Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 17 No. 20 (2023) | OPEN ACCESS

https://doi.org/10.3991/ijim.v17i20.42445

PAPER

MOOCMaven: Bridging M-Learning and Open Data for Enhanced Unified MOOC Exploration

Ahmad Fajar Tatang(✉), Abdullah M. Algarni

King Abdulaziz University, Jeddah, Saudi Arabia

atatang@stu.kau.edu.sa

ABSTRACT

We live in a modern environment where learning does not have any barriers. Our work aims to become the main gateway for people of any interest to take advantage of the full opportunities of e-learning. We also built a dataset of e-learning courses from various sources and developed a platform that can understand search queries. We introduce a unified massive open online courses (MOOCs) searching platform using semantic methodology called MOOCMaven. Our platform can be accessed both via web and mobile. This paper demonstrates the platform's efficiency in generating a list of courses relevant to search queries. Additionally, MOOCMaven offers an application programming interface (API) collection to ensure access to the data. Our approach could find the gaps in e-learning course discovery and highlight the solution to the public. This paper provides a detailed description of the algorithms, methodologies, and technologies utilized to augment the usefulness and efficacy of the platform.

KEYWORDS

unified massive open online courses (MOOCs), natural language processing (NLP), semantic search engines, m-learning applications, mobile educational technology

1 INTRODUCTION

Massive open online courses (MOOCs) growing steadily in recent years have significantly altered the education landscape, as seen in Figure 1. MOOCs provide unlimited access to educational resources to anyone interested in taking a course, irrespective of their physical location. MOOCs have gained immense popularity as they provide a flexible and convenient option for learners to access top-quality educational content from leading institutions across the globe [1] and have the potential to be an alternative to the traditional education system [2]. Not only have they become an ideal choice for learners seeking to upgrade or enhance their skill sets, but they have also become a preferred option for educators looking to expand their influence to a larger audience. MOOCs have also become one of the best options for school students and

Tatang, A.F., Algarni, A.M. (2023). MOOCMaven: Bridging M-Learning and Open Data for Enhanced Unified MOOC Exploration. *International Journal of Interactive Mobile Technologies (iJIM)*, 17(20), pp. 4–20. https://doi.org/10.3991/ijim.v17i20.42445

Article submitted 2023-06-19. Revision uploaded 2023-08-17. Final acceptance 2023-08-21.

teachers to learn new things that can improve their skills to master something [3]. Uncertain situations such as COVID-19 are also one of the fuels for MOOCs' growth [4].

A study by Mohamad et al. [5] shows online course materials, activities, and tools have an influence on students' performance. Aggregator MOOCs have emerged as a solution to simplify the process of finding and accessing online courses. By offering a centralized platform that gathers courses from various sources, learners can conveniently explore a wide range of topics without the need to navigate multiple websites. These aggregator platforms provide a comprehensive selection of courses, personalized recommendations, and additional features such as course reviews, progress tracking, and enhancing the overall learning experience. As the demand for online learning grows, aggregator MOOCs serve as valuable resources, connecting learners with diverse educational opportunities in a user-friendly and efficient manner.

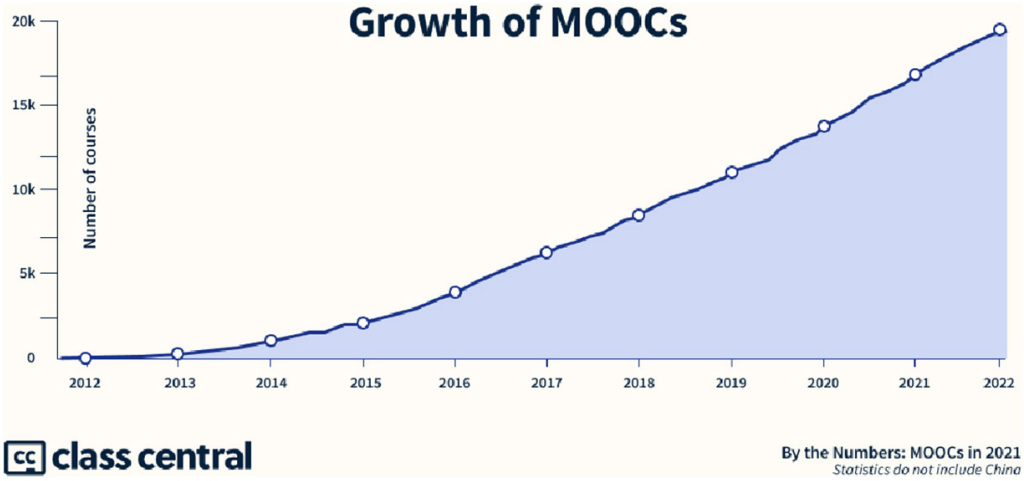

Fig. 1. Growth of MOOCs by numbers [6]

All aggregator MOOC websites offer information about MOOCs, allowing users to search for courses using keywords. Currently, these platforms primarily rely on full-text searching, which presents a significant challenge. Full text searching techniques rely on linguistic matching, in which a word or phrase in a search query must precisely match a word or phrase in a document within the database being searched [7]. However, this approach has limitations as it may fail to match variant terms, resulting in incomplete or inaccurate search results. Therefore, the reliance on linguistic matching in full-text searching poses potential limitations and can lead to suboptimal search results.

The MOOCMaven platform was specifically developed to address the need for accurate course discovery based on the exact meaning of user keywords. It aims to assist people in overcoming information overload and efficiently finding courses that align with their specific requirements. By partnering with a growing network of MOOC providers, universities, and research institutes, MOOCMaven has successfully curated a comprehensive and publicly accessible collection of online courses. This platform serves as a valuable public service, enabling users to discover and comprehend the most relevant courses with ease.

The platform incorporates semantic features, such as a semantic search engine, enabling users to explore the available courses by translating the meaning of the keyword. Additionally, the platform provides data download and API services, facilitating access to its comprehensive dataset. There are many factors that have influenced the promotion of open data access, including MOOC data. Studies [8], [9]

highlight the significant potential of open data in fostering increased participation and collaboration. However, challenges persist in ensuring the effective reuse of open data, which is sometimes overlooked in open data definitions.

The remainder of this paper is organized as follows: Section 2 provides an overview of the related work in the field. Section 3 details the methodology employed, including the data processing pipeline. Section 4 presents a comprehensive discussion of the research findings. Section 5 concludes the paper and outlines potential areas for future work.

1.1 Broader aim, novelty, and contributions

The broader aim of this paper is to enhance course discovery in unified MOOCs by applying the new approach of semantic search engines. We evaluate using approximate nearest neighbors as an indexing algorithm to improve precision. The contributions of this paper can be succinctly stated as follows:

1. We investigated the possibility of combining approximate nearest neighbors (ANN) with Facebook AI similarity search (FAISS) as our indexing technology.
2. We restructured the unorganized data into organized data to make it more valuable.
3. We compiled an exhaustive dataset for unified MOOCs, which includes 73,825 MOOCs in 75 languages from 65 different sources.
4. We developed a unified MOOC searching platform that understands the meaning of the keyword.
5. We discovered the application of a combination of three web scraping techniques to modern web technology.

