IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 30, 2021, accepted November 12, 2021, date of publication November 25, 2021, date of current version December 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3130841

MOOCs One-Stop Shop: A Realization of a Unified MOOCs Search Engine

KHADIJAH M. ALZAHRANI AND MARAM MECCAWY
Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
Corresponding author: Khadijah M. Alzahrani (Kalzahrani0193@stu.kau.edu.sa)

ABSTRACT Massive open online courses (MOOCs) have gained great popularity among adult learners, as many professionals use them to advance their careers. However, many MOOCs providers offer a large number of courses across many subjects, making the search and selection of the appropriate course cumbersome for the learner through a regular Web search. Hence, there is a crucial need for a unified search engine that acts as a one-stop-shop for all MOOCs. This paper presents the design and implementation of a unified MOOCs search model for retrieving and recommending courses from multiple MOOCs. The evaluation showed that the proposed model is reliable and convenient. Initial results showed that the system could present courses from multiple MOOCs that suit users' needs. Moreover, the recommendation functionality is applied for a better user-centric orientation.

INDEX TERMS MOOCs, search engines, recommender systems, personalization, recommendations, clustering algorithms.

I. INTRODUCTION

Online learning platforms have gained high popularity worldwide as a solution to empower education and employability. This is also true for massive open online courses (MOOCs), especially during the current Covid-19 pandemic, which has precluded in-person learning for millions of people worldwide.

MOOCs are online learning platforms that offer open admission to educational content for anyone from anywhere at any time. They offer courses and training programs to individuals and institutions via open platforms that provide free or paid courses [1], such as edX [2], Khan academy [3], and Udemy [4].

Top-ranked universities, such as MIT and Harvard, have invested in MOOCs to enhance education through MOOCs campuses for enrolled students only (e.g., MITx) or via open-platform MOOCs for global students (e.g., edX).

Many MOOCs have been launched since 2013, and some have stood out [5], [6], such as edX, Udacity [7], and Coursera [8]. Follow by example, Arabic MOOCs such as Edraak [9] and Rwaq [10]; the largest MOOC platforms in the Arab region [11]. However, with the transformation of the educational process and the movement to follow the MOOCs' approach, MOOCs' number has dramatically increased year by year. According to Shah, around 13.5K MOOCs were launched by the end of 2018, and about 2.5K more were added in 2019 by over 900 universities worldwide [12], [13]. Thus, online learners are facing an increasing number of course choices on the Web [14]. As a result, some individuals find it difficult to navigate and locate suitable materials among multiple MOOCs and their educational content, leading them to get overloaded with too many options and make poor decisions [15]. Eventually, they get ''lost in hyperspace'' (MOOCs in this case) while searching through MOOCs hypertext systems [16], [17].

As the Web carries a large amount of information that can cause an information overload situation, search engines have become an important tool that guides people to access the appropriate information. However, search engines retrieve a wide variety of results that often leave users with information overload [18]. On the other hand, recommender systems are developed to narrow the information for each user. However, the recommender needs the availability of correct data and user preferences in order to tailor each user's recommendations. Those information retrieval techniques are enhanced to support Cross language information retrieval (CLIR) to process queries expressed in many languages regardless of the actual data language [19]. However, CLIR techniques are challenged by the properties of extremely derivational and inflectional languages like Arabic [20].

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

Overall, search engines can be more precise and get more specific information through data scaling and clustering based on a set of interests. Based on those sets of interests, recommendation and personalization approaches can add more value by tailoring each user's results. Some studies have proposed the idea of a unified search approach to MOOCs [21]–[28]; such studies are still limited. Where many of them present models and platforms with implementation deficiencies. From a technical perspective, they suffer either from manual scraping and indexing or because data are limited to no more than five MOOCs. Moreover, their focus is on specific subjects in either English or Chinese. This is a weakness in CLIR with respect to the Arabic language, because available hybrid MOOCs locator systems do not target Arabic speakers. Moreover, since accessing useful MOOCs data is still a challenge, this may affect such models' performance.

This research aimed to overcome the overload of MOOCs information when a user searches for a specific course to study; it proposed a model that searches and recommends courses for users.

This is achieved by demonstrating a model of unified MOOCs Recommender Search Engine (MRSE), that acts as the central hub for searching and locating various MOOCs, then recommending the most suitable courses for a certain user-learner [29]. A set of research questions has been formulated as follows:

Q1) What are the main components needed for designing a unified recommender search engine?

Q2) What is the most suitable form of implementation using open source tools and platforms?

Q3) How will the implemented tool perform in real-world settings?

Q4) How accurate are the search engine results, and how useful are the recommendations for the user?

Q5) How do the MRSE and other unified MOOCs search models compare in terms of outcomes?

To answer the first and second questions, this study reviews current MOOCs search engines along with related work throughout the inspection and implementation phases. Examining other search models related to MOOCs or any other field helps to design an optimal architecture based on the available capabilities.

This study answers questions three and four by analyzing system performance through experimental querying, and it measures the system's convenience with a User Acceptance Test. This test evaluates whether the search model meets user requirements. This means it can measure the system's acceptance from the perspective of a real user.

Finally, the paper addresses question five with further evaluation by comparing the MRSE model with other MOOC unified recommender search models.

II. REVIEW OF THE LITERATURE

This section reviews MOOCs retrieving and mining techniques relevant to this work and discusses related work about MOOC unified search models.

A. INFORMATION RETRIEVAL TECHNIQUES AND INFORMATION OVERLOAD

Information retrieval (IR) and data mining techniques gained Web users' trust by delivering the information that matched their needs based on high recall and precision [30]. Such techniques include search engines and recommender systems, focusing on delivering precise results to users based on their needs. Moreover, cross-language information retrieval (CLIR) is established to support the process of finding results written in natural languages with IR systems that can perform queries expressed in other languages [19], [31], [32]. However, CLIR tools rely on translation systems, which are challenged by the different properties of languages [33]. Thus, many researchers are rapidly studying CLIR for diverse languages, including Arabic [20], [34]–[39].

Applications of IR techniques are diverse depending on the scale of the data and diversity. For instance, search engines can cover overall Web content and act as Web search engines such as Google, or they can be optimized to cover a smaller scale, such as websites search bars. Moreover, search engines' results are characterized through search-result-clustering-engines for further relevancy and accuracy. These engines cluster search results by grouping similar documents into groups and then re-generating results based on terms and topics [40], [41]; such as Carrot2 [42].

Search engines are also characterized according to their data collection (type and domain). Such are Vertical or topical search engines, through an explicit segment of online content based on a particular domain or knowledge [30], [43]; for example, Google Scholar. This characterizing provides many benefits over regular Web searches by offering more relevant information and greater precision due to a scope limitation that supports specific user tasks [44]. Likewise, recommender systems have many approaches based on resource filtering techniques [45]–[47]. These systems are applied on a specific scale of data, usually limited to a website based on explicit criteria and relations through content analysis filtrations [48], [49]. Both the industry and the academia have proposed several recommendation techniques that are widely used, particularly in Web applications, such as Netflix and Amazon. Recommender systems promote recommendations for a more user-centric approach. Moreover, these systems benefit from users' interactions (expressed as user preferences or inferred information) to enhance personalized recommendations through user action analysis [50], [51].

However, search engines and recommender systems both have their shortcomings along with their advantages. Given the rapid growth of Web content and website numbers, for instance, Web search engines return an enormous set of results for any given search query [43]. Moreover, the retrieving is highly affected by unclear queries. Thus, search engines are continuing to retrieve some irrelevant results. This forces the user to traverse many pages to reach the desired results, and he/she might have difficulty in locating content relevant to the initial query [52]. In contrast, recommender systems suffer from the cold-start phenomenon, which occurs in the absence of user information in the case of new users or users with only a few preferences [53], [54].

