductory pages) before using a unified strategy for data extraction from the documents. Additionally, it uses heuristic rules to pull papers from digital online libraries. Additionally, by utilizing the power of big data, a straightforward yet extremely successful strategy is established for profiling Web users.

2. Integration: connects and combines the profiles of the collected researchers with the extracted articles. The researcher’s name is used as an identification by the application. The name ambiguity issue in the integration has been addressed using a probabilistic model and thorough framework. The combined data are then kept, organized, and indexed in a research database.

3. Storage and Access: It enables the researcher network knowledge base to store and index the extracted and combined data. In particular, it uses Jena, a tool for storing and retrieving ontological data, for storage, and the inverted file indexing method, a tried-and-true technique for speeding up information retrieval, for indexing.

4. Modeling: models multiple types of information sources simultaneously using a generative probabilistic model. The system makes an estimation of the subject distribution spread over many information sources.

5. Services: based on the modeling results, offers a number of powered services, including user management, conference analysis, sub-graph search, profile search, course search, expert finding and subject browser.

2.4 Microsoft Academic Search

To give users a more powerful way to explore academic content, Microsoft Academic makes use of data mining technologies, semantic analysis and machine learning. A user can register and make a public profile by claiming the works they contributed to. Microsoft Academic offers "follow" features that are more comprehensive. Users can follow scholars, books, magazines, journals, gatherings, groups, and research areas. The most pertinent items and news are displayed on a user's personalized homepage by Microsoft Academic based on the publication history of the user and the events they are following. Additionally, Microsoft Academic presents pertinent results and recommendations instead of just a keyword-based search engine to assist users in finding additional academic information resources that may be of interest to support a more thorough learning and research experience.

According to the help page, the papers are prioritized in the search results based on two factors: their relevance to the query and their overall importance (similar to Google Scholar). The search method is based on objects rather than documents, which are usually recognised entities like authors, publications, conferences, or journals. Actually, this makes it possible to show the results in a variety of ways by making efficient use of the object metadata. For example, co-authors, a journal, a keyword tag cloud or a conference can be used to organize the results. The results can be filtered and sorted by year. But you may sort the outcomes for a particular author by year, citation, and rank.

A decent profile of a certain author can be found in Academic Search, which includes details like the number of publications, citations, collaborative authors throughout g- and h-indices, interest domains and time. Academic Search offers a number of excellent visualization tools, including co-author graphs and paths of co-author, domain trends and citation graphs. Additionally, it provides a variety of APIs (application programming interfaces) that can be used, for example, to build your own rankings of educational institutions or a visual explorer for browsing academic publications in a given topic.

2.5 CiteSeerX

CiteSeer is a digital library for academic and scientific papers, primarily in the areas of computer and information science, as well as a free public search engine. Academic search engines like Microsoft Academic Search and Google Scholar are thought to have been inspired by CiteSeer. CiteSeer-like archives and search engines typically only collect documents from websites that are open to the public and do not crawl publisher websites. As a result, the index is more likely to include authors whose works are available for free.

The aim of CiteSeer is to enhance access to and dissemination of academic and scientific literature. It has been regarded as a component of the open access movement, which seeks to alter academic and scientific publication to provide more access to scientific material. It is a non-profit service that anybody may use without charge. CiteSeer links indexed documents to additional metadata sources, such as ACM and DBLP Portal, and provides Open Archives Initiative metadata of all indexed publications for free.

It has the following features:

- By using autonomous citation indexing, a citation index that can be utilized for literature searches and evaluation was automatically produced.
- Citation statistics and related documents were calculated for all articles in the database, not just indexed articles.
- Reference links enable database browsing via citation links.
- A researcher can readily examine what other academics would like to talk on an item of interest thanks to citation context, which displays the context of citations to a particular work.
- A live, constantly up-to-date bibliography is displayed for every document, and related documents were identified using citation and word counts.

3. Proposed System

Our proposed system i.e. Scholarly include functionalities such as:

- To create a search engine specifically for professors based on name, topic, paper title, etc.
- To create a working prototype of a line of text scholarly search engine. The proposed approach dynamically labels each sentence. All of the sentences and their related purpose labels are indexed by the system.
- A querier may decide to look for sentences that best fit a specific purpose by focusing on those features.
- Ranking and filtering based on h-index, citations over the last five years, conferences, etc.
- Ranking metric based on the data.

