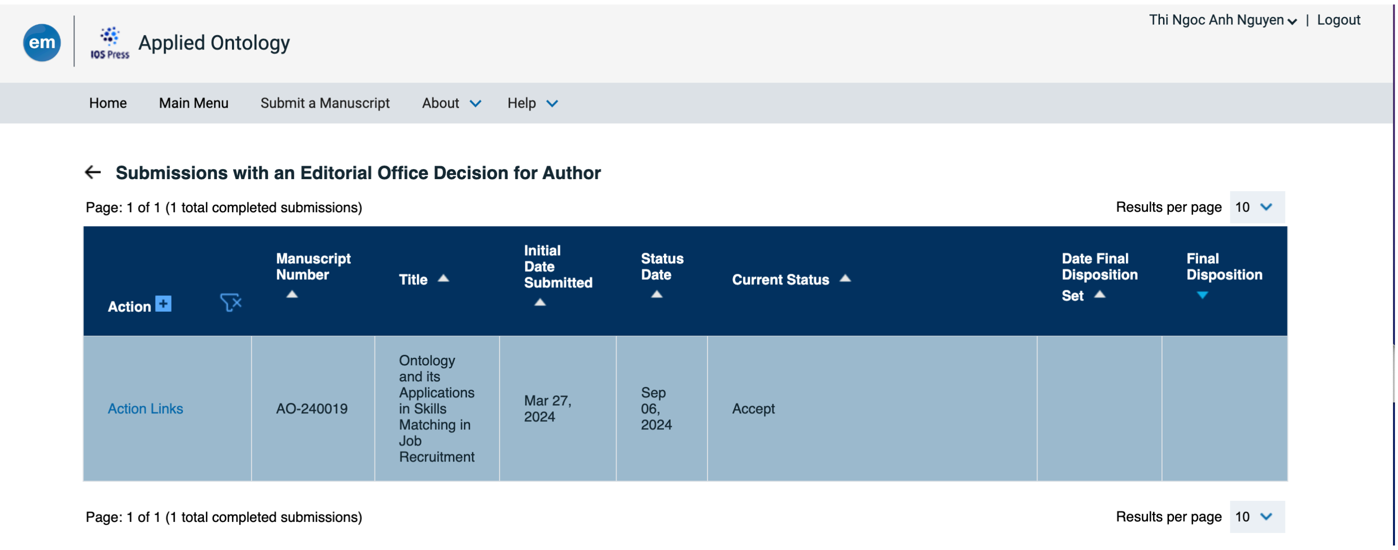


Applied Ontology

← Submissions with an Editorial Office Decision for Author

Page: 1 of 1 (1 total completed submissions)

Results per page 10

Action	Manuscript Number	Title	Initial Date Submitted	Status Date	Current Status	Date Final Disposition Set	Final Disposition
Action Links	AO-240019	Ontology and its Applications in Skills Matching in Job Recruitment	Mar 27, 2024	Sep 06, 2024	Accept		

Page: 1 of 1 (1 total completed submissions)

Results per page 10

Your Submission AO-240019R1

em.ao.0.8de658.13ba050f@editorialmanager.com
<em.ao.0.8de658.13ba050f@editorialmanager.com>
on behalf of
Applied Ontology <em@editorialmanager.com>
Fri 06/09/2024 8:54 PM

To:Nguyen Thi Ngoc Anh <anh.nguyenthingoc@hust.edu.vn>

Ref.: Ms. No. AO-240019R1
Ontology and its Applications in Skills Matching in Job Recruitment
Applied Ontology

Dear Dr Nguyen,

I am pleased to tell you that your work has now been accepted for publication in Applied Ontology. It was accepted on Sep 02, 2024, comments from the Editor and Reviewers can be found below.

Your paper will now be sent to the publisher's typesetting office. Assuming there are no technical errors with the manuscript then, after the paper has been typeset, you will be informed by e-mail and asked to proofread the final copy. At this stage the galley proof of your paper will also be published online in the 'pre press' section of the journal available at https://content.iospress.com/journals/applied-ontology/Pre-press/Pre-press. After final publication in an issue of the journal you will receive a complimentary PDF copy of the final published article which will include the final page numbers.

Should you not receive your proof copy within 6-8 weeks from now, please contact the publisher at editorial@iospress.nl and ensure to mention your manuscript number in the e-mail subject line: AO-240019R1.

Thank you for submitting your work to the journal, we look forward to receiving your future work as well.

With kind regards

Mihai Pomarlan
Assistant Editor
Applied Ontology

Comments from the Editors and Reviewers:

In compliance with data protection regulations, you may request that we remove your personal registration details at any time. (Use the following URL: https://www.editorialmanager.com/ao/login.asp?a=r). Please contact the publication office if you have any questions.

