Framework for a Knowledge-Based Course Recommender System Focused on IT Career Needs

Hien Pham Thi Xuan[1,2], Le Nguyen Hoai Nam[1,2][a] and Cuong Pham-Nguyen[1,2,*][b]
[1]*Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam*
[2]*Vietnam National University, Ho Chi Minh City, Vietnam*

Keywords: Course Recommendation Method, Data Collection Process, Knowledge Base.

Abstract: This paper presents an approach for a knowledge-based recommender system that provides relevant courses based on learners' profiles, requirements, and career needs. The framework integrates an automatic data collection process, ensuring that the knowledge base reflects the latest job market and course information. The recommendation method relies on a set of rules that combine various matching techniques, incorporating user requirements, skill and knowledge gaps, contextual information, and the course weight indicating its relevance to the career or market. An experiment was conducted to measure the satisfaction of the approach through a survey of users who used the system. The results reveal that the approach is deemed acceptable. This framework contributes to ongoing discussions surrounding the application of technology in building recommender systems for education.

1 INTRODUCTION

Nowadays, there has been a significant increase in job opportunities within the IT industry, attracting numerous students and professionals. A survey conducted in the US across 43 schools and institutes, involving 32,000 students, revealed that only 34% of students believe they graduate with the skills and knowledge that meet market demands (Gallup, 2017). It also highlighted a disparity in perspectives: while 96% of schools believe that their training aligns with career needs, only 11% of businesses agree with this opinion. Efforts have been made to bridge the gap between training, the job market, and business needs through activities such as job fairs, business visits, internships, etc. Many enterprises also offer refresher courses to new graduates to familiarize them with the knowledge and skills specific to their business. Another survey conducted on social forums, including Quora, Stack Overflow, and Stack Exchange, consisting of 2,860 questions related to IT careers showed that the number of questions seeking guidance on learning paths is a majority (83.4%) (Nguyen et al, 2022). This indicates that students often require additional time for retraining or supplementary courses to become employable, and a high demand for professionals in acquiring skills that meet the new occupational requirements and facilitate career transitions.

Two major challenges emerge: Firstly, the gap between schools and businesses, where recruitment needs evolve and change rapidly. School's curricula have not kept pace with market demands. Schools may lack complete information about market requirements to make necessary adjustments to their curricula. Secondly, the overwhelming information overload in the vast online course landscape (O'Mahony and Smyth, 2007). There is a wide range of courses available on various e-learning platforms and MOOCs. For instance, Edumall (https://edumall.vn/) offers more than 2,000 courses, Udemy (https://www.udemy.com/) provides over 55,000 courses, edX (https://www.edx.org/) has over 2,900 courses, and Coursera (https://www.coursera.org/) offers over 1,000 courses. Each skill or learning subject can have numerous providers, with hundreds of courses dedicated to teaching it. Platforms such as Edumall, Funix, Coursera, Nordic, VTC Academy,

[a] https://orcid.org/0000-0001-9675-2191
[b] https://orcid.org/0000-0002-7057-753X
[*]Corresponding author

Hien, P., Nam, L. and Pham-Nguyen, C.
Framework for a Knowledge-Based Course Recommender System Focused on IT Career Needs.
DOI: 10.5220/0012892500003838

In *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2024) - Volume 2: KEOD*, pages 15-26
ISBN: 978-989-758-716-0; ISSN: 2184-3228

Athena, Unica, Stanford all offer Python courses covering various topics like "Selenium Webdriver with Python", "Python and computer vision", “Python Excel for professionals”, “Learning about Python frameworks with Selenium 3. x”. These courses range from basic to advanced levels. Consequently, the process of finding relevant courses from multiple websites is time-consuming and poses a challenge of making the right choice among the numerous options available that align with individual needs (Bhumichitr et al., 2017; Huang et al., 2017). Two primary difficulties are encountered: the overwhelming number of available courses and a lack of knowledge about which courses to take and in what order (O’Mahony and Smyth, 2007). With the motivation to support learners in overcoming these challenges, the main contributions of this paper are:

- A knowledge-based recommender framework incorporates a recommendation method that offers relevant courses for learning paths based on learner requirements, market needs and profiles. It uses a set of rules and combines various matching techniques. These techniques consider learner requirements, skill gaps, contextual information, and the course weight, which represents the course's significance in the learner's career.
- A data collection process ensures that the knowledge base remains up to date with the latest information on the job market and course information. This is achieved by automatically extracting data from recruitment and learning websites to build and update the knowledge base.

The paper is organized as follows: Section 2 summarizes a background, and describes related works and the motivation of our approach. Section 3 provides the proposed framework, comprising a general architecture, a knowledge model, a data collection process, and a recommendation method. Section 4 focuses on the implementation of the framework. Section 5 presents the experiments and evaluation and finally, Section 6 highlights the main results and provides some perspectives.

2 THEORICAL BACKGROUND AND RELATED WORK

2.1 Recommender Systems

Recommender systems often rely on users' past preferences collected through surveys to provide future recommendations. Specifically, content-based recommendations recommend users on items (i.e. products or services) similar to those they liked in the past (Javed at al., 2021). However, this approach is often criticized for its lack of diversity in the recommendation set, leading to potential user boredom. Another approach is collaborative filtering, where systems rely on the preferences of multiple related users to advise a particular user. Two prominent models in this approach are latent factor models and neighbor models (Nam, 2023). Latent factor models are trained on the preferences of all users to learn hidden features representing users and items. This new representation facilitates predicting whether to recommend an item to a user through matching mechanisms. On the contrary, the neighbor model identifies a set of users with similar preferences to the target user using similarity metrics in preference, referred to as neighbors. The preferences of these neighbors help inferring the preferences of the target users, thereby recommending suitable items.