A. Block Diagram

In this section, we explain the steps involved during the various phases of our project using block diagram.

The steps for the above block diagram are explained in detail:

1. Scraping and Retrieving Text: This module uses the list of Google Scholar IDs of professors (from CSRankings, split across 10 files), scrapes data from their Google Scholar pages and stores it as CSV.
2. Cleaning Data: This module cleans the scraped data from the previous module and stores it as CSV.
3. Building Index: This module uses the cleaned data to build two inverted indices (first for name and affiliation, second for topics and paper titles) and stores them as JSON.
4. Query Processing and Ranking: This module receives query information from the "Backend", processes it and returns an ordered list of professors depending upon the specifications provided by the user.
5. Backend: (Web Server) This module forwards the user's query to the "Query Processing and Ranking" module and acts as an intermediate to the user's browser.
6. User: The user types a query, specifies the retrieval method (Boolean, phrase or TF-IDF) and the context in which he wants the results (names and affiliations or topics and paper titles).
7. Computing Data Statistics: It computes and plots various statistics of the cleaned dataset.
8. Evaluation: It generates its own queries, runs the queries using the "Querying and Ranking" module, evaluates search results and plots the evaluation metrics.

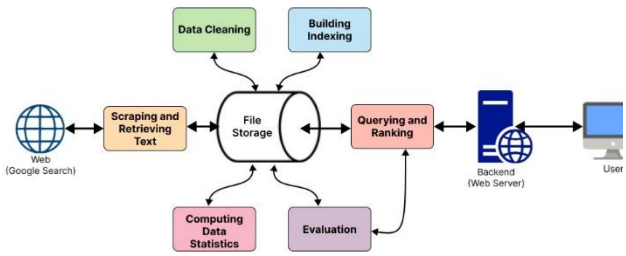

Fig 1: Block Diagram.

4. Implementation Methodology

4.1 Querying and Ranking

Along with the search query, we also take the retrieval method and the search context i.e. whether they want to search in name and affiliation space or research topics and paper title space, as inputs. We then use the corresponding index and retrieval method.

We implemented three different techniques for querying:

4.1.1 Boolean Retrieval

Some of the following variants of Boolean Retrieval are:

- AND

Only those professors would be shortlisted who had all the words from the search query on their scholar page. To do this, we simply compared the postings lists of all the words in the search query pairwise, and stored the intersection of all of them.

- Optimisation

Since we are only interested in the final postings list, that is, the list containing all the words in the phrase query, the order in which we build this does not matter. Hence, to speed up the process, we sort the postings lists in the increasing order of length. Since we use the intersection of two postings lists in the next iteration, this ensures the fewest number of computations overall.

- OR

Here we return the pages of all the professors who even contain one word from the search query. However, we sort matches according to the number of matches they have, that is, if a professor's page matches 4 out of 5 words from the search query, it will be displayed higher than a professor's page which matches 3 out of 5 words from the query. In case two pages match the same number of words from a search query, higher precedence will be given to the page that matches all the words in the query more times. For example, the word science may appear on a page more than once, this will increase the tiebreaker score.

4.1.2 Phrase Retrieval

We also added an extra feature, allowing a user to search for exactly the phrase that they have typed. This is done by storing a new variable called distance in our ordinary Boolean Retrieval, which denotes the distance at which the two words appear in the search query. This is the same distance at which they must appear in the postings lists. The second parameter in every element of our postings list (inverted index) denotes the position the word appears in the given web page. So once we know that two words belong to the same page, we can check if they are at the required distance from each other.

4.1.3 TF-IDF Scores

Search results can also be based on the numeric scores, such as the Term Frequency-Inverse Document Frequency (TF-IDF) which are available to users. It is a numerical statistic that is frequently used in text mining and information retrieval to reflect how significant a word is with respect to a document in a corpus or collection. Professors are treated like documents by us. The frequency of a word in a document is measured using the term frequency, which is normalized by the size of the document. While inverse document frequency determines how much information a word conveys, i.e., how frequent or uncommon a word is throughout all the documents.