Nevertheless, the CLIR concept is well established in both search engines and recommender systems, as can be noticed in the Google search engine and Amazon recommender system [55]. However, many studies have shown that search engines designed for English speakers are often less effective with Arabic queries [20], [34], [36]–[39], [56]. Thus, it is essential to be further addressed by academic research.

B. MOOCs AND INFORMATION RETRIEVAL TECHNIQUES

Like many other Web platforms, MOOCs use information retrieval (IR) techniques in order to reach the correct user and improve the learning quality [57], [58]. Most MOOCs recommend courses based on the user's enrollment history or potential interest if the user has no previous history recorded by a certain provider [59]. One example is Course-Match [60], a machine learning tool launched by Coursera due to the Covid-19 pandemic. The aim is to help universities enrolled students fulfill their curriculum requirements online instead of via traditional schooling. Hence, Course-Match matches Coursera courses relevant to the universities' taught courses and then recommends relevant courses to students [61]. However, since the main challenge of e-learning is to locate and deliver the right content to the right person, it is still a challenge to match the educational content with the learner [57], [62].

Many researchers have proposed the implementation of recommender systems and the personalization of MOOCs platforms. Such studies include [63]–[66], where they introduced a diversity of implementations and recommendations for MOOC platforms based on many criteria, such as user behavior, course content, and filtering and clustering techniques. Others suggested or applied artificial intelligence (AI) algorithms to improve the personalization of MOOCs and enhance personalized learning [67]. MOOCs movement towards personalization aims to create an adaptive MOOCs environment that relates courses to the correct learner in order to decrease dropout rates [68]. This is applicable if users make poor choices due to MOOCs information overloads with regular searches. Hence, using a vertical search engine for a MOOCs search instead of a regular Web engine would be more convenient.

C. UNIFIED MOOCs RETRIEVING TECHNIQUES

In order to overcome MOOCs information overload and increase searching efficiency, MOOCs search engines and unified search approaches have been proposed. The top MOOC search engines are Class Central [69], MOOC.org [70], and My MOOC [71]. These tools retrieve results from the most common MOOCs. Recently, these engines started presenting courses from Arabic MOOCs, such as Rwaq and Edraak. However, both Class Central and My MOOC lean towards classifying and customizing results rather than personalization. Therefore, improved personalization is needed.

An and Qu [25], proposed "MOOCsoso" to demonstrate a Chinese MOOCs' vertical search engine that retrieves results limited to Chinese courses from Chinees MOOCs only.

Likewise, Lee *et al.* [21], proposed "Courserush" as a unified MOOCs search engine based on manual data scraping and BM25 ranking function from edX, Udemy, and Coursera only. However, the search suffers from the manual assembling of data sets as well as indexing. Courducate is also a unified MOOCs search engine implemented based on manual data scraping and self-developed search engine and algorithms to rank the results. In addition, the BM25 ranking function, along with a self-built ranking function based on Apache Lucene are used for indexing. The model scraped data from edX, Udemy, Udacity, Coursera, and Khan Academy. Moreover, the model customized the results based on the universities' and tutors' information [22]. However, both models are keen to introduce more personalized outputs and user interactions [21], [22].

Further studies in this area include unified MOOCs recommender systems. Bousbahi and Chorfi [24], proposed a unified MOOCs recommender system (MOOC-Re), patterned on the Cased-based recommender system [72], and crawled data. However, the search did not state which MOOCs were crawled. Moreover, the system is still under implementation, and more real-world testing is needed.

He *et al.* [26], proposed a unified MOOC recommender system based on collaborative filtering and commodity filtering to analyze users' historical preferences. Nevertheless, the system recommends Chinese courses only in a social work major based on self-built data sets.

On the other hand, MoocRec.com is an implementation of a hybrid approach by integrating both the search engine and the content recommender system, along with a Matrix Factorization model. However, the engine retrieves data only from edX and Coursera. More testing and accuracy improvements regarding recommendations are needed [23].

Likewise, Aryal *et al.* [28], proposed a hybrid platform named MoocRec1 that recommends courses by integrating MOOCs video styles and users' learning styles through a content analysis of MOOCs data (videos and transcripts). Moreover, allowing users to customize their results by course filtration based on topic parameters. Though, the system allows the user to search only Computer Science courses from edX and Coursera. As an attempt to upgrade the system, Fazuludeen *et al.* [27], proposed a second version named MoocRec2 for Humanities courses based on the same concept. Nevertheless, both systems still need to be validated with actual users and real-world data [27], [28].

Table 1 shows the results of MOOCs unified retrieving models using different techniques.

In summary, many studies have proposed several unified MOOCs search models. Some used search engines only, while others used recommender systems. In contrast, others offered integration of both the search engine and the recommender system. Some studies suffer from manual data collecting and structuring, as the researchers had to use manual

TABLE 1. Summary of unified MOOCs retrieving models.

Proposed model	IR techniques	Shortcomings
MOOCsoso [26]	MOOCs search engine	• Chinese courses only • Chinees MOOCs • No recommendations
Courserush [21]	MOOCs search engine	• Manual techniques • English courses only • edX, Udemy, and Coursera • No recommendations
Courducate [22]	MOOCs search engine	• Self-developed • English courses only • edX, Udemy, Udacity, Coursera, and Khan Academy • No recommendations
MOOC-Re [24]	MOOCs recommender system	• Real-world testing • Still under development
He et al. proposed model [27]	MOOCs recommender system	• Manual techniques • Chinese courses for social work major only
MoocRec.com [23]	MOOCs recommender and search engine	• Real-world testing • English courses for computer science major only • edX, and Coursera
MoocRec1 [28]	MOOCs recommender and search engine	• Real-world testing • English courses for humanities major only • edX, and Coursera
MoocRec2 [27]	MOOCs recommender and search engine	• Real-world testing • English courses • edX, and Coursera

scraping and self-built data sets and indexing. Moreover, most systems lack a diversity of resources, user interactions, and additional real-world experimentation where regular users can rate the systems. In addition, revised studies have shown unsupported CLIR regarding Arabic, where the proposed MOOCs-locator systems are fully designed for English or Chinese speakers only. Hence, this research proposes an architecture that can retrieve and recommend courses from multiple MOOCs for users in one platform. Moreover, it can search for MOOCs using two different languages, Arabic and English.

III. ARCHITECTURE AND IMPLEMENTATION

A. MOOCs RECOMMENDER SEARCH ENGINE PROPOSED MODEL

In [29], we proposed a hybrid personalized search model: the MOOCs Recommender Search Engine (MRSE). The model integrates the recommender system and the search engine to provide a personalized MOOCs search service. The MRSE model aims to help users browse MOOCs courses in a single platform as a "one-stop-shop for MOOCs" and recommends courses that suit users' needs based on their user profile. The MRSE model first retrieves information by capturing or crawling the MOOCs Web pages and then indexing the Web pages as data structures to retrieve the required content. The structured data is then fed into the recommender system, and then filtering algorithms are applied to employ user modeling for personalization later on. This approach has a lot of potential for empowering each tool's characteristics: a) the search engine results can be more relevant through the vertical search, and b) the recommender system can overcome the cold-start problem by using the engine-crawled data. The MRSE model is detailed in Fig 1.

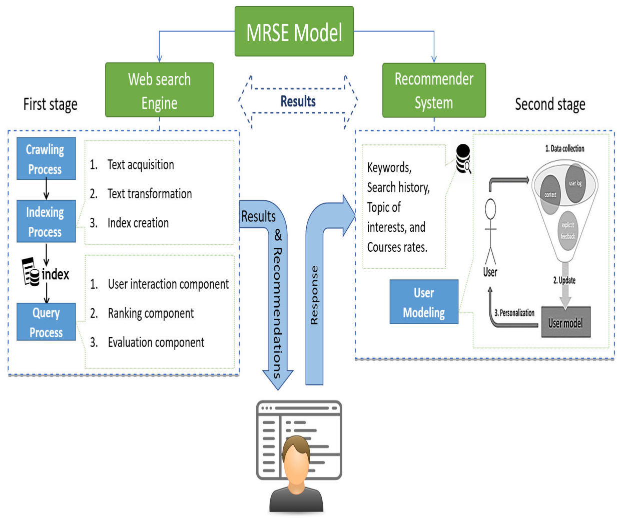

FIGURE 1. The MRSE proposed model [20].

