IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 November 2024, accepted 11 December 2024, date of publication 17 December 2024,
date of current version 27 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3519263

RESEARCH ARTICLE

Development of a Knowledge Graph-Based Model for Recommending MOOCs to Supplement University Educational Programs in Line With Employer Requirements

VALIYA RAMAZANOVA[1], MADINA SAMBETBAYEVA[1], SANDUGASH SERIKBAYEVA[1], ZHANNA SADIRMEKOVA[2], AND AIGERIM YERIMBETOVA[2]

[1]L. N. Gumilyov Eurasian National University, 010000 Astana, Kazakhstan
[2]Institute of Information and Computation Technologies, 050000 Almaty, Kazakhstan

Corresponding author: Sandugash Serikbayeva (inf_8585@mail.ru)

This work was supported by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan under Grant AP22783030.

ABSTRACT The modern labor market demands that educational institutions prepare specialists capable of effectively responding to rapidly changing professional standards and technologies. In this regard, the use of innovative approaches to adapt educational programs has become a key factor. This study is dedicated to developing a methodology for using heterogeneous knowledge graphs to create a recommendation system aimed at bridging the gap between existing educational courses and the dynamically changing requirements of the labor market. The central element of the study is the use of knowledge graphs to aggregate and analyze diverse data on skills, job vacancies, and educational courses. Knowledge graphs not only structure large volumes of information but also visualize complex connections between various educational modules and professional requirements. This approach fosters a deeper understanding of how educational programs can be adjusted to match the market specifics. An important aspect of the study is the application of multilingual semantic similarity algorithms to analyze and match skills. These algorithms play a key role in determining the degree of correspondence between the skills listed in educational programs and courses, and those required for specific job vacancies. The use of natural language processing techniques allows not only capturing explicit keyword matches, but also recognizing deep semantic connections, which is an integral part of accurate matching in educational and professional domains. The results of the study demonstrate that the proposed methodology can effectively analyze the multilingual relationships between educational and professional skills, which improves personalized courses and job recommendations. Our study contributes to the literature by proposing a new methodology for building recommendations that improves the accuracy of personalized educational and career recommendations, and facilitates the adaptation of educational programs to dynamic changes in the labor market.

INDEX TERMS Knowledge graphs, recommendation system, integration of education and labor market, recruitment websites, curriculum, skills, job vacancies, massive open online courses (MOOCs), semantic similarity, machine learning algorithms, natural language processing, skill similarity, data analysis, ontological model, skill embeddings, vector representation, reskilling, professional development, career advancement.

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

I. INTRODUCTION

In light of the rapidly changing requirements of the labor market and applied technologies, modern education faces the

necessity of revising its methods and approaches to designing educational courses. Standard educational programs often fail to adapt to the individual needs of students and quickly become outdated, not reflecting the latest achievements in various fields of knowledge. There is a gap between the content of educational programs and the real needs of the modern IT sector, leading to the insufficient preparation of graduates for professional activities. To close this gap and provide students with relevant skills, innovative approaches to recommending educational courses based on current employer requirements are required. Reports, surveys, and studies by leading organizations point to significant changes in the labor market and the need to adapt educational programs to new realities. According to the World Economic Forum's Future of Jobs report, by 2025, 50% of workers will need to be retrained owing to the impact of technology, automation, and artificial intelligence [1]. However, according to McKinsey, 43% of employers cannot find employees with the required skills, especially in areas such as information technology, engineering, and data analysis, which highlights the lack of digital and technical competencies that educational institutions cannot fill. Educational systems in many countries, including Kazakhstan (according to the Ministry of Labor and Social Protection of the Population), face inertia [2], [3], which makes it difficult to prepare specialists for modern requirements. Although massive open online courses (MOOCs) offer flexibility and accessibility, they do not solve all problems associated with low program completion, insufficient practical focus, lack of personalized guidance, and non-recognition of certificates by employers. All these factors highlight the need to adapt educational programs to rapidly changing labor market standards.

In the educational landscape, knowledge graphs play an important role in structuring and organizing educational content. Knowledge graphs, which represent networks of interconnected concepts and ideas, provide a deep understanding of educational material, allowing students to visualize and explore various connections between knowledge. Knowledge graphs foster the development of critical thinking and independent learning through the ability to integrate and link diverse information resources. Machine learning is a revolutionizing learning method that personalizes the educational processes. This technology enables the creation of intelligent systems capable of analyzing learner data and offers individualized learning materials and methods that meet learners' needs and knowledge levels. Owing to machine learning, educational platforms can adapt learning materials in real time, taking into account the speed of information assimilation and learner characteristics. Furthermore, machine learning has been used to analyze the effectiveness of educational programs, identify trends in educational processes, and develop new teaching methods based on large volumes of data.

However, the issues of integrating education, the labor market, and MOOCs using knowledge graphs and machine learning have not been fully explored, as some interdomain connections between skills acquired by modern multilingual transformers for mixed sentences in Russian and English are missing. Knowledge graphs allow structuring information about required skills and competencies, as well as identifying connections between them and labor market requirements. Machine learning facilitates the analysis of large volumes of data on demanded professional skills and predicts their relationship with educational skills. By integrating these tools into the process of designing educational courses, educational institutions can promptly respond to changes in labor market requirements and provide graduates with the necessary knowledge and skills for successful careers. Thus, this study proposes an approach supported by skill embeddings from modern multilingual transformers and text embeddings to address the problem of interdomain recommendations in higher education, the labor market, and MOOCs from the perspective of a heterogeneous knowledge graph. These three domains have several types of vertices that interact through different types of connections. A meta-path for recommending job vacancies and courses is created: educational program-skill-skill-job vacancy-skill-skill-course, which can be used for various practical scenarios. For example, for an instructor creating a new course for the elective discipline catalog, a student or job seeker can identify the unmastered skills of an interested job vacancy and receive a recommended list of courses based on their skill set.

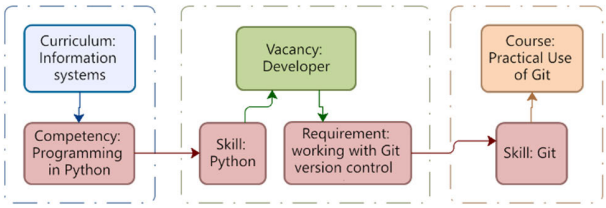

FIGURE 1. Heterogeneous scheme of the knowledge graph.

Fig. 1 shows a simplified example of integrating three disparate data sources – higher education, job vacancies, and MOOCs – into a heterogeneous knowledge graph using skills as the linking element. Consider the use of the meta-path: educational program-skill-skill-job vacancy-skill-skill-course in the example of the educational program "Information Systems," the vacancy "Developer" with the skill set Python and Git, and the MOOC course on Git. 1. Program1 (Information Systems): Teaches Skill1 (Python). 2. Skill1 (Python): Linked to Skill2 (data analysis using Python). 3. Skill2 (data analysis using Python): Required for Vacancy1 (Developer). 4. Vacancy1 (Developer): Also requires Skill3 (working with version control systems Git). 5. Skill3 (working with version control systems like Git): Linked to Skill4 (Git). 6. Skill4 (Git): Taught in the MOOC course MCourse1 (practical use of Git). Knowledge graphs are complemented by sentence vectorization algorithms, which are used to solve sentence-similarity tasks based on modern transformers. Skill phrase embeddings are compared pairwise using cosine similarity to calculate the similarity between skills from different domains, helping to link data

on higher education, job vacancies, and MOOCs. Nodes of the educational program are represented by blue squares, skill nodes by red squares, vacancies by green squares, and course nodes by orange squares. Relationships between program and skill are shown with blue-directed arrows and job-skill relationships (skills required for the job) with green-directed arrows. Skill-skill relationships are depicted with red-directed arrows, and skill-course relationships are represented by orange-directed arrows. The goal of this study was to develop a methodology for using knowledge graphs and machine learning based on multilingual data sources in Russian and English to recommend job vacancies and courses that build missing skills and competencies. These skills and competencies were not acquired in the university's educational program or at a previous educational stage, but were demanded by employers, as exemplified by the IT labor market. This model acts as a bridge between academic training and the practical needs of the market, using data from educational programs, recruitment websites, and massive open online courses (MOOCs) to identify specific skills and knowledge that are most relevant and required in a professional IT environment. It recommends courses aimed at bridging the gap between university educational programs and the modern labor market. In this context, knowledge graphs and machine learning algorithms serve as tools for skill analysis, domain integration, and recommendation of required qualifications. They enable educational institutions to tailor and adapt their programs more effectively to prepare graduates to meet the current demands of the IT sector.

II. METHODS

Knowledge graphs have found the most diverse applications in education for student audiences, helping in knowledge representation and organization, personalized learning, and learning analytics. Knowledge graphs provide a more structured and organized way of presenting knowledge, making it easier for students to link and understand concepts [4], [5], [6], [7], [8]. They are used for personalized learning [6], [7], and offer students tailored learning materials based on their learning preferences. Knowledge graphs are also employed for learning analytics [11], collecting and analyzing data on student performance, to identify areas where they need additional support or where learning materials can be improved. For educators, they are used to help structure educational materials [12], [13], for managers to build educational trajectories [14], and as tools for decision-making by all participants in the educational process [15]. Several key trends and common approaches to the use of knowledge graphs in educational processes have been highlighted. An important trend is the desire to unify the standards for knowledge representation. Authors such as [6], [9], [10], and [15] note the use of ontologies, including standard schemas such as Schema.org, which facilitate more efficient data exchange and mutual understanding in the educational context. Many researchers, including [6], [7], [8], [10], [12], [17], and [18], have successfully integrated machine learning methods to enhance educational processes. The use of models such as Word2vec, BERT, and Graph Convolutional Networks (GCN) underscores the importance of applying advanced technologies to adapt education to each student's needs. Expert knowledge, as presented in [4], [11], and [16], plays a significant role in constructing problem-oriented knowledge graphs. Beyond the purely educational context, knowledge graphs have found applications at the intersection of various subject areas such as the integration of different levels of education and the integration of education and the labor market. In the context of employment, knowledge graphs are used for recommendation systems [19], [20], which determine the most suitable job vacancies based on user skill sets. There are also applications of knowledge graphs that link education, science, and employment. For example, constructing or using knowledge graphs to connect job vacancies and MOOC education through entity alignment [21]; linking job classifiers and educational courses at SAE Institute using entity recognition and classification methods with the distilbert-base-german-cased algorithm [22]; linking MOOCs with university courses and job vacancies for job and university course recommendations using skill similarity determination methods with BM25, Word2vec, and Bert algorithms [23]; and analyzing the mismatch between scientific publications, job vacancies, and MOOC education [24]. These works are based on integration through the extraction, classification, and alignment of skills, applying algorithms such as FastText, BM25, Word2Vec, Bert, all-mpnet-base-v2 [19], [23], [25], and others, and discovering new connections between skills and professions using Node2Vec and Preferential Attachment [20]. To determine node similarity, measures such as Jaccard, cosine similarity, and Euclidean distance [23], [25] were used. In skill analysis, various correlation methods, such as Pearson correlation, Kullback-Leibler divergence, and multidimensional Hawkes processes, are applied [24]. Algorithms such as Dijkstra's shortest path, random walk algorithms, community detection, graph neural networks, policy-based path reasoning, genetic algorithms, and ant colony optimization are used for recommendations. The TF–IDF term-weighting method is employed to determine the most relevant skills for different professions [20]. In recent years, graph neural networks (GNNs) have become an important tool for building recommender systems based on knowledge graphs. One of the main advantages of GNNs is their ability to model complex relationships between entities such as users, courses, and skills, which allows for personalized and more accurate recommendations. For example, the study [26] presents methods for using GNNs to solve recommendation problems on a large scale, demonstrating their high efficiency compared with traditional approaches such as collaborative filtering. This study showed that GNNs can successfully handle complex relationships and provide more relevant recommendations. The work [27] highlights the potential of GNNs to improve recommendation accuracy by aggregating information from neighboring nodes in