The literature review (Section 2) established that the proposed ANN in a semantic search engine, the developed platform for searching unified MOOCs, and the unified MOOCs dataset are novel contributions to the field.

2 RELATED WORK

2.1 Data mining in MOOCs

Data mining in the context of MOOCs involves the analysis and extraction of valuable insights from the available data. Researchers have examined the standard data architecture and data management strategies for MOOCs. Du, Chen, and Jiang [10] categorized the MOOC data into three entities: courses, students, and teachers. This approach allows for the evaluation of students' learning outcomes and teachers' teaching effectiveness by utilizing the data associated with these entities. Other than that, most researchers use the data from MOOC activities to do several kinds of research, such as predicting students' performance [11], placement prediction [12], dropout prediction [13], learning analytics [14], and more.

2.2 Open data in MOOCs

Open data refers to publicly accessible data that anyone can use and reuse without restrictions imposed by any copyright or patent. The primary objective of open

data is to ensure the availability of non-personal and non-commercial data, with a particular focus on data collected and processed by governmental bodies. This movement draws inspiration from the principles of the Open Source and Open Access movements. Typically, open data is made accessible through websites, such as the United States government data portal (https://data.gov), World Bank Data (https://data.worldbank.org), Kingdom of Saudi Arabia data (https://data.gov.sa/en/home), and Indonesia One Data (https://data.go.id).

Braunschweig et al. [15] investigated the quality of data and metadata in a selected set of five repositories. Their research findings indicated that most platforms needed more standards and APIs and published data in formats that were not easily readable by machines or were proprietary. As a result, many open datasets were not truly accessible to automated programs. The integration of open datasets from various platforms with disparate information levels was proposed to address this issue.

Veeramachaneni et al. [9] proposed a standardized format for sharing MOOC data. Standardizing the data structure implies that all course-related data would be stored in a uniform database. This format conforms to conventions such as standardized event timestamps, tables, fields, and relational linkages between tables. In a separate study, Yu et al. [8] introduced MOOCCube, a data repository that combines MOOC courses with external resources. Notably, this repository is currently available only in Chinese. Moundridou et al. [16] present SlideWiki as an open collaboration resource for educational materials.

2.3 Semantic search engines

Search engines are widely used software programs that allow a user to find information online by entering keywords or phrases. While traditional search engines match keywords to retrieve relevant documents [17], semantic search engines employ natural language processing (NLP) and machine learning (ML) to understand the context and intent behind a query [18], providing more accurate and meaningful results.

Jiang et al. [19] proposed a hybrid indexing approach that calculates semantic similarity based on lexical vocabularies and adjusts the scores using an ontology model that captures the relatedness of concepts. This method enhances retrieval accuracy by incorporating the semantic relationships between terms.

Pan [20] focused on optimizing retrieval systems in the context of digital libraries. Employing a semantic search engine algorithm addressed the sorting problem of retrieval results, leading to improved retrieval effectiveness. The algorithm was tailored to leverage semantic understanding for enhanced search outcomes. Lukas and Katharina [21] employ unsupervised embedding learning in conjunction with graph convolutional neural networks to learn a math representation that enables efficient retrieval of semantically-related expressions.

2.4 MOOC search engine

In the modern era, the search engine has transformed magically with blazing technology. Multiple researchers have conducted data mining and information retrieval studies in the realm of MOOCs. [9–12] are among those who have explored this area. They have typically employed web crawlers to collect MOOC course data and develop unified MOOC repositories. Web crawlers, also known as spiders or robots, are systems that bulk-download web pages. They play a crucial role in

building web page indexes for search engines and can be utilized for data replication, competitive analysis, or academic research purposes.

Various approaches have been employed to facilitate the retrieval and recommendation of MOOC course data. An et al. [1] developed a virtual search engine for MOOC courses in Chinese. Lee et al. [26] introduced Courserush, a unified MOOC search engine that scraped data manually from edX, Udemy, and Coursera, employing the BM25 ranking algorithm. Courducate, created by [24], utilized manual data scraping, self-developed search engine technology, and the BM25 ranking mechanism based on Apache Lucene to index data from Coursera, Udemy, and edX data.

MoocRec.com adopted a hybrid approach by combining a search engine and a content recommendation system using a matrix factorization model. However, it only pulls data from edX and Coursera [23]. Kagemann and Bansal [25] developed an ontology to generate linked data from multiple MOOC providers, enabling users to explore courses across various platforms. Conversely, Alzahrani and Meccawy [22] proposed a hybrid search engine and recommender system model that assists users in browsing courses on a single platform. Their approach combines vertical search and clustering components, focusing primarily on vertical search with limited recommender system integration. Xu [27] developed a vertical MOOC search engine using the POS-weighted TF-IDF method for improving information retrieval in various contexts.

3 METHODOLOGY

The MOOCMaven open data platform aims to create and disseminate MOOCMaven Data (M2D), which is a high-quality, disambiguated course knowledge graph consisting of nodes representing courses, instructors, and providers. M2D is constructed by integrating multiple data sources into the data processing pipeline and is made available to the public through various APIs and datasets, as depicted in Figure 2 below.