Author Details for Manuscript Number: AO-240019
Ontology and its Applications in Skills Matching in Job Recruitment

Close

Corresponding Author Status

Order	Author Name	Email Address	ORCID Identifier	Academic Degree(s)	Affiliation	Options
7	Thi Ngoc Anh Nguyen	anh.nguyenthingoc@hust.edu.vn	0000-0002-6555-9740		Hanoi University of Technology: Hanoi University of Science and Technology	

Other Author Status

Order △▽	Author Name △▽	Added in Revision △▽	Email Address	ORCID Identifier	Academic Degree(s)	Affiliation △▽	Confirmed? △▽	Options
1	Anh Chi Tuan	R0	chi.ta194232@sis.hust.edu.vn Edit			Faculty of Mathematics and Informatics, Hanoi University of Science and Technology (Hanoi, Vietnam)	Yes Mar 27, 2024	
2	Minh Tuan Dang	R0	tuandm1@cmc.com.vn Edit		PhD	CMC Applied Technology Institute (Hanoi, Vietnam); Posts and Telecommunications Institute of Technology (Hanoi, Vietnam)	No Response	Resend Letter
3	Hai Nam Do	R0	dhnam3@cmc.com.vn Edit			CMC Applied Technology Institute (Hanoi, Vietnam)	No Response	Resend Letter
4	Vijender Kumar Solanki	R0	spesinfo@yahoo.com Edit			Department of Computer Science Engineering, CMR Institute of Technology (Hyderabad, India)	Yes Mar 27, 2024	
5	Jorge Torres	R0	jorge.torres@unir.net Edit			International University of La Rioja (La Rioja, Spain)	No Response	Resend Letter
6	Ruben Gonzalez Crespo	R0	ruben.gonzalez@unir.net Edit		PhD	International University of La Rioja (La Rioja, Spain)	Yes Mar 27, 2024	

To update the Co-Author's e-mail address that is associated with the submission, please click on the ‘Edit’ link. To save your changes please click on the 'Save' link.

To allow a Co-Author to change their responses to the questions they may have already answered (if any), use the 'Resend Letter' link. To request verification from all Co-Authors who have not confirmed their authorship and present the most recent questionnaire (if any) to them, use the 'Send Letters' button.

Send Letters

Collaborative Author Groups

Ontology and its Applications in Skills Matching in Job Recruitment

Anh Chi Tuan [a], Minh Tuan Dang [b,c], Hai Nam Do [b], Vijender Kumar Solanki [d], Jorge Torres [e], Ruben Gonzalez Crespo [e] and Thi Ngoc Anh Nguyen [a,*]

[a] *Faculty of Mathematics and Informatics, Hanoi University of Science and Technology, Hanoi, Vietnam*
E-mails: chi.ta194232@sis.hust.edu.vn, anh.nguyenthingoc@hust.edu.vn
[b] *CMC Applied Technology Institute, Hanoi, Vietnam*
E-mails: tuandm1@cmc.com.vn, dhnam3@cmc.com.vn
[c] *Posts and Telecommunications Institute of Technology, Hanoi, Vietnam*
[d] *Department of Computer Science Engineering, CMR Institute of Technology, Hyderabad, TS, India*
E-mail: spesinfo@yahoo.com
[e] *International University of La Rioja, La Rioja, Spain*
E-mails: ruben.gonzalez@unir.net, jorge.torres@unir.net

Abstract. In the recruitment process, manually selecting suitable candidates from curriculum vitae (CVs) for a job description (JD) is both time-consuming and expensive. Traditional keyword-based methods struggle to capture skill semantics, prompting the development of more advanced JD-CV matching systems. This paper aims to investigate and construct an ontology-based skills recommendation system, with objectives including creating a skills ontology and developing skills matching methods for JD-CV pairs. The objective of our approach is to enhance the accuracy and contextual relevance of recommendations by utilizing the proposed score. The proposed skills ontology and skills matching strategies are applied to a real dataset in Vietnam. The results of our study can automatically recommend a list of CVs for a given JD. Furthermore, the findings indicate that our proposed model surpasses comparative approaches by a margin of at least 1% to 5%. Overall, the study demonstrates the potential of utilizing ontology-based approaches to offer a practical solution for enhancing hiring practices.

Keywords: Skills matching, Skills Ontology, Job recruitment, Recommendation

1. Introduction

Nowadays, the demand for hiring fitting candidates for specific job positions is increasing [1]. LinkedIn is the largest business and employment social media platform, with 2.7 million companies listed, 61 million people searching for jobs weekly, and 117 job applications submitted every second[2]. These statistics demonstrate a significant demand for recruitment services and highlight the necessity of developing an efficient system that can optimize resource allocation for companies.