However, gathering user preferences with sufficient quality and quantity can be challenging in certain domains where items are infrequently purchased, or users make one-time purchases and use items over an extended period (Nam, 2021). Users may rarely disclose their preferences on the system after using items, and some deliberately provide misleading preferences in text reviews (Charu, 2016). In such contexts, knowledge-based recommendation systems (KBRSs) emerge as the most suitable choice. In these systems, users submit their requirements, which the system integrates with user information, product descriptions, and an underlying knowledge model to generate a list of relevant items (Guo et al., 2020). Based on the arguments above, we choose a knowledge-based approach for our course recommender system. Some main characteristics of a KBRS are as follows:

- *C1 - Domain Knowledge Integration*. KBRS leverages detailed domain knowledge from experts or collecting from online resources, represented in structured databases or ontologies, and relies on a knowledge base with rules, constraints, and heuristics.
- *C2 - Explicit User Requirements.* Users provide explicit inputs about their preferences and requirements through questionnaires, forms, or direct interaction via system interface, including specific features, attributes, or constraints.
- *C3 - Rule-Based.* Recommendations are generated using rules or an inference engine that applies logical reasoning to the knowledge base, such as if-then statements or decision trees.

- *C4 - Conversational Interface*. KBRS includes an interactive interface that allows users to iteratively refine their preferences and adjust inputs based on system explanations.
- *C5 – Content Update*. KBRS can be adapted to different contexts by regularly updating the knowledge base and rules.
- *C6 - Handling Cold Start Problem*. The system effectively addresses the cold start problem by recommending new users and items based on explicit attributes, rules, and domain knowledge.

2.2 Knowledge-Based Recommender Systems for Education

For an effective course recommendation system tailored to learners' vocational needs and businesses' demands, two criteria must be met: alignment with learners' career direction and learning journey, and effective matching of professional skills sought by businesses with skills acquired from courses.

Several studies (Majidi and Newfoundland, 2018; Obeid et al., 2018; Ilkou et al., 2021; Agarwal et al., 2022; Tarus et al., 2017) have developed knowledge-based course recommendation systems in line with our focus. Majidi and Newfoundland (2018) proposed a system that combines association rule mining and genetic algorithms to recommend courses that cover required skills set for a career path of a specific job position using data from sources such as educational websites and MOOC platforms. Obeid et al. (2018) proposed an ontology-based recommender system enhanced with machine learning techniques to guide students in higher education. This system assesses students' vocational strengths, weaknesses, interests, and capabilities to recommend the appropriate major and university. Ilkou et al. (2021) introduced EduCOR ontology for educational and career-oriented ontology representing online learning resources for personalized learning systems. Agarwal et al. (2022) introduced a system combining collaborative filtering and rule-based recommendation, integrating learning styles for personalized recommendations in MOOCs. Tarus et al. (2017) developed a hybrid knowledge-based recommender system based on the learner and resource ontologies, and sequential pattern mining algorithm for recommendation of e-learning resources to learners. The ontologies are used to model and represent the domain knowledge whereas the algorithm discovers the learners' sequential learning patterns. Subramanian and Ramachandran (2019) proposed Student Career Guidance (SCG) system that gathers learners' data from an input questionnaire including school results, students' school/home activities and academic interests. Based on this information, the system employs if then else rules to infer an appropriate study program for learners.

To provide tailored course suggestions that meet both learners' and business needs, it's essential to align the required skills in job postings, student profiles, and course outcomes. Mochol et al. (2015) developed a human resources ontology and used semantic matching techniques to improve online recruitment processes. Other studies (Paudel and Shakya, 2017; Straccia et al., 2009) employed skills ontology to model job seeker profiles and recruitment advertisements, enabling semantic matching for candidate suggestions. Authors often applied deductive matchmaking based on description logic to rank job seeker profiles according to various criteria such as work experience and competency level. Straccia et al. (2009) used ontology to rank profiles by translating logical queries into SQL query language. Corde et al. (2016) used bird mating optimization to match job seeker profiles with business job postings. Montuschi et al. (2015) matched student profiles with business recruitment needs, utilizing ontology to represent profiles, recruitment information, and course learning outcomes. Lexical matching of student skills with job requirements was then performed based on these representations.

Table 1: Summary of the KBRS for education approaches.

Approaches	C1	C2	C3	C4	C5	C6
Majidi and Newfoundland (2018)	+				+	+
Obeid C. et al. (2018)	+	+				+
Subramanian and Ramachandran (2019)	+	+	+			
Agarwal et al. (2022)	+		+	+		
Tarus, J. et al. (2017)	+	+		+		+
Our approach	+	+	+	+	+	+

Table 1 compares different approaches to KBRS for education based on six characteristics outlined in Section 2.1. In general, most approaches do not fully address these characteristics. This observation also confirms that these six characteristics are frequently used to identify a KBRS for education. All approaches focus on a knowledge base that represents learner profiles, learning domain and resources using ontologies. Some explicitly address the cold start problem by considering the learner's profile and asking learners to register it in the system to identify their interests and preferences. However, most studies lack real-time data updates and coherent integration of semantic and lexical elements. Additionally, reasoning capabilities using rules or constraints are rarely proposed in current studies to provide recommendations for learners.

To address this gap, our approach proposes a framework that offers courses or learning paths based on learners' skills and interests, focusing on six key characteristics. This system implements an ontology-based knowledge model for representing IT jobs, involving the extraction and analysis of online job postings and courses. The system constructs a structured knowledge framework for occupational requirements and course information. In our previous studies, a context-aware knowledge (CAK) model was developed to build the knowledge base for smart service systems (Le Dinh et al., 2022; Nguyen et al., 2022). This study extends and uses that knowledge base to build a KBRS for education, focusing primarily on an automated data collection process to ensure the knowledge base is always up to date, and a recommendation method based predominantly on a set of rules and various matching techniques.

3 FRAMEWORK FOR EDUCATIONAL KNOWLEDGE-BASED RECOMMENDER SYSTEM

This section outlines the principles of the proposed framework for a KBRS for education. Firstly, it encompasses the overall architecture, elucidating how the system components are organized and elaborated. The knowledge base model subsequently depicts the knowledge components and their relationships. Next, the primary data collection process is detailed, illustrating how it constructs and updates the knowledge base. Finally, the recommendation method relies on the knowledge base and rules, incorporating various matching techniques to provide course recommendations.