B. UNIFIED MOOCS SEARCH ENGINE ARCHITECTURE

This paper presents the implemented architecture of the MRSE. The MRSE architecture consists of three modules: Crawling module, indexing module, and querying module. However, the system is still under implementation, and this architecture is considered as the first version of MRSE (Fig 2).

The MRSE prototype is based on integrating open source technology that runs on distributed platforms. To adapt the system structure to any search engine, we used the open source Apache Nutch version 1.16 [73] as the crawler and Apache Solr version 8.1 [74] as the indexing server and search engine. For recommendation functionality empowering, we activated Solr to build in the results clustering engine—Carrot2 [42] and Solr's Suggester component [75]—to generate a clustering-based recommendation. In contrast, the interface is linked and powered by Solr and PHP and Cascading Style Sheets (CSS).

First, the crawling module is responsible for MOOCs data crawling and then parsing data segments. Second, the indexing module is responsible for constructing the index of the crawled Web pages. The indexed documents are clustered to generate clustering-based recommendations. Third, the querying module provides a search interface and conducts

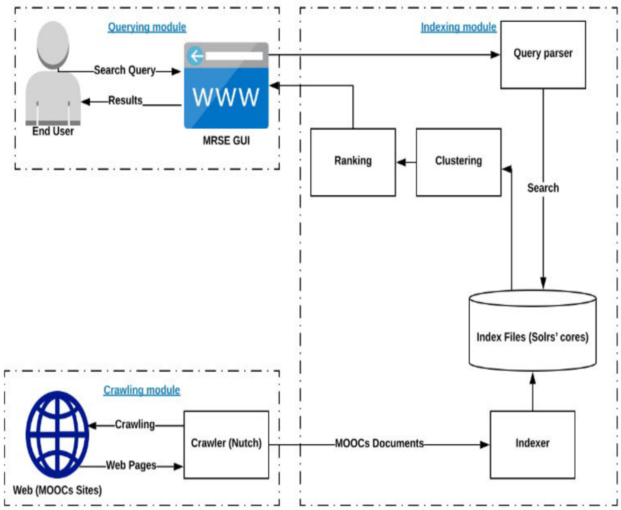

FIGURE 2. The MRSE modules (beta version).

searching, after which the query results are displayed in this module.

The core of the data collection process is a Web crawler in the *crawling module*, which is injected with a set of MOOCs URLs' webpages. The crawler collects data from MOOCs webpages and sub-pages and then generates and parses the collected data to be indexed. Nutch, in this case, is injected with the URLs of eight MOOCs platforms—edX [2], Coursera [8], Udacity [7], Edraak [9], Rwaq [10], Khan Academy [3], Harvard Online Courses [76], and Udemy [4]—to activate the vertical search and ensure the diversity of results. Then it generates data segments to the indexer. Subsequently, the crawler generates data segments to the indexer.

To create data segments, Nutch first injects the MOOC URLs into the crawling database "crawldb." The crawldb maintains information on all injected URLs (fetch schedule, fetch status, metadata, ...). Then Nutch generates a link database "linkdb" that includes source URLs and the anchor text of each link. Those links are then fetched by Nutch's fetcher to create a fetched list from the URL's content. Afterwards, Nutch parses and processes the fetched content of each web page to generate a set of crawled data segments. Each segment is a set of URLs fetched as a unit and saved based on time and date. Those segments have subdirectories that maintain information on a set of URLs to be fetched, the status of URL fetching, the raw content retrieved from URLs, the parsed text of URLs, and outlinks and metadata parsed from URLs. These four processes form a cycle that is repeated for a few rounds to enhance data collection by adding new data and removing all repetitive data. With each round, the crawldb is updated using the segment's information.

Thereafter, Nutch prepares the data for indexing by crawling the final segment file that contains the last fetched data labeled with the latest date to ensure data recency. Moreover, Nutch inverts all the links in linkdb, so the indexer scoring function can process and rank incoming anchor text within MOOC pages. By this phase, the indexer uses all data from crawldb, linkdb, and segments for indexing.

The *indexing module* starts indexing and building the data set and stores the data as index files to be ready for the query process. At this point, Solr indexes the collected data and creates index documents. Then, each document is analyzed and prepared for the querying process. In this stage, the system parses and transforms the textual data into tokens by applying analysis filters, such as "tokenizers" and "filters". Then, the output of this process is used to match the query.

Through the querying process, Solr's request handler handles the search query and states the logic to be used when processing a search request. The request handler initiates the query processing by calling the Solr's default query parser known as the Lucene query parser, which understands the terms of a query. The standard query parser has a filter query that runs the query over the entire index and stores results. Meanwhile, Solr's default query function, term frequency-inverse document frequency (TF-IDF), is applied within the query parser to rank the results. The term frequency function rate results correspondingly to the number of times a query term appears in the document. At this stage, the search results are arranged into categories, and the clustering groups the search results. Afterward, Solr's default response writer presents the final response of the search query.

Simultaneously, clustering components are activated within the indexing module to perform a clustering-based recommendation to overcome the recommender system's absence in this version by combining the vertical search with the clustering components' results, as shown in Fig 2. Carrot2 and Solr's Suggester component, in this case, generated the clustering-based recommendation.

The clustering plugins automatically discover groups of related search documents by applying a clustering algorithm on top of each search hit and assigning human-readable labels to these groups.

The clustering process has two rounds. The first round of recommendations is generated by the results clustering component—Carrot2—through the Lingo algorithm [77] and the STC algorithm [78]. The component regenerates results as a tree from the seed query term.

The second round of recommendations is generated by Solr's Suggester through "AnalyzingInfix suggesters" algorithms [79]. The Suggester returns fields that have exactly the same terms as the query.

Finally, the querying process is performed through the *querying module* in cooperation with the *indexing module*. The querying process starts with a user's query request via the system interface (Fig 3). Then input terms are analyzed and processed by the *indexing module* via query parsing through the index files, and then it starts clustering and ranking the results for the user.

In summary, the unified MOOCs search system can retrieve MOOCs information and recommend courses to

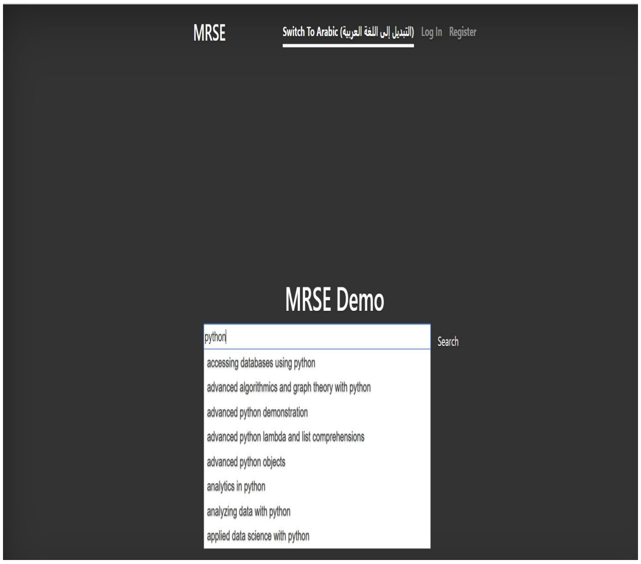

FIGURE 3. The MRSE interface.