The manual selection of candidates during recruitment can be time-consuming and laborious. The conventional method of keyword-based searching fails to capture the semantics (Negi and Kumar, 2014)

[*] Corresponding author. E-mail: anh.nguyenthingoc@hust.edu.vn.
[1] ManpowerGroup's 2024 Global Talent Shortage Survey.
[2] LinkedIn Statistics 2023.

and complexities of skills, resulting in unsuitable matches. Therefore, many matching systems between JDs and CVs have been developed.

JD-CV matching systems (recommendation systems) are intelligent technologies that help recruit talent by identifying the best candidates for a specific job role. There are four main types of recommendation systems:

- Content-based system: A content-based recommendation system focuses on the attributes and characteristics of the CVs and JDs (Francis C. Fernández-Reyes, 2019; Najjar et al., 2022).
- Collaborative filtering system: It relies on the behaviour and preferences of multiple users to make recommendations. In JD-CV matching, collaborative filtering can analyze the interactions between CVs and JDs (Cabrera-Diego et al., 2019).
- Knowledge-based system: A knowledge-based recommendation system, known for ontology technologies, leverages domain-specific knowledge, such as industry standards, job requirements, and qualifications (Guo et al., 2016).
- Hybrid system: A hybrid recommendation system combines multiple recommendation approaches to provide more accurate recommendations (Wang et al., 2022; Huang et al., 2023).

Each method has its merits and there is no one-size-fits-all approach. In this paper, an ontology-based matching system between JDs and CVs is investigated since it offers several benefits:

- Ontology-based methods enable semantic understanding of data (Grimm et al., 2011; Embley, 2004), ensuring that matches are not solely based on exact keyword matches but consider the relevance and relatedness of concepts, leading to meaningful matches.
- Ontologies are designed to be flexible and extensible (Zhao and Meersman, 2005), allowing for the addition of new skills as the job market evolves. This adaptability ensures that the matching system remains up-to-date and relevant over time.
- Ontologies can be developed with input from domain experts, ensuring that the matching process aligns closely with industry-specific requirements (Tang et al., 2023).

The paper is structured as follows: Section 1 provides a broad perspective of the research and outlines the core issue addressed. Section 2 analyzes some related works. Section 3 introduces the conceptualization of ontology and its application in constructing a skills ontology. Additionally, it describes the methods for skills matching. Section 4 is the practical implementation and outcomes of experiments on real datasets. Our conclusion is described in Section 5.

2. Related Works

2.1. JD-CV matching

Talent acquisition is a significant, intricate, and time-consuming task in human resources (HR) (Derous and De Fruyt, 2016). In addition, there is a substantial number of employees stepping in and out each month. For example, in the United States alone, in April 2024, there were 5.64 million hires and 5.37 million separations, reflecting significant workforce movement[3]. Recruiters have two types of sourcing: active and passive candidates. The fit of applications and a JD must be evaluated by a domain expert for effective CV screening. Because there are so many different positions and the human resources department receives so many applications, shortlisting CVs is challenging.

[3] Statista's monthly job hires and separations in the United States.

Table 1

Summary of previous literature on JD-CV matching and skills matching.

Publication	**Year**	**Objective**	**Methodology**	**Case Studies**
JD-CV matching				
Guo et al. (2016)	2016	Extracting information, constructing an ontology for skills, matching JDs and CVs	Natural Language Toolkit (NLTK) (Bird, 2006), Finite state transducer, Skills ontology, Statistical similarity measure	Private datasets to evaluate different modules
Cabrera-Diego et al. (2019)	2019	Ranking CVs without depending on JDs	Dice's coefficient (Cabrera-Diego et al., 2015), Inter-Résumé proximity, Relevance feedback	A private dataset of 171 JDs, each with at least 20 CVs, from which at least 5 were from relevant applicants and 5 from irrelevant ones
Francis C. Fernández-Reyes (2019)	2019	Retrieving relevant CVs based on a JD	Word2vec, Principal Component Analysis (PCA) (Rao, 1964), Hybrid word embedding space (Mikolov et al., 2013)	A private dataset of over 580 CVs for different JDs
Wang et al. (2022)	2022	Matching degree between JDs and CVs using historical data	Recurrent neural network (RNN), mashRNN (Jiang et al., 2019), Co-attention neural networks, Attention-based BiLSTM (Bi-directional Long Short Term Memory) (Alfattni et al., 2021), Graph matrix	A private, real-world dataset provided by an online recruitment company, which includes nearly 4500 CVs and 270,000 JDs
Najjar et al. (2022)	2022	Constructing I-Recruiter, an intelligent decision support system for screening CVs	Skip-Gram model, Natural Language Toolkit (NLTK), NER, Word Embedding, Cosine Similarity	A private dataset of 101 CVs and 4 JDs
Huang et al. (2023)	2023	Depicting the hidden potential preference, capturing the dynamic interactive importance, evaluating the suitability of CVs for jobs	A lite version of BERT (ALBERT) (Lan et al.), Text Convolutional Neural Network (TextCNN) (Kim, 2014), Co-attention mechanism, Aspect-attention mechanism	A private dataset of 5238 CVs and 62 JDs
Skills matching				
Fareri et al. (2021)	2021	Mining and mapping soft skills	NER, Text Mining	Not available
Fallahnejad and Beigy (2022)	2022	Attention-based skill translation models for expert finding	Long Short-Term Memory(LSTM), Attention Mechanism, Deep Learning	Not available