3.1 General Architecture

The general architecture of the framework is depicted in Figure 1. It comprises the following components:

- User Interface: allows learners to use the web interface for account registration, login, course consultation, view purchase history, and system evaluation.
- Data collection: consists of a process that automatically gathers data from online recruitment and educational platforms to build and update the knowledge base.
- Recommendation: handles requests from learners, applying course recommendation methods and returning relevant courses for learning paths.

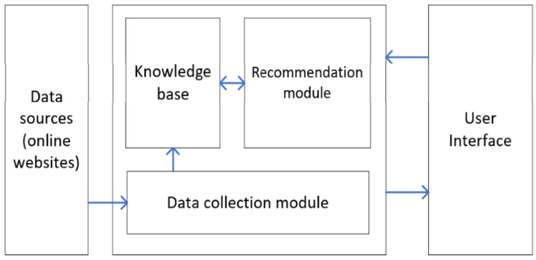

Figure 1: General architecture.

3.2 Knowledge Base Model

The incorporation of ontology to model the knowledge base is a crucial component of the system as it embodies the shared common understanding of the domain of interest. This component enables the system to interpret and enhance the integration of various online learning resources (Younten and Kristina, 2021). This study refines our prior knowledge model outlined in (Nguyen et al., 2022; Thi et al., 2020) for an educational recommender system. The knowledge model comprises three ontologies: the occupation ontology, the course ontology, and the learner ontology. Figure 2 illustrates the comprehensive knowledge model and the interconnections among these three models.

3.2.1 Occupation Model

In the field of occupation ontology, it was observed that the JobPosting Ontology (Thi et al., 2020) is suitable for reuse in this study. This ontology provides definitions tailored to the IT industry's needs, and the available dataset aligns well with job postings, assisting learners in understanding the skills and qualifications companies require for various positions. However, job postings for different roles can have varied requirements from each employer. While some skills are specific to certain positions, there are also additional skills that may be required. The JobPosting Ontology does not allow for specifying the relative importance of different skills within a job requirement. We reused its definitions to

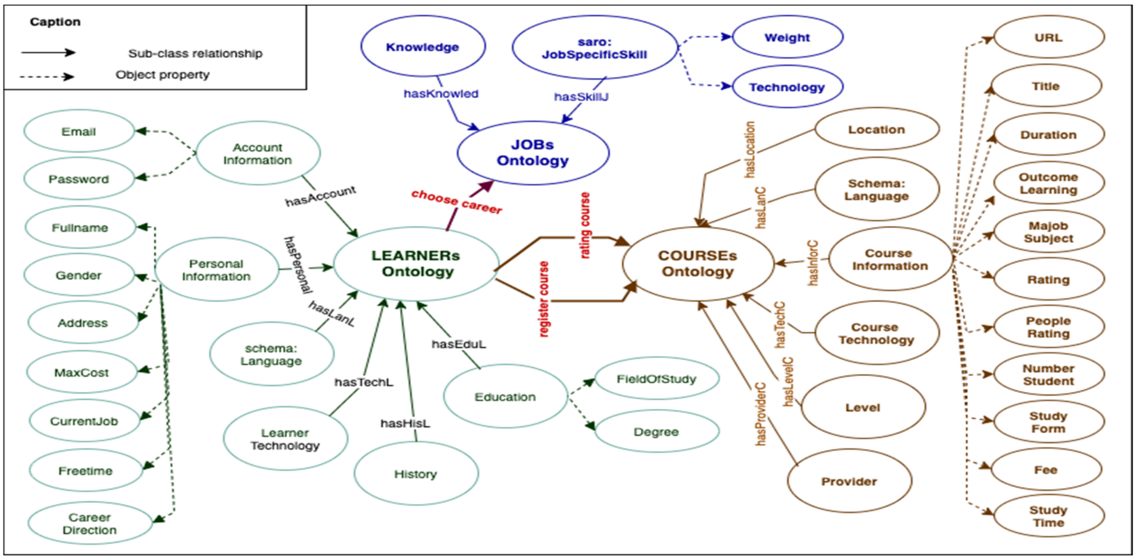

Figure 2: The high level of the knowledge base model.

represent entities within the same knowledge domain, specifically employing the TechnologySkill and KnowledgeSkill classes of the JobSpecificSkill class in the model to propose our occupation ontology.

Our occupation model presents concepts and relationships pertinent to the needs of the IT industry. The concept “JobSpecificSkill” describes job-related skills, comprises two sub-classes:

- The class “Knowledge” manages the knowledge requirements of an occupation, encompassing aspects like “deep understanding of Android SDK/Kotlin for Backend/willing to study Kotlin”.
- The class “Skill” denotes the necessity to be proficient in specific IT technologies, with a weight assigned to describe the importance of these technologies in the context of the related occupation. Technologies may include software, libraries, programming languages, databases, AI, Python, Google Colab, ETL, BI, C++, and others.

3.2.2 Course Model

This model identifies the most influential factors that impact learners when making decisions about course selection. These factors then become the primary classes of the model. Each class in the course profile corresponds to an equivalent class in the learner profile (Younten and Kristina, 2021). In this work, the course model encompasses concepts describing learning courses and their relationships.

- The class "Course Information" incorporates attributes such as title, URL, duration, fee, learning outcomes, major subject, rating, people rating, number of students participating in the course, study form, and study time.
- The class "Technology" encompasses technological skills required for the related course.
- The class "Level" characterizes the course level, such as Beginner, Elementary, Intermediate, Proficient. Additionally, the class "Provider" represents course providers such as Edumall, Coursera, Edx, etc.