TF(i,j) = (count of ith term in jth document) / (total terms in jth document)

IDF(i) = (total number of documents) / (number of documents containing ith term)

TF − IDF Score(i,j) = TF(i,j) × IDF(i)

Table 1 shows the different types of retrieval methods along with their advantages and disadvantages

.**Table 1:** Different Retrieval Techniques

Retrieval Method	**Description**	**Advantages**	**Disadvantages**
Boolean Operators	Use of logical operators like AND, OR, NOT to refine search queries	Precise results, allows for complex search queries.	Requires knowledge of operators, can miss relevant results
Natural Language	Allow users to input queries as they would ask a question in every language	Easy to use, no need of technical knowledge	Can lead to imprecise results, limited to specific search engines
Concept-based	Uses a predefined set of concepts or terms to find relevant contents	Allows for more comprehensive search results, can help find hidden connections	Limited to specific topics or domains, can miss relevant search results
Fuzzy Logic	Allows for the inputs of imprecise queries and returns results based on similarities	Can find relevant results that other methods might miss, can handle misspellings and variations in language	Results may be less precise, requires advance algorithms and processing power

5. Data Statistics

Figure 2 shows that the frequency distribution follows a power law, i.e., the number of words (in the topics and paper titles index) with frequency k is C/kn where C and n are constants. Statistics related to number of citations, i10-index and h-index are presented in Figure 3, Figure 4 and Figure 5 respectively. Figure 6 shows that most professors have mentioned their affiliated institution, verified their email address and provided the URL to their homepage.

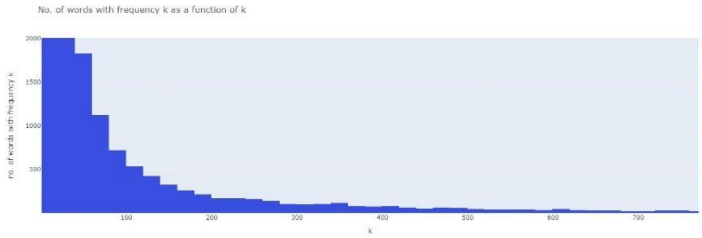

Fig 2: No. of words with frequency 'k' as a function of 'k'