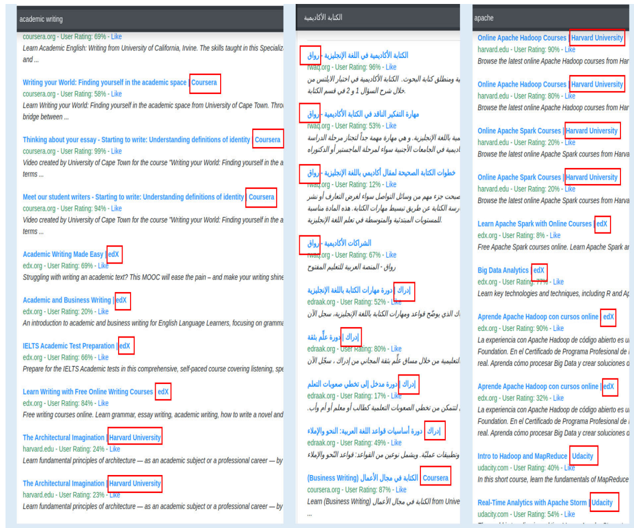

FIGURE 4. Samples of queries displaying the diversity of MOOCs (Coursera, edX, Harvard University, Edraak (إدراك), and Rwaq (رواق)) and courses in MRSE.

users. The system retrieves Arabic and English courses from different MOOCs with many subjects and recommends the results in a convenient GUI to provide a good user experience. The system is deployed on two virtual machines in order to simulate a production environment. The system's performance is monitored throughout the implementation to ensure reliability, alliance, and integration.

The proposed model integrates both the vertical search engine and the recommender system to achieve personalized experiences. However, the absence of actual user history and data does not apply to the recommender system's integration. Hence, the implementation is limited to the vertical search scope, deployed with clustering components in order to empower recommendation functionality.

IV. EVALUATION

A. REALIZATION OF THE SYSTEM

Aimed at MOOCs search requirements, this paper mentioned earlier the deficiency of the general search engines and MOOCs information overload and therefore designed and implemented a unified MOOCs recommender search system. The implemented system was verified through a real data set test of model performance (Fig 4 and Fig 5).

The implementation of the MRSE model managed to automate data retrieving instead of manual scraping and indexing. The system retrieved different courses (Arabic and English) with a variety of subjects from multiple MOOCs (edX, Coursera, Udacity, Edraak, Rwaq, Khan Academy, and Harvard Online Courses) and then offered them in a single search platform instead of many platforms. The main results are retrieved based on keyword matching generated by Solr's typical ranking by transforming the original text into tokens. Most results are close to user queries, as shown in Fig 6 and Fig 7.

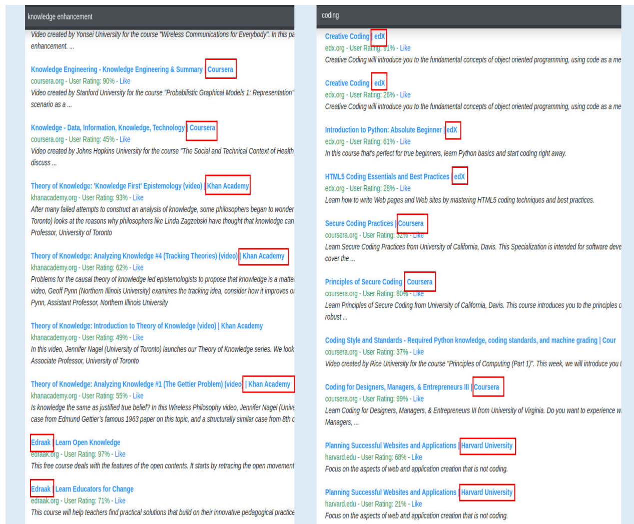

FIGURE 5. Samples of queries displaying the diversity of MOOCs (Coursera, Khan Academy, Edraak, edX, and Harvard University) and courses in MRSE.

However, some of the retrieved pages are not related to the desired courses; hence, accuracy and course-restricted results are still required (Fig 8 and Fig 9).

On the other hand, recommended courses are more relevant to users' queries than the main results since they are based on term correlations. Carrot2 generated the first round of recommendations by building a correlation between documents with the same terms. After auto-organizing documents from search results into thematic categories, machine learning algorithms were applied to those categories. This technique was ideal for English terms; however, it was not for Arabic terms, even though Carrot2 displayed solutions for the Arabic clustering in 2015. Hence, Arabic recommendations were irrelevant, and the content was unclear compared with

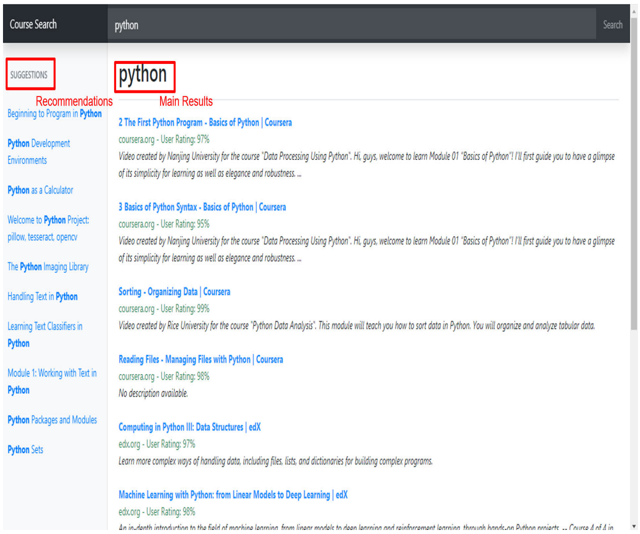

FIGURE 6. "Python" search results' relevancy from both keyword matching and term correlation.

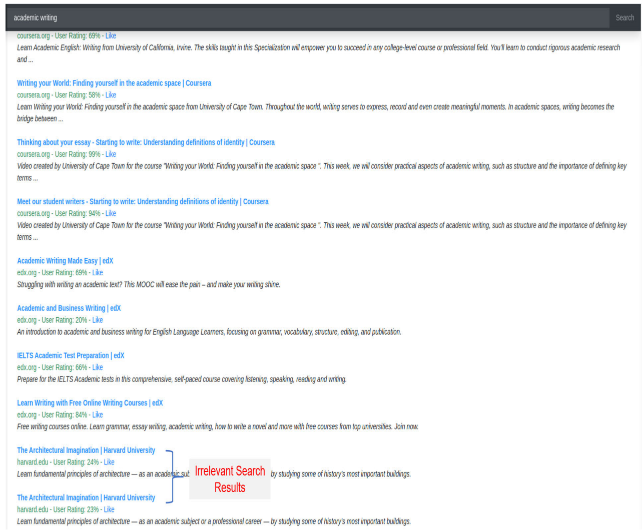

FIGURE 8. Irrelevant results in English search.

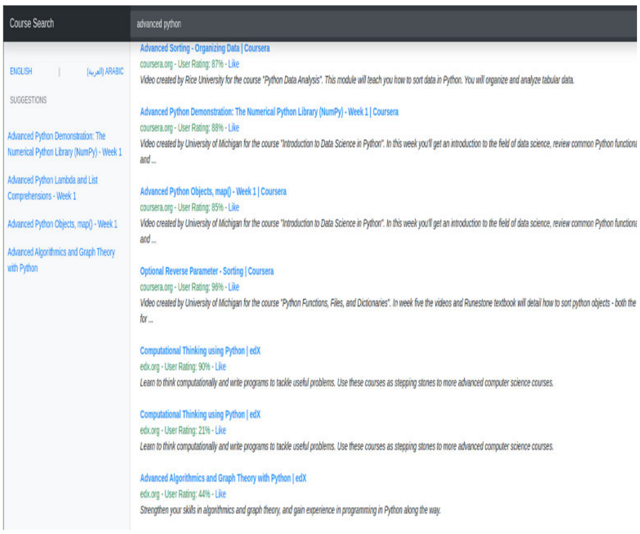

FIGURE 7. "Advanced python" search results' relevancy for both keyword matching and term correlation.