There are several JD-CV matching methods. Previous studies have incorporated JD-CV matching systems using a content-based approach. Francis C. Fernández-Reyes (2019) focused on training the hybrid word embedding space to better represent JDs and CVs. To be more specific, they trained a word embedding space and prepared another pre-trained space to make a hybrid word embedding space. Three techniques including word embedding addition, linear combination and selection were utilized to create the hybrid space. Then CVs and JDs were both mapped to the corresponding average word embeddings and then they were used to compute ranking scores, using the cosine similarity. Besides, Najjar et al. (2022) suggested an intelligent decision support system (I-Recruiter) to rank CVs. The recommender system includes (i) a training block to train word embeddings, (ii) a matching block to match between JDs and CVs using cosine similarity, and (iii) an extracting block to retrieve the details of top-ranked CVs. Nevertheless, word embeddings failed to incorporate semantic information and were not considered historical data, potentially leading to adverse effects on the output of the system.

On the other hand, Cabrera-Diego et al. (2019) utilized a collaborative-centered approach and proposed two novel methods for evaluating CVs without relying on job offers or any semantic resources, namely Inter-Résumé proximity and Relevance feedback. Inter-Résumé proximity refers to the lexical similarity observed among CVs for a specific JD, while Relevance feedback is a means to enhance the ranking of CVs. The application of Relevance feedback involves the utilization of techniques that rely on similarity coefficients and vocabulary scoring. Nevertheless, due to the semantic independence of this approach, the ranking of CVs or more specifically, the scores of CVs do not accurately represent the compatibility between CVs and JDs.

Other studies employed a knowledge-based method to address the JD-CV matching process. Guo et al. (2016) introduced RésuMatcher, a personalized job-résumé matching system that extracts information from JDs and CVs and constructs a domain-specific ontology for skills. The system subsequently employs a unique statistical similarity measure to compute the degree of similarity between JDs and CVs. However, the assessment was performed on a limited dataset and only a single-domain ontology was constructed. Hence, numerous domain-specific ontologies are an essential expansion to better and effectively incorporate JDs and CVs from various domains (i.e. other skills and disciplinaries).

Recently, hybrid ontology methods have gained more attention in the science community. Wang et al. (2022) are the pioneers in incorporating graph neural networks into the person-job fit task, thereby introducing a novel approach to modelling the experience of recruiters. They established PJFCANN (Person-Job Fit from candidate profile and related recruitment history with Co-Attention Neural Networks) - a model considering both JDs and CVs content and the successful matches in the past. When provided with a given targetted CV-job post pair, PJFCANN initially generates local semantic representations using a Recurrent Neural Network (RNN). Simultaneously, it generates global experience representations for the pair from historical actual employment records, using a Graph Neural Network (GNN). Consequently, the ultimate matching degree is computed by concatenating these two representations. Though the model is robust, it heavily depends on historical data and more semantic information should be incorporated. Huang et al. (2023) also attempted hybrid methods and used neural networks and integrated historical recruitment data in their model, Attentive Implicit Relationship-Aware Neural Network (AIRANN). ALBERT and TextCNN were used to represent text features while the co-attention mechanism was used to represent non-text features. Additionally, hired CVs also contributed to the overall representation via an implicit relationship mechanism. Three of them were aggregated using an aspect-attention mechanism and then went through a prediction layer to generate a final prediction.

2.2. *Skills matching*

The research on recruitment is always developed with the aim of improving the match between employers' skills and job requirements quickly and accurately. Recent research in the field of HR has been increasingly developed to improve the match between candidates and job requirements. Research focuses mainly on the quantitative relevance of candidates' skills, including soft skills (Fareri et al., 2021) and expert-level skills (Fallahnejad and Beigy, 2022).

The above studies focus mainly on the amount of skills included in a candidate's profile and their compatibility. The approaches are also very diverse such as using text mining (Fareri et al., 2021) to help extract skills, or more complex using deep learning networks combined with attention technology to get similarities between two different CVs (Fallahnejad and Beigy, 2022), (Wang et al., 2022).