3.2.3 Learner Model

This model offers information about learners in the field of IT, encompassing conditions set by learners. The system uses this information to generate personalized course suggestions tailored to the learner's profile. The learner model consists of the following classes:

- The class "Account Information" delineates the learner's account details, including email and password.
- The class "Personal Information" provides information about the learner's personal details, such as full name, gender, address, current job, available time, maximum cost, and career direction.
- The class “Education” handles information about the learner's field of study and current degree.
- The class “Learner Technology” encompasses the current learner's technology skills.
- The class “History” describes the learner's order history and course completion status.

3.2.4 Semantic Rules for Recommendations

One of the key components of the KBRS is to use constraints to generate personalized recommendations for learners. This approach specifically identifies and applies semantic rules to refine and generate course recommendations based on specific conditions. To achieve this, 12 rules were built that combine attributes such as languages, maximum cost, learning period, distance, course type, study format, available time, and current occupation. For instance, if a learner is seeking offline courses and is currently employed, Rule 1 is applied to suggest offline courses scheduled after office hours and located near the learner's address, with a preference for part-time courses. Conversely, if a learner is seeking employment, Rule 2 is applied, offering a broader range of options including full-time and part-time courses.

Rule 1: languages = **Vietnamese**; free_time; maximum_cost; type_of_couses; current_job = **Employed**; address; free_time ⇒ type_of_courses = **Offline**; study_format = **Part-time**; study_time = 18:00 ‐ 23:59
Rule 2: languages = **Vietnamese**; free_time; maximum_cost; type_of_couses; current_job = **Not Employed**; address, free_time ⇒ type_of_couses = Offline; study_format = (Part-time or Full-time)

3.3 Data Collection Process and Datasets

The purpose of data collection is to ensure that the knowledge base remains up-to-date and aligns with the current market requirements. We use Apache Airflow to automate the process and schedule DAG activations[3]. The process is described in Figure 3 encompasses four main tasks as follows:

- *Raw Data Retrieval.* Initially, raw data items are extracted from various online educational websites using Scrapy[4] and BeautifulSoup[5] libraries. This task can be configurable to be scheduled or executed manually.
- *Duplicate or Missing Data Removal.* Potential duplicate data items are identified and eliminated based on criteria built by combining ontological attributes, such as provider, instructor, course name, and taught skills for course data, as well as description, occupation name, required skills for occupation data. Additionally, items with missing mandatory attributes, such as courses without content or jobs without submission date are removed. If an item represents unique value in the dataset, an item imputation process is applied to fill in its missing data.
- *Skills and Knowledge Identification.* Potential skills and knowledge of each item are identified based on two dictionaries: job position and technology skills. To achieve this, we apply the Stanza library[6] to measure the similarity between the potential skills and the terms in the dictionaries using Bert Embedding. Then we add these skills if they are new, or merge them if they are not by increasing their weight.
- *Data Consolidation.* Finally, to synthesize skills and knowledge of collected items, similar items are identified and classified into the same cluster using the Kmeans unsupervised learning technique. In each cluster, technology skills and knowledge are aggregated using the GPT-4 model, incorporating Insight prompt templates[7]. Each cluster represents a data object or instance of the knowledge base.

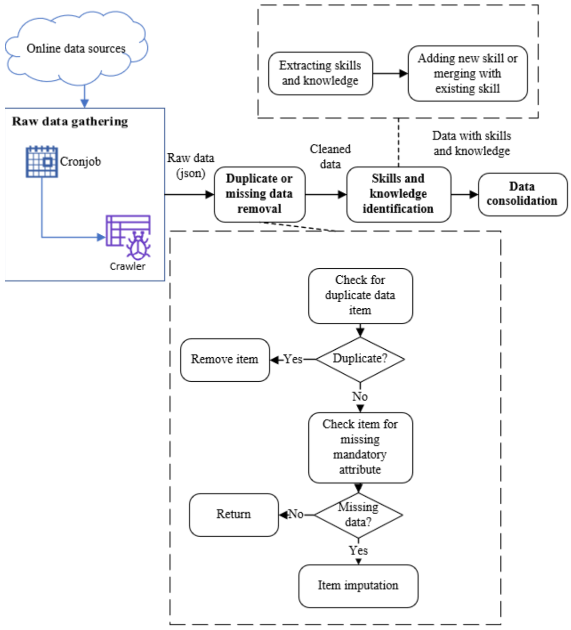

Figure 3: Automated data collection process.

[3] https://airflow.apache.org/docs/
[4] https://docs.scrapy.org/en/latest/
[5] https://beautiful-soup-4.readthedocs.io/en/latest/
[6] https://stanfordnlp.github.io/stanza/
[7] https://www.promptingguide.ai/models/gpt-4

3.3.1 Datasets

To initiate the datasets for implementation, testing and evaluation purposes, we used the data collection process to build occupation and course datasets. The occupation dataset was compiled from 350 job postings gathered from 4 websites: VietnamWorks, ITViec, TopITWorks and Indeed VN. Subsequently, these job postings were consolidated into 80 occupations, categorized into 9 groups such as Software and Application Development, System Administration and Networking, Information Security, Data Analysis and Data Science, etc. The course dataset was sourced from diverse platforms including CodeCademy, Udemy, Edx, Datacamp, FPT Software Academy, and Unica, resulting in the acquisition of 600 courses.

3.4 Knowledge-Based Recommendation Method

The recommendation method is fully illustrated in Figure 4, comprising five primary steps. Each step is described in detail in the following subsections.

