A Unified Educational Resource Search Engine with Intelligent Ranking for Smart Education

Nithin M S
Dept.of Information Science and Engineering
Presidency University
Bengaluru,India
nithin.20221ISE0066@presidencyuniversity.in

Chiranthan T C Nadig
Dept.of Information Science and Engineering
Presidency University
Bengaluru,India
chiranthan.20221ISE0062@presidencyuniversity.in

Aditya V S
Dept.of Information Science and Engineering
Presidency University
Bengaluru,India
aditya.20221ISE0032@presidencyuniversity.in

Mr.John Bennet J
Assistant Proffessor,School of Computer Science and Engineering
Presidency University
Bengaluru,India
john.bennet@presidencyuniversity.in

***Abstract*— The learning process has become more fragmented due to the increasing availability of instructional content on websites like YouTube, GitHub, arXiv, and Kaggle. Long searches across multiple platforms make it more difficult to find re- sources, and an excessive cognitive load raises the opportunity cost of ineffective search. We created Study Search, a client- side federated search application designed to offer a cohesive learner-centric multi-source search experience, in order to ad- dress these problems. In addition to using a heuristic ranking system based on GitHub Stars/Forks, YouTube Likes/Views, arXiv Recency/Citations, and other quality indicators, our system integrates modular APIs. To ensure that the search results are reliable and pertinent to the user's information needs, these cross- reference multi-metric ratings. JavaScript, CSS, and plain HTML were used to create the user interface, which included usability features. A skeleton loader and dynamic result cards. This project shows how to combine, dynamically rank, and classify various resources to create an easier-to-navigate digital learning environment. Index Terms: Intelligent Ranking, Information Retrieval, Federated Search, Smart Education, and Educational Technology.**

***Index Terms*—Federated Search; Smart Education; Information Retrieval; Intelligent Ranking; Educational Technology.**

I. Introduction

An unprecedented amount of learning resources has been created by the growth of digital education, but they are dispersed throughout disparate silos, such as academic archives (arXiv), code repositories (GitHub), dataset hubs (Kaggle), and video platforms (YouTube). Students and researchers spend most of their time in combining these results and drawing conclusions. This fragmentation results in incomplete information retrieval.

This inefficiency of the existing systems is the main idea behind developing the Study Search Engine project. According to the surveys approximately 40 percent of the students spend their time over the internet in browsing, navigating and combining these results. Additionally without any ranking system students recieve a huge amount of information more than 500 hours of videos are uploaded on YouTube, and there are over 100 million repositories on GitHub. This causes them to give up on the process of finding a relevant content over the web.

The Study Search platform uses a unified search solution to overcome this problem of fragmented results. By querying several special API's and displaying the results on a single platform where the results are ranked, this helps in finding the best educational content. Our method reduces the burder of accessing the resources from various sources and also it emphasizes efficiency, relevance and trustworthiness by providing ranked results.

Technical and academic results are treated as regular webpages by the existing commercial search engine. These mainly rely on view counts and generalized page authority. By the the most popular result is out of date or lacks information .By incorporating few platform related calculation metrics such as GitHub commits or recent arXiv publication ,Study Search silver this problem. By providing appropriate results for advanced learning and research purpose our system place a major role in the field of smart education.

II. Related Work And Problem Definition

Semantic Search and Context: Approaches using NLP and RDF [1], [4] demonstrate success in highly structured environments they completely rely on the metadata and semantic graphs. This limits their direct application on resources like video transcripts and code files .This type of manual or deep learning based annotation remains a challenge for cross platform federation.

Domain-Specific Recommendation Systems (RS): Highly effective in dedicated contexts (e.g., Collaborative Filtering in Online Judge systems [2] or specialized learning platforms [6]). Yet the features and algorithms and retrieved from only one domain. This cannot be generalized fundamentally for all different categories of data like papers or datasets.

This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

Metadata and Quality Ranking: Research into quality proxies [3], [8] reinforces the need to move beyond simple keyword matching. Generic Search Engine Optimization (SEO) has the main goal to make itself a popular platform commercially rather than academic excellence. But our system by combining both commercial popularity and academic excellence to overcome the inherent bias of commercial engines.

Distributed Search and Scalability: Prior work [5], [9].Confirms the challenges faced in creating a federated search engine. A truly scalable solution for this problem must handle the dynamic API results and perform strict schema normalization and maintain consistency and multiple source data retrieve.

A. *Limitations of Commercial Search Engines:* Traditional commercial search engines fail because they place more emphasis on PageRank variations and general SEO than on intrinsic quality metrics unique to technical content (such as GitHub's star count or arXiv's submission date), traditional commercial search engines fall short in specialized educational discovery. For advanced learners, this results in subpar, highly popular, but possibly low-quality outcomes.