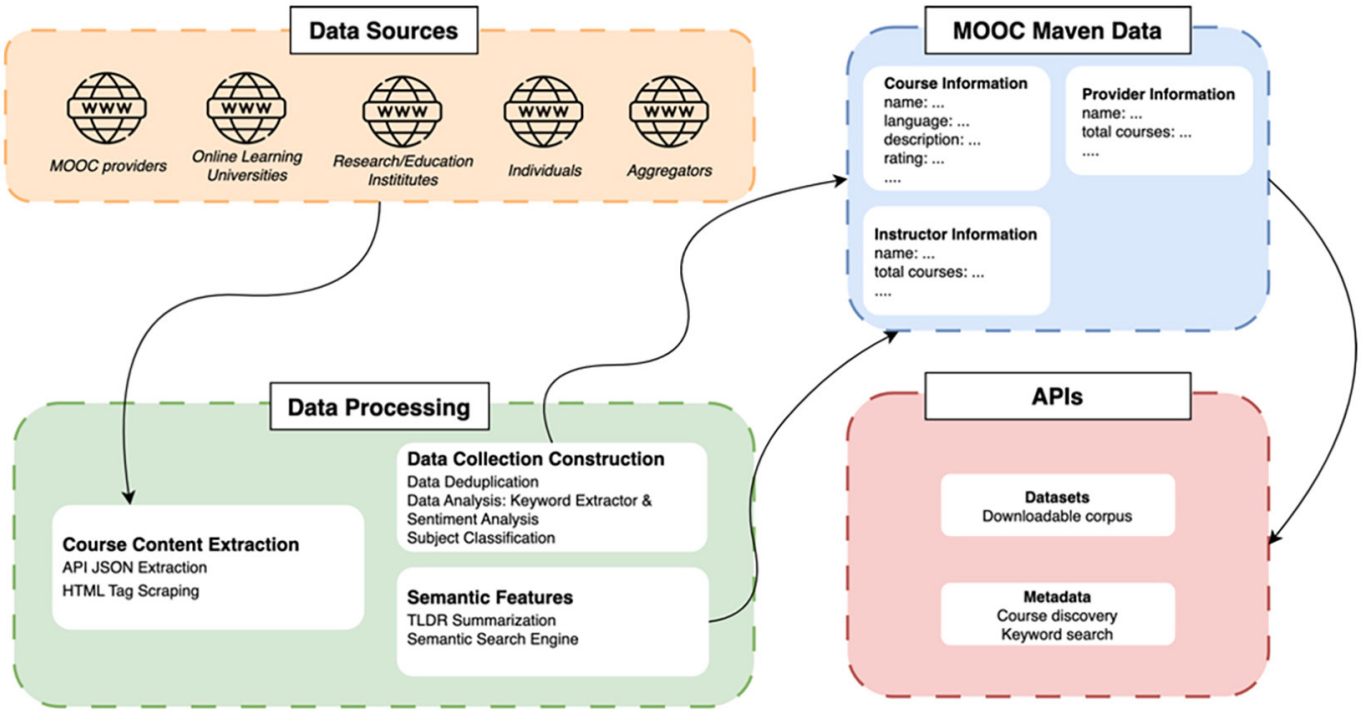

Fig. 2. MoocMaven data pipeline

3.1 Overview

The most important activity of this platform is data collection and extraction. Collects the data continuously and gets meaningful information from the collected data. The pipeline builds M2D by ingesting course metadata from various data sources. We built a custom crawler for this research because not all MOOC providers provide an API to get their data. The crawler will scrape the course data from the listed providers.

The course content extraction system retrieves organized course information from various structures provided by different sources. The data collection construction system, which constructs M2D, then analyzes the information. This data model is enhanced by a group of models that add semantic features to the graph, such as a semantic search engine and recommendations. A number of our pipeline processes are accessible as open software or models.

3.2 Data sources

The pipeline consists of 65 input sources that were meticulously selected to guarantee the acquisition of a comprehensive dataset. Our decision to include 65 sources was based on a strategic approach prompted by the discovery of Class Central, a leading aggregator that houses the largest collection of educational courses. Utilizing Class Central as a reflective surface, we utilized its vast database to increase the diversity and depth of the pipeline. Each of the 65 sources, ranging from Amazon Web Services skill builders to Udemy, Coursera, and OpenLearn, contributes unique and valuable content to the dataset. This organized selection process enables the pipeline to deliver a robust and representative dataset, which perfectly aligns with our research objective of providing valuable insights into the ever-changing online educational landscape.

3.3 Data processing

In this section, the data we got from various sources will be processed to become more valuable and readable to the public. We use several NLP techniques, such as text summarization and keyword extraction.

Course content extraction. We enhance the structured metadata of MOOCs by transforming unstructured information from a course into a structured format, which provides comprehensive data about the entire course. We extract essential data, such as title, description, and instructor. Extracting structured information from different sources is challenging since they are primarily designed for themselves and susceptible to errors. However, courses are the established representation of a MOOC's official content, and we have made significant investments in our extraction technology. There are two steps to course content extraction: data collection and data structuring.

Algorithm 1: Data Collection Algorithm

Input: *mooc sources*
Output: *courses data visualization*
1. *check the provider if offering API*
2. *hit API then store in the database*
3. *if API not available, then scrape the json-ld schema*
4. *if json-ld is not contain the designated information or available,then scrape the html tag*
5. *store in csv file*

API JSON (JavaScript Object Notation) extraction is the process of getting the course metadata from the publicly available API of its providers. We prefer this step since it is easier to convert into our data model. We select only the appropriate data that fits our prepared structure of bank data.

In Algorithm 1, we present a multi-step approach for data collection and storage from various sources, focusing on educational course information. The process begins by verifying whether the provider offers an accessible API to extract data. If an API is available, the crawler queries the API and subsequently stores the retrieved data in a database for further analysis and processing.

When an API is unavailable, the algorithm employs an alternative data retrieval strategy. Specifically, it attempts to scrape data from the web page's JSON-LD (JavaScript Object Notation for Linked Data) schema, which often contains structured information about the course offerings. JSON-LD provides a standardized and machine-readable way to express data within web pages [28]. Several researchers also do web crawling or scraping to get their research data [29].

If the JSON-LD schema does not contain the designated information or is unavailable, the algorithm further resorts to scraping the HTML tags of the webpage. The crawler directly extracts course-related data from the web page's content by analyzing the HTML structure. Finally, regardless of the data source (API, JSON-LD schema, or HTML tags), the algorithm stores the collected course information in a CSV file. This step ensures that the extracted data is consolidated and easily accessible for subsequent analyses, processing, or sharing.

The final output of the course content extraction pipeline is a unified structured data object suitable for encoding in JSON format. It includes core metadata such as title, description, subject, rating, language, institute, and provider, as well as the full body text description with detailed structural information about the text and the other supporting analyzed data from the data collection construction process.

Data collection construction. The content processing above results in the bank data used for the M2D fuel. Several methodologies have been implemented in this phase. Course deduplication is necessary because we process data from many sources, especially from aggregators offering multiple providers. Course titles are not unique and can vary in their expression. According to L. Hao and F. Farzad [30]. Their study examined the effectiveness of deduplication algorithms on data streams with approximate repeats, which are common in practice.

Data analysis was also performed on the dataset to provide an overview and support the primary entities. Based on the title and description, we conduct keyword extraction and sentiment analysis in data analysis. Keyword extraction needs to find the keywords of the courses that can be linked to similar courses in the same group of keywords. Keyword extraction is a technique that involves the identification of the most relevant words or phrases in a course description. The extracted keywords can summarize the text, categorize it, or facilitate the search for relevant courses. W. Xinyun and N. Hongyun [31] proved that using TF-IDF (term frequency-inverse document frequency) in the keyword extraction method is useful for semantic classification tasks. Sentiment analysis identifies the sentiment expressed in a course, such as positive, negative, or neutral. It is beneficial to evaluate the description of the course because new people will read the title and description before they decide to subscribe. Chihab et al. studied using BiLstm and TextBlob and showed better accuracy and F1 scores [32].