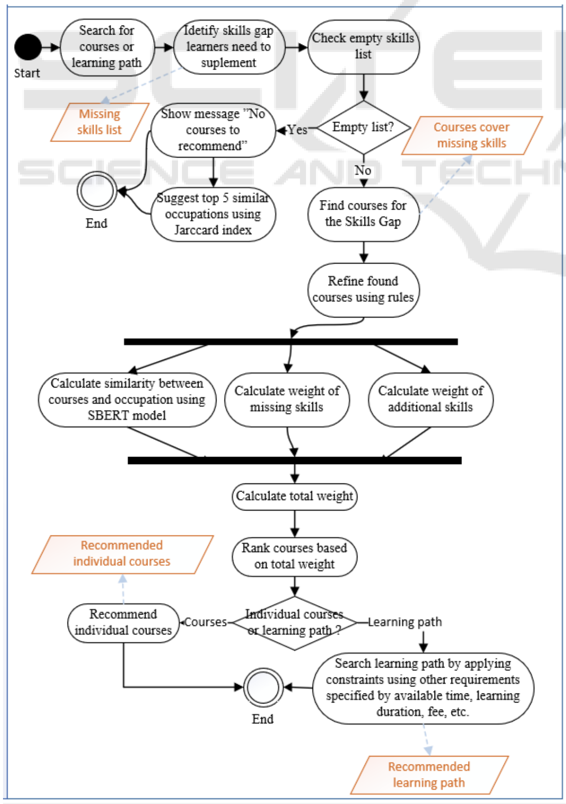

Figure 4: Recommendation process.

3.4.1 Identify Skills Gap

This step consists of matching the skills gap of the learner's profile with the skills required for the chosen occupation. The skills gap are computed by the equation (1) as follows:

$$\text{skillsGap}(l,o) = \text{skills}(o) - \text{skills}(l) \qquad (1)$$

Where l and o are respectively the learner and occupation. The formula *skillsGap (l,o)* calculates the set containing lacking skills of the user l with the occupation o. For example, an user l wants to explore 'Data Analytics'. This occupation o requires proficiency in skills O = *{SQL, Python, R, Machine Learning, Apache Hadoop, Apache Spark}*. The learner l currently possesses skills L = {Microsoft Excel}. The system identifies the skills gap by *skillsGap(l, o)* = $O \setminus L$ = *{SQL, Python, R, Machine Learning, Apache Hadoop, Apache Spark}*.

3.4.2 Find Courses Suitable for the Missing Skills Set

The system searches for courses that match the missing skills. This matching method is carried out using lexical matching technique proposed in (Gupta and Garg, 2014) to identify the courses that cover those skills. The technique focuses solely on the skills set of each course in mapping with the missing skills. This mapping results in a set of courses that match one or more of the learner's missing skills. For each course it gives the set of missing skills it covers (a course may offer additional skills that the learner already has or does not require). All courses that cover one or more of the missing skills are identified.

3.4.3 Filter Courses Using Rules

This step mainly applies the constraint-based method that uses rules to refine the selection of courses (Charu, 2016). As mentioned in section 4.2.4, these rules are built based on the combination of the learner profiles, learning requirements input from the user interface such as desired occupation, learning mode (online/offline), learning period, and the type of recommendation (top high ranking individual courses/bundled learning paths). The system applies these rules to refine the courses obtained from the previous step.

3.4.4 Calculate Weights and Rank Courses

if the result still includes a large number of courses, refinement is necessary to generate more concise and accurate recommendations. We propose the formula (3) to calculate weights for ranking courses based on the combination of three features:

- Weight of similarity between the courses' learning outcomes and the required knowledge of the selected occupation. Courses with high similarity to the required knowledge for occupation are prioritized. To achieve this, we convert the courses' learning outcomes and the required knowledge for occupation from text into vectors and apply the SBERT model (Reimers and Gurevych, 2019) to calculate the similarity for each pair (course, occupation) using cosine similarity. The equation is described in (2), where R_c and R_o denote the course learning outcomes and required knowledge for occupation, respectively.

$$\text{sim\_course}(c,o) = \cos(R_c, R_o) = \frac{\sum_{i \in I_c \cap I_o} r_{i,c} \times r_{i,o}}{\sqrt{\sum_{i \in I_c} r_{i,c}^2} \times \sqrt{\sum_{i \in I_o} r_{i,o}^2}} \quad (2)$$

- Weight of the missing skills for the selected occupation. In the occupation model, each skill has a weight representing its importance of the occupation. A course is preferable if it offers multiple skills that learners currently need to supplement, with each skill given a higher priority weight captured in the occupation model. For example, consider two courses A and B, which provide the missing skills *{R:4, Python:3, C#:2}* and *{R: 4, C:1, C#:2}* for an occupation, respectively. The weight of the skills provided by two courses is calculated as 4 + 3 + 2 = 9 for course A and 4 + 1 + 2 = 7 for course B. Thus, course A would be more suitable to recommend to learners than course B. This calculation is based on the equation (3) where $skill(a)_c$ represents the missing skill a of course c, and $weight(a)_o$ represents its weight in occupation o.

$$sumWeightSkills(c,o) = skill(a)_c \times weight(a)_o + \cdots + skill(n)_c \times weight(n)_o \quad (3)$$

- In addition to the skills required by an occupation, a course may also provide additional skills to support that occupation but are not listed in the occupation's skills list. This feature is also considered in the course selection process. For example, Course A provides 3 skills required by an occupation, namely *{R, C, C#}*, and includes 2 additional skills, *{Python, SQL}*. Course B provides the same three required skills, *{R, C, C#}*, and one additional skill, *{Python}*. The course A would be more suitable to recommend to learners as it not only covers the required professional skills but also offers more additional skills for learning.

Finally, the total weight of a course for an occupation, denoted as *weight(c,o)*, is computed using the formula (4) and two parameters, α and β. The parameter α represents the influence of *sumWeightSkill(c,o)* on the final result (with a default value of α = 0.4), while the parameter β indicates the importance of *sim_cos(c,o)*, knowledge similarity on the final result (with a default value of β = 0.4). The optimization of these parameters can be achieved through bridge regression, which is calculated based on learners' feedback.

$$weight(c,o) = \alpha \times sumWeightSkills(c,o) + \beta \times sim\_\text{course}(c,o) + (1-\alpha-\beta) \times \text{additional}Skills(c, o) \quad (4)$$

3.4.5 Select Suitable Courses Based on the Calculated Weights

Depending on the type of recommendation selected by learners, the system returns either (1) the top high ranking individual courses or (2) the relevant learning path. For the learning path, certain rules are applied to recommend courses given the same weight and from the same provider as well as from different providers. In this case, other attributes such as course rating, number of participants, learner's available time, fee, duration, etc. are considered for ranking.