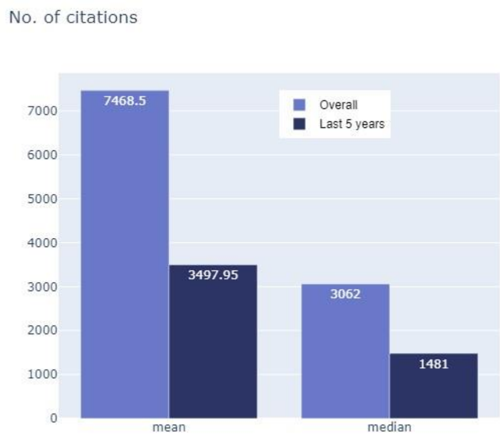

Fig 3: Number of citations

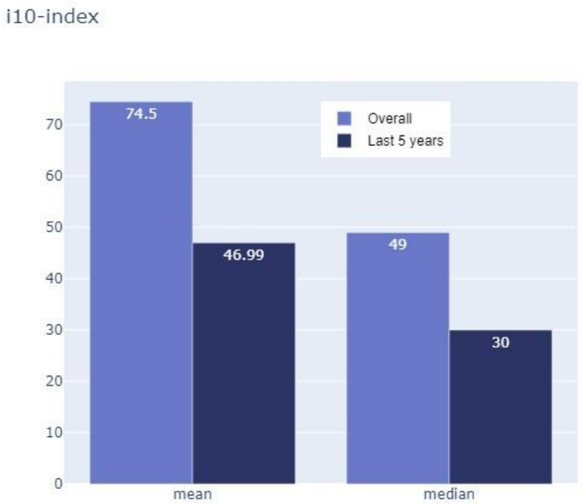

Fig 4: i10–index

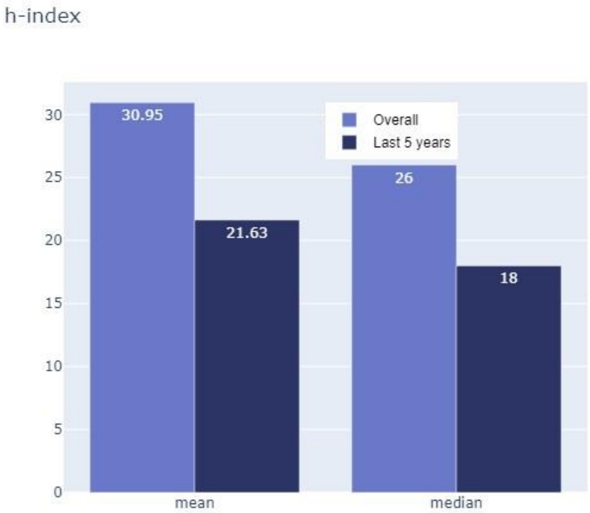

Fig 5: h-index

Fig 6: Professor's information

Table 2 shows comparison of different search engines.

Table 2: Search Engine Comparison

Search Engine	Coverage	Parameter considered for retrieval method	Features	User Interface	Unique Advantages
Scholarly	Broad	Citation analysis, h-index, i10-index, TF-IDF, recent publications, Keyword matching, full-text search, author profiles	Visualization, author profiles, top 10 papers, author and affiliation, topic and paper title	User-friendly, simple, clean	Parameters considered will help to refine search results to better meet users need
Google Scholar	Broad	Keyword matching, Citation analysis, full-text search, author profiles	Citation tracking, alerts, metrics	Simple, familiar	Large index, integration with other Google services
Microsoft Academic	Broad	Query parsing, relevance ranking, citation, semantic analysis	Citation tracking, metrics, visualization	Clean, modern	Includes conference papers, patents, and grey

					literature
Scopus	Broad	Relevance, Citation impact, timeliness, content	Citation tracking, alerts, metrics	Robust, customizable	Extensive international coverage
Web of Science	Broad	Search terms, document type, citation index, subject categories	Citation tracking, alerts, metrics	Powerful, advanced	Includes social sciences and arts and humanities
PubMed	Biomedic al	Text word search, author search, related articles, clinical queries	Medical subject headings (MeSH), filters, alerts	Basic, clinical	Comprehensive biomedical literature

6. Result

Consider a user who has a particular professor in his mind, he using some information of that professor like name, affiliation, or title of a paper of that professor, and by choosing appropriate querying method (mentioned in Querying and Ranking) searches and gets results. Now, we define the rank as the position where the professor he had in mind shows up in the search results. Here, the professor in his mind is the ground truth.

We generated random 500 Professors and queried the search engine using the search query and retrieval method pairs given below. The appropriate index i.e. name and affiliation or research topics and paper titles was used for each combination. Since, we already had the unique ID of the Professors before querying, we look up for that unique ID in the matched Professors IDs returned by Querying and Ranking module.

The parameters and its initials used in the below graphs are as follows:

N - Name

A - Affiliation

P - Paper & Topics

B - Boolean Retrieval

Ph - Phase Retrieval

T - TF-IDF

6.1. Median Rank

The median of the ranks obtained by the ground truth is called the median rank. Figure 7 shows the median rank for each combination. The X-axis represents the search query and method type pair, and Y-axis represents the median rank obtained. Since phrase retrieval utilizes proximity of words appearing in query and the information present in data set, it performs much better even on paper titles. TF-IDF on the other hand considers all the professors where at least one word from the search query appears hence, the number of search results is very large. As there can be multiple professors where words from paper title could appear and could cause larger TF-IDF score. Hence we get slightly higher median rank for TF-IDF with paper title as search query. The definition of median rank suggests that lower the median rank, the better the search results. A median rank of one means the user gets what they were looking for in the first position.

6.2 Recall Rate

Along with median rank, we calculated the percentage (out of 500 queries) of times that the ground truth appears in top X search results. This is called as Recall Rate at X. We computed Recall Rate at 5 and 10.

Figure 8 shows the recall rates for each combination. The X-axis represents the search query and method type pair. The recall rate follows similar pattern as median rank. A recall rate of 100 percent would mean that user finds their desired result in top 5 or 10 every time they query.

6.3 Average time per query

Figure 9 shows the average query time for each combination. For each search query and query method we evaluated the average time required by the algorithm to return the matched document (professor) ids. This time does not include the time to read the entire data of the matched professor ids, since, Querying and Ranking module returns only list of matched ids. Phrase and Boolean Retrieval perform well on search query of type professor name, affiliation and paper title. However, TF-IDF is relatively slower on paper title since it computes scores for all documents where at least one word from the search query appears.