B. *Problem Statement:*

The fragmentation of educational resources is the main issue. In order to satisfy the comprehensive academic or technical quality requirements of advanced users, a unified system that retrieves, normalizes, and intelligently ranks heterogeneous resources based on resource-specific quality and popularity indicators is critically lacking.

III. SYSTEM ARCHITECTURE AND IMPLEMENTATION

A. *High-Level Architecture and Data Flow*

The Study Search engine employs a hybrid client-server architecture, with the frontend (HTML/JS) handling multi-source data aggregation and the backend (PHP/MySQL) handling user persistence (authentication, history).

Scalability and maintainability are given top priority in the system's three-layered structure, which consists of Presentation, Application, and Data .First the user starts searching for the required content by entering the query in the search bar. The query is accepted through the input control modules. This query is normalized and searched for the required repositories, datasets and papers. But for videos the query is executed with the help of api and the retrieved results are ranked using the ranking algorithm. The the ranked results are catagorised and displayed at the front end. At the time of retrieval while the contents are still loading due to network or the server issues, the skeleton loader module plays a major role in better user engagement where animated placeholders are displayed.

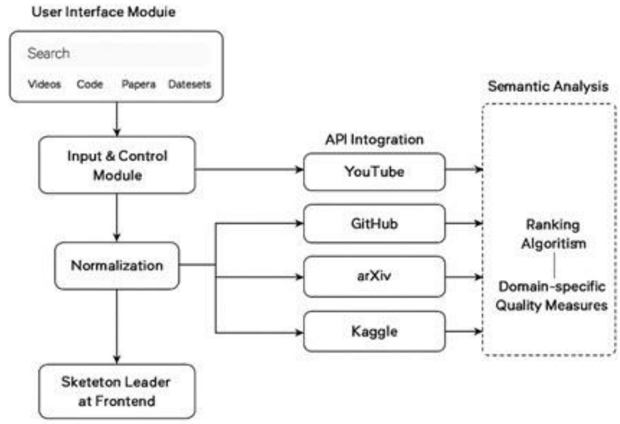

Fig. 1. The Study Search Engine's System Architecture,which displays the federated search flow from the User Interface Module to the platform-specific API Integrations

User Interaction and Routing: The user chooses a category and enters a query. Using JavaScript functions, the Input and Control Module verifies the query and forwards it to the client-side API Integration Layer.

1) Concurrent data fetching from external APIs (YouTube Data API v3, GitHub REST API, arXiv API, and Kaggle API) is handled by modular functions in asynchronous fetching and normalization. Before moving on to the ranking module, raw data is instantly processed and normalized into a consistent `.Resource Title`, `Source Platform`, `Resource URL`, and metrics) in- ternal data structure that includes metrics item Persis- tence Layer: The MySQL database ('study_search_db') manages:
   - keeps user credentials and hashed passwords.
   - History Tracking (`search history` table): Serves as the basis for the customized recommendation system by logging each query, type, and user ID.
   - Feedback and Rate Limiting (`feedback` and `feedback rate limits` tables): Used to gather qualitative feedback for system enhancement and to stop misuse.
2) Latency Management: To conceal network latency, the Skeleton Loader Module is shown during asynchronous fetching.

B. *Tier-1 Standard Ranking Methodology.*

The core technical contribution is the implementation of domain-specific scoring formulas that prioritize resources based on quality proxies inherent to each platform.

1) YouTube (Videos): Balances audience reach (×0.6) with direct engagement (×0.4).

$\text{Score}_{\text{YouTube}} = (0.6 \times \text{Views}) + (0.4 \times \text{Likes})$ (1).

2) GitHub (Code Repositories): Prioritizes community

endorsement (×0.7 for Stars) over contribution potential (×0.3 for Forks).

$$Score_{GitHub} = (0.7 \times Stars) + (0.3 \times Forks) \quad (2)$$

3) arXiv (Research Papers): Prioritizes Recency to ensure the latest available research is presented.

$$Score_{arXiv} = 1/(CurrentYear - PublishedYear + 1)$$

4) Kaggle (Datasets): Balances Downloads (×0.5) with Usability Rating (×0.5, scaled by 1000).

$$Score_{Kaggle} = (0.5 \times Downloads) + (0.5 \times Usability) \times 1000 \quad (4)$$

C. *Tier-2: The "Ultra Premium" Recommendation System.*

Based on the user's search history, this system offers users a feature of recommendations on the home screen layout.