a knowledge graph. Other studies have focused on comparing GNNs with traditional methods, where GNNs have shown significant advantages in recommendation accuracy for online courses, particularly when integrating data from multiple sources [28]. These studies show that GNNs not only improve accuracy, but also provide more interpretable and diverse recommendations by leveraging rich contextual information [29], [30]. The quality of machine-learning algorithms is usually assessed using precision (P), recall (R), and F1 measures, for example, for entity recognition, classification, and linking tasks [22]. Recommendation quality measures included Mean Average Precision (MAP@K), precision (P@K), and normalized discounted cumulative gain (NDCG@K). For example, [23] evaluated ranking effectiveness at a specific threshold level, considering only the top K-courses returned as candidates in the experiment. However, because of the lack of studied connections along the program-skill-skill-job vacancy-skill-skill-course path in mixed phrases in Russian and English to integrate higher education, the labor market, and MOOCs using knowledge graphs and machine learning, additional research is required on skill similarity using modern text vectorization transformers for job and MOOC course recommendations. Additionally, unlike previous studies, career data, in addition to job skills, are supplemented with descriptions of requirements and responsibilities, enriching the job context. Our research proposes a multilingual methodology that uses a heterogeneous knowledge graph that combines educational and career data to recommend jobs and corresponding additional courses based on the content of the educational program or experience for teachers or educational program managers, as well as for students or employees at different levels. Consider the description of the proposed methodology, including data collection on education and job vacancies (Section II-A), ontological modeling of subject areas (Section II-B), and data integration based on semantic similarity (Section II-C).

A. DATA COLLECTION

In this study, three datasets were collected and analyzed: competency maps of a university educational program and discipline annotations, job vacancies from the online recruitment site hh.kz, and MOOCs from Coursera.

The availability of open data, such as job vacancies from the hh.kz website and free courses from the Coursera platform, ensured ease of collection and transparency of information. Access to specialized corporate-licensed platforms requires proprietary data. In this study, we focus on publicly available resources. The datasets were selected to ensure the relevance and representativeness of the labor market, educational programs, and online course offerings. Job vacancy data from hh.kz are frequently updated and cover a wide range of professions, ensuring a comprehensive representation of job-market trends. The Coursera dataset was selected to ensure access to internationally recognized high-quality MOOCs. Competency data from university programs were collected to align with national educational standards and higher education requirements. This approach guarantees that the integrated knowledge graph reflects the current labor market and educational needs.

TABLE 1. List of information technology professions on headhunter.

№	Profession ID	Profession name
1.	10	Analyst
2.	12	Art Director, Creative Director
3.	25	Game Designer
4.	34	Designer, Artist
5.	36	Chief Information Officer (CIO)
6.	73	Product Manager
7.	96	Programmer, Developer
8.	104	Development Team Lead
9.	107	Project Manager
10.	112	Network Engineer
11.	113	System Administrator
12.	114	Systems Engineer
13.	116	Information Security Specialist
14.	121	Technical Support Specialist
15.	124	Tester, QA Engineer
16.	125	Chief Technology Officer (CTO)
17.	126	Technical Writer
18.	148	Systems Analyst
19.	150	Business Analyst
20.	155	Methodologist
21.	156	BI Analyst, Data Analyst
22.	157	Head of Analytics
23.	160	DevOps Engineer
24.	164	Product Analyst
25.	165	Data Scientist

The educational program dataset contains phrases of competency names, learning outcomes, and brief annotations of courses in the specialty ‘‘Information Systems’’ in Russian for 2022 intake.

The hh.kz dataset consists of job titles, company names, industry, experience, salary, publication date, job link, skills, part of the requirements, and responsibilities in Russian and English. The data were downloaded from the HeadHunter website (https://hh.kz/) using the API (https://api.hh.ru/vacancies) and Python programming language. Job vacancies were selected for 25 professions in the information technology field (Table 1) from 177 settlements in Kazakhstan over two months, from February 1, 2024, to March 31, 2024. The number of job vacancies is 5248.

The Coursera course dataset consists of free courses in English on topics related to computer science, information technology, and data science, as of May 3, 2024. The data included course titles, difficulty levels, ratings, lists of skills, names of modules and lessons with their study durations, and brief descriptions. The course data were collected using Beautiful Soup scraping tools. The total number of courses is 953. All three datasets were loaded into the graph database management system Neo4j as graphs.

B. DOMAIN MODELING

This section presents developing a knowledge graph model that describes entities, their relationships, and attributes related to skills and competencies in information technology. This model also considers the relationships between these skills with educational programs, job vacancies, and MOOCs.

The modeling process was performed using set theory, articles [31], [32], the knowledge graph standard [33], and the conceptual mapping visualization tool Cmap [34]. According to the standard, the conceptual model of the knowledge graph is divided into the ontology level and the instance level.

We define the ontological model of the knowledge graph of the subject area as a heterogeneous, labeled, directed graph $G = \{E, R, A, V, r, f, p\}$, where: E - is the set of entity classes; R is the set of relationship types; A is the set of attributes of entities or relationships; V is the set of attribute values; the function $r: R \rightarrow E \times E$ means that each relationship type is assigned a pair of entity classes; the function $f: E \cup R \rightarrow 2A$ means that each entity class or relationship type has an attribute from the set A; the function $p: E \cup R \rightarrow 2A \times V$ means the correspondence of an entity class or relationship type to a set of attribute-value pairs.This section analyzes the subject area of the university's educational program. The educational program contains a competency map and a catalog of elective courses. Competency maps include competencies, learning outcomes, and courses. Learning outcomes are units of competencies. Learning outcomes are formed by courses, for which there is a brief description in the catalog of elective courses.

The ontological model of the graph representing the university's educational program is analyzed in this section. The graph includes a set of entity classes: $E_{\text{university}} = \{\text{Curriculum, CUA}\}$. The educational program (EP) is a class representing a structured plan of study modules and activities. CUA consists of a class of competency phrases, learning outcomes, and annotations.

Competency: Reflects the specific abilities that a student must develop within the educational program.

Learning Outcomes (LO): These are the core knowledge, skills, and competencies that students are required to demonstrate upon the completion of the educational program.

Annotation: Provides a brief description of the course content, allowing students and instructors to quickly assess its main topics and significance in the context of the curriculum.

We define the set of relationships $R_{\text{university}} = \{\text{HAS}\}$. The characteristics of the relationships and attributes are presented in the Table 2.

The ontological model is visually depicted in Fig. 2.

TABLE 2. Description of the ontology of the knowledge graph of the university's educational program.

Relationships and attributes	Description
Curriculum $\xrightarrow{\text{HAS}}$CUA	The educational program has phrases of competencies, units of study and annotations of disciplines.
f(Curriculum) = {code, title}	The educational program is characterized by attributes such as code and name.
f(CUA) = {code, description, embedding}	Each competence has a unique code, competencies are divided into universal competencies (CC), general professional competencies (MIC), professional competencies (PC), discipline annotations, the description is divided into phrases, which are also presented in the form of embedding.

Fig. 3 provides a schematic representation of the process for constructing a knowledge graph from a file containing descriptions of disciplines, competencies, and learning outcomes. Non-essential text was manually removed from the educational program of the specialty, retaining only the descriptions of competencies, their corresponding units of study, and the annotations associated with the disciplines.

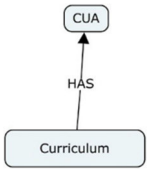

FIGURE 2. Visual representation of the ontological model of the university's knowledge graph with a class of phrases of competencies, learning outcomes and annotations.

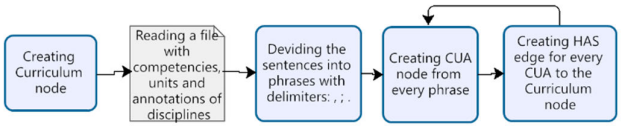

FIGURE 3. The process of building a knowledge graph EP.

The text data was converted to the.txt format. Using Python, the lines containing descriptions of competencies, units of study, and annotations were segmented into individual phrases, with commas, semicolons, and periods serving as delimiters. Each phrase was represented as a distinct node. All phrase nodes were connected to the main node of the educational program, referred to as "Curriculum", via the HAS relation. Embeddings for each phrase were generated using the paraphrase-multilingual-mpnet-base-v2 model, a Sentence Transformer, and were stored as properties of the corresponding nodes. This process resulted in the construction of a graph in Neo4j, with Fig. 4 presenting a subgraph containing four nodes.

FIGURE 4. The EP subgraph, consisting of CUA nodes and HAS links, is provided in two versions: a) in the original language; b) in its English translation.

HeadHunter is one of the largest online recruitment companies, established in 2000. As of October 2024, SimilarWeb ranks HeadHunter as the fourth most popular job and employee search portal globally. The platform publishes an average of over 933,000 vacancy ads daily.

This section analyzes the structure of vacancy announcements. The key entities include vacancies, skills, requirements, and responsibilities. Vacancies are open positions

corresponding to the professions that employers seek to fill. Skills are the knowledge or abilities that employers expect from applicants. Requirements and responsibilities refer to additional skills, qualifications, and specific duties that are essential for a given vacancy.

The ontological model of a vacancy graph is considered next. The graph consists of multiple classes of entities: $E_{Recruitment}$ = {Vacancy, Skill, Requirements, Responsibility} and a set of relationships $R_{Recruitment}$ = {REQUIRE, HAS_REQUIREMENT, HAS_RESPONSIBILITY}. The characteristics of these relationships, along with their respective attributes, are presented in Table 3.

TABLE 3. Description of the ontology of the knowledge graph for vacancy announcement.

Relationships and attributes	Description
Vacancy $\xrightarrow{\text{REQUIRE}}$ Skill	The job requires certain skills.
Vacancy $\xrightarrow{\text{HAS\_REQUIREMENT}}$ Requirement	The vacancy has requirements.
Vacancy $\xrightarrow{\text{HAS\_RESPONSIBILITY}}$ Responsibility	The vacancy has responsibilities.
f(Vacancy) = {title, profession, city, company, industry, experience, salary, url, publish_date}	The vacancy is characterized by such attributes as the name, profession name, city, company, field of activity of the company, experience in years, salary level, link to the vacancy, date of publication.
f(Skill) = {name, embedding}	The skill has a name and its embedding
f(Requirement) = {description, embedding}	The requirements contain the phrase of the text description and its embedding.
f(Responsibility) = {description, embedding}	Responsibilities contain the phrase of the text description and its embedding.

The names of skills and the descriptions of requirements are transformed into vector embeddings generated by the paraphrase-multilingual-mpnet-base-v2 Sentence Transformer. The ontological model is depicted in Fig. 5.

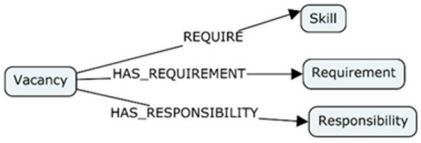

FIGURE 5. Visualization of the ontological model for the knowledge graph of vacancy announcements.

The process of constructing a knowledge graph on vacancies and skills is schematically depicted in Fig. 6.

The primary function responsible for job parsing establishes a connection with the Neo4j database. It iterates through a predefined list of cities, making API calls for each city. The script sends HTTP requests to https://api.hh.ru/vacancies to retrieve detailed information about vacancies. The request parameters specify the region, the number of results per page, and the page index. Upon receiving a response, the function verifies the response status and returns the structured data in JSON format.