If there are no suitable courses or no courses containing required skills, the system suggests the top 5 occupations with similar skills set to the selected occupation by applying the Jaccard similarity index proposed by Gupta and Garg (2014).

4 IMPLEMENTATION

Developing a web application to facilitate users in using the recommendation system more conveniently is a key objective. Additionally, the creation of a user interface enables users to register accounts and update necessary personal information, thereby enhancing the effectiveness of the system. The system leverages the ReactJS framework for the front-end, the ExpressJS framework for the back-end, and Python libraries for recommender systems. JSX is employed to embed HTML code into JavaScript, using Props to delegate tasks to different components and ensuring data immutability through the use of Immutable. The Docker platform is used for building, deploying, and running the application more efficiently via containerization. MySQL is chosen for database management. The source code is available on GitHub at https://github.com/ptxhien/KRSHien. The application is hosted at https://services.fit.hcmus.edu.vn:250/#/.

The learners can create their profile in the system including current skills, occupation, major, language, desired learning time, budget, etc. The new skills are updated when the learners complete a course, reflecting the system's capability to the knowledge base.

Figure 5 depicts the main page of the system, allowing learners to submit a requirement and search for recommendations. At the top of the page, the learners specify the desired job position for their future career and the type of recommendation, indicating whether the results should focus on a learning path or individual courses. Additional inputs including specifying the study mode (online, offline or both) and learning period, can also be selected.

The main page presents a list of recommended courses based on the specified requirements. It also displays the skills required for the chosen occupation on the top-right, with the skills to be learned for that occupation on the bottom-right. The importance level of each skill is indicated by the size of the skills, where skills of the same size are considered equally important. Detailed information about a course is showed when learners click on a course and view it. The important information includes the skills acquired upon completing the course. Additionally, learners can also see courses that are currently studying and add courses to their cart.

5 EXPERIMENTS AND EVALUATION

In this section, experiments are conducted to assess our system based on two aspects: user satisfaction and the accuracy of the data extraction process.

5.1 User Satisfaction

The effectiveness is assessed by considering the relevance of responses, which reflects how well users discover accurate answers while using the system, alongside the system's response speed. To achieve this, a survey involving 45 participants was conducted, with 67% being computer science students and 33% professionals who come from different schools and working environments. This diverse participant group helps to cover a wide range of job positions and avoids bias.

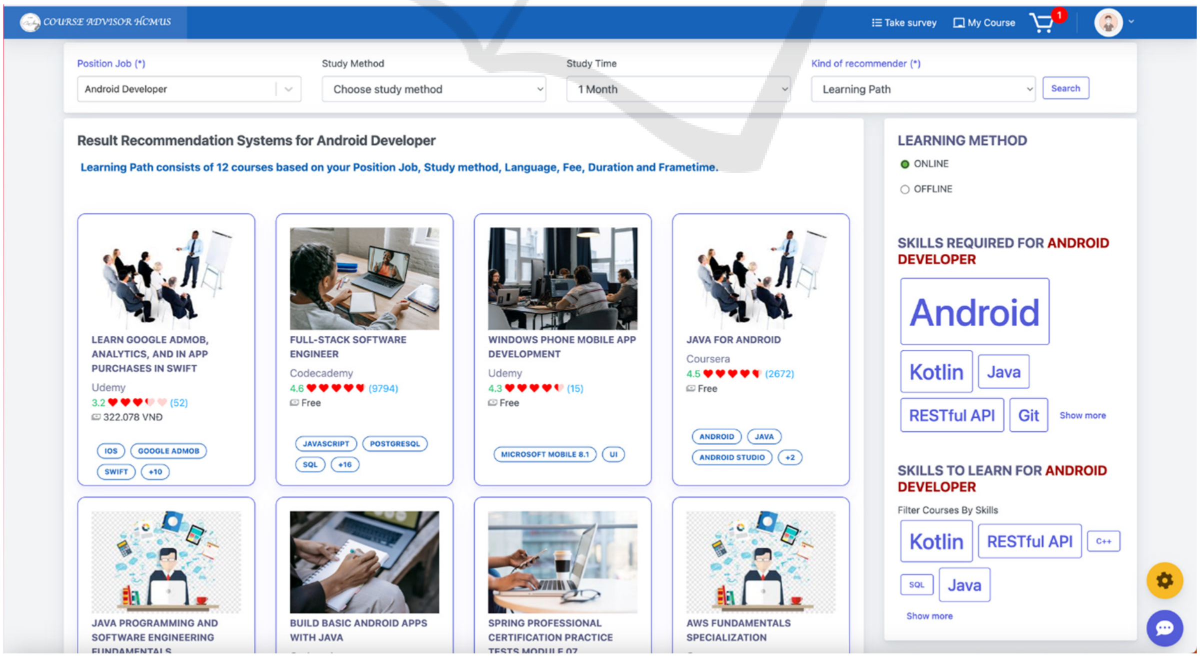

Figure 5: Main screen showing recommend courses or learning path, in descending order of relevance.