1) Retrieval & Query Enhancement: The most recent query is obtained using the search history. Next this query is extended using get Enhanced Queries function. This function adds keywords to the data saved in the history and enhances the search results.(e.g 'ai' is the key word stored in the search history is enhanced and searched as artificial intelligence).This helps in finding better search results across all platforms.
2) Advanced Elements of Scoring: Once the results are retrieved. Results from various domains like videos, code, datasets and research papers are combined and displayed in the homepage as Ultra Premium recommendations .There are certain evaluation metrics that are used to rank them:

- YouTube (Premium):The likes , views, shares, comments and recency are used to calculate the score for the YouTube premium content.
- GitHub (Elite):For providing the premium recommendations in GitHub, the number of account watchers, forks, and stars are used to provide the elite recommendations contents.
- arXiv(Research):The length of the paper title, author count, year of publication are used to provide the premium content for the research papers.
- Kaggle(Premium):The premium datasets are recommended using account votes, views, downloads, and usability ratings.

IV. RESULTS AND ASSESMENT

The Study Search Engine is evaluated in terms of usability, performance and functional requirements.

A. *Functional Validation*

Accuracy of Ranking Algorithm : Using the Tier-1 (Section 3.2) ranking methodology ,the results were appropriately ranked for the GitHub repositories using the stars and fork counts and provides top ranked GitHub repositories.

User Management and Authentication: The use of PHP/MySQL allowed the system to track the user session and efficiently record the user preferences and then provide the recommendations based on the search history.

B. *Experimental Configuration and Comparative Evaluation.*

Comparative ranking analysis and functional validation were the two main stages of the evaluation. A blind pool of N = 20 technical queries (such as "Transformer model PyTorch implementation" and "GANs research paper 2024")was submitted to the Study Search engine and two baseline methods: direct YouTube search (Baseline 2: Pure Popularity)and Google Search (Baseline 1: General Purpose) for the Comparative Ranking Analysis. Each search's top ten results were compiled. On a scale of 1 (Lowest Quality/Relevance)to 5 (Highest Quality/Relevance), three domain experts were hired to independently rate the results' quality and relevance. The Ground Truth for determining ranking metrics was this expert scoring.

C .*Performance and Usability*

Performance: The API response from the server side to the client side was about 2-4 seconds. The addition of skeleton loader makes it more interactive and results in highly perceived response.

Usability: Usability was considered as to be a fundamental strength because ,by mimicking the structure of the incoming results in the form of a loading container, the Skeleton Loader Module (Fig. 2) helps to avoid users getting annoyed during network latency or low server. The platform improvements are later made by reviewing the user feedbacks.

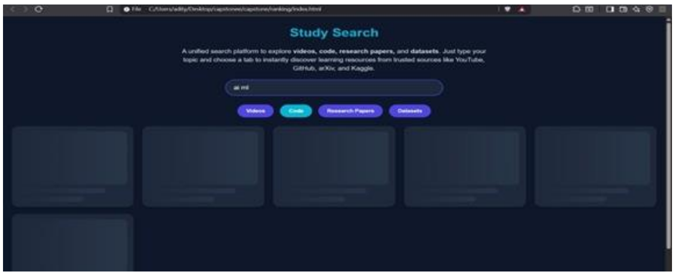

Fig. 2. Design of Skeleton Loader. During the content retrieval process, animated placeholders in the shape of loading contents helps to improve the perceived loading speed.

V. ETHICAL, LEGAL, AND SOCIAL ASPECTS

The Study Search Engine complies with important non-technical requirements. It offers direct links to navigate to the contents. This platform ensures rule and regulations to intellectual property laws and avoids providing copyrighted content. Unauthorized content scraping is prevented by controlling access to external data through usage policies of the APIs. In addition to this HTTPS is used in the GitHub Pages for the deployment, ensuring safe access to the repositories and user protection by external data breach.

By classifying resources according to the evaluation metrics (views, stars, and recency), this platform encourages transparency. Beyond the login process and history tracking, other personal information of the user is not gathered. This guarantees the user data privacy and ethical responsibility. By this the academic advancement is promoted around the world.

VI. Final Thoughts and Upcoming Projects

The Study Search Engine addresses the problem of vast spread fragmented resources in online learning.
This project provides categorized and ranked results for the learning process through a unified and user-friendly interface by combining two tiered intelligent ranking algorithms with a unified search approach.

The main methodology contributing in this idea is the tiered ranking system, which makes the system an intelligent and efficient platform for academic purposes and technical research purposes by going beyond the limits of the basic old style or default API result ordering to offer highly relevant content through ranking and personalized content recommendations using the search history.