Fallahnejad et al. approached the issue by leveraging context-aware data to identify semantically related translations for skills or by employing machine translation models to bridge the semantic gap between applicants and skills (Fallahnejad and Beigy, 2022). The models mentioned above have partially solved the issue of locating qualified applicants and have also demonstrated the significance of skill fields in the process in large part due to their vast number of skills and ongoing updates and also specifically identified how appropriate the application is for the position.

2.3. Ontology-based approach

Table 2

Summary of previous literature on ontology-based approach.

Publication	Year	Objective	Methodology	Case Studies
Zaouga et al. (2019)	2019	Developing a domain ontology for human resource management to improve team selection based on competencies and provide a common understanding of process concepts	Ontology development process (Lopez, 1999), Web Ontology Language (OWL)	Not available
Lv and Peng (2021)	2021	Propose a novel periodic learning ontology matching model based on interactive grasshopper optimization algorithm	Grasshopper optimization algorithm (Dinh, 2021), Roulette wheel approach, Reward and punishment mechanisms	Public datasets provided by Ontology Alignment Evaluation Initiative (OAEI)
Ntioudis et al. (2022)	2022	Building the job recommendation framework of the Integration of Migrants MatchER SErvice (IMMERSE)	Resource Description Framework (RDF), Web Ontology Language (OWL), Simple Protocol and RDF Query Language (SPARQL), Native OWL 2 RL reasoning, GraphDB	A private dataset of 100 JDs and 30 CVs
Sharma and Kumar (2023)	2023	Retrieving semantic documents using Word2vec model	CONtextual QUery-awarE Ranking (CONQUER) (Chen and Saad, 2009), Word2vec model, Horse Herd optimization, K-nearest neighbour (KNN)	A private dataset

The application of the ontology-based approach is widespread across various disciplines due to its provision of a formal representation and semantic lucidity. Zaouga et al. (2019) suggested employing an ontological methodology to construct a unified and shared representation within the field of human resource management (HRM) domain. Following the ontology development process, the authors built a human resource ontology named HR-Ontology. The HR-Ontology provides a structured and universally accepted lexicon of HRM processes and concepts to address the knowledge gap. It also lets us evaluate each offer role based on skills, authority, and other factors. The proposed ontology allows for the development of decision support systems that improve human resource management.

On the other hand, Lv and Peng (2021) focused on ontology matching. The primary contribution of this article lies in the introduction of a novel periodic learning ontology matching model based on an interactive grasshopper optimization algorithm. This model incorporates user engagement and regular feedback to enhance the accuracy of the matching process. The authors additionally present the roulette wheel method, for selecting the most challenging mappings, and mechanisms for rewarding and punishing in order to effectively disseminate user feedback to the evolving population. The model was examined using two interactive tracks from the Ontology Alignment Evaluation Initiative and it is expected to facilitate enterprises in achieving harmonization of product catalogues and enhancing data integration.

Ntioudis et al. (2022) shared the same interest with our research when they developed an ontology-based personalized job recommendation framework for migrants and refugees. The framework uses an ontology to semantically represent the CV of an applicant, which is a migrant or refugee, and the details of a JD. It also includes a matchmaking service that provides relevant job recommendations considering the full CV of a job seeker and the details of the available jobs. They also utilized Simple Protocol and RDF Query Language (SPARQL) (Steve Harris, 2013) to give suitable recommendations. However, the framework was experimented on a limited dataset with only 100 JDs and 30 CVs of migrants and refugees.

Recently, semantic retrieving has received more attention. Sharma and Kumar (2023) devised a semantic knowledge-based retrieval system to overcome challenges encountered in previous approaches, including limited vector dimensions and prolonged execution time. The unique point in this model is using a word2vec model enhanced by Horse Herd Optimization (HHO). This integration was used to support the accuracy level performance of the information retrieval system, which aids in the extraction of vectors as features for classification.

3. Methodology

3.1. Skills Ontology

Based on a skills taxonomy, a skills ontology can be built. Gallagher et al. (2022) constructed a UK skills taxonomy, extracted from over 100,000 JDs in the UK in 2022 with four levels. Using this result, an ontology for skills is built.

Definition 1. *A skills ontology can be defined as the following tuple:*

$$skillsOntology =< S, R, L, C >, \tag{1}$$

where S is the set of skills (as classes in ontology), R is the set of relationships, L is the set of class labels, and C is the set of comments on the skillsOntology.

Set of skills Set S contains all skills (act as classes in *skillsOntology*) in the UK skills taxonomy.

Set of relationships Between skills (classes in *skillsOntology*), there is only the relationship "has subskill". For example, skill *webDevelopment* has a subskill $django$. Hence, set of relationships R

contains pairs of skills that have the relationship "has subskill", such as $(webDevelopment, django)$ or $(webDevelopment, flask)$. Note that $(webDevelopment, django)$ means skill *webDevelopment* has subskill $django$.