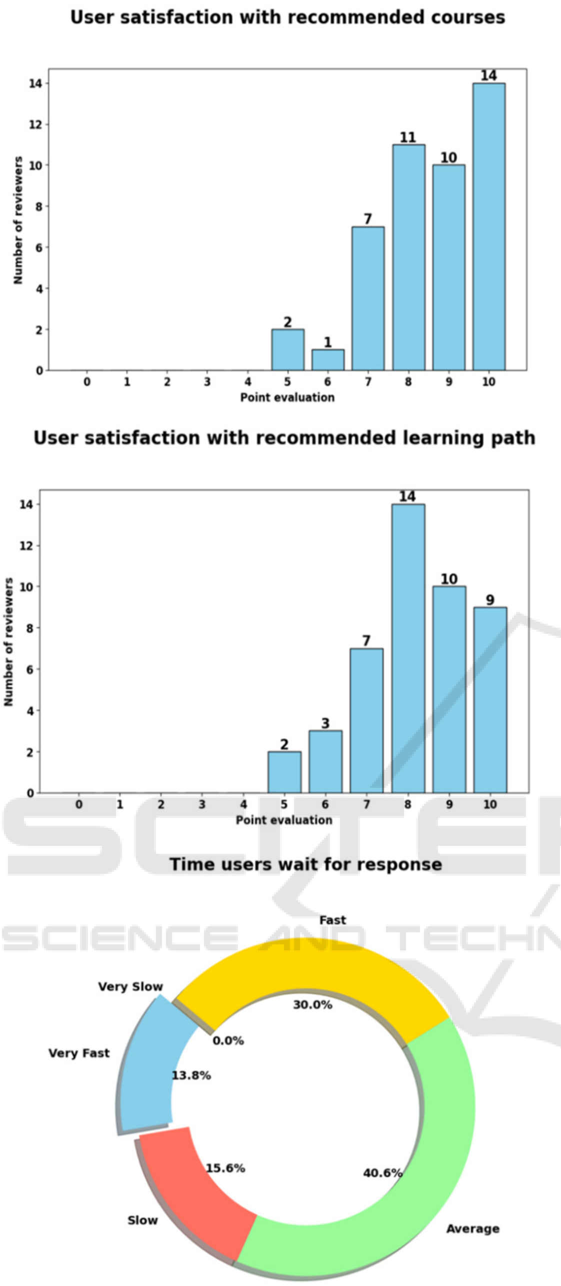

Figure 6: (a) User satisfaction with recommended courses, (b) User satisfaction with recommended learning path, (c) Time users wait for response.

Participants were first asked to sign up for the system and register their existing skills, preferred learning languages, budget, desired time for courses, current location, occupation, and more. Then, they logged in and used the system to search for courses or learning paths, check all skills and courses recommended by the system for relevance, add courses to their cart, select and join courses if any.

After a period of using the system, all participants were asked to complete a survey based on their experience. The survey results were then collected, revealing that a small-scale test effectively demonstrates the system's ability to provide accurate responses. Users were given the opportunity to express their satisfaction using a 10-point scale.

Figure 6(a) shows that users exhibited substantial satisfaction with course recommendations based on their professions. Notably, 31% of respondents awarded a perfect score of 10, indicating a robust positive sentiment. In contrast, only 2% provided a moderate rating with a score of 5 and 6. Moving on to Figure 6(b), satisfaction with learning path recommendations also showed positive trends, particularly concerning users' professional backgrounds. Here, 31% expressed satisfaction with a score of 8, while an impressive 20-22% rated the recommendation at the highest levels (9-10). In conclusion, both Figure 6(a) and Figure 6(b) confirm the system's effectiveness in meeting user preferences and professional needs. The overwhelmingly positive feedback indicates a promising acceptance of the system's utility in guiding users through their educational paths.

Figure 6(c) shows that the participants' contentment with the system's response speed varies. While 43.8% were happy, 15.6% were disappointed with it. The main drawback concerned the calculation speed of the recommendation method, especially for participants using the system for the first time. Fortunately, this issue can be technically enhanced in our future work. Additionally, 40.6% expressed a neutral level of satisfaction when giving an average point, indicating that the performance is not an issue for them. Although the response speed may vary based on the loaded data, system caching techniques, etc., it reflects the initial effectiveness when the recommendation method integrated into the system, which is considered acceptable.

5.2 Accuracy of the Skills Extraction

The significance of the data collection process lies in identifying skills within courses that serve as candidates for the recommendation process. Accordingly, we measure the accuracy of the skills extraction step using the course dataset, which comprises 600 courses, using the F1-score metric. A total of 970 technology terms were manually extracted from these courses. Table 2 presents the overall F1-score, which is recorded as 0.73.

Table 2: Evaluation of the results of feature extraction for the Technology class.

Manual extraction	Auto extraction	Precise extraction	Recall	Precision	F1-score
970	1202	788	0.81	0.66	**0.73**

Currently, no other studies have used a dataset comparable to ours, making it challenging to establish a basis for comparing the effectiveness of our proposed method. While the Stanza library demonstrates effectiveness in sentence segmentation and word analysis, occasional inaccuracies in sentence segmentation pose challenges in identifying the intended technology skill names targeted by our rules.

6 CONCLUSIONS

This paper has explored a framework for a course KBRS that fulfils all six aforementioned characteristics. It includes a data collection process that reschedules gathering data from different online educational platforms to represent the knowledge base in graph database, ensuring that the collected data is always up to date that reflects the current market requirements and course information (C1, C5). The implemented system provides an interactive interface that enables learners to explicitly input and refine their requirements and preference (C2, C4). In this way, it can address the cold start problem by recommending courses to new learners based on rules and onlotigies (C6). The recommendation method incorporates a set of rules and various matching techniques to propose relevant courses or learning paths according to job requirements, learners' needs and profiles (C3). A system implementation was performed to demonstrate its functionalities and usage capability in a real environment.

During the evaluation, we observed that the effectiveness of the proposed solution achieved a high ranking from the learners' perspective. Most participants rated it above average, with particularly high ratings for its course recommendations.

However, the limitations of this study include its reliance on a specific IT field and a small-scale laboratory setting. It is neccesary to test the approach in a broader environment to further prove its stability and usability. Additionally, applying the approach to developing a course recommendation system for domains other than IT needs to be considered to extend the research findings.

With the rapid advancement of Artificial Intelligence, particularly Large Language Models (LLMs), we anticipate that these models can enhance both the recommendation and data collection processes. The Retrieval-Augmented Generation (RAG) model has emerged as promising model that combine the knowledge from LLMs and the local once to address specific domain problems (Gao et al., 2024). We believe that incorporating RAG into our approach will strengthen our solution. In this study, we have already begun applying GPT-4, an LLM, in the data consolidation step, as described in section 3.3, and we will continue this work in future research.