It is advised that future work concentrate on switching to a server-side architecture in order to provide customized features and sophisticated ranking capabilities:

1) Backend Integration: To improve security and depend- ability, implement a strong server-side solution for intricate tasks like server-side caching and sophisticated API rate limit handling.
2) Commercial Viability and User Management: To monitor the search history for better recommendations and usage limits. Implementation of a comprehensive User Management System for the user enhancement and subscription tiers (e.g., Premium with 100 searches per day and Pro with unlimited searches) for the better recommendations. These factors are essential for handling high API rate limits, by enabling the content based recommendations and guaranteeing the service's long term availability.
3) Semantic Enrichment (LLMs): Incorporate Large Language Models (LLMs) into the query processor for better search results and information retrieval. This is a crucial next step for federated search. This will all the process of semantic embedding and advance query analysis. This will advance the platform to perform beyond its multi-metrics ranking and content based relevance scoring [9].
4) Multi-Modal Search Integration: Extend the search capacity that can support the model to take inputs like chemical structures and mathematical equations (in Latex) or image diagram to receive the information based on the the image input. This can be accomplished by combining the special APIs together (e.g PubChem, Chem Spider) and convert the text that has been entered in text queries to machine readable text. This facilities advanced resource retrieval.
5) Advanced Filtering and Taxonomy Mapping: By creating and adaptive filter to map the results across different platforms into a single educational taxonomy (e.g. Bloom's Taxonomy levels or particular knowledge graphs such as Wikidata). This would help the users to search for the required content based on the difficulty level or the required knowledge. This will allow the user to set the difficulty level and then browse for the contents by selecting varities of contents. This will enhance the usefulness of the aggregated resources and helps in better information retrieval.

References

[1] U. Yadav and N. Duhan, Efficient Retrieval of Data Using Semantic Search Engine Based on NLP and RDF," in *proc. Volume 6*, Oct. 27, 2021.

[2] D. M. Muepu and Y. Watanobe, From Code to Ratings – Enhancing Collaborative Filtering in OJ Systems," in *proc. Volume 7*, Dec. 31, 2022.

[3] R. Zhou, D. Xia, J. Wan, and S. Zhang, An Intelligent Video Tag Recommendation Method for Improving Video Popularity in Mobile," in *proc. Volume 8*, Jan. 14, 2020.

[4] M. M. El-Gayar, N. E. Mekky, A. Atwan, and H. Soliman, Enhanced Search Engine Using Proposed Framework and Ranking Algorithm Based on Semantic Relations," in *proc. Volume 7*, Oct. 7, 2019.

[5] I. Saif, A. S. Doukkali, and A. Enaanai, GENAUM: NEW SEMANTIC DISTRIBUTED SEARCH ENGINE," *Mobile Multimedia*, vol. 12, no. 3&4, pp. 210-221, 2017.

[6] P. P. Ingave´lez-Guerra, V. E. Robles-Bykbaev, A. Pe´rez-Mun˜oz, J. Hilera- Gonza´lez, and S. Oto´n-Tortosa, Automatic Adaptation of Open Ed- ucational Resources: An Approach From a Multilevel Methodology Based on Students' Preferences, Educational Special Needs, Artificial Intelligence and Accessibility Metadata," in *proc. Volume 10*, Jan. 27, 2022.

[7] C. Limongelli, M. Lombardi, A. Marani, and D. Taibi, A Semantic Approach to Ranking Techniques: Improving Web Page Searches for Educational Purposes," in *proc. Volume 10*, Jun. 27, 2022.

[8] J.B. Killoran, How to Use Search Engine Optimization Techniques to Increase Website Visibility," in *proc. Volume 56*, no. 1, Mar. 2013.

[9] K. A. Hambarde and H. Proenc¸a, Information Retrieval: Recent Ad- vances and Beyond," *Volume 11*, Jul. 14, 2023.

[10] D. Spinellis, P. Louridas, G. Gousios, and I. Chatzigiannakis, Using Research Literature to Generate Datasets of Implicit Feedback (LI- BRETTI)," in *proc. Volume 7*, Dec. 23, 2019.

[11] M. Tavakoli, M. Elias, G. Kismiho´k, and S. Auer, Metadata Analysis of Open Educational Resources: Towards Data-Driven Quality Scoring and Recommendation," *arXiv preprint arXiv:2101.10915*, Jan. 2021.

[12] D. Mukhopadhyay, A. Banik, S. Mukherjee, J. Bhattacharya, and Y. C. Kim, A Domain Specific Ontology Based Semantic Web Search Engine," *arXiv preprint arXiv:1102.0402*, Feb. 2011.

[13] N. Polatidis and C. K. Georgiadis, A multi-level collaborative filtering method that improves recommendations," in *proc. Volume 15*, Feb. 1, 2025.

[14] A. F. A. Mahmoud, Enhancing Semantic Search Precision through the CBOW Algorithm in the Semantic Web," in *proc. Volume 15*, Feb. 1, 2025.

[15] D. Bonino, DOSE: a distributed open semantic elaboration platform," in *proc. Volume 11*, 2003.

[16] P. Ingave´lez-Guerra, Automatic Adaptation of Open Educational Re- sources: An Approach From a Multilevel Methodology Based on Stu- dents' Preferences," in *proc. Volume 10*, Jun. 7, 2022.