The next stage of the process involves issuing an API request to obtain detailed information about a specific

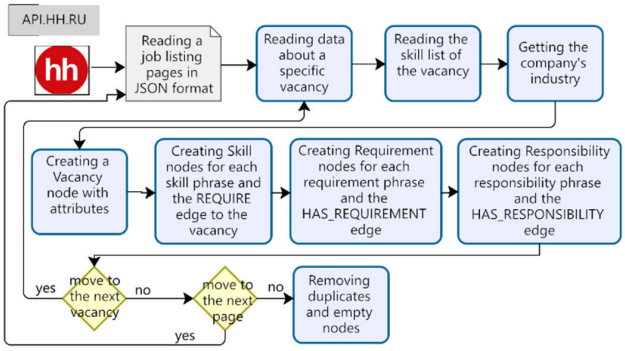

FIGURE 6. Schematic diagram of the construction process for nodes and edges in the vacancy knowledge graph.

vacancy. The script extracts a list of skills from the API response and formats it as a comma-separated string. Subsequently, an additional request is made to the API to retrieve company information using its unique identifier (ID), from which the company's industry details are extracted. The collected job-related information is stored in the Neo4j database. A Vacancy node is created to represent the job position, while distinct nodes are generated for each associated skill, requirement, and responsibility. These nodes are linked to the corresponding Vacancy node. To ensure data consistency, several queries are executed to eliminate duplicate Vacancy nodes, as well as empty or incomplete nodes for skills, requirements, and responsibilities.

Fig. 7 schematically depicts the vacancy subgraph. The subgraph's central node corresponds to the position "Senior IT Project Specialist (Aksai)". This node is connected to other nodes representing skills, requirements, and responsibilities, which collectively describe the key attributes of the role and the required qualifications.

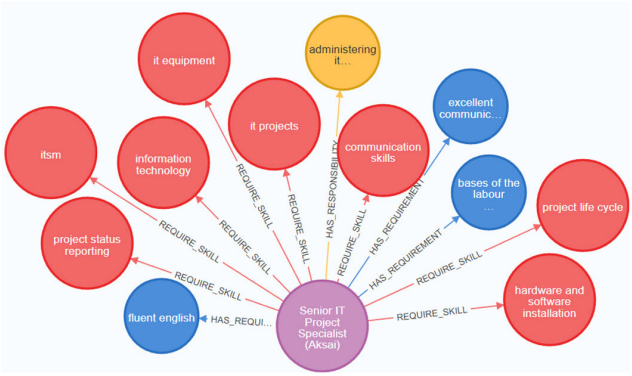

FIGURE 7. Subgraph of the vacancy. The node representing the vacancy is highlighted in purple, skill nodes are red, requirement nodes are blue, and responsibility nodes are yellow.

Structured data in JSON format was utilized to construct the job graph. Additionally, job descriptions can be parsed using the GPT model to extract key skills, thereby improving the accuracy in future.

A massive open online course (MOOC) is an online course designed for unrestricted participation and open access via

the Internet. Coursera serves as the primary data source. Established in 2012, Coursera Inc. is a leading provider of MOOCs. By 2023, more than 300 universities and companies (such as Google, IBM, and Meta) had partnered with Coursera to offer online courses. By 2024, the total number of available courses had increased to approximately 7,000.

The key entities include MOOC courses, MOOC skills, MOOC modules, and MOOC lessons. A MOOC course is a course on the MOOC platform designed for an unlimited number of participants via the Internet. MOOC skills refer to specific knowledge and competencies that students acquire upon completing a course. The MOOC module is a structural unit of the course, representing a separate block or section dedicated to a specific topic or set of learning objectives. The MOOC lesson is an integral part of the module, each of which contributes to achieving the educational goals defined for that module. The ontological model of the MOOC domain is presented as a graph that includes the following entity classes: $R_{MOOC} = \{MCourse, MSkill, Module, MLesson\}$ and many relationships $R_{MOOC} = \{DEVELOPED\_BY, CONSISTS\_OF\}$. The characteristics of these relationships, as well as their attributes, are outlined in Table 4.

TABLE 4. Ontology of the knowledge graph for the MOOC course: Description.

Relationships and attributes	Description
$MSkill \xrightarrow{DEVELOPED\_BY} MCourse$	Skills are developed in MOOC courses
$MCourse \xrightarrow{CONSISTS\_OF} Mmodule$	The MOOC course consists of modules.
$Mmodule \xrightarrow{CONSISTS\_OF} MLesson$	The module consists of lessons.
f(MCourse) = {title, url, organization, certificate_type, rating, reviews_num, difficulty, students_enrolled}	The course is characterized by attributes such as name, link to the course, organization, type of course, user rating, number of reviews, difficulty level, number of subscribed students.
f(MSkill) = {name, embedding}	The skill has a name and its embedding.
f(Mmodule) = {title, description, duration, embedding}	The module contains a title, a phrase of the text description, a duration in hours or minutes, embedding of the description phrase.
f(MLesson) = {title, type, embedding}	The activity has a title, type of activity, embedding of the title.

The names of skills, as well as the names and descriptions of modules, are additionally transformed into vector embeddings using the paraphrase-multilingual-mpnet-base-v2 model. The structure of the model is visually illustrated in Fig. 8.

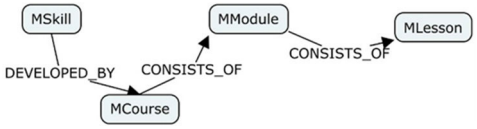

FIGURE 8. Ontological model of the MOOC course knowledge graph: Visual representation.

The process of constructing a knowledge graph from vacancies and skills is schematically shown in Fig. 9.

To retrieve webpage content, an HTTP request is issued to the server. The returned page content is processed using the BeautifulSoup library. The data collection process is organized as a cyclic web scraping procedure that calls auxiliary functions to collect and process the data. For each page and each element within the page, a designated function identifies the data to be extracted and defines its processing method. After the data collection is complete, the results are stored in a DataFrame for further analysis.

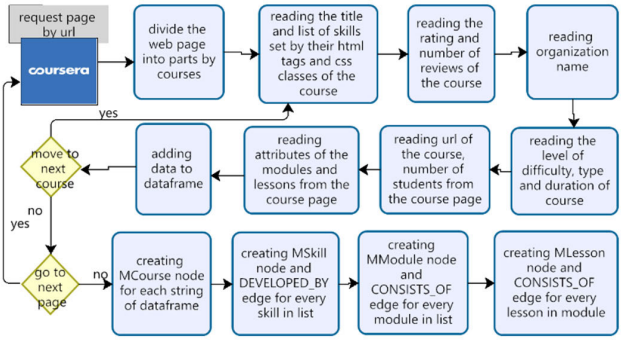

FIGURE 9. Visualization of the process of constructing nodes and edges in the MOOC knowledge graph.

For each row in the DataFrame, an MCourse node is generated. For each skill, module, and lesson within the course, corresponding MSkill, MModule, and MLesson nodes are created. Relationships are established between the nodes to reflect their connections, including the development of specific skills within each course. In Fig. 10, a subgraph of a MOOC course is illustrated, showing three skills linked to other courses and five modules, each containing multiple lessons.

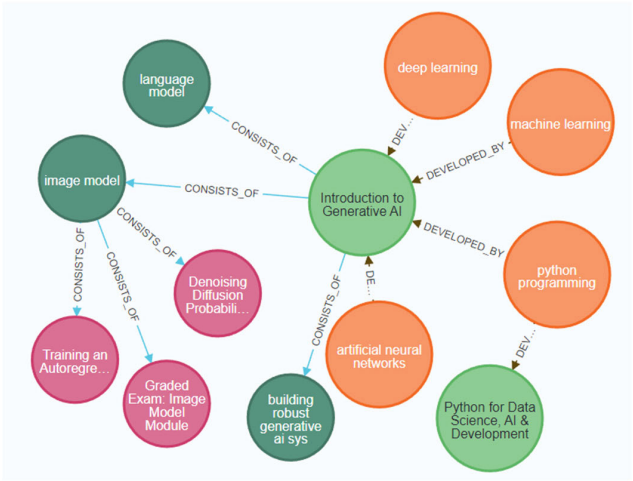

FIGURE 10. Green nodes represent MOOC courses, orange nodes represent course skills, dark green nodes represent modules, and pink nodes represent module lessons.

C. INTEGRATION OF THREE KNOWLEDGE GRAPHS

The ontologies of the three knowledge graphs should be integrated at the skill level by calculating the cosine similarity of skill embeddings from different sources. Embeddings are generated using a multilingual pre-trained model of Sentence Transformers, such as paraphrase-multilingual-mpnet-

base-v2. Paraphrase-multilingual-mpnet-base-v2 is a pre-trained Sentence Transformer model designed to generate dense text embeddings by mapping sentences and paragraphs into a 768-dimensional vector space. The model is trained on more than 50 languages and optimized to create vectors that effectively represent the semantic content of the text. This makes the embeddings particularly useful for tasks such as paraphrase retrieval, semantic search, and text clustering. The choice of this specific multilingual model is due to the high performance of the mpnet family (from all-mpnet-base-v2 to paraphrase-multilingual-mpnet-base-v2), its relative novelty (2021), and its high rating on the official website https://sbert.net/.

To determine the optimal embedding approach, a comparative evaluation of three models was conducted: BERT, Word2Vec, and paraphrase-multilingual-mpnet-base-v2. Testing was performed on an English dataset of skill paraphrases with binary labels (0 and 1). The evaluation of model performance was based on precision, recall, and F1-score metrics. The results, as shown in Table 5, reveal that paraphrase-multilingual-mpnet-base-v2 achieved the highest F1-score, indicating a balanced performance in terms of precision and recall.

TABLE 5. Comparison of test results for three models.

Model	Precision	Recall	F1-Score
BERT	0,5479	0,9645	0,6988
Word2Vec	0,991	0,6338	0,7731
Paraphrase-multilingual-mpnet-base-v2	0,983	0,8159	0,8917

Next, the cosine similarity between the embeddings of educational program competencies and job skills, as well as between job skills and MOOC skills, was calculated. Additional connections between the three ontologies were established when the skill similarity exceeded a predefined threshold, as presented in Table 6.

To formalize this logic mathematically, we define a set of operations that describe each step of the process. The main components are defined as follows:

Let:

$E(e_i)$ – the embeddings of phrases e_i, of CUA nodes in $E_{university.}$ graph, generated by the paraphrase-multilingual-mpnet-base-v2 model.

$E(e_j)$ – the embeddings of phrases e_j in Skill, Requirements, and Responsibilities nodes in $E_{Recruitment}$ job graph, generated by the paraphrase-multilingual-mpnet-base-v2 model.

The cosine similarity between the vector representations $E(e_i)$ and $E(e_j)$ is calculated as the dot product of the vectors divided by the product of their lengths, according to the Equation (1):

$$\textbf{CosineSimilarity}\left(E(e_i), E(e_j)\right) = \frac{E(e_i) \cdot E(e_j)}{\|E(e_i)\| \cdot \|E(e_j)\|} \quad (1)$$

TABLE 6. Generated links for the integration of three knowledge graphs.