Not all providers, especially unique ones, look into the details of which subject or category this course belongs to. Possibility conflict also occurs in the courses if the data from the aggregator does not match the original provider. Subject classification is a mechanism to classify courses based on the subject. Prashasti et al. compare the

accuracy of subject classification systems in three popular academic databases—Web of Science, Scopus, and Dimensions—through a large-scale user-based study. While article-based subject classification schemes are believed to be superior, the study found that Web of Science, which uses a journal-based classification system, had the most accurate subject classification. Previous studies have compared subject classification schemes in the Web of Science, but this study is the first to compare article-based and journal-based systems in different academic databases [33].

Semantic features. We now describe the models that provide semantic features and data collection. The dataset has undergone cleaning and preparation in readiness for sentence transformation using a multi-language DistilBERT model, a smaller and faster NLP model derived from BERT [34]. To enhance comprehension and streamline decision-making while perusing a list of courses, we provide condensed summaries, often referred to as "TLDRs" (an acronym for "too long, didn't read"). This approach was introduced by Chachola et al. for scientific literature purposes [35].

Algorithm 2 outlines the semantic search algorithm that runs MOOCMaven backend services, where the input is the search keywords that will be encoded using Bi-Encoder, then produces a query vector [36]. Before encoding the search keywords, the system loads data based on the selected language. We created a custom index using FAISS [37] and ANN [38]. The index provides more accurate, relevant, and fast search results, as shown in Figures 3 and 5. This detailed research and report about our custom index can be found in another paper by the author. By searching data in the index, we get candidate data that will be re-ranked using Cross-Encoder, then returning the result.

Algorithm 2: Semantic Search Algorithm

Input: *keyword*
Output: *array json object courses*
1. *bi_encoder, cross_encoder, passages* ←*loadDataLanguage(language)*
2. *query_vector* ← *doEncodingQuery(keyword)*
3. *index* ← *getIndex(language)*
4. *top_ids* ← *retrievePassages(query_vector)*
5. *CN_scores* ← *getReRankScores(cross_encoder, top_ids)*
6. *result* ← *createResponse(CN_scores)*

MOOCMaven data. MOOCMaven is an open data platform designed to support online learners in discovering a wide range of online courses. MOOCMaven combines various public and proprietary MOOC data sources. Our data processing pipeline and semantic model output are made available through a suite of APIs and datasets described below. In this phase, we differentiate between three types of data: course, provider, and instructor information. We are linking all the data to enable the platform to receive a data filter. At this point, the proposed model offers brief information on the course graph and the relationship between courses.

APIs, datasets, and technology. Our research combines a range of technologies in order to provide an API that maximizes user efficiency. The platform allows users to access a diverse array of datasets and utilize its search functionality. In our data processing pipeline, we use several tools, such as Python, FastAPI [39], Redis [40], and OAuth 2.0 [41], to facilitate secure access to the resources.

Python not only has a function in data processing but also in accessing semantic models. The rich library environment facilitates the seamless integration of pre-existing models into the workflow of researchers, eliminating the necessity for laborious implementation or development endeavors.

FastAPI is a software library that prioritizes the optimization and effectiveness of the services integrated into our system. The system is constructed based on ideas

aimed at efficiently managing requests and generating prompt answers. This feature improves the overall user experience by enabling effortless retrieval of data. In addition, Redis is employed as a caching technique to enhance the speed of data retrieval and increase the responsiveness of the API by storing frequently retrieved information. This methodology reduces latency, thereby improving the accessibility of datasets for users promptly and efficiently.

In order to provide a secure and authorized method for accessing our APIs, we have implemented the OAuth 2.0 authorization mechanism. We establish a trusted mechanism for accessing protected resources by requiring users to obtain authentication keys. This approach guarantees that researchers can securely retrieve datasets while adhering to our terms of use.

To delve deeper into our API documentation, researchers can refer to the link [42]. It should be noted that while limited access is available for unauthenticated users in terms of API requests and dataset samples, researchers can obtain an authentication key at no cost to unlock the full potential of our datasets and enjoy higher request volumes.

Our research paper includes open-source code, allowing for complete transparency, reproducibility, and collaborative improvement of our platform. Researchers and developers are encouraged to access and contribute to the codebase, enhancing the platform's capabilities and making it more valuable for the academic community. The code can be found in the repository **semantic-search-ann-moocs** at [43], providing an opportunity for others to build upon our work and advance the field of unified MOOCs or semantic search engines.

4 DISCUSSION

4.1 Overview

MOOCMaven was created to provide the best experience for students in finding courses according to their needs. Therefore, MOOCmaven's design is kept simple and clean by eliminating elements that are not too important in the search process. Figure 3 presents the user interface of the platform. Figure 4 demonstrates the search result; it shows the title along with the subject or category. That page also shows the provider and the instructor.