Set of class labels For each class in *skillsOntology*, there is a label included. Set of class labels L contains class names in *skillsOntology* with its exact English name, for instance, $(webDevelopment$, "web development"$)$.

Set of comments In set of comments C, all comments on *skillsOntology* are stored, namely title, description, version info, et cetera.

3.2. Skills Matching

In this part, techniques to match skills between a JD and CVs, namely cosine similarity and semantic similarity are discussed.

3.2.1. Cosine similarity

The most common approach is to use cosine similarity or its variants to evaluate skills in CV and JD. In this paper, a cosine similarity-based approach is introduced. Note that, this approach is the most common, however, it is not an ontology-based approach.

First, the "closeness" between pairs of skills is calculated. The idea is that if two skills are related, it is highly likely to appear in the same JD. Thus, a count matrix is built to record the appearance of every skill in all JDs in the dataset.

Given $JD = \{JD_1, JD_2, \ldots, JD_m\}$ is a set of m JDs, $S^J = \{S^J_1, S^J_2, \ldots, S^J_t\}$ is a set of t required skills extracted from all JDs in JD. A count matrix $\boldsymbol{F}_{m\times t}$ is defined by:

$$\boldsymbol{F}[k][l] = \begin{cases} 1 & \text{if } S^J_l \text{ appears in } JD_k, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

in which $k = \overline{1,m}$, $l = \overline{1,t}$. JD_k, S^J_l corresponding to the kth row and lth column of $\boldsymbol{F}$ respectively.

From matrix $\boldsymbol{F}$, the vector represented for skill S^J_l is lth column in matrix $\boldsymbol{F}$, denoted by $\boldsymbol{F}_l$. Create a matrix $\boldsymbol{FS}_{t\times t}$ to denote a matrix of cosine scores between pairs of skills. For two skills S^J_k and S^J_l, the score between them is calculated using cosine similarity:

$$\boldsymbol{FS}(S^J_k, S^J_l) = \boldsymbol{FS}[k][l] = \frac{\boldsymbol{F}_k.\boldsymbol{F}_l}{|\boldsymbol{F}_k|.|\boldsymbol{F}_l|}, \tag{3}$$

where $\boldsymbol{F}_k$ is the k column vector of $\boldsymbol{F}$.

Algorithm 1 describes how to construct matrices $\boldsymbol{F}$ and $\boldsymbol{FS}$ based on (2) and (3).

For arbitrary JD_i, skill scores between JD_i and different CVs are calculated. Given $S^J_i = \{S^J_{i,1}, S^J_{i,2}, \ldots, S^J_{i,p_i}\}$ is a set of p_i required skills in JD_i, $S^C_j = \{S^C_{j,1}, S^C_{j,2}, \ldots, S^C_{j,q_j}\}$ is a set of q_j skills listed in CV_j. Note that if a skill in a CV does not appear in S^J, the similarity equals zero. The skill score between JD_i and CV_j is computed as follows:

$$cScore(JD_i, CV_j) = \frac{1}{p_i} \sum_{k=1}^{p_i} \max_{1\leqslant l\leqslant q_j} \boldsymbol{FS}(S^J_{i,k}, S^C_{j,l}), \tag{4}$$

Algorithm 1 Constructing matrix of cosine scores between skills $\boldsymbol{FS}$

```
Input: $JD = \{JD_1, JD_2, \ldots, JD_m\}$.
Output: Cosine scores matrix $\boldsymbol{FS}$.
Function $scoreMatrix(JD)$:
Extract all skills from $JD_k$ and add to set $S^J$
Initialize matrices $\boldsymbol{F}$, $\boldsymbol{FS}$
for each $JD_k \in JD$ do
    for each $S_l \in S^J$ do
        if $S_l \in JD_k$ then
            $\boldsymbol{F}[k][l] = 1$
        else
            $\boldsymbol{F}[k][l] = 0$
        end if
    end for
end for
for $i = k$ to $t$ do
    for $j = l$ to $t$ do
        $\boldsymbol{FS}[k][l] = \frac{\boldsymbol{F}_k.\boldsymbol{F}_l}{|\boldsymbol{F}_k|.|\boldsymbol{F}_l|}$
    end for
end for
return $S^J$, $\boldsymbol{FS}$
```

Algorithm 2 Cosine similarity

```
Input: $S^J_i = \{S^J_{i,1}, S^J_{i,2}, \ldots, S^J_{i,p_i}\}$, $S^C_j = \{S^C_{j,1}, S^C_{j,2}, \ldots, S^C_{j,q_j}\}$, $S^J$, $\boldsymbol{FS}$.
Output: cScore between $JD_i$ and $CV_j$.
Function $cScore(JD_i, CV_j)$:
score = 0
for each $S^J_{i,k} \in S^J_i$ do
    temp = 0
    for each $S^C_{j,l} \in S^C_j$ do
        if $S^C_{j,l} \in S^J$ then
            temp = max(temp, $\boldsymbol{FS}(S^J_{i,k}, S^C_{j,l})$)
        end if
    end for
    score = score + temp
end for
cScore = score / $p_i$
return cScore
```

where $\boldsymbol{FS}(S^J_k, S^C_l)$ is the score between skill S^J_k and S^C_l.