Relationships and Properties	Description
Let: CUA∈ $E_{university.}$ Skill, Requirement, Responsibility ∈ $E_{Recruitment}$ CUA $\xrightarrow{IS\_SIMILAR\_TO}$ Skill CUA $\xrightarrow{IS\_SIMILAR\_TO}$ Requirement CUA $\xrightarrow{IS\_SIMILAR\_TO}$ Responsibility	A link between the EP graph and the job graph is established if the embedding of a competence description, learning outcome (LO), or discipline annotation phrase has a high cosine similarity with the embedding of a skill name, requirement description, or responsibility.
Let: Skill, Requirement, Responsibility ∈ $E_{Recruitment}$ MSkill, MModule ∈ E_{MOOC} Skill $\xrightarrow{IS\_SIMILAR\_TO2}$ MSkill Skill $\xrightarrow{IS\_SIMILAR\_TO2}$ MModule Requirement $\xrightarrow{IS\_SIMILAR\_TO2}$ MSkill Requirement $\xrightarrow{IS\_SIMILAR\_TO2}$ MModule Responsibility $\xrightarrow{IS\_SIMILAR\_TO2}$ MSkill Responsibility $\xrightarrow{IS\_SIMILAR\_TO2}$ MModule	A relationship between the job graph and the MOOC graph is established if the embedding of a skill name, requirement description, or responsibility has a high cosine similarity with the embedding of a skill name or module of the MOOC course.
f(IS_SIMILAR_TO) = {range} f(IS_SIMILAR_TO2) = {range}	The similarity link attribute takes values in the range from 0 to 1.

Algorithm for Calculating Similar Skills in the EP Graph and the Job Graph:

1. Convert each e_i, e_j to lowercase.
2. Generate embeddings $E(e_i)$, $E(e_j)$ for each e_i and e_j.
3. For each pair of elements $(E(e_i), E(e_j))$, calculate CosineSimilarity$(E(e_i), E(e_j))$ using the Equation (1).
4. If CosineSimilarity$(E(e_i), E(e_j)) >$ threshold1:
   a. Create an IS_SIMILAR_TO edge in the graph;
   b. assign the CosineSimilarity$(E(e_i), E(e_j))$ value to the range attribute of the created edge using the function p: p(IS_SIMILAR_TO$(E(e_i), E(e_j))) =$ CosineSimilarity$(E(e_i), E(e_j))$.

A simplified diagram of the process of using sentence similarity to generate embeddings to create edges between nodes corresponding to discipline phrases and skill phrases is shown below in Fig. 11.

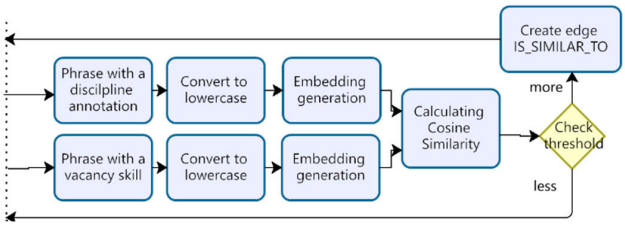

FIGURE 11. The process of creating a connection between two nodes with high cosine similarity between embeddings generated by the Sentence Transformers language model.

Creating links between job skills and MOOC skills follows a process similar to the previous algorithm. The main steps of the algorithm are as follows:

Let:

$E(e_j)$ – embeddings of phrases e_j of Skill, Requirement or Responsibility nodes in the job graph, generated using the paraphrase-multilingual-mpnet-base-v2 model.

$E(_{em})$ – embeddings of phrases em of MSkill or MModule nodes in the MOOC graph, generated using the paraphrase-multilingual-mpnet-base-v2 model.

Steps of the algorithm:

1. Convert each phrase e_j, e_m to lowercase.
2. Generate embeddings $E(e_j)$, $E(e_m)$ for each phrase e_j, e_m.
3. For each pair of embeddings ($E(e_j)$, $E(e_m)$, calculate CosineSimilarity($E(e_j)$, $E(_{e_m})$) using the Equation (1).
4. If CosineSimilarity($E(e_j)$, $E(e_m)$) > threshold2:
   a. create an IS_SIMILAR_TO2 edge in the graph;
   b. assign the CosineSimilarity($E(e_j)$, $E(e_m)$) value to the range attribute of the created edge using the function p: p(IS_SIMILAR_TO2($E(e_j)$, $E(e_m)$)) = CosineSimilarity($E(e_j)$, $E(e_m)$).

The integration of the three knowledge graphs – educational programs, job market data, and MOOCs – enables the creation of a heterogeneous knowledge graph. To achieve an optimal balance of precision the threshold1 >=0.87 and threshold2 >=0.81 were empirically established. Thus, three graphs are combined based on two sets of connections $\xrightarrow{\text{IS\_SIMILAR\_TO}}$, which are created when their calculated range attribute reaches a certain threshold value. Fig. 12 shows a visualization of the integration of three knowledge graphs ontologies.

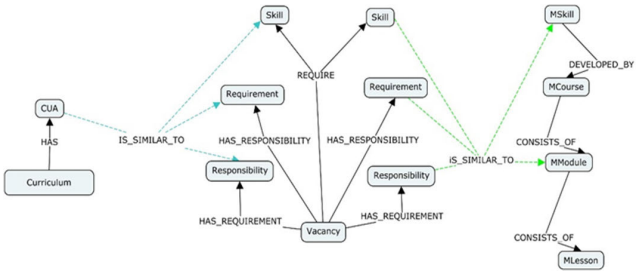

FIGURE 12. Integration of three ontologies at the skill level.

The integration of ontologies from the three knowledge graphs at the skill level facilitates the creation of a heterogeneous knowledge graph that unifies information from educational programs, job vacancies, and MOOC courses. By leveraging cosine similarity to assess the similarity between vector representations of competencies, learning modules, and annotations with the skills, requirements, and responsibilities from job vacancies, as well as with the skills and modules from MOOC courses, the approach ensures the formation of meaningful connections between these elements. The introduced IS_SIMILAR_TO relations, equipped with a range attribute representing the cosine similarity score, enable the precise classification and integration of corresponding elements. This approach contributes to the construction of a unified and interconnected knowledge space, where the elements from educational programs, job descriptions, and MOOCs are linked through semantically meaningful relationships.

Next, the EP graph is enriched with words from a dictionary built from the EP using the TF-IDF method, and the concept words from the EP that have a cosine similarity of ≥ 0.95 with job skills are added to the EP graph. Thus, the proposed approach enhances the accuracy and relevance of recommendations in a system focused on educational and career trajectories.

The graph has a clear modular structure, which makes it easy to add new entities and relationships. For example, if new courses or skills need to be added in the future, this can be done without disrupting the existing graph structure. Neo4j, along with other graph databases, generally performs well with an increase in the number of nodes and relationships, taking into account the optimization of indexes to maintain performance.

III. CROSS-DOMAIN RECOMMENDATIONS BASED ON A HETEROGENEOUS KNOWLEDGE GRAPH

The cross-domain heterogeneous knowledge graph can operate on several meta-paths for recommendations. The possible meta-paths are summarized in Table 7.

TABLE 7. Possible path scenarios in an integrated heterogeneous graph.

Recommendation Type	Script	Meta-path	Hypothesis being tested
Recommendation of similar skills from the EP and vacancies	1	CUA->Skill	The skills of the vacancy are semantically close to a subset of the skills from the EP or the user.
Job recommendation	2	CUA->Skill/Requirement/Responsibility<-Vacancy	A required vacancy is displayed if a subset of the skills, requirements, or responsibilities of the vacancy is semantically close to a subset of the skills from the EP/user.
Recommendation of undeveloped skills	3	Vacancy->Skill	A required skill is displayed if it is related to the vacancy of interest.
	4	CUA->Skill/Requirement/Responsibility<-Vacancy->Skill	An undeveloped job skill is displayed if it includes a subset of skills that are semantically close to the skills from the EP/user
Course recommendations	5	Vacancy->Skill/Requirement/Responsibility->MSkill->MCourse	A MOOC is displayed if a subset of its skills is semantically close to a subset of the skills from the vacancy.
	6	CUA->Skill/Requirement/Responsibility<-Vacancy->Skill/Requirement/Responsibility->MSkill->MCourse	A MOOC is displayed if a subset of its related skills is semantically close to a subset of the undeveloped skills of the vacancy, provided that the mastered skills of the vacancy are semantically close to the skills from the EP/user.

Let us consider the main mechanisms of recommendation scenarios for the resulting heterogeneous knowledge graph using the Cypher query language.

A. SCENARIO 1

The EP manager analyzes the compatibility of the EP with the labor market. The query outputs a ranked list of semantically similar job skills to the skills of the EP/user.

Cypher Query 1.

```
MATCH (c:CUA)-[r:IS_SIMILAR_TO]->(related:Skill)
WITH c, related.name AS RelatedSkill, r.cos_sim AS SimilarityStrength
ORDER BY SimilarityStrength DESC, c.description
RETURN c.description AS CUADescription, collect({Skill: RelatedSkill, Similarity: SimilarityStrength}) AS RelatedSkillsWithSimilarity
```

The first line of the query identifies all IS_SIMILAR_TO relationships between CUA and Skill, where each relationship has a cosine similarity property called cos_sim. The second line prepares the data for aggregation, retaining the CUA description, the name of the related skill, and the strength of the semantic similarity. The third line sorts the data, first by descending cosine similarity to ensure that the most similar skills appear at the top of the aggregated list, and then by the CUA description. The fourth line returns the description of each CUA and aggregates the related skills and their cosine similarities into a list of dictionaries. Each dictionary contains the skill name and its corresponding cosine similarity.

B. SCENARIO 2

A graduate or specialist is planning a career path. This query identifies which job openings include the greatest number of acquired skills that are semantically similar to the user's skills, and which specific acquired skills are required for each job opening.

Cypher Query 2.

```
MATCH (c:CUA)-[r:IS_SIMILAR_TO]->(related:Skill)
WITH c, related, r.cos_sim AS SimilarityStrength
ORDER BY SimilarityStrength DESC
WITH c, collect({Skill: related.name, Similarity: SimilarityStrength}) AS RelatedSkillsWithSimilarity
UNWIND RelatedSkillsWithSimilarity AS SkillData
MATCH (related:Skill {name: SkillData.Skill})
<-[:REQUIRE_SKILL]-(v:Vacancy)
WITH v, collect(DISTINCT related.name) AS RelatedSkills
WITH v, RelatedSkills, size(RelatedSkills) AS NumberOfRelatedSkills
ORDER BY NumberOfRelatedSkills DESC
RETURN v.title AS VacancyTitle, NumberOfRelatedSkills, RelatedSkills
```

The process begins by identifying IS_SIMILAR_TO relationships between CUA and Skill nodes. These relationships indicate which skills (Skill) are associated with specific conceptual units (CUA). The process continues by passing CUA nodes (c), related Skill nodes (related), and the cosine similarity (cos_sim) of each relationship as SimilarityStrength for further analysis. The identified relationships are sorted in descending order of cosine similarity, ensuring that the most relevant relationships are processed first. For each CUA, information about each associated skill and its similarity is grouped into a structured list (RelatedSkillsWithSimilarity). This list is then expanded into a stream of individual elements (SkillData), allowing for the separate processing of each skill. Subsequently, the system searches for job vacancies (Vacancy) that require each of the processed skills. This approach enables the identification of job vacancies associated with each skill. For each vacancy (v), the unique skill names (related.name) required for that vacancy are collected. The total number of unique skills required for each vacancy is calculated as NumberOfRelatedSkills, determined by the size of the RelatedSkills list. The vacancies are then sorted in descending order based on the number of unique required skills, prioritizing vacancies with the highest diversity of required skills. The final output includes the vacancy title (VacancyTitle), the number of required skills (NumberOfRelatedSkills), and the list of these skills (RelatedSkills). This approach provides valuable insights into which job vacancies require the most acquired skills and identifies the specific skills demanded for each vacancy. Such information is beneficial for labor market analysis and career planning.

C. SCENARIO 3

A graduate, educator, program manager, or specialist analyzes the labor market to identify key trends and demands. The query generates a list of the most in-demand skills required for the target groups of job vacancies, providing valuable insights for career planning, and curriculum development.