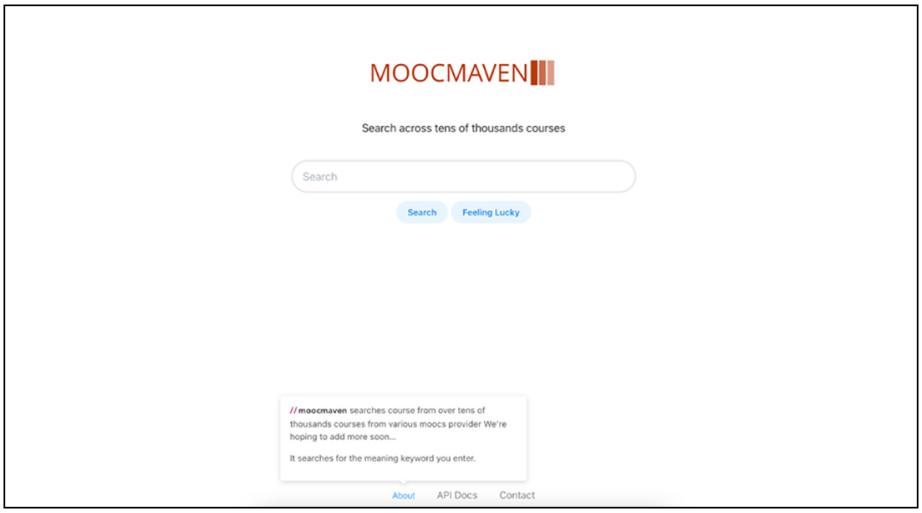

Fig. 3. Design of main page of MOOCMaven platform

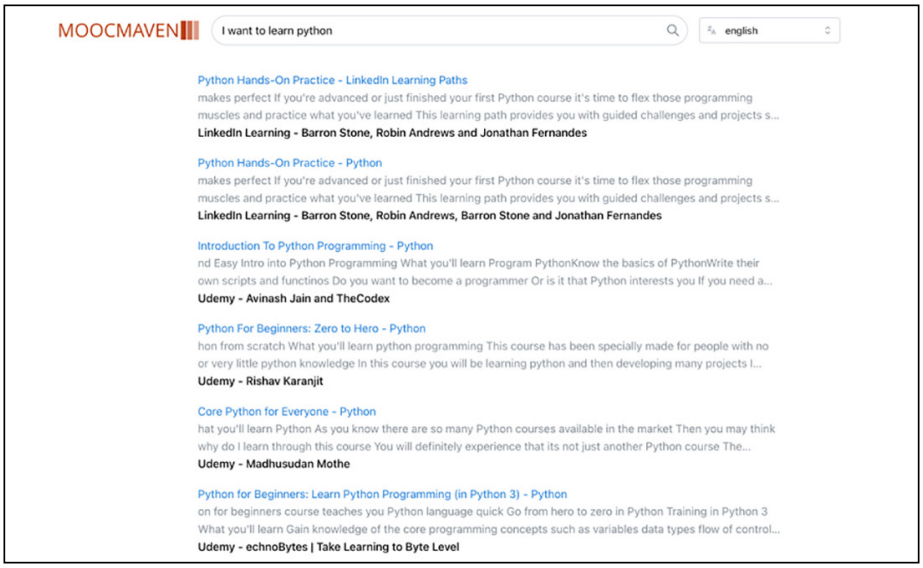

Fig. 4. Design of search result page of MOOCMaven platform

4.2 API list

Our API documentation can be accessed at https://api.moocmaven.com/docs. We wrote the documentation based on the Open API specification [44]. In Table 1, the API documentation outlines various endpoints and functionalities of the platform aimed at facilitating access to a comprehensive list of courses. The first endpoint, "Searching courses," allows users to retrieve course details by submitting a POST request with a keyword and language preference. The API response includes a list of matching courses, each described by its unique identifier, title, description, instructor, subject, provider, and URL. Another endpoint, "Download a csv file," permits users to download the dataset in CSV format for a specific language, authenticated via a Bearer token. Additionally, the platform offers endpoints to obtain a list of supported languages, providers, and functionalities to register and log in to the platform securely. These API endpoints enable seamless interaction with the platform's vast course repository, empowering users to explore and access educational resources easily.

4.3 Load performance test

Our platform runs on the server with 2GB of RAM, 1 vCPU, and 50GB of NVMe; it is enough to serve the traffic for now. To ensure the optimal performance of our platform even under heavy traffic conditions, we conducted a load performance test utilizing the open-source Locust framework [45]. This rigorous testing involved simulating 500 user accesses, carefully distributed over time, while utilizing our largest English course dataset. The total number of requests reached an impressive count of 16,731, with an average request rate of 63 requests per second.

The performance test results indicate significant potential, as illustrated in Figure 5. The platform demonstrated outstanding reliability during the testing phase, effectively managing the assigned workload without any failure. The platform demonstrated its competence by achieving the fastest response time at 176 milliseconds, notably low latency. On average, the response time was measured at 1,172 milliseconds. The prompt response might be credited to installing a

proficient caching strategy utilizing Redis. The decision to incorporate Redis into our platform has proven prudent, especially in efficiently providing static data.

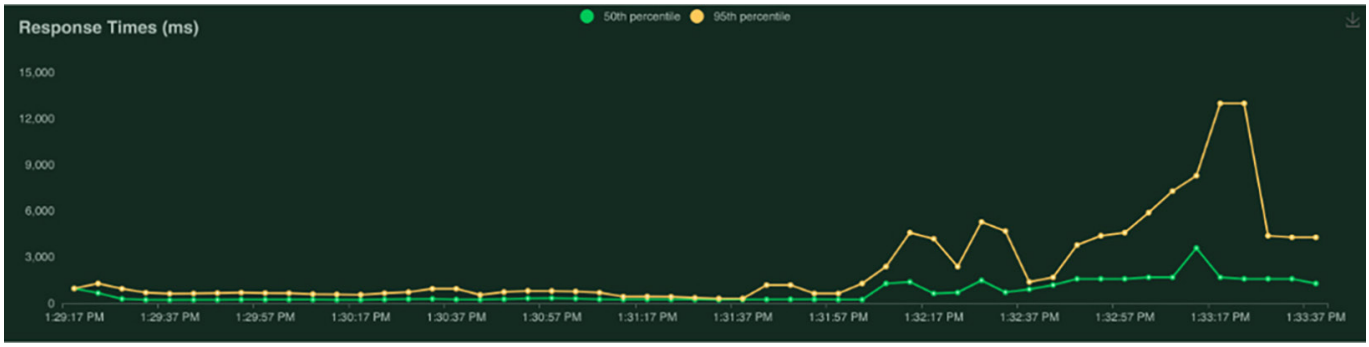

Fig. 5. The response times of load performance test with Locus (ms)

Table 1. Detail list of API offered by MOOCMaven

No	Description	Endpoint	Method	Authentication	Parameter	Response

1.	Searching a list of courses that match with the keyword	/search	POST	–	Request body:
					{
					“query”: “string”,
					“lang”: “string”,
					“skip”: 0,
					“limit”: 10
					}
						{
						“results”: [
						{
						“id”: int,
						“title”: “string”,
						“description”: “string”,
						“instructor”: “string”,
						“subject”: “string”,
						“provider”: “string”,
						“url”: “string”
						}
						],
						“total_results”: int
						}

2.	Download dataset (csv file)	/download/{language}	GET	Bearer token	Path variable: {language} e.g. english	text/csv

3.	Get supported languages	/languages	GET	–	–	{
						“languages”: [
						“string”,
						“string”
						]
						}

4.	Get provider list	/providers/{language}	GET	–	Path variable:
					{language}
					e.g. english
						{
						“providers”: [
						“string”,
						“string”
						]
						}

5.	Register to the platform	/register	POST	–	Request Parameters:
					username = “string”
					Password = “string”
						{
						“message”: “string”
						}

6.	Login to the platform	/login	POST	–	Request Parameters:
					username = “string”
					Password = “string”
						{
						“access_token”: “string”,
						“token_type”: “string”
						}

4.4 Comparison other platform

Table 2 presents an in-depth analysis of major aggregators that collect unified data from MOOCs, specifically emphasizing two key dimensions: accessibility and the range of services provided. Many of these aggregators do not offer unrestricted access to their data or the option to download it; instead, they present the data only on their own websites. Interestingly, the most popular aggregator, Class Central, does not have semantic search technology, potentially limiting its search capabilities. In contrast, MOOCMaven distinguishes itself from other platforms by providing open access to detailed data. Furthermore, MOOCMaven provides a range of services, establishing itself as an extensive platform within the MOOC industry. The services provided include data retrieval, search capabilities, categorization of courses, and access to the API. The rich data and additional features offered by the platform distinguish it from other aggregators in the MOOC market. A comparison analysis was conducted on the search results obtained from MOOCMaven and Class Central, focusing on the keyword "how to learn business". It was observed that there was a significant disparity in the quality and precision of the search results matching the keyword. The results indicated that MOOCMaven performed better than Class Central in providing more accurate and pertinent search results for the given keyword, as seen in Figure 6 for the MOOCMaven search result and Figure 7 for the Class Central search result.