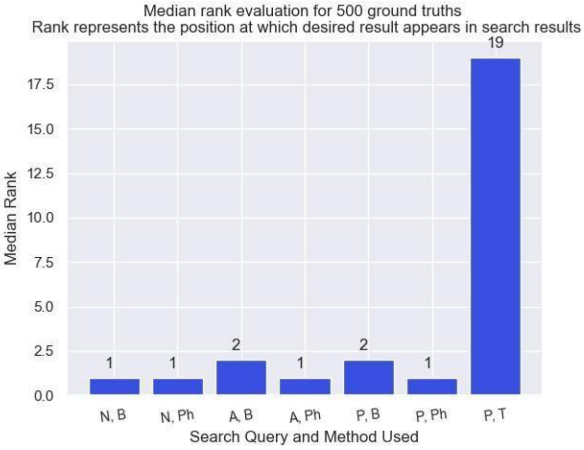

Fig 7: Median Bank

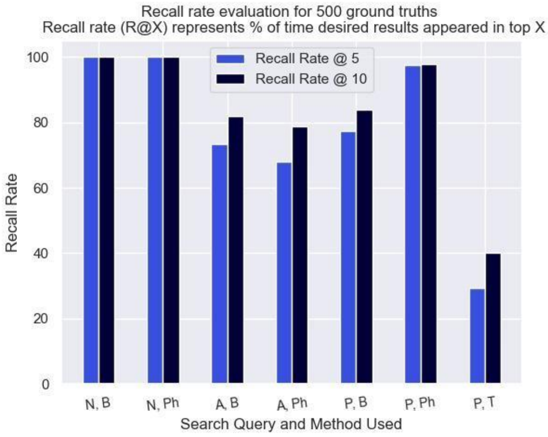

Fig 8: Recall Rate

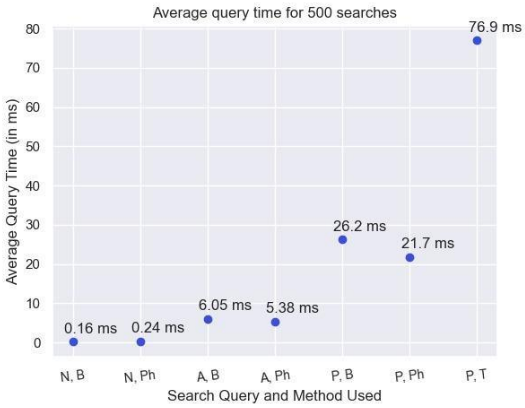

Fig 9: Average Time per query

7. Conclusion

A light-weight and crowd-sourced search engine has been proposed for all those students and aspirants who would be looking up for professors. Table 2 shows the comparison between the different existing search engines along with the proposed search engine Scholarly. The table gives information about the parameters considered for retrieval method by the search engines, features and their advantages. The data about the scholars and professors has been scraped from Google Scholar, all the required cleaning and preprocessing tasks upon the scraped data have been performed and for each of these preprocessed data, its inverted index has been created successfully. On some of the types of information retrieval techniques like Boolean AND & OR Retrieval, Phase Retrieval and TF-IDF scoring, a detailed study has been done and attempted to understand its comparative retrieval performance which can be useful in the further implementation of these retrieval techniques in the application.

References

[1] Suzanne Fricke, "Semantic scholar". In:Journal of the Medical Library Association: JMLA 106.1 (2018). [Online; accessed 22-November-2022]., p.145. URL:https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5764585/

[2] Necia Parker Gibson, "Research Guides: Getting Better Results with Google Scholar: What is Google Scholar? What is it for?" [Online; accessed 22-November-2022]. 2022. URL: https://uark.libguides.com/googlescholar

[3] Li-Yuan Hsu et al, "Toward building an academic search engine understanding the purposes of the matched sentences in an abstract", IEEE Access 9 (2021), pp. 109344–109354.

[4] Shah Khalid et al, "An effective scholarly search by combining inverted indices and structured search with citation networks analysis", IEEE Access 9 (2021), pp. 120210–120226.

[5] Bethany A Myers and Katherine L Kahn, "Practical publication metrics for academics", Clinical and Translational Science 14.5 (2021), pp. 1705–1712.