Cypher Query 3.

```
:params {vacancies: ["разработчик", "дизайнер"]}
MATCH (v:Vacancy)-[:REQUIRE_SKILL]->(s:Skill)
WHERE ANY(keyword IN $vacancies WHERE
toLower(v.title) CONTAINS toLower(keyword))
WITH toLower(s.name) AS Skill, COUNT(*) AS MentionCount
ORDER BY MentionCount DESC
LIMIT 30
RETURN Skill, MentionCount
```

The parameters accept a list of job titles (e.g., "developer" and "designer") and utilize them in the request to filter relevant vacancies. These parameters are passed into the query to refine the selection of job vacancies. The relationship between vacancies (v:Vacancy) and skills (s:Skill) is established through the:REQUIRE_SKILL link, indicating that specific skills are required for each vacancy. Vacancies are filtered so that their titles (v.title) contain any of the specified keywords from the parameters ($vacancies). To ensure case-insensitive matching, the toLower() function is applied, allowing for consistent string comparison regardless of capitalization. The results are grouped by skill names (s.name), converted to lowercase, and the total number of mentions for

each skill (Mention Count) is calculated across the filtered vacancies. The results are sorted in descending order of Mention Count, prioritizing the most frequently required skills. To maintain focus on the most in-demand skills, the output is limited to the top 30 results. The query returns the names of the most popular skills (Skill) and their corresponding Mention Count among the vacancies that match the specified job title keywords.

D. SCENARIO 4

The EP manager develops the Educational Program (EP) in alignment with a targeted group of job vacancies. This process includes generating a recommended list of underdeveloped skills or user skills that are essential for the selected vacancies. The identification of these skill gaps allows for the enhancement of the EP to better align with labor market demands and improve graduate employability.

Cypher Query 4.

```
MATCH (c:CUA)-[r:IS_SIMILAR_TO]->(related:Skill)
WITH c, related, r.cos_sim AS SimilarityStrength
ORDER BY SimilarityStrength DESC
WITH c, collect({Skill: related.name, Similarity: SimilarityStrength}) AS RelatedSkillsWithSimilarity
UNWIND RelatedSkillsWithSimilarity AS SkillData
MATCH (related:Skill {name: SkillData.Skill})
<-[:REQUIRE_SKILL]-(v:Vacancy)
WITH v, collect(DISTINCT related.name) AS RelatedSkills
WITH v, RelatedSkills, size(RelatedSkills) AS NumberOfRelatedSkills
MATCH (v)-[:REQUIRE_SKILL]->(allSkills:Skill)
WITH v, RelatedSkills, NumberOfRelatedSkills, collect(DISTINCT allSkills.name) AS AllSkills
WITH v, RelatedSkills, NumberOfRelatedSkills, AllSkills, size(AllSkills) AS TotalNumberOfSkills,
  [skill IN AllSkills WHERE NOT skill IN RelatedSkills] AS UnrelatedSkills
WITH v, RelatedSkills, NumberOfRelatedSkills, AllSkills, TotalNumberOfSkills, UnrelatedSkills, size(UnrelatedSkills) AS NumberOfUnrelatedSkills
ORDER BY NumberOfRelatedSkills DESC
RETURN v.title AS VacancyTitle, NumberOfRelatedSkills, RelatedSkills,
NumberOfUnrelatedSkills, UnrelatedSkills,
TotalNumberOfSkills, AllSkills
```

At the initial stage, the query identifies nodes of type CUA that are connected to Skill nodes through IS_SIMILAR_TO relationships. The parameter r.cos_sim represents the cosine similarity between CUA and Skill nodes. The results are sorted in descending order based on SimilarityStrength (cosine similarity), prioritizing the most relevant relationships. A collection of dictionaries is then created, where each dictionary contains the name of a skill and its corresponding similarity to CUA. This collection is expanded, presenting each ''skill-similarity'' pair as a separate line item. The query also identifies job vacancies that require skills previously linked to CUA. For each vacancy, a list of unique related skills is generated, capturing all required skills that are associated with CUA. Additionally, a complete list of all required skills for each vacancy is compiled. From this comprehensive list, the system identifies unrelated skills, which are defined as those that are not part of the list of related skills. The total number of unrelated skills is then calculated for each vacancy. Finally, the vacancies are sorted in descending order based on the number of related skills. The output includes the job title, the number and list of related skills, the number and list of unrelated skills, the total number and the full list of skills.

E. SCENARIO 5

A graduate or specialist seeking retraining opportunities can utilize the system to receive tailored recommendations. The system suggests a list of courses aimed at developing the skills required for targeted groups of job vacancies.

Cypher Query 5.

```
:params {vacancies:["разработчик"]}
MATCH (v:Vacancy)-[:REQUIRE_SKILL]->(s:Skill)
WHERE ANY(keyword IN $vacancies WHERE
toLower(v.title) CONTAINS toLower(keyword))
WITH toLower(s.name) AS VSkill, COUNT(*) AS MentionCount
ORDER BY MentionCount DESC
LIMIT 30
WITH collect(DISTINCT VSkill) AS TopSkills
MATCH (skill:Skill)-[:IS_SIMILAR_TO2]-
>(mskill:MSkill)
WHERE toLower(skill.name) IN TopSkills
MATCH (mskill)-[:DEVELOPED_BY]-
>(mcourse:MCourse)
WITH mcourse, collect(DISTINCT mskill.name) AS MatchedSkills, TopSkills, COUNT(DISTINCT mskill.name) AS NumberOfMatchedSkills
WHERE NumberOfMatchedSkills > 0
RETURN mcourse.title AS MCourseTitle,
  MatchedSkills,
  NumberOfMatchedSkills,
  TopSkills
ORDER BY NumberOfMatchedSkills DESC
LIMIT 30
```

This Cypher query is designed to analyze job vacancies and identify courses that teach the skills required for these vacancies, taking into account the semantic proximity of skills. The query is divided into several stages of data processing to ensure accuracy and relevance. Firstly, ehe query identifies all skills (s) required for job vacancies (v). It filters vacancies to include only those whose titles contain any of the keywords specified in the $vacancies parameter. To ensure case-insensitive matching, both vacancy titles and skill names are converted to lowercase. The system counts the number of mentions for each skill across the filtered vacancies. Skills are then sorted in descending order of frequency and the top

30 most frequently mentioned skills are selected. The names of these top 30 skills are stored in the TopSkills list.

The query identifies semantic links (IS_SIMILAR_TO2) between ordinary skills (Skill) and course skills (MSkill). These links are filtered so that only skills whose names appear in the TopSkills list are included. The query identifies courses (MCourse) that develop the skills from the MSkill set. For each course, it collects the unique names of the course skills and counts the total number of matching skills. Courses that do not have any matching skills with the TopSkills list are excluded from the results.

The query returns the course name, a list of matched skills, the number of matched skills, and a list of the top 30 required skills. The results are sorted in descending order based on the number of matched skills. To maintain focus on the most relevant courses, the output is limited to the top 30 courses.

F. SCENARIO 6

A graduate or specialist seeking professional development opportunities, or an educational program (EP) instructor designing a course aligned with a specific group of job vacancies, can utilize the system for targeted recommendations. The system identifies and suggests a list of courses specifically focused on addressing underdeveloped skills within the EP or user's skill set.

Cypher Query 6.

```
MATCH (c:CUA)-[r:IS_SIMILAR_TO]->(related:Skill)
WITH c, related, r.cos_sim AS SimilarityStrength
ORDER BY SimilarityStrength DESC
WITH c, collect({Skill: related.name, Similarity: SimilarityStrength}) AS RelatedSkillsWithSimilarity
UNWIND RelatedSkillsWithSimilarity AS SkillData
MATCH (related:Skill {name: SkillData.Skill})
<-[:REQUIRE_SKILL]-(v:Vacancy)
WITH v, collect(DISTINCT related.name) AS RelatedSkills
WITH v, RelatedSkills, size(RelatedSkills) AS NumberOfUniqueRelatedSkills
MATCH (v)-[:REQUIRE_SKILL]->(allSkills:Skill)
WITH v, RelatedSkills, NumberOfUniqueRelatedSkills,
collect(DISTINCT allSkills.name) AS AllSkills
WITH v, RelatedSkills, NumberOfUniqueRelatedSkills, AllSkills,
size(AllSkills) AS TotalNumberOfSkills,
  [skill IN AllSkills WHERE NOT skill IN RelatedSkills]
AS UnrelatedSkills
WITH v, RelatedSkills, NumberOfUniqueRelatedSkills, AllSkills,
TotalNumberOfSkills, UnrelatedSkills, size(UnrelatedSkills)
AS NumberOfUnrelatedSkills
ORDER BY NumberOfUnrelatedSkills DESC
MATCH (unrelated:Skill)-[:IS_SIMILAR_TO2]
->(mskill:MSkill)-[:DEVELOPED_BY]
->(mcourse:MCourse)
WHERE unrelated.name IN UnrelatedSkills
WITH v, NumberOfUnrelatedSkills, UnrelatedSkills,
mcourse, collect(DISTINCT mskill.name) AS RelatedMSkills
ORDER BY size(RelatedMSkills) DESC
RETURN v.title AS VacancyTitle, NumberOfUnrelatedSkills,
UnrelatedSkills, mcourse.title AS MOOCTitle,
size(RelatedMSkills) AS CountOfRelatedMSkills, RelatedMSkills
```

This query identifies MOOCs that develop underdeveloped skills closely related to those required for specific vacancies but missing in potential candidates. The process begins by finding connections between CUA nodes and skills, where an IS_SIMILAR_TO relationship indicates semantic proximity. These relationships are sorted in descending order of cosine similarity. The related learned skills and their similarities are aggregated into a collection and expanded for further processing. Next, the query identifies job vacancies that require these related mastered skills and compiles a list of unique related mastered skills for each vacancy. It also retrieves the complete set of skills required for the vacancy. By comparing these lists, the query generates a list of ''unrelated undeveloped skills'' — skills required for the job but not present in the list of related mastered skills.

The query then identifies MOOCs that develop skills similar to these unrelated undeveloped job skills. It specifies that an unrelated undeveloped skill must be present in the vacancy's list of unrelated skills. The skills developed by MOOCs that relate to these undeveloped job skills are collected. MOOCs are subsequently sorted by the number of relevant skills they address. Finally, the query returns information about the vacancies, the number of unrelated undeveloped skills, their corresponding lists, as well as the MOOCs and the number of skills they cover.

This query supports HR professionals and educational institutions in identifying educational needs based on labor market demands and facilitates the development of targeted training programs to bridge ≪skill gaps≫.

Thus, the submitted Cypher queries demonstrate a robust approach to data analysis and management within heterogeneous knowledge graphs, enabling a comprehensive understanding of labor market dynamics and training needs. By employing a structured and semantic method to link skills, job vacancies, and educational courses, it becomes possible not only to identify the most in-demand skills and professions but also to pinpoint key areas for training and professional development. These queries support educational institutions and HR specialists in enhancing educational programs and selecting courses that most effectively align with current market demands. This approach increases the employability of graduates by equipping them with the skills most sought after by employers, thereby facilitating successful career outcomes.

IV. RESULTS AND DISCUSSIONS

Queries from the first to the sixth scenario were executed. Queries 1, 2, 4, 5, and 6 are suitable for evaluating

recommendations. Three experts assessed the relevance of the recommendations for five scenarios, and the results of this assessment were tabulated for each query. Following the methodology outlined in [19], we evaluated Precision at K (P@K). (P@K) is the level of false positives, which is an accuracy calculated taking into account only a subset of our recommendations from rank 1 to K. Let's set K to 3 and 5. (P@K) is calculated as the ratio of the sum of all positive relevant recommendations to the total number of k, according to the Equation (2):

$$\mathrm{P@}K\ (q) = \frac{\sum_{k=1}^{K} acc_q(k)}{K}, \tag{2}$$

where $acc_q(k)$is a binary indicator function that returns 1 if the k-th recommendation is relevant for the q-th query, and 0 otherwise. Accuracy is important in our experiments, without regard to order.