Table 2. Comparison of existing unified MOOC aggregator

Source	URL	Total Courses	Access Type	Services
Class Central	classcentral.com	100,000	Open	S, C
Edukatico	edukatico.org	8,193	Open	S, C
Course Buffet	coursebuffet.com	N/A	Open	S, C
Course Talk	coursetalk.com	–	Terminated	–
Degreed	Degreed.com	–	Subscription	S, C
MOOC List	mooc-list.com	N/A	Open	S, C
MOOC.org	mooc.org	3,000	Open	C
MOOCMaven	**moocmaven.com**	**73,825**	**Open**	**D, S, C, A**

Notes: Key: D = data download; S = search; C = classification; A = API access.

MOOCMaven utilizes powerful semantic search techniques that consider the significance and contextual nuances of the user's inquiry. By utilizing ANN within the semantic model, MOOCMaven can comprehend the nuances and purpose underlying the given term. By including semantic features, the platform is able to provide course matches that are highly relevant, even in cases where the precise term is not explicitly included in the course names or descriptions. This feature extends its ability to accommodate users' personalized learning needs beyond keyword matching, enhancing the search experience.

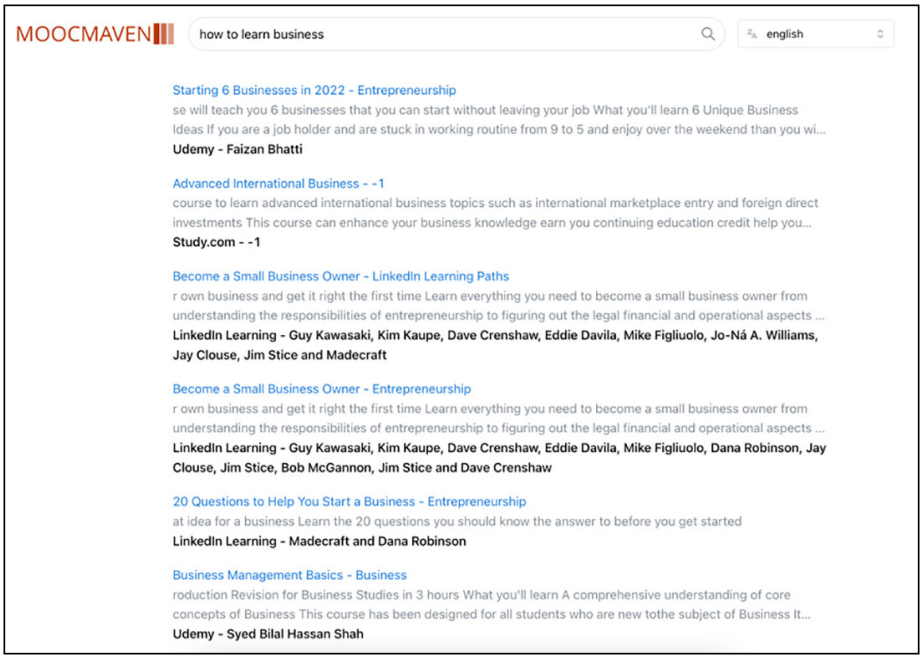

Fig. 6. Search results of the keyword "how to learn business" in MoocMaven

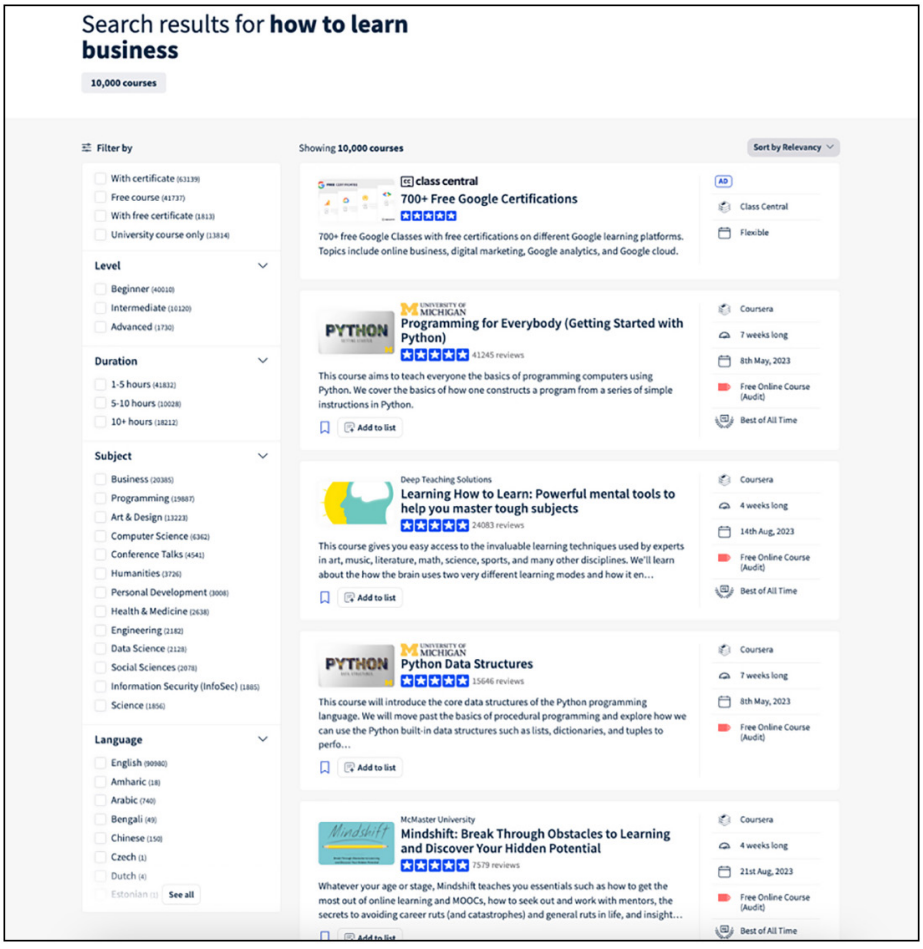

Fig. 7. Search results of the keyword "how to learn business" in Class Central

In contrast, Class Central's search algorithm may place greater emphasis on traditional keyword-based matching, thus leading to a wider range of search outcomes that exhibit varied degrees of pertinence. Although Class Central is a useful platform for collecting courses from different providers, its lack of advanced semantic capabilities might affect its ability to accurately understand the user's inquiry.

5 CONCLUSION AND FUTURE WORK

We live in a modern environment where learning does not have any barriers. There is clear evidence that online learning through MOOCs is the future of education for the next generation. The recent COVID-19 pandemic is a major example of the global movement from traditional to innovative approaches to learning. MOOCMaven platform offers a new experience of searching courses as users' think by understanding the keyword meaning. It also offers an expansive collection of datasets and APIs covering a wide range of online courses. Leveraging a state-of-the-art content extraction and collection graph pipeline, the platform houses thousands of online courses, making it a rich and diverse resource for learners and researchers alike. A standout feature of MOOCMaven is its incorporation of advanced semantic capabilities, including a powerful semantic search engine, which revolutionizes the way users discover and access course offerings.