Algorithm 2 shows the cosine similarity approach. In general, it is clear that cosine similarity is easy to implement and not related to *skillsOntology*. However, from the word embeddings to the ranking strategy, the semantics of skills are not involved much. That is why ontology-based approaches are chosen to deal with this problem.

3.2.2. Semantic similarity

With the aim of using the semantics of skills, the *skillsOntology* is used to rank candidates. Many skills matching techniques are proposed for skills ontology (Balachander and Moh, 2018). The underlying of many skills matching techniques is the Semantic Similarity Measure suggested by Alani and Brewster (2005). This similarity measures the "closeness" of two classes in ontology. Note that in our *skillsOntology*, skills are acting as classes in ontology.

Before the semantic similarity is introduced, first, the *skillsOntology* is converted into an undirected graph $G = (V, E)$ where vertices are skills (V is set of skills S) and edges are relationships between skills (E is set of relationships R) with all weights equal 1. Note that skills in *skillsOntology* are taken from the UK skills taxonomy so that some other skills in CVs might not appear in set of skills S in *skillsOntology*. Let $\tilde{\boldsymbol{P}}$ denote a set of paths between skill S_k and S_l in graph G. Inspired by the Semantic Similarity Measure by Alani and Brewster (2005), the semantic similarity between two skills S_k and S_l can be defined as:

$$semantic(S_k, S_l) = \begin{cases} 0, \text{if } \tilde{\boldsymbol{P}} = \emptyset; S_k, S_l \notin V, \\ 1, \text{if } S_k \equiv S_l, \\ \dfrac{1}{\min\limits_{p \in \tilde{\boldsymbol{P}}} length(p) + 1}, \text{otherwise.} \end{cases} \tag{5}$$

To find $\min\limits_{p \in \tilde{\boldsymbol{P}}} length(p)$, the length of the shortest path between S_k and S_l, many algorithms might be chosen. For example, Dijkstra's method can be used to find the shortest path between two skills in the graph. Algorithm 3 illustrates the Dijkstra's method to find the shortest path from S_k to S_l.

Algorithm 3 Dijkstra's algorithm to find the shortest path

```
Input: $G = (Vertices, Edges)$, source vertex $S_k$, target vertex $S_l$.
    ($G$ has a set of vertices $Vertices = \{v_0, \ldots, v_n\}$, $v_0 = S_k$, $v_n = S_l$)
Output: Length of the shortest path from $S_k$ to $S_l$.
Function MinLength($G$, $S_k$, $S_l$):
for i=1 to n do
    $L(v_i) = \infty$
end for
$L(v) = 0$
$S = \emptyset$
while $v_n \notin S$ do
    $u$ = a vertex not in $S$ with $L(u)$ minimal
    $S = S \cup \{u\}$
    for all vertices $v \notin S$ do
        if $L(u) + w(u, v) < L(v)$ then
            $L(v) = L(u) + w(u, v)$
        end if
    end for
end while
```

Algorithm 4 Semantic similarity

```
Input: $S_i^J = \{s_{i,1}^J, s_{i,2}^J, \ldots, s_{i,p_i}^J\}$, $S_j^C = \{s_{j,1}^C, s_{j,2}^C, \ldots, s_{j,q_j}^C\}$, $G$.
Output: sScore between $JD_i$ and $CV_j$.
Function $sScore(JD_i, CV_j)$:
score = 0
for each $S_{i,k}^J \in S_i^J$ do
    semantic = 0
    for each $S_{j,l}^C \in S_j^C$ do
        if $s_{i,k}^J \equiv s_{j,l}^C$ then
            semantic = 1
        else
            if $S_{i,k}^J, S_{j,l}^C$ in graph $G$ then
                semantic = $\max\left(\text{semantic}, \frac{1}{MinLength(G, S_{i,k}^J, S_{j,l}^C) + 1}\right)$
            end if
        end if
    end for
    score = score + semantic
end for
sScore = score / $p_i$
return sScore
```

Finally, for arbitrary JD_i, skill scores between JD_i and different CVs are calculated. Given $S_i^J = \{S_{i,1}^J, S_{i,2}^J, \ldots, S_{i,p_i}^J\}$ is a set of p_i required skills in JD_i. $CV = \{CV_1, CV_2, \ldots, CV_n\}$ denotes a set of n CVs. For each CV_j ($j = \overline{1,n}$), $S_j^C = \{S_{j,1}^C, S_{j,2}^C, \ldots, S_{j,q_j}^C\}$ is a set of q_j skills listed in CV_j. The skill score between JD_i and CV_j is computed:

$$sScore(JD_i, CV_j) = \frac{1}{p_i}\sum_{k=1}^{p_i} \max_{1 \leqslant l \leqslant q_j} semantic(S_{i,k}^J, S_{j,l}^C). \tag{6}$$

Algorithm 4 describes the complete semantic similarity approach to calculate the skill score between JD_i and CV_j.