Let's consider examples of recommendations based on real data.

Scenario 1: The query generates a ranked list of semantically similar job skills corresponding to the skills of the OP/user. We evaluated three and four recommendations for mastered skills, considering ranks up to 3 and 5, respectively, as presented in Table 8.

TABLE 8. Results of query 1 with a score of P@3, P@5.

Skills from the Curriculum	Recommended job similar skills					P@K
php	php	php4	php5			1
sql	sql	основы sql	и база данных sql			1
ответственность	ответственность	высокая ответственность	ответственность и исполнительность			1
английский	английский	english	английский язык	english language	разговорный английский	1
отладка тестирование	отладка тестирование	смр проведение тестирований	mbedos нагрузочное тестирование	nestjs тест-кейсы	сапр функциональное тестирование	0.2 1
общие понятия информационной безопасности	информационная безопасность	information security	внедрение систем информационной безопасности	аудит информационной безопасности	технические средства информационной защиты	1

Scenario 2: The query identifies job vacancies that include the largest number of mastered skills semantically related to the skills of the EP/user. For each vacancy, the query outputs the mastered skills that match. Table 9 presents the top five vacancies for the EP ≪Information Systems≫.

TABLE 9. The first five vacancies with a list of skills similar to the survey.

Job title	Number of similar skills	List of mastered, similar skills with Curriculum	P@5
Графический дизайнер	16	лендинг, photoshop, дизайн, design, реклама, баннерная реклама, интернет-реклама, illustrator, adobe illustrator, ui, графика, инфографика, иллюстрация, графический дизайн, иллюстрирование, обработка изображений	1
Аналитик-Данных	14	sql, английский язык, автоматизация процессов, python, математический анализ, 1с, базы данных, аналитические исследования, аналитика, сводные таблицы, работа с базами данных, сбор и анализ информации, умение работать в коллективе, умение принимать решения	1
Project Engineer	14	английский язык, проектная деятельность, проектирование, первичные документы, ввод в эксплуатацию, проектный менеджмент, project management, ведение документации, ms project, разработка проектной документации, составление сметно-проектной документации, работа с проектной документацией, проектная документация, работа с документами	1
Системный администратор	13	html, html5, sql, php, php5, коммуникабельность, internet, java, css, 1с программирование, информационные технологии, информационные системы, разработка мобильных приложений	1
3D Motion designer / Моушн дизайнер	13	adobe photoshop, дизайн, 3d, 3d-графика, анимация, видеомонтаж, видеосъемка, обработка видео, монтаж видео, инфографика, работа в команде, графический дизайн, разработка концепции	1

Scenario 3: The query outputs the number of the most popular skills required for specific job groups of interest. As an example, a query was executed for job groups containing the term "developer." Table 10 presents the first ten rows of the results, highlighting the most in-demand skills for these roles.

TABLE 10. The top ten skills according to the ≪Developer≫ job group filter.

Skill	Number of mentions in vacancies
git	139
javascript	114
sql	101
php	96
postgresql	84
html	82
mysql	75
ооп	61
css	58

Scenario 4: The query generates a list of undeveloped skills of the EP/user that are required for vacancies of interest. Table 11 presents the undeveloped skills for the top five vacancies, sorted according to the number of mastered skills.

TABLE 11. The first five vacancies with a list of undeveloped job skills.

Job title	Number of undeveloped skills	List of undeveloped skills vacancies	P@5
Графический дизайнер	14	web-дизайн, ms powerpoint, smm, presentation skills, графические редакторы, illustration, web design, ux, figma, дизайн полиграфии, веб-дизайн, usability, полиграфический дизайн, подготовка презентаций	0,6
Аналитик-Данных	16	ms sql server, исследования рынка, деловая коммуникация, конкурентная аналитика, swot анализ, прогнозирование, ms access, аналитическое мышление, бизнес-анализ, data analysis, аналитические способности, анализ данных, анализ бизнес показателей, анализ финансовых показателей, ms sql, работа с большим объемом информации	0,6
Project Engineer	15	leadership skills, техническая документация, инженерные системы, negotiation skills, организаторские навыки, техническая экспертиза, business process management, risk management, sap project, sales management, teambuilding, team management, procurement, business planning, чтение чертежей	1
Системный администратор	12	разработка технических заданий, 1с: предприятие 8, mysql, разработка инструкций, javascript, техническая поддержка, windows 7, администрирование сетевого оборудования, настройка сетевых подключений, web application development, разработка регламентов, web design	0,4
3D Motion designer / Моушн дизайнер	10	coreldraw, adobe premier pro, adobe after effect, креативность, motion design, чувство стиля, blender 3d, cinema 4d, adobe premier, уверенный пользователь пк	0,8

Scenario 5: The query generates a list of courses that enable users to master the skills required for specific vacancy groups of interest. As an example, a query was executed for job groups containing the term ''developer.'' Table 12 presents the first five rows of the results, showcasing the courses that offer training for the relevant skills needed for these roles.

Scenario 6: The query outputs a list of courses that address the undeveloped skills from the Curriculum/user required for vacancies, along with a list of undeveloped related skills provided by the MOOCs. Table 13 presents the first five rows of the results.

TABLE 12. The first five courses are based on the ≪Developer≫ job group filter.

MOOC Header	A list of matching skills vacancies and MOOCs	The number of matching skills	P@K
Responsive Website Basics: Code with HTML, CSS, and JavaScript	computer programming, html, javascript, html and css, programming principles, jquery	6	1
Programming Foundations with JavaScript, HTML and CSS	computer programming, html, javascript, html and css, programming principles, html5	6	1
Introduction to Web Development	computer programming, html, html and css, javascript, css code	5	1
Scripting with Python and SQL for Data Engineering	python programming, computer programming, programming principles, sql	4	1
Using Python to Access Web Data	python programming, computer programming, html and css, programming principles	4	1

The accuracy of the recommendation system model was evaluated using the Precision@K (P@K) metric, with K set to 3 and 5. This approach enabled the assessment of the system's accuracy based on the top three and top five recommendations, respectively.

In Scenario 1, where recommendations of semantically similar skills were analyzed, the P@3 metric demonstrated high accuracy, indicating that the first three recommendations were highly relevant in most cases. Similarly, the P@5 metric maintained a satisfactory level of accuracy, further confirming the model's effectiveness in generating relevant recommendations within the top five positions.

The analysis of the results for scenarios 2, 4, 5, and 6 demonstrated positive accuracy estimates. In Scenario 2, the system effectively identified vacancies with the maximum number of similar skills, aligning the user's acquired skills with labor market requirements. Scenario 4 highlighted the need for clustering to better structure and group undeveloped skills, enhancing recommendation precision. In Scenario 5, which focused on recommending courses to master required skills, the model showed good performance, with P@5 demonstrating high accuracy, confirming its ability to suggest relevant courses aligned with market demands. Similarly, Scenario 6, which aimed to recommend courses for undeveloped skills, also produced positive results with P@5.

The conclusions on the evaluation of the model's accuracy indicate its effectiveness and relevance in providing educational and professional path recommendations, especially for the top positions in the recommendation list. The results highlight the system's ability to offer precise and meaningful guidance, with some scenarios emphasizing the need for skill clustering to enhance recommendation quality. These findings confirm the validity of the developed methodology for comparing skills with labor market requirements,

TABLE 13. The result of the query for the sixth scenario.

Job title	Number of undeveloped skills	List of undeveloped job skills	Name of recommended MOOC	Number of related skills	List of undeveloped skills from MOOC	P@5
Software Engineer (onsite)	18	.net framework, software development, react, agile, oracle pl sql, data analysis, system & technology integration, power apps model driven, ms sql, teamplayer, angular, analytical skills, azure sql, asp.net framework, data visualization, data modeling, azure technologies, devops	Exploratory Data Analysis with MATLAB	8	data analysis, data analysis software, exploratory data analysis, software visualization, statistical visualization, interactive data visualization, data visualization software, data visualization	0,8
AI/ML Technical Lead / Senior Developer / Data Scientist	14	tensorflow, big data, pandas, flask, deep learning, pytorch, machine learning, ai, nlp, data analysis, computer vision, data science, data mining, ml	Machine Learning Foundations: A Case Study Approach	8	deep learning, machine learning concepts, statistical machine learning, machine learning software, applied machine learning, machine learning algorithms, machine learning, data analysis	1
AI/ML Technical Lead / Senior Developer / Data Scientist	14	tensorflow, big data, pandas, flask, deep learning, pytorch, machine learning, ai, nlp, data analysis, computer vision, data science, data mining, ml	Machine Learning With Big Data	8	machine learning concepts, statistical machine learning, machine learning algorithms, machine learning, data analysis, data analysis software, exploratory data analysis, data mining	1
AI/ML Technical Lead / Senior Developer / Data Scientist	14	tensorflow, big data, pandas, flask, deep learning, pytorch, machine learning, ai, nlp, data analysis, computer vision, data science, data mining, ml	Computational Social Science Methods	7	big data, statistical machine learning, applied machine learning, machine learning algorithms, machine learning, data analysis, data model	1
AI/ML Technical Lead / Senior Developer / Data Scientist	14	tensorflow, big data, pandas, flask, deep learning, pytorch, machine learning, ai, nlp, data analysis, computer vision, data science, data mining, ml	Introduction to Artificial Intelligence (AI)	7	deep learning, machine learning software, applied machine learning, machine learning algorithms, machine learning, data science, computer vision	1

representing a significant contribution to the advancement of recommendation systems in education and professional development.

The integration of the three knowledge graphs at the skill level offers substantial benefits for faculty, students, alumni, and staff in the educational domain. Faculty can leverage data on current skill and competency requirements extracted from job vacancies to adapt courses and curricula to align with in-demand knowledge. The incorporation of MOOC data enables the introduction of modern teaching methods and tools, enhancing the interactivity and effectiveness of the learning process. Students receive personalized course recommendations that align with their existing skills and career aspirations, allowing them to use their time and resources more efficiently by focusing on the most relevant learning materials. Job requirements analysis enables students to identify knowledge gaps and address them, thereby increasing their competitiveness in the labor market. Graduates benefit from job recommendations that match their skills and competencies, along with suggestions for additional courses to further enhance their qualifications. IT employees receive recommendations for MOOC courses and educational programs that help them stay current with emerging technologies and industry trends. This system supports students, graduates, and employees in identifying the most promising vacancies and career paths while offering tailored educational resources to achieve their goals. Overall, the integration of educational programs, job vacancies, and MOOC courses into a unified knowledge system provides a more precise and relevant recommendation space, fostering continuous learning and career development in the IT sector.

Our methodology is compared with other methods and approaches, and Table 14 is presented to outline the data sources for skills and competencies relevant to similar jobs.

The table presents a comparison of various authors and their contributions to the integration of data sources and the analysis of skills and concepts, including the use of knowledge graphs. Several key aspects distinguish our model from others.

The considered models primarily integrate different domains within the same language, most often English, which limits their applicability in multilingual contexts. Unlike similar approaches, our methodology addresses the lack of multilingual integration between the three data sources by exploring connections along the program-skill-skill-vacancy-skill-skill-course route. This is achieved through the analysis of mixed phrases in Russian and English from various domains using a pre-trained transformer model. The model performs paraphrasing tasks with the paraphrase-multilingual-mpnet-base-v2 architecture to facilitate multilingual processing.