The implications of our findings are multifaceted and hold great promise for the field of online education. By harnessing the power of semantic search, MOOCMaven is poised to personalize the learning experience for users based on their unique profiles and learning histories. This personalized approach, in line with the research findings of Katsaris and Vidakis [46], can significantly enhance the effectiveness of e-learning platforms, leading to heightened learner performance and engagement.

We set a roadmap to expand the dataset by providing auto-indexing and crawling for the web with course content. Providing a course recommendation based on search history is also the future research of this paper. These advancements are expected to further elevate the platform's ability to provide tailored course recommendations, catering to users' diverse learning needs and preferences. The envisioned future for MOOCMaven is one where it plays a pivotal role in driving global scientific progress, enabling the development of applications and research that leverage the vast potential of unified MOOC data.

Our research brings to the forefront the transformative impact of the MOOCMaven platform on online education. As we refine and expand the platform's offerings, we envision MOOCMaven becoming a cornerstone resource, revolutionizing how learners access knowledge and helping researchers advance human understanding. The future holds exciting possibilities for MOOCMaven, as it paves the way for a more inclusive, accessible, and effective era of online education.

6 REFERENCES

[1] B. An, T. Qu, H. Qi, and T. Qu, "Chinese MOOC search engine," in *Intelligent Computation in Big Data Era*, vol. 503, 2015, pp. 453–458. https://doi.org/10.1007/978-3-662-46248-5_55

[2] S. Papadakis, "MOOCs 2012–2022: An overview," *Adv. Mobile Learn. Educ. Res.*, vol. 3, no. 1, pp. 682–693, 2023. https://doi.org/10.25082/AMLER.2023.01.017

[3] F. Lazarinis, A. Karatrantou, C. Panagiotakopoulos, V. Daloukas, and T. Panagiotakopoulos, "Strengthening the coding skills of teachers in a low dropout Python MOOC," *Adv. Mobile Learn. Educ. Res.*, vol. 2, no. 1, pp. 187–200, 2022. https://doi.org/10.25082/AMLER.2022.01.003

[4] D. Y. Mohammed, "The web-based behavior of online learning: An evaluation of different countries during the COVID-19 pandemic," *Adv. Mobile Learn. Educ. Res.*, vol. 2, no. 1, pp. 263–267, 2022. https://doi.org/10.25082/AMLER.2022.01.010

[5] N. Mohamad, A. Othman, T. S. Ying, N. Rajah, and N. Samsudin, "The relationship between Massive Online Open Courses (MOOCs) content design and students' performance," *Int. J. Interact. Mob. Technol.*, vol. 15, no. 04, p. 4, 2021. https://doi.org/10.3991/ijim.v15i04.20201

[6] S. Dhawal, "By The numbers: MOOCS in 2022," *ClassCentral*, 2021. https://www.classcentral.com/report/mooc-stats-2021. [Accessed: Jan. 21, 2023].

[7] A. Fatima, C. Luca, and G. Wilson, "New framework for semantic search engine," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, Cambridge, United Kingdom: IEEE, 2014, pp. 446–451. https://doi.org/10.1109/UKSim.2014.114

[8] J. Yu *et al.*, "MOOCCube: A large-scale data repository for NLP applications in MOOCs," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, 2020, pp. 3135–3142. https://doi.org/10.18653/v1/2020.acl-main.285

[9] K. Veeramachaneni, S. Halawa, F. Dernoncourt, U.-M. O'Reilly, C. Taylor, and C. Do, "MOOCdb: Developing standards and systems to support MOOC data science," *arXiv*, 2014. [Online]. Available: http://arxiv.org/abs/1406.2015. [Accessed: Oct. 01, 2022].

[10] Z. Du, H. Chen, and J. Jiang, "Research on the big data system of massive open online course," in *2016 IEEE International Conference on Big Data (Big Data)*, Washington DC, USA: IEEE, 2016, pp. 1931–1936. https://doi.org/10.1109/BigData.2016.7840813

[11] S. Keskin, M. Şahin, and H. Yurdugül, "Online learners' navigational patterns based on data mining in terms of learning achievement," in *Learning Technologies for Transforming Large-Scale Teaching, Learning, and Assessment*, D. Sampson, J. M. Spector, D. Ifenthaler, P. Isaías, and S. Sergis, Eds., Cham: Springer International Publishing, 2019, pp. 105–121. https://doi.org/10.1007/978-3-030-15130-0_7

[12] M. S. Surya, M. S. Kumar, and D. Gandhimathi, "Student placement prediction using supervised machine learning," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India: IEEE, 2022, pp. 1352–1355. https://doi.org/10.1109/ICACITE53722.2022.9823648

[13] Z. Shou, P. Chen, H. Wen, J. Liu, and H. Zhang, "MOOC dropout prediction based on multi-dimensional time-series data," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–12, 2022. https://doi.org/10.1155/2022/2213292

[14] A. Hadioui, N. El Faddouli, Y. Benjelloun Touimi, and S. Bennani, "Machine learning based on big data extraction of massive educational knowledge," *Int. J. Emerg. Technol. Learn.*, vol. 12, no. 11, p. 151, 2017. https://doi.org/10.3991/ijet.v12i11.7460

[15] K. Braunschweig, J. Eberius, M. Thiele, and W. Lehner, "The state of open data limits of current open data platforms," in *Proceedings of the 21st World Wide Web Conference 2012*, Web Science Track at WWW'12, Lyon, France, April 16–20, 2012. ACM. http://wwwdb.inf.tu-dresden.de/opendatasurvey

[16] M. Moundridou, E. Zalavra, K. Papanikolaou, and A. Tripiniotis, "Collaboratively developing open educational resources for engineering educators in SlideWiki," *Int. J. Eng. Ped.*, vol. 9, no. 2, pp. 99–116, 2019. https://doi.org/10.3991/ijep.v9i2.9959

[17] J. Beall, "The weaknesses of full-text searching," *The Journal of Academic Librarianship*, vol. 34, no. 5, pp. 438–444, 2008. https://doi.org/10.1016/j.acalib.2008.06.007

[18] S. Pandiarajan, V. M. Yazhmozhi, and P. Praveen Kumar, "Semantic search engine using natural language processing," in *Advanced Computer and Communication Engineering Technology*, H. A. Sulaiman, M. A. Othman, M. F. I. Othman, Y. A. Rahim, and N. C. Pee, Eds., in Lecture Notes in Electrical Engineering, vol. 315. Cham: Springer International Publishing, 2015, pp. 561–571. https://doi.org/10.1007/978-3-319-07674-4_53