ACKNOWLEDGEMENT

This research is funded by the University of Science, VNU-HCM under grant number CNTT 2023-02.

REFERENCES

Gallup inc. (2017). A Nationally Representative Survey of Currently Enrolled Students, https://safesupportivelearning.ed.gov/resources/2017-college-student-survey-nationally-representative-survey-currently-enrolled-students, visited 2022/12/24.

Nguyen, D., Dinh , N., Pham-Nguyen, C., Le Dinh, T., Nguyen Hoai Nam, L. (2022). ITCareerBot: A Personalized Career Counselling Chatbot. In: *Recent Challenges in Intelligent Information and Database Systems. ACIIDS 2022*. Communications in Computer and Information Science, vol 1716. Springer. Singapore. https://doi.org/10.1007/978-981-19-8234-7_33.

O'Mahony, M. P., & Smyth, B. (2007). A recommender system for on-line course enrolment: An initial study. RecSys'07: *Proceedings of the 2007 ACM Conference on Recommender Systems*, 133–136. https://doi.org/10.1145/1297231.1297254

Bhumichitr, K., Channarukul, S., Saejiem, N., Jiamthapthaksin, R., & Nongpong, K. (2017). Recommender Systems for university elective course recommendation. *Proceedings of the 2017 14th International Joint Conference on Computer Science and Software Engineering, JCSSE 2017*

Huang, T., Zhan, G., Zhang, H., & Yang, H. (2017). MCRS: A course recommendation system for MOOCs. In *Multimedia Tools and Applications* (Vol. 68, pp. 1–19). https://doi.org/10.1007/s11042-017-4620-2

Javed, U., Shaukat, K., Hameed, I. A., Iqbal, F., Alam, T. M., & Luo, S. (2021). A review of content-based and context-based recommendation systems. *International Journal of Emerging Technologies in Learning (iJET),* 16(3), 274-306.

Nam, L. N. H. (2023). A Robust Approach for Hybrid Personalized Recommender Systems. In *International*

Conference on Theory and Practice of Digital Libraries (pp. 160-172). Cham: Springer Nature Switzerland.

Nam, L. N. H. (2021). Latent factor recommendation models for integrating explicit and implicit preferences in a multi-step decision-making process. *Expert Systems with Applications*, 174.

Charu C. A. (2016). *The book*, Recommender Systems. 1st edn. Springer Cham Heidelberg New York Dordrecht London. DOI 10.1007/978-3-319-29659-3

Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., & He, Q. (2020). A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 34(8), 3549-3568.

Majidi N. and John's Newfoundland S. (2018), A Personalized Course Recommendation System Based on Career Goals, Newfoundland, no. April, 2018.

Obeid C. et al. (2018). Ontology-based Recommender System in Higher Education. *Track: The Third Edition of Educational Knowledge Management Workshop. WWW 2018*, April 23-27, 2018, Lyon, France.

Ilkou, E. et al. (2021). EduCOR: An Educational and Career-Oriented Recommendation Ontology. In: Hotho, A., et al. *The Semantic Web – ISWC 2021. ISWC 2021*. Lecture Notes in Computer Science, vol 12922. Springer, Cham. https://doi.org/10.1007/978-3-030-88361-4_32.

Agarwal A., Mishra S., and Kolekar S. V., (2022). Knowledge-based recommendation system using semantic web rules based on Learning styles for MOOCs. *Cogent Eng.,* vol. 9, no. 1, 2022, doi: 10.1080/23311916.2021.2022568.

Tarus, J. K., Niu, Z., & Yousif, A. (2017). A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Future Generation Computer Systems*, 72, 37–48. https://doi.org/10.1016/j.future.2017.02.049

Subramanian E. K., Ramachandran (2019). Student career guidance system for recommendation of relevant course selection. International Journal of Recent Technology and Engineering (IJRTE), vol. 7, no. 6, pp. 493–496, Issue-6S4, April 2019.

Mochol, M., Wache, H., & Nixon, L. (2006). Improving the recruitment process through ontology-based querying. *CEUR Workshop Proceedings*, 226, 59–73.

Paudel, S., & Shakya, A. (2017). Ontology based Job-Candidate Matching using Skill Sets. 8914, 251–258. Retrieved from http://conference.ioe.edu.np/ioegc2017/papers/IOEGC-2017-34.pdf.

Straccia, U., Tinelli, E., Colucci, S., Di Noia, T., & Di Sciascio, E. (2009). A system for retrieving top-k candidates to job positions. *CEUR Workshop Proceedings*, 477.

Corde, S., Chifu, V. R., Salomie, I., Chifu, E. S., & Iepure, A. (2016). Bird Mating Optimization method for one-to-N skill matching. Proceedings - 2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing, ICCP 2016, 155–162. https://doi.org/10.1109/ICCP.2016.7737139.

Younten T., Kristina T. (2021). Ontology-Based Recommender System of Online Courses. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*. Volume 9 Issue VIII.

Le Dinh, T., Pham Thi, T. T., Pham-Nguyen, C., Nam, L. N. H. (2022). A knowledge-based model for context-aware smart service systems. *Journal of Information and Telecommunication*, 6(2), 141–162. https://doi.org/10.1080/24751839.2021.1962105.

Thi P., Diep H., Nguyen Dinh T., Pham-Nguyen C., Le Dinh T., Nam L. (2020). Towards An Ontology-Based Knowledge Base for Job Postings. *7th NAFOSTED NICS 2020. Nov. 2020, pp. 267–272*. doi: 10.1109/NICS51282.2020.9335876.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Gupta A. and Garg D. (2014). Applying data mining techniques in job recommender system for considering candidate job preferences. *2014 International Conference on Advances in Computing, Communications and Informatics* (ICACCI), Delhi, India, 2014, pp. 1458-1465, doi: 10.1109/ICACCI.2014.6968361.

Gao Y. et al., (2024). Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997v5 [cs.CL] for this version). https://doi.org/10.48550/arXiv.2312.10997