This methodology, which relies on heterogeneous knowledge graphs and machine learning, can be effectively transferred to other contexts with multilingual challenges, such as countries with multilingual education systems, including Canada, India, and Switzerland. The use of multilingual

TABLE 14. Comparison or integration of skills from different data sources.

Authors	Scientific publications	University courses	MOOCs	Classifiers SOC, ESCO, DWA	Vacancies	Resumes	Language	Knowledge graphs	Language models	Similarity metrics
Katy Börner [24], 2018	+		+		+		English	+	MaxMatch	Kulback-Leibler Distance
Hung Kim Chau [25], 2024		+			+		English	-	DistilBERT, all-mpnet-base-v2	Cosine proximity
Jibril Frej [35], 2018			+	+	+	+	English	-	JobBERT	Cosine proximity
Guoqing Zhu [23], 2022		+	+		+		English	+	BM25, Word2Vec, Bert	Cosine proximity
Albert Weichselbraun [22], 2022		+		+			English	+	Distilbert-base-german-cased	-
Yousra Fettach [21], 2022			+		+		English	+	-	-
Anna Giabelli [19], 2020				+	+		English, German, French	+	FastText	Cosine proximity
Our model		+	+		+		Russian, English	+	Paraphrase-multilingual-mpnet-base-v2	Cosine proximity

transformers, such as paraphrase-multilingual-mpnet-base-v2, enables the integration of data from educational programs, labor markets, and online courses in different languages, facilitating accurate and personalized recommendations. In the context of globalization, the methodology proves particularly valuable for companies operating in international markets, where a multilingual approach is essential. As a result, this system supports retraining and upskilling, aligning with global professional requirements.

To evaluate the effectiveness of the recommendation system, feedback was collected from graduates and students who completed the proposed MOOC courses. As part of this process, questionnaires were administered, allowing users to share their perspectives on the usefulness of the courses for their professional development. Participants were asked to rate their satisfaction with the quality of recommendations and their relevance to current career goals and professional needs. Pilot tests demonstrated that the system effectively addresses the challenge of aligning skills with employer requirements.

The use of personal data, such as student performance and career-related information, in knowledge graphs raises significant ethical concerns, particularly with regard to privacy. To ensure data protection, it is recommended to implement measures such as data anonymization, minimization of collected information, encryption during storage and transmission, and access restrictions for authorized users only. An essential aspect of this process is obtaining informed consent from users for the use of their data. These measures aim to balance the effectiveness of the recommendation system with the protection of users' rights to privacy and personal data security.

The knowledge graph model can be personalized for individual learners based on their current skills, career goals, and learning preferences. For instance, course recommendations can be tailored according to the skills already mastered, with suggestions for new skills aligned with the learner's career objectives. Personal learning preferences can also be considered through the use of filters, enabling the selection of courses with varying levels of difficulty and diverse formats, such as video lectures or hands-on assignments, depending on the learner's needs. This approach facilitates a flexible and personalized system that supports learners in achieving their educational and career goals in an efficient and targeted manner.

The visualization of knowledge graphs plays a crucial role in facilitating the understanding of relationships between skills, courses, and job requirements for universities and students. Open-source libraries such as Neovis.js, Cytoscape.js, D3.js, and Sigma.js are well-suited for the seamless integration of Neo4j and graph visualization into web applications. These tools enable the development of user-friendly interfaces that cater to non-technical audiences.

In addition to analyzing the skills described in related works, the model also examines job vacancies. The methodology takes into account certain requirements and responsibilities specified in vacancies, integrating them with prior educational experience and MOOC course recommendations.

The main limitations of the methodological model are associated with the quality and completeness of data sources. The accuracy of the analysis and conclusions is highly dependent on the quality, completeness, and relevance of the data used. Errors or omissions in the data, as well as incomplete information on skills, may result in inaccurate recommendations. In rapidly evolving industries, such as IT, skills data can become outdated quickly, necessitating continuous updates to maintain the relevance of the analysis. To address this limitation, it is essential to implement modules for the regular

updating and verification of data, as well as to rely on reliable and up-to-date information sources.

The integration of three large knowledge graphs — educational programs, job vacancies, and MOOCs — requires substantial computational resources to calculate the cosine similarity between the embeddings of skills, requirements, and responsibilities. As the number of nodes increases, the computational load grows accordingly. To optimize performance, parallel processing and incremental update methods can be employed, enabling the integration of new data without the need for a complete recalculation of links. This approach ensures real-time responsiveness of the system. In graph databases, such as Neo4j, complex query execution is accelerated up to 40% compared to single-threaded systems. Incremental updates enable addition of new data into the graph without rebuilding the structure, hence considerably reducing processing time. This brings update times for graphs with millions of nodes down from many hours to mere minutes. These methods also optimize the performance and thus reduce the computational overhead. According to various studies, incremental processing decreases CPU usage by as much as 30-50% compared to traditional full-rebuild approaches. This will be particularly important for systems that have to handle very frequent updates, such as adding 1,000 new nodes and connections each hour, without sacrificing performance in terms of responsiveness.

The performance of the model depends on the specifics of the domain. The methodology can be optimized for a particular field or industry, which limits its applicability in other areas without additional adaptation. Semantic proximity models trained on data from one domain may demonstrate reduced effectiveness when applied to data from another domain due to differences in terminology and contextual usage. To address this challenge, the paraphrase-multilingual-mpnet-base-v2 model was further trained on domain-specific data, specifically IT job vacancies from hh.kz. This approach allows for the consideration of professional terminology and industry-specific jargon, thereby adapting the model to frequently used terms in the IT sector. Future research should focus on experiments involving the use of modern language models, such as GPT, for the task of skill extraction. Additionally, it is necessary to explore the potential of specialized models trained on technical data, such as CodeBERT, GraphCodeBERT, CuBERT, DevBERT, and CodeT5. These models are capable of effectively handling the industry-specific jargon present in job vacancies. For other domains, customized specialized models may be required to achieve similar performance.

Another key limitation of the model is its scalability and performance. The construction and analysis of large knowledge graphs can be computationally intensive, particularly as the volume of data increases. This poses a challenge for ensuring system scalability in real-time applications. Queries to the graph database can become slow with increasing graph size, which requires careful optimization and the use of additional caching or indexing technologies is possible. To overcome this limitation, it is necessary to implement scalable architectures using cloud-based infrastructures and distributed computing frameworks such as Apache Spark or Hadoop. Additionally, incremental graph updates will be adopted to avoid full recomputations, thus enhancing the system's responsiveness in real time. Empirical validation of these methods will be supported by computational performance metrics (processing time, throughput) on datasets of varying sizes.

V. CONCLUSION

As part of this study, a modern methodology for applying heterogeneous knowledge graphs has been developed and comprehensively described to create a recommendation system for MOOC courses and job vacancies. The primary objective of this system is to minimize the gap between current labor market requirements and the educational programs offered. The use of knowledge graphs enables not only the structuring and aggregation of large volumes of data but also facilitates a deeper understanding of the interrelationships between professional skills and educational requirements.

The primary advantage of graph data structures lies in their high degree of abstraction and flexibility, enabling the representation of complex networks of relationships. This feature is particularly significant in the context of dynamically evolving labor market requirements and constantly updated educational standards. Systematic modeling and integration of data from various sources, such as university courses, job market vacancies, and open online courses, have established a robust analytical foundation that ensures the relevance and accuracy of the proposed recommendations.

The integration of Sentence Transformers algorithms into the recommendation system has not only expanded the base of potential comparisons but also improved the quality of recommendations by enabling a deeper understanding of the context and semantics of skills. This approach, in turn, results in more accurate and personalized recommendations, thereby supporting effective learning and promoting the professional development of users.

Future development will focus on enhancing the functionality and adaptability of the proposed knowledge graph-based recommendation system. One key direction is the international expansion of the system's capabilities through the integration of data from various global labor markets and educational platforms. This will involve the automatic updating of data sources to maintain the relevance of the knowledge graph, as well as the clustering of skills to improve recommendation accuracy. Another important area will be to refine the recommendation algorithm, which will take into account not only the current skills of users, but also the personal preferences of students, using filters to select courses of varying complexity and formats. This approach will create a flexible and personalized system that will help students more effectively achieve their educational and career goals. This will make recommendations more personalized and useful. Other areas will be effective graph visualization for a

non-technical audience in a web application, as well as graph scaling and real-time processing optimization.

Thus, the proposed methodology not only increases the effectiveness of educational and professional recommendations but also contributes to the formation of adaptive educational programs that are responsive to changes in market requirements. This is a key factor in training qualified specialists who can successfully compete in the labor market.

REFERENCES

[1] World Economic Forum. (2023). *The Future of Jobs Report 2023. World Economic Forum*. [Online]. Available: https://www3.weforum.org/docs/WEF_Future_of_Jobs_2023.pdf

[2] Ministerstvo Truda I Sotsial'noi Zashchity Naseleniia Respubliki Kazakhstan. (Jan. 20, 2024). *Tretii Natsdoklad Po Rynku Truda Vypushchen V Kazakhstane*. [Online]. Available: https://www.gov.kz/memleket/entities/enbek/press/news/details/687443?lang=ru

[3] (2024). *Ministerstvo Truda I Sotsial'noi Zashchity Naseleniia Respubliki Kazakhstan*. Natsional'nyi Doklad O Zaniatosti I Sotsial'noi Zashchite. [Online]. Available: https://erdo.enbek.kz/storage/news_files%D0%9D%D0%B0%D1%86%D0%B4%D0%BE%D0%BA%D0%BB%D0%B0%D0%B4%20%D1%80%D1%83%D1%81%D1%81.pdf

[4] T. Zhao, Y. Huang, S. Yang, Y. Luo, J. Feng, Y. Wang, H. Yuan, K. Pan, K. Li, H. Li, and F. Zhu, "Mathgraph: A knowledge graph for automatically solving mathematical exercises," in *Database Systems for Advanced Applications*, G. Li, J. Yang, J. Gama, J. Natwichai, Y. Tong, Eds., 2019, doi: 10.1007/978-3-030-18576-3_45.

[5] W. Li, H. Zhou, J. Dong, Q. Zhang, Q. Li, G. Baciu, J. Cao, and X. Huang, "Constructing low-redundant and high-accuracy knowledge graphs for education," in *Learning Technologies and Systems*, C. S. González-González, Ed., 2022, doi: 10.1007/978-3-031-33023-0_13.

[6] G. Agrawal, Y. Deng, J. Park, H. Liu, and Y.-C. Chen, "Building knowledge graphs from unstructured texts: Applications and impact analyses in cybersecurity education," *Information*, vol. 13, no. 11, p. 526, Nov. 2022, doi: 10.3390/info13110526.

[7] Y. Su and Y. Zhang, "Automatic construction of subject knowledge graph based on educational big data," in *Proc. 3rd Int. Conf. Big Data Educ.*, Apr. 2020, pp. 30–36, doi: 10.1145/3396452.3396458.

[8] Y. Qin, H. Cao, and L. Xue, "Research and application of knowledge graph in teaching: Take the database course as an example," *J. Phys., Conf.*, vol. 1607, no. 1, Aug. 2020, Art. no. 012127, doi: 10.1088/1742-6596/1607/1/012127.

[9] M. Rizun, "Knowledge graph application in education: A literature review," *Acta Universitatis Lodziensis. Folia Oeconomica*, vol. 3, no. 342, pp. 7–19, Aug. 2019, doi: 10.18778/0208-6018.342.01.

[10] Y. Sun, J. Liang, and P. Niu, "Generation of personalized knowledge graphs based on GCN," *J. Comput. Commun.*, vol. 9, pp. 108–119, Jan. 2021, doi: 10.4236/jcc.2021.99008.