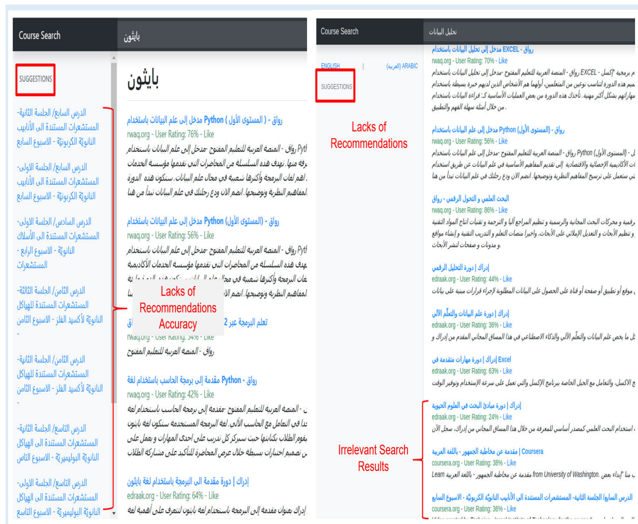

FIGURE 9. Irrelevant results and recommendation deficiencies in Arabic search.

the English recommendations (Fig 9). Also, manual Arabic segmentation is required.

The system still needs a recommendation process. Hence, the Suggester component was applied to generate the second round of recommendations based on query results instead of using Carrot2. The Suggester component triggers recommendations by forming clusters of strings, so each suggestion presumably is a string that occurs in multiple documents. The Suggester recognizes Arabic and English terms equivalently and manages to present meaningful results.

In addition to using a basic retrieval function, users can alternate between the Arabic and English searches and get results and recommendations on the same page. The system also offers some auxiliary functions, such as auto-complete and spell-checking. To enhance the users' experience throughout trials, the system mimics courses' ratings through pseudo-scoring that generates random ratings for each course. The system's ranking of results is affected by click-throughs counting of visited links and the percentages of likes. These impacts have a potential need for future work, which may generate real user data and preferences.

B. THE SYSTEM REQUIREMENTS FULFILLMENT

This paper's primary goal was to present a proposed unified search model for MOOCs that functions well in real-world settings, as was explained in the previous section. For a more comprehensive evaluation, we performed a User Acceptance Test to measure how well the system fulfilled the model's requirements and to investigate other aspects such as usability and the human impact. In this case, the participants willingly agreed to test the system online during the COVID-19 national lockdown. However, due to limitations of time and the recruits' schedules, only 10 adults participated in this experiment. These participants were aged between 25 and 40, and they were Arabic speakers with different levels of competence in English. Moreover, they all had advanced degrees in either computing or engineering.

The system should allow each user to query and retrieve different MOOCs courses related to what they searched for in one platform. Then receive recommended results based on their queries.

The user experience with the system proves the reliability and alliance of the system. Users managed to search and view multiple Arabic and English courses from different MOOCs that matched their queries in a single platform (Fig 10). Where ninety-six percent of users agreed or strongly agreed that the system met their functional requirements. They also managed to incorporate with the system by creating profiles, liking courses, and viewing courses through brief descriptions and direct links.

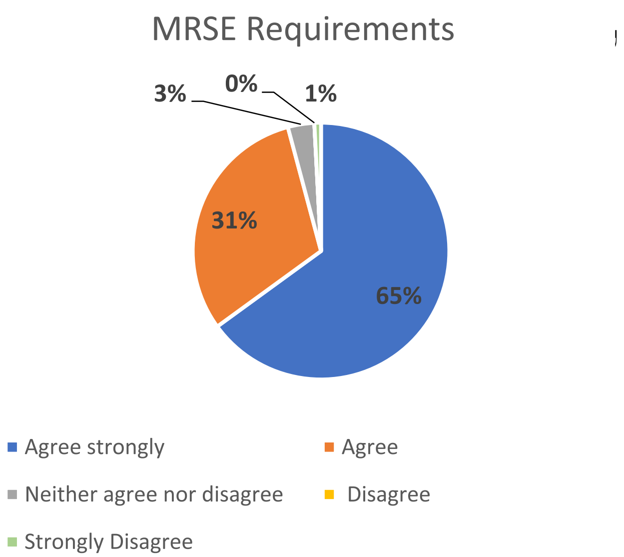

FIGURE 10. Concluded rating of users' requirements achievement.

In general, the system experiments are proof of the need for a MOOCs unified search engine. The overall effectiveness and convenience of the MRSE model are shown in Fig 11.

About one hundred percent of users agreed or strongly agreed on the system's convenience and enhanced their experience with the system Arabic search. And about ninety percent agreed or strongly agreed with the high quality of the suggested courses.

V. DISCUSSION

Based on MOOCs resources search and recommendation requirements during online learning, this paper discussed the deficiency of unified MOOCs search models, designed and implemented a MOOCs unified search model, and verified

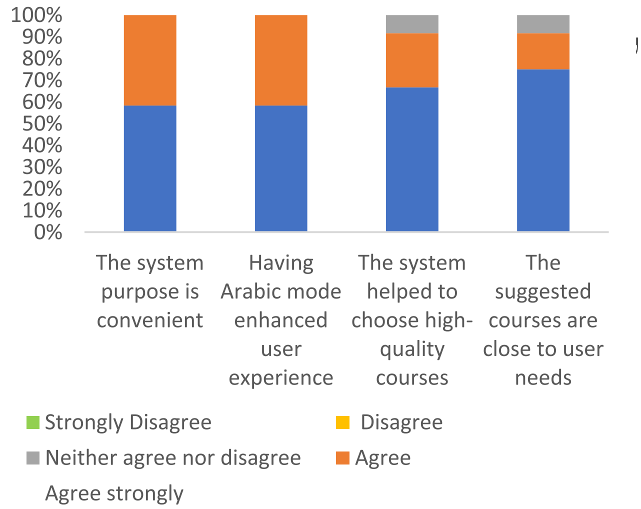

FIGURE 11. Users' overall gratification with MRSE.

the system's outcome through a real-world test of the model's performance.

The system's architecture is designed to retrieve courses from different MOOCs platforms and recommend Arabic and English results in a one-stop-shop. As noted in the literature, each tool and model has many negatives in contrast to their positive possibilities. Hence, the proposed model embraced techniques that best fit the need for an ideal MOOCs searching platform and should serve well in future adaptations and deployments. The model consists of a vertical search engine as the retrieving functionality empowered by clustering components to activate recommendation functionality.

The system's implementation went through new high-performance, open sourced tools and a combination of high-ranked Arabic and English MOOCs. To serve our purpose, we empowered each tool's characteristics to reach optimum performance. Although the system is distributed, its alliance, reliability, response, and throughput were optimistic.

Overall, the system managed to collect MOOCs data and present diversified results in both Arabic and English in a convenient and friendly GUI. It also managed to present clustering-based recommendations for users based on their queries. The results of both Arabic and English searches were promising and relevant to users' queries. However, English results and recommendations were more accurate and topic-related since all the used tolls are mainly developed based on English syntax. Even though some IR tools addressed Arabic syntax recognition, it is still minor.

The main challenge was to generate recommendations despite the deficiency of users' data preferences. Hence, the initial recommendations were generated by build-in clustering components instead of by applying a recommender system. These applied techniques could be vital in achieving later-on personalization of future work [52]. Moreover, users made positive observations about the system's trials.