[19] S. Jiang, T. F. Hagelien, M. Natvig, and J. Li, "Ontology-based semantic search for open government data," in *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, Newport Beach, CA, USA: IEEE, 2019, pp. 7–15. https://doi.org/10.1109/ICOSC.2019.8665522

[20] Z. Pan, "Optimization of information retrieval algorithm for digital library based on semantic search engine," in *2020 International Conference on Computer Engineering and Application (ICCEA)*, Guangzhou, China: IEEE, 2020, pp. 364–367. https://doi.org/10.1109/ICCEA50009.2020.00085

[21] L. Pfahler and K. Morik, "Semantic search in millions of equations," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Virtual Event, CA, USA: ACM, 2020, pp. 135–143. https://doi.org/10.1145/3394486.3403056

[22] K. M. Alzahrani and M. Meccawy, "MOOCs one-stop shop: A realization of a unified MOOCs search engine," *IEEE Access*, vol. 9, pp. 160175–160185, 2021. https://doi.org/10.1109/ACCESS.2021.3130841

[23] S. Aryal, A. S. Porawagama, M. G. S. Hasith, S. C. Thoradeniya, N. Kodagoda, and K. Suriyawansa, "MoocRec: Learning styles-oriented MOOC recommender and search engine," in *2019 IEEE Global Engineering Education Conference (EDUCON)*, 2019. https://doi.org/10.1109/EDUCON.2019.8725079

[24] Q. Cheng and Y. Gao, "Courducate—An MOOC search and recommendation system," p. 10.

[25] S. Kagemann and S. Bansal, "MOOCLink: Building and utilizing linked data from massive open online courses," in *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, Anaheim, CA, USA: IEEE, 2015, pp. 373–380. https://doi.org/10.1109/ICOSC.2015.7050836

[26] S. Lee, R. Girish, and Y. U. Kim, "Courserush a MOOC search engine," 2017.

[27] R. Xu, "POS weighted TF-IDF algorithm and its application for an MOOC search engine," in *2014 International Conference on Audio, Language and Image Processing*, Shanghai, China: IEEE, 2014, pp. 868–873. https://doi.org/10.1109/ICALIP.2014.7009919

[28] M. Lanthaler, "Creating 3rd generation web APIs with hydra," in *Proceedings of the 22nd International Conference on World Wide Web*, Rio de Janeiro, Brazil: ACM, 2013, pp. 35–38. https://doi.org/10.1145/2487788.2487799

[29] D. Zeber *et al.*, "The representativeness of automated web crawls as a surrogate for human browsing," in *Proceedings of the Web Conference 2020*, Taipei, Taiwan: ACM, 2020, pp. 167–178. https://doi.org/10.1145/3366423.3380104

[30] H. Lou and F. Farnoud, "Data deduplication with random substitutions," *IEEE Trans. Inform. Theory*, vol. 68, no. 10, pp. 6941–6963, 2022. https://doi.org/10.1109/TIT.2022.3176778

[31] X. Wang and H. Ning, "TF-IDF keyword extraction method combining context and semantic classification," in *Proceedings of the 3rd International Conference on Data Science and Information Technology*, Xiamen, China: ACM, 2020, pp. 123–128. https://doi.org/10.1145/3414274.3414492

[32] M. Chihab, M. Chiny, N. Mabrouk, H. Boussatta, Y. Chihab, and M. Y. Hadi, "BiLSTM and multiple linear regression based sentiment analysis model using polarity and subjectivity of a text," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 10, 2022. https://doi.org/10.14569/IJACSA.2022.0131052

[33] P. Singh, R. Piryani, V. K. Singh, and D. Pinto, "Revisiting subject classification in academic databases: A comparison of the classification accuracy of Web of Science, Scopus & Dimensions," *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 2, pp. 2471–2476, 2020. https://doi.org/10.3233/JIFS-179906

[34] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," *arXiv*, 2020. https://doi.org/10.48550/arXiv.1910.01108

[35] I. Cachola, K. Lo, A. Cohan, and D. Weld, "TLDR: Extreme summarization of scientific documents," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 4766–4777. https://doi.org/10.18653/v1/2020.findings-emnlp.428

[36] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," *arXiv*, 2019. https://doi.org/10.48550/arXiv.1908.10084

[37] H. Jegou, M. Douze, and J. Johnson, "Faiss: A library for efficient similarity search," 2018. https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/. [Accessed: Nov. 10, 2022].

[38] M. Aumüller, E. Bernhardsson, and A. Faithfull, "ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms," *Information Systems*, vol. 87, p. 101374, 2020. https://doi.org/10.1016/j.is.2019.02.006

[39] S. Ramires, "FastAPI Documentation," https://fastapi.tiangolo.com/. [Accessed: Jul. 10, 2023].

[40] D. Eddelbuettel, "A brief introduction to Redis," 2022. https://10.48550/ARXIV.2203.06559

[41] P. Siriwardena, "OAuth 2.0," in *Advanced API Security*, Berkeley, CA: Apress, 2014, pp. 91–132. https://doi.org/10.1007/978-1-4302-6817-8_7

[42] A. F. Tatang and A. M. Algarni, "MOOCMaven API documentation," *MOOCMaven API Documentation*. https://api.moocmaven.com/docs

[43] Secondl1f3 and A. F. Tatang, "Secondl1f3/semantic-search-ann-moocs: Initial release of MoocMaven Open Data Platform," Zenodo, 2023. https://10.5281/ZENODO.7990811

[44] I. OpenAPI, "OpenAPI Specification 3.0.0," 2017. https://github.com/OAI/OpenAPI-Specification. [Accessed: Jul. 10, 2023].

[45] "Locust Documentation." https://docs.locust.io/en/stable/. [Accessed: Jul. 10, 2023].

[46] I. Katsaris and N. Vidakis, "Adaptive e-learning systems through learning styles: A review of the literature," *Adv. Mobile Learn Educ. Res.*, vol. 1, no. 2, pp. 124–145, 2021. https://doi.org/10.25082/AMLER.2021.02.007

7 AUTHORS

Ahmad Fajar Tatang received B.Eng. degree in Information Technology from the University of Muhammadiyah Prof. DR. HAMKA, Jakarta, Indonesia, in 2015. He is pursuing a Master's at King Abdulaziz University, Saudi Arabia. Besides that he is also working as a professional Software Engineer. His research interests include software engineering, web engineering, and natural language processing.

Abdullah M. Algarni received the Ph.D. degree in Computer Science from the College of Natural Sciences, Colorado State University, USA, in 2016, and his master's degree in Computer Science from Colorado State University in 2014, and another master's degree in Software Systems Engineering from the University of Melbourne, Australia, in 2008. He is currently working as an Associate Professor in the Computer Science Department, King Abdulaziz University, Jeddah, Saudi Arabia. His research interests include software engineering, software security, and cybersecurity.