[11] R. Meissner and A. Thor. (2021). *Creation and Utilisation of Domain Specific Knowledge Graphs (DSKG) for E-Learning*. DELFI. [Online]. Available: https://dl.gi.de/server/api/core/bitstreams/b99e4bf0-10aa-4e4e-a0f1-96c1497bf3ee/content

[12] P. Chen, Y. Lu, V. W. Zheng, X. Chen, and B. Yang, "KnowEdu: A system to construct knowledge graph for education," *IEEE Access*, vol. 6, pp. 31553–31563, 2018, doi: 10.1109/ACCESS.2018.2839607.

[13] Q. Ke and J. Lin, "Dynamic generation of knowledge graph supporting STEAM learning theme design," *Appl. Sci.*, vol. 12, no. 21, p. 11001, Oct. 2022, doi: 10.3390/app122111001.

[14] X. Yu, M. Stahr, H. Chen, and R. Yan, "Design and implementation of curriculum system based on knowledge graph," in *Proc. IEEE Int. Conf. Consumer Electron. Comput. Eng. (ICCECE)*, Oct. 2021, pp. 767–770.

[15] I. Aliyu, A. Kana, and S. Aliyu, "Development of knowledge graph for university courses management," *Int. J. Educ. Manage. Eng.*, vol. 10, no. 2, pp. 1–10, Apr. 2020, doi: 10.5815/ijeme.2020.02.01.

[16] V. P. Tel'nov and Y. A. Korovin, "Semantic web and knowledge graphs as an educational technology of personnel training for nuclear power engineering," *Izvestiya Wysshikh Uchebnykh Zawedeniy, Yadernaya Energetika*, vol. 2019, no. 2, pp. 219–229, Jun. 2019, doi: 10.26583/npe.2019.2.19.

[17] L. S. Nair, M. K. Shivani, and S. Jo Cheriyan, "Enabling remote school education using knowledge graphs and deep learning techniques," *Proc. Comput. Sci.*, vol. 215, pp. 618–625, Jan. 2022, doi: 10.1016/j.procs.2022.12.064.

[18] Y. Chi, Y. Qin, R. Song, and H. Xu, "Knowledge graph in smart education: A case study of entrepreneurship scientific publication management," *Sustainability*, vol. 10, no. 4, p. 995, Mar. 2018, doi: 10.3390/su10040995.

[19] A. Giabelli, L. Malandri, F. Mercorio, M. Mezzanzanica, and A. Seveso, "Skills2Job: A recommender system that encodes job offer embeddings on graph databases," *Appl. Soft Comput.*, vol. 101, Mar. 2021, Art. no. 107049, doi: 10.1016/j.asoc.2020.107049.

[20] M. de Groot, J. Schutte, and D. Graus, "Job posting-enriched knowledge graph for skills-based matching," 2021, *arXiv:2109.02554*, doi:

[21] Y. Fettach, M. Ghogho, and B. Benatallah, "Knowledge graphs in education and employability: A survey on applications and techniques," *IEEE Access*, vol. 10, pp. 80174–80183, 2022, doi: 10.1109/ACCESS.2022.3194063.

[22] A. Weichselbraun, R. Waldvogel, A. Fraefel, A. van Schie, and P. Kuntschik, "Building knowledge graphs and recommender systems for suggesting reskilling and upskilling options from the web," *Information*, vol. 13, no. 11, p. 510, Oct. 2022, doi: 10.3390/info13110510.

[23] G. Zhu, Y. Chen, and S. Wang, "Graph-Community-Enabled personalized course-job recommendations with cross-domain data integration," *Sustainability*, vol. 14, no. 12, p. 7439, Jun. 2022, doi: 10.3390/su14127439.

[24] K. Börner, O. Scrivner, M. Gallant, S. Ma, X. Liu, K. Chewning, L. Wu, and J. A. Evans, "Skill discrepancies between research, education, and jobs reveal the critical need to supply soft skills for the data economy," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 50, pp. 12630–12637, Dec. 2018, doi: 10.1073/pnas.1804247115.

[25] H. K. Chau, "Explainable course recommendation: Connecting college education to knowledge and careers through skills," Ph.D. dissertation, Dept. Inform. Networked Syst., School Comput. Inf., Univ. Pittsburgh, Pittsburgh, PA, USA, 2024.

[26] C. Gao, Y. Zheng, N. Li, Y. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, X. He, and Y. Li, "A survey of graph neural networks for recommender systems: Challenges, methods, and directions," *ACM Trans. Recommender Syst.*, vol. 1, no. 1, pp. 1–51, Mar. 2023, doi: 10.1145/3568022.

[27] C. Zhang, Y. Li, S. Chen, S. Fan, and W. Li, "Knowledge graph driven recommendation system algorithm," 2023, *arXiv:2401.10244*.

[28] Toolify AI. (2024). *Revolutionizing Course Recommendations: Graph Neural Networks Vs. Traditional Approaches*. Toolify AI News. [Online]. Available: https://www.toolify.ai/ai-news/revolutionizing-course-recommendations-graph-neural-networks-vs-traditional-approaches-743103

[29] P. Guy. (2024). *Knowledge Graphs for Recommendation Systems: A Deep Dive Into Graph Neural Networks*. Pi Guy Blog. [Online]. Available: https://the-pi-guy.com/blog/knowledge_graphs_for_recommendation_systems_a_deep_dive_into_graph_neural_networks/

[30] Stanford University. (2024). *GNN for Recommender Systems: Lecture Notes*. CS224W: Machine Learning with Graphs. [Online]. Available: https://web.stanford.edu/class/cs224w/slides/05-GNN3.pdf

[31] A. Hogan, E. Blomqvist, M. Cochez, C. D'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C.-N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann, "Knowledge graphs," *ACM Comput. Surveys*, vol. 54, no. 4, pp. 1–37, May 2022, doi: 10.1145/3447772.

[32] V. S. Ramazanova, M. A. Sambetbayeva, and A. Y. Zagorulko, "Obzor i analiz predstavleniy grafov znaniy trudy universiteta," Tech. Rep., 2024, pp. 414–422, doi: 10.52209/1609-1825_2024_1_414.

[33] *IEEE Standard for Framework of Knowledge Graphs*, IEEE Standard 2807-2022, pp. 1–52, doi: 10.1109/IEEESTD.2022.10017167.

[34] G B. Zagorulko, "Methods, and tools for comprehensive support of DSS development in poorly formalized subject areas," Ph.D. dissertation, A. P. Ershov Inst. Inform. Syst., Siberian Branch Russian Acad. Sci., Novosibirsk, Russia, 2020.

[35] J. Frej, A. Dai, S. Montariol, A. Bosselut, and T. Käser, "Course recommender systems need to consider the job market," 2024, *arXiv:2404.10876*, doi:

[36] Z. Shakhbanov. *Kak Ispol'zovat' API Hh.Ru Dlya Parsinga (Sbora) Vakansiy V Oblasti Data Science [How To Use the Hh.ru API for Parsing (collecting) Vacancies in the Field of Data Science*. [Online]. Available: https://shakhbanov.org/kak-ispolzovat-api-hhru-dlya-parsinga-sbora-vakansiy-v-oblasti-data-science/

[37] M. Siddharth. *Web Scraping and Coursera*. [Online]. Available: https://medium.com/analytics-vidhya/web-scraping-and-coursera-8db6af45d83f
[38] A. Mahadan. *Coursera Course Scraping*. [Online]. Available: https://www.kaggle.com/code/azraimohamad/coursera-course-scraping
[39] *SentenceTransformers Documentation (nd)*. [Online]. Available: https://sbert.net/
[40] *Model Card for Paraphrase-Multilingual-Mpnet-Base-V2*. [Online]. Available: https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2
[41] *The Neo4j Graph Data Science Library Manual v2.6*. [Online]. Available: https://neo4j.com/docs/graph-data-science/current/
[42] *Scikit-Learn: Machine Learning in Python*. [Online]. Available: https://scikit-learn.org/stable/

VALIYA RAMAZANOVA received the master's degree in specialty information systems from Al-Farabi Kazakh National University, Almaty, Kazakhstan, in 2013. She is currently pursuing the Ph.D. degree in information systems with L. N. Gumilyov Eurasian National University, Kazakhstan.

She is a Main Engineer with the IT Department, M. Utemisov West Kazakhstan University. She has more than ten publications, one article in SCOPUS base journals, three articles in the journals of the Higher Attestation Commission of the Republic of Kazakhstan, two international conferences, and others. Her research interests include knowledge graphs, machine learning, and NLP.

MADINA SAMBETBAYEVA received the Ph.D. degree in information systems from L. N. Gumilyov Eurasian National University, Astana, Kazakhstan. She is a Project Manager of the Grant Financing Project "Development of the fundamentals of the technology for creating models and software for building distributed information systems to support scientific and educational activities (taking into account the morphology of the Kazakh language)." She is also involved in interstate projects, working together with scientists from Novosibirsk State University and the Institute of Computational Technologies SB RAS. She has over 50 publications, of which one educational-methodical manual, 11 articles in Scopus, 16 articles in the journals of the list of Higher Attestation Commission of the Republic of Kazakhstan and the Russian Federation, and five certificates of official registration of computer programs used in pedagogical and scientific practice. She has an H-index of 5. Her research interests include information systems, information retrieval systems, information retrieval processes, decision support systems, and linguistic support of information systems. She is working on research and development of the fundamentals of the technology for creating models and software for building distributed information systems to support scientific and educational activities, taking into account the morphology of the Kazakh language, identified in the form of electronic libraries that are consistent with international standards and trends in the development of national and international information infrastructure.

SANDUGASH SERIKBAYEVA received the Ph.D. degree in specialty information systems from L. N. Gumilyov Eurasian National University, Astana, Kazakhstan. Her dissertation theme is "Creation of models and technologies for building distributed information systems to support scientific and educational activities." She has more than 30 publications, including one academic book, eight articles in SCOPUS base journals, three articles in Web of Science base, six articles in the journals of the Higher Attestation Commission of the Republic of Kazakhstan, and the Higher Attestation Commission of the Russian Federation. Her Scopus H-index is 4 and Web of Science H-index is 1. Her research interests include distributed information systems, thesaurus, information retrieval, digital library, and ontology.

ZHANNA SADIRMEKOVA received the Ph.D. degree in information systems from L. N. Gumilyov Eurasian National University, Astana, Kazakhstan. She successfully defended her thesis on "Development of technology for integration of information systems to support scientific and educational activities based on metadata and ontological model of the subject area." She is currently pursuing the master's degree in the field of 03.06.01-physics and astronomy with the Federal State Autonomous Educational Institution of Higher Education "Novosibirsk National Research State University." She is an Associate Professor with the Information Systems Department, M. Kh. Dulaty Taraz Regional University, Taraz, Kazakhstan. Her research interests include information technology in education, artificial intelligence, search, digital library, and ontology. She has more than 30 publications, including four articles in Scopus base journals, two articles in Web of Science base, seven articles in journals of the Higher Attestation Commission of the Republic of Kazakhstan; and an H-index of 3.

AIGERIM YERIMBETOVA is currently pursuing the Ph.D. degree in technical sciences. She is an Associate Professor with the Institute of Information and Computational Technologies, CS MSHE RK, Almaty, Kazakhstan. She is engaged in the research and creation of linguistic algorithms for semantic search and retrieval of information and mathematical and logical methods. She is the author of more than 80 articles on computational linguistics and the development of information systems: one textbook, two monographs, 32 articles in the journal SCOPUS, and 20 articles in the journals of the CQAES list. She holds one patent and four certificates of entry in the State Register of Copyright Rights.