Along with the MRSE model, online MOOC locaters such as Class Central [69], MOOC.org [70], and My MOOC [71], have precise foundations to establish a personalization environment. These platforms enable users to customize their interests, where each system can use each user's customized data to predict further interests and offer more personalized results. However, some platforms still suffer from lack of proper querying, such as in MOOC.org or CLIR deficiency regarding Arabic language. Although Class Central does support Arabic term retrieval, the results are minimal, and no feedback is stored to those courses. This emphasizes the research goal of having a personalized experience that supports Arabic as well as other languages retrieval.

The MRSE model aims to overcome the deficiencies in the literature. Particularly, comparing to the proposed hybrid solutions [23], [27], [28], the MRSE supports the CLIR by collecting a diversity of recent Arabic and English courses from eight high-ranked MOOCs. In comparison, the previous studies retrieved only English courses in narrow domains from only two MOOCs, edX and Coursera. Although we enhanced CLIR regarding Arabic, the study did not fully overcome the Arabic recognition compering to English.

Then MRSE successfully recommends courses that fit users' queries. However, the recommendation functionality through clustering substitutes for the actual recommender system, which is established in all the proposed hybrid solutions. Moreover, the study also presents real-world users' experimentation to measure the approach's achievement, where other studies stated that the system needs to be used tested by real-world users. Table 2 presents a comparison of the MRSE model with other proposed hybrid solutions.

TABLE 2. Comparison of the MRSE model with other hybrid retrieving models.

MOOCs Hybrid Search Model	MOOCs variety	Results variety	Basis of recommendation	Real users' experimentation
MoocRec.com [23]	Only two MOOCs	Variety of English results	Content filtration and Matrix Factorization algorithm	X
MOOC Rec1 [28]	Only two MOOCs	English results in computer Science subject only	Content filtration of video styles and users' learning styles	X
MOOC Rec2 [27]	Only two MOOCs	English results in Humanities subject only	Content filtration of video styles and users' learning styles	X
MRSE	Eight MOOCs	Variety of subjects in English and Arabic	Clustering techniques of MOOCs data	√

Some lessons were learned. Throughout the implementation, we faced many complications. In order to activate the automation without going through multiple clusters, we had to mimic the production environment and configure each tool to activate automated focused-crawling and automated indexing. All of this was needed to offer a usable search platform for users.

The accuracy of the recommendations and some results in terms of courses is not what we expected. Hence, more alterations to improve accuracy are still needed.

The collected data still needs manual segmentation since the quality of segmentation is essential for search engines, as it might affect the accuracy of the text extension extraction and the recommendation accuracy, especially with Arabic syntax. With these changes, a full-function recommender system will be realized.

VI. CONCLUSION AND FUTURE WORK

For almost two decades, MOOCs have been evolving along with the educational process, offering learners opportunities to increase their knowledge and experiences. This has been notably noticed for the past few years, especially within the latest COVID-19 pandemic. These MOOCs' rapid increase hinders learners while searching and choosing proper courses, leaving people to get overloaded with too many choices over the Web and therefore get lost in hyperspace.

Many researchers have attempted to assist users' navigation for desired courses by offering personalized experiences through recommender systems for MOOCs. However, researchers are not able to access essential MOOCs data to generate users' preferences, provide recommendations, and perform actual personalization. Moreover, the individual efforts on MOOC platforms are not enough to overcome information overload and the "lost in hyperspace" situation.

Some studies have proposed unified retrieving models for MOOCs. These models differentiate between search engines and recommender systems or either combine both to emphasize each other's characteristics. However, such studies are still limited due to the challenge of accessing useful MOOCs data. Moreover, their orientation is on either English or Chinese courses in specific subjects from at most five MOOCs. This is a weakness in the diversity of results and in CLIR with respect to the Arabic language, where those MOOCs locator systems do not target Arabic speakers.

Since vertical search engines provide many benefits over regular search engines by offering more relevant information and greater precision as a result of a scope limitation, it is appropriate to propose a vertical search engine for MOOCs.

The proposed system provides users with courses from different MOOCs by integrating open source tools and technologies (Solr-Nutch). Initial results showed that the system can collect and present courses from several MOOCs as intended. Moreover, the system can retrieve courses in both the Arabic and English languages. In a further effort, the model was combined with clustering techniques (Carrot2 suggester component) to generate recommendations based on query terms to add more value and precision to the engine results. However, the applied techniques did not fully present high-quality results in Arabic compared to English. Hence, it is encouraged to continue the enhancement of the Arabic retrieval in this field.

This work contributed to the current MOOCs retrieving innovation in academia by presenting a unified MOOCs search model that serves Arabic and English users. The model can retrieve Arabic and English courses from multiple MOOCs using the latest tools and techniques in diverse subjects. Furthermore, the research shows how the model prepares recommendations out of no users' patterns. Currently, the beta version can collect users' data to obtain better recommendations and future personalization. This version will be the base for future work by integrating an actual recommender system as the second stage of the proposed model to provide personalization that includes Arabic orientation.

REFERENCES

[1] D. Jansen and K. Lizzie. (2018). *The 2018 Open Education Trend Report on MOOCs*. Amsterdam, The Netherlands. Accessed: Dec. 24, 2018. [Online]. Available: https://www.eadtu.eu%7Cwww.openuped.eu

[2] (2020). *EdX*. Accessed: Dec. 29, 2020. [Online]. Available: https://www.edx.org

[3] (2020). *Khan Academy*. Accessed: Dec. 29, 2020. [Online]. Available: https://www.khanacademy.org/

[4] (2020). *Udemy*. Accessed: Jan. 11, 2020. [Online]. Available: https://www.udemy.com

[5] E. Ossiannilsson, F. Altinay, and Z. Altinay, "MOOCs as change agents to boost innovation in higher education learning arenas," *Educ. Sci.*, vol. 6, no. 4, p. 25, Aug. 2016, doi: 10.3390/educsci6030025.

[6] D. Shah. (2013). *MOOCs in 2013: Breaking Down the Numbers, EdSurge News*. Accessed: Dec. 29, 2018. [Online]. Available: https://www.edsurge.com/news/2013-12-22-moocs-in-2013-breaking-down-the-numbers

[7] (2020). *Udacity*. Accessed: Dec. 29, 2020. [Online]. Available: https://mena.udacity.com

[8] (2020). *Coursera*. Accessed: Dec. 22, 2020. [Online]. Available: https://www.coursera.org

[9] (2020). *Edraak*. Accessed: Dec. 24, 2020. [Online]. Available: https://www.edraak.org/en/

[10] (2020). *Rwaq*. Accessed: Dec. 24, 2020. [Online]. Available: https://www.rwaq.org/

[11] R. S. Adham and K. O. Lundqvist, "MOOCS as a method of distance education in the Arab world—A review paper," *Eur. J. Open, Distance E-Learn.*, vol. 18, no. 1, pp. 123–138, Jul. 2015, doi: 10.1515/eurodl-2015-0009.

[12] D. Shah. (2019). *By the Numbers: MOOCs in 2019*. Class Central. Accessed: Nov. 23, 2020. [Online]. Available: https://www.classcentral.com/report/mooc-stats-2019/

[13] D. Shah. (Nov. 30, 2020). *By the Numbers: MOOCs in 2020*. The Report-Class Central. Accessed: Feb. 17, 2021. [Online]. Available: https://www.classcentral.com/report/mooc-stats-2020/

[14] A. Vasinis. (Apr. 19, 2019). *The Wide Wide Web and the Information Overload Conundrum*. Brainsfeed, Medium. Accessed: Aug. 26, 2020. [Online]. Available: https://medium.com/brainsfeed/the-wide-wide-web-and-the-information-overload-conundrum-2c37fd7728be

[15] H. Jain and Anika, "Applying data mining techniques for generating MOOCs recommendations on the basis of learners online activity," in *Proc. 6th IEEE Int. Conf. MOOCs Innov. Technol. Educ.*, Nov. 2018, pp. 6–13, doi: 10.1109/MITE.2018.8747056.