3.3. Proposed method

3.3.1. Proposed semantic similarity

To better the semantic similarity introduced in 3.2.2, an improved version is suggested. Instead of converting an ontology to a graph and then using Dijkstra's method, the *skillsOntology* stays unchanged. The formula (5) is reused but instead of Dijkstra's method, SPARQL, a powerful query language used to retrieve and manipulate data stored in ontologies, is utilized.

First, consider the situation shown in Figure 1. In this case, $semantic(arimaModel, laplaceTransformer) = 0.25$. However, it is obvious that if a recruiter requires skill *arimaModel* and an applicant has skill *laplaceTransforms*, the similarity score between them should not be counted since they are too irrelevant. That observation suggests a better semantic similarity version.

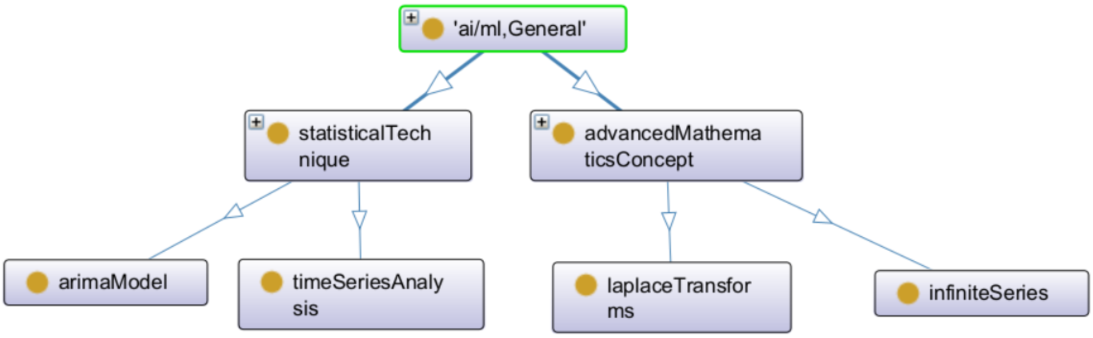


Fig. 1. An example leads to the improved semantic similarity.

The proposed version originates from the idea that it only matters if two skills have the relationship $hasSubSkill$ directly or indirectly. That means, if one skill is not a subskill of the other, the semantic similarity should be zero. This leads to the $semantic(arimaModel, laplaceTransformer) = 0$ instead of 0.25 in Figure 1 as they do not have a $hasSubSkill$ relationship neither directly nor indirectly.

Let $\widehat{\boldsymbol{P}}$ denote a set of paths between skill S_i and S_j if they have relationship $hasSubSkill$ either directly or indirectly in *skillsOntology*. The proposed semantic similarity measure between two random skills S_k and S_l is:

$$pSemantic(S_k, S_l) = \begin{cases} 0, \text{if } \hat{\boldsymbol{P}} = \emptyset; S_k, S_l \notin S, \\ 1, \text{if } S_k \equiv S_l, \\ \dfrac{1}{\min\limits_{p\in\hat{\boldsymbol{P}}} length(p) + 1}, \text{otherwise.} \end{cases} \tag{7}$$

To find $\min\limits_{p\in\widehat{\boldsymbol{P}}} length(p)$, the length of the shortest path between skill S_i and S_j if they have relationship $hasSubSkill$ either directly or indirectly in *skillsOntology*, SPARQL can be used to make the most of ontologies technology.

First, as presented in Definition 1, each class has a label corresponding to its exact English name. For example, a class displaying *webDevelopment* (its class name) can represent the skill "web development" (corresponding label). Since a class might have many superclasses and subclasses, we will use SPARQL to query all superclasses and subclasses of a class based on its labels (exact English skill name). Consider class *skillX* representing a skill named "skill X". Figure 2 illustrates its superclasses and subclasses. Remember that querying superclasses and subclasses of a class named *skillX* is the same as finding all skills that have the relationship $hasSubSkill$ directly or indirectly with "skill X".

Let *QSub_1("name of skill")* denote the query in SPARQL for finding subclasses (subskills) level 1 of an arbitrary skill. Class *skillX* represents a skill named "skill X". *QSub_1("skill X")* can be expressed in Owlready2[4] as follow:

[4] A package for manipulating OWL 2.0 ontologies in Python.