[16] M. Otter and H. Johnson, "Lost in hyperspace: Metrics and mental models," *Interacting Comput.*, vol. 13, no. 1, pp. 1–40, 2000, doi: 10.1016/S0953-5438(00)00030-8.

[17] T. Nelson, "Lost in hyperspace," *New Sci.*, vol. 191, no. 2561, p. 26, 2006, doi: 10.1002/elsc.200620112.

[18] A. Mitra and A. Awekar, "On low overlap among search results of academic search engines," in *Proc. 26th Int. Conf. World Wide Web Companion*, 2017, pp. 823–824, doi: 10.1145/3041021.3054265.

[19] D. W. Oard and A. R. Diekema, "Cross language information retrieval," *Annu. Rev. Inf. Sci. Technol.*, vol. 33, pp. 223-256, Dec. 1998, Accessed: Nov. 24, 2020. [Online]. Available: https://eric.ed.gov/?id=EJ610156

[20] B. Elayeb and I. Bounhas, "Arabic cross-language information retrieval: A review," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 15, no. 3, pp. 1–44, Mar. 2016, doi: 10.1145/2789210.

[21] S. Lee, R. Girish, and Y. U. Kim, "Courserush a MOOC search engine," vol. 91, pp. 399–404, Dec. 2017. [Online]. Available: http://sifaka.cs.uiuc.edu/~wang296/Course/IR_Fall/past-projects/3.pdf

[22] Q. Cheng and Y. Gao. *Courducate-An MOOC Search and Recommendation System*. Accessed: Dec. 24, 2018. [Online]. Available: http://www.cs.virginia.edu/~hw5x/Course/IR2015/docs/Projects/Samples/2.pdf

[23] P. Symeonidis and D. Malakoudis. (2016). *MoocRec: Massive Open Online Courses Recommender System*. Accessed: Dec. 24, 2018. [Online]. Available: http://ceur-ws.org/Vol-1688/paper-01.pdf

[24] F. Bousbahi and H. Chorfi, "MOOC-Rec: A case-based recommender system for MOOCs," in *Proc. World Conf. Technol., Innov. Entrepreneurship*, vol. 195, 2015, pp. 1813–1822, doi: 10.1016/j.sbspro.2015.06.395.

[25] B. An, T. Qu, and H. Qi, "Chinese MOOC search engine," *Commun. Comput. Inf. Sci.*, vol. 503, pp. 453–458, Dec. 2015, doi: 10.1007/978-3-662-46248-5_55.

[26] X. He, P. Liu, and W. Zhang, "Design and implementation of a unified mooc recommendation system for social work major: Experiences and lessons," in *Proc. 7 IEEE Int. Conf. Comput. Sci. Eng. (CSE)*, Jul. 2017, pp. 219–223, doi: 10.1109/CSE-EUC.2017.46.

[27] F. Fazuludeen, G. Vijayakumaran, Z. A. Mahroof, N. Kodagoda, and K. Suriyawansa, "MOOCRec 2 for humanities–learning style based MOOC recommender and search engine," in *Proc. Int. Conf. Adv. Comput. (ICAC)*, Dec. 2019, pp. 470–475, doi: 10.1109/ICAC49085.2019.9103397.

[28] 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 *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, Apr. 2019, pp. 1167–1172, doi: 10.1109/EDUCON.2019.8725079.

[29] K. Alzahrani and M. Meccawy, "A hybrid personalization model for searching multiple MOOCs," in *Proc. 12th Int. Conf. Lang., Educ., Hum. Innov. Int. Conf. Open Learn. Educ. Technol.*, 2019, pp. 42–52. [Online]. Available: https://icsai.org/Proc./1icolet-014-014/.

[30] W. B. Croft, D. Metzler, and T. Strohman, *Search Engines: Information Retrieval in Practice*. London, U.K.: Pearson, 2015.

[31] G. Grefenstett, *Cross-Language Information Retrieval*. Grenoble, France: Springer, 2012.

[32] P. Scháuble and P. Sheridan, "Cross-language infor-mation retrieval (CLIR): Track overview," in *Proc. 6th Text Retr. Conf.*, 1997, pp. 31–44. Accessed: Dec. 23, 2020. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.7262

[33] L. Ballesteros and W. B. Croft, "Phrasal translation and query expansion techniques for cross-language information retrieval," *ACM SIGIR Forum*, vol. 31, no. 1, pp. 84–91, 1997, doi: 10.1145/278459.258540.

[34] N. AbdulJaleel and L. S. Larkey, "Statistical transliteration for English-Arabic cross language information retrieval," in *Proc. 12th Int. Conf. Inf. Knowl. Manage.*, Nov. 2003, pp. 139–146, Accessed: Jan. 9, 2021. [Online]. Available: https://doi/abs/10.1145/956863.956890.

[35] K. Darwish, "Arabic information retrieval," *Found. Trends Inf. Retr.*, vol. 7, no. 4, pp. 239–342, 2014, doi: 10.1561/1500000031.

[36] M. Aljlayl, O. Frieder, and D. Grossman, "On bidirectional English-Arabic search," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 53, no. 13, pp. 1139–1151, Nov. 2002, doi: 10.1002/asi.10143.

[37] A. Bellaachia and G. Amor-Tijani, "Enhanced query expansion in English-Arabic CLIR," in *Proc. Int. Workshop Database Expert Syst. Appl.*, 2008, pp. 61–66, doi: 10.1109/DEXA.2008.52.

[38] J. Xu, A. Fraser, and R. Weischedel, "Empirical studies in strategies for Arabic retrieval," in *Proc. SIGIR Forum*, 2002, pp. 269–274, doi: 10.1145/564376.564424.

[39] M. Aljlayl and O. Frieder, "Effective Arabic-English cross-language information retrieval via machine-readable dictionaries and machine translation," in *Proc. 10th Int. Conf. Inf. Knowl. Manage.*, 2001, p. 295, doi: 10.1145/502585.502635.

[40] C. Carpineto, S. Osiáski, G. Romano, and D. Weiss, "A survey of web clustering engines," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–38, Jul. 2009, doi: 10.1145/1541880.1541884.
[41] M. Alam and K. Sadaf, "A review on clustering of web search result," *Adv. Intell. Syst. Comput.*, vol. 177, no. 2, pp. 153–159, 2013, doi: 10.1007/978-3-642-31552-7_17.
[42] (2019). *Carrot2*. Accessed: Nov. 24, 2020. [Online]. Available: https://doc.carrot2.org/
[43] K. Curran and J. Mc Glinchey, "Vertical search engines," *ITB J.*, vol. 8, no. 2, 2007, Art. no. 4, doi: 10.21427/D7MT83.
[44] Z. Guoyou, Y. Houlong, and Z. Guoqiang, "Vertical search engine: Its value analysis and development trend prospect," in *Proc. Int. Conf. Manage. E-Commerce E-Government*, Oct. 2014, pp. 179–182, doi: 10.1109/ICMeCG.2014.44.
[45] N. Hug, "Surprise: A Python library for recommender systems," *J. Open Source Softw.*, vol. 5, no. 52, p. 2174, Aug. 2020, doi: 10.21105/joss.02174.
[46] M. Quadrana, D. Jannach, and P. Cremonesi, "Tutorial: Sequence-aware recommender systems," in *Proc. World Wide Web Conf.*, May 2019, p. 1316, doi: 10.1145/3308560.3320091.
[47] A. Karatzoglou and B. Hidasi, "Deep learning for recommender systems," in *Proc. 11th ACM Conf. Recommender Syst.*, Aug. 2017, pp. 396–397, doi: 10.1145/3109859.3109933.
[48] C. A. Gomez-Uribe and N. Hunt, "The Netflix recommender system: Algorithms, business value, and innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, pp. 1–19, Jan. 2016, doi: 10.1145/2843948.
[49] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender System Handbook*, Boston, MA, USA: Springer, 2011, pp. 1–35.
[50] R. Burke, "Hybrid web recommender systems," in *The Adaptive Web Tech*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Germany: Springer, 2007, ch. 2, sec. 12, pp. 377–408. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-540-72079-9_12#citeas
[51] J. Henskens, "Recommender systems for E-commerce websites," M.S. thesis, Dept. Econ. Bus. Admin, Ghent Univ., Ghent, Belgium, 2016. [Online]. Available: https://libstore.ugent.be/fulltxt/RUG01/002/304/661/RUG01-002304661_2016_0001_AC.pdf
[52] A. Khaleghi, K. Mahmoudi, and S. Mozaffari, "Design of a vertical search engine for synchrotron data: A big data approach using Hadoop ecosystem," *Social Netw. Appl. Sci.*, vol. 1, no. 12, pp. 1–7, Dec. 2019, doi: 10.1007/s42452-019-1582-1.
[53] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2065–2073, Mar. 2014, doi: 10.1016/j.eswa.2013.09.005.
[54] M. Volkovs, G. Yu, and T. Poutanen, "DropoutNet: Addressing cold start in recommender systems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 4957–4966, Accessed: Nov. 24, 2020. [Online]. Available: https://github.com/layer6ai-labs/DropoutNet
[55] J. Chen and Y. Bao, "Information access across languages on the web: From search engines to digital libraries," *Proc. Amer. Soc. Inf. Sci. Technol.*, vol. 46, no. 1, pp. 1–14, 2009, doi: 10.1002/meet.2009.1450460278.
[56] M. Beseiso, A. R. Ahmad, and R. Ismail, "A survey of Arabic language support in semantic web," *Int. J. Comput. Appl.*, vol. 9, no. 1, pp. 35–40, Nov. 2010, doi: 10.5120/1348-1818.
[57] A. S. Sunar and S. White. (Apr. 2019). *Investigating Personalised Applications in MOOCs: The Challenge of Achieving Transparency*. Accessed: Aug. 18, 2020. [Online]. Available: https://www.researchgate.net/publication/332979645_Investigat-ing_Personalised_Applications _in_MOOCs_the_challenge_of_achieving_transparency
[58] A. S. Sunar, N. A. Abdullah, S. White, and H. Davis, "Personalisation in MOOCs: A critical literature review," *Commun. Comput. Inf. Sci.*, vol. 583, pp. 152–168, 2016, doi: 10.1007/978-3-319-29585-5_9.
[59] N. El Mawas, J.-M. Gilliot, S. Garlatti, R. Euler, and S. Pascual, "Towards personalized content in massive open online courses," in *Proc. 10th Int. Conf. Comput. Supported Educ.*, 2018, pp. 1–9, doi: 10.5220/0006816703310339.
[60] (2020). *Coursematch*. Accessed: Mar. 14, 2021. [Online]. Available: https://www.coursematch.io/
[61] Coursera. (2020). *Campus CourseMatch*. Accessed: Apr. 19, 2020. [Online]. Available: https://www.coursera.org/campus-coursematch
[62] S. Downes, *Connectivism and Connective Knowledge-Essays on Meaning and Learning Networks*. Ottawa, ON, Canada: National Research Council Canada, 2012.
[63] Z. Gulzar, A. A. Leema, and G. Deepak, "PCRS: Personalized course recommender system based on hybrid approach," *Proc. Comput. Sci.*, vol. 125, pp. 518–524, Jan. 2018, doi: 10.1016/j.procs.2017.12.067.
[64] H. Zhang, T. Huang, Z. Lv, S. Liu, and Z. Zhou, "MCRS: A course recommendation system for MOOCs," *Multimedia Tools Appl.*, vol. 77, no. 6, pp. 7051–7069, Mar. 2018, doi: 10.1007/s11042-017-4620-2.
[65] Y. Pang, Y. Jin, Y. Zhang, and T. Zhu, "Collaborative filtering recommendation for MOOC application," *Comput. Appl. Eng. Educ.*, vol. 25, no. 1, pp. 120–128, Jan. 2017, doi: 10.1002/cae.21785.
[66] Y. Li and H. Li, "MOOC-FRS: A new fusion recommender system for MOOCs," in *Proc. IEEE 2nd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Mar. 2017, pp. 1481–1488, doi: 10.1109/IAEAC.2017.8054260.
[67] H. Yu, C. Miao, C. Leung, and T. J. White, "Towards AI-powered personalization in MOOC learning," *npj Sci. Learn.*, vol. 2, no. 1, pp. 1–5, Dec. 2017, doi: 10.1038/s41539-017-0016-3.
[68] V. Garg and R. Tiwari, "Hybrid massive open online course (MOOC) recommendation system using machine learning," in *Proc. Int. Conf. Recent Trends Eng., Sci. Technol.*, Apr. 2018, pp. 1–5, doi: 10.1049/cp.2016.1479.
[69] (2020). *Class Central*. Accessed: Dec. 29, 2020. [Online]. Available: https://www.classcentral.com/
[70] (2020). *MOOC*. Accessed: Dec. 29, 2020. [Online]. Available: https://www.mooc.org/
[71] (2020). *My Mooc*. Accessed: Dec. 29, 2020. [Online]. Available: https://www.my-mooc.com/en/
[72] D. Bridge, M. H. Göker, L. McGINTY, and B. Smyth, "Case-based recommender systems," *Knowl. Eng. Rev.*, vol. 20, no. 3, pp. 315–320, Sep. 2005, doi: 10.1017/S0269888906000567.
[73] (2019). *Apache Nutch*. Accessed: Dec. 28, 2019. [Online]. Available: http://nutch.apache.org/
[74] (2019). *Apache Solr*. Accessed: Dec. 29, 2019. [Online]. Available: http://lucene.apache.org/solr/
[75] (2019). *Suggester Apache Solr Reference Guide 8.0*. Accessed: Dec. 18, 2020. [Online]. Available: https://lucene.apache.org/solr/guide/8_0/suggester.html
[76] (2020). *Harvard Online Courses*. Accessed: Dec. 24, 2019. [Online]. Available: https://online-learning.harvard.edu/
[77] S. Osinski and D. Weiss, "Lingo: Search results clustering algorithm based on singular value decomposition," in *Proc. Intell. Inf. Process. Web Mining*, Jun. 2004, pp. 359–368.
[78] STC. *STC Clustering Algorithm (Carrot2 V3.17.0-SNAPSHOT API Documentation (JavaDoc)*. Accessed: Dec. 28, 2020. [Online]. Available: https://get.carrot2.org/head/javadoc/org/carrot2/clustering/stc/STCClusteringAlgorithm.html
[79] E. Erickson. (Nov. 21, 2019). *Improving the Search Experience With Suggester*. Lucidworks. Accessed: Dec. 20, 2020. [Online]. Available: https://lucidworks.com/post/solr-suggester/

KHADIJAH M. ALZAHRANI received the B.S. degree in information systems from the Faculty of Science, Al-Bahah University, Saudi Arabia, and the M.S. degree in information systems from the Faculty of Computing and Information Technology, King Abdulaziz University (KAU), Saudi Arabia. She keens to pursue her research in blended education technologies to support the education for everyone movement.

MARAM MECCAWY received the Ph.D. degree from the University of Nottingham, U.K. She is currently an Associate Professor with the Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University (KAU), Saudi Arabia. After her graduation, she spent a few years working in the industry before joining KAU as an Assistant Professor, in 2014. She is currently a Visiting Scholar with the University of Durham, U.K., working on research related to the use of XR technologies and application in education. Her research interests include adaptive educational hypermedia, user modeling, LMS, MOOCs, and teaching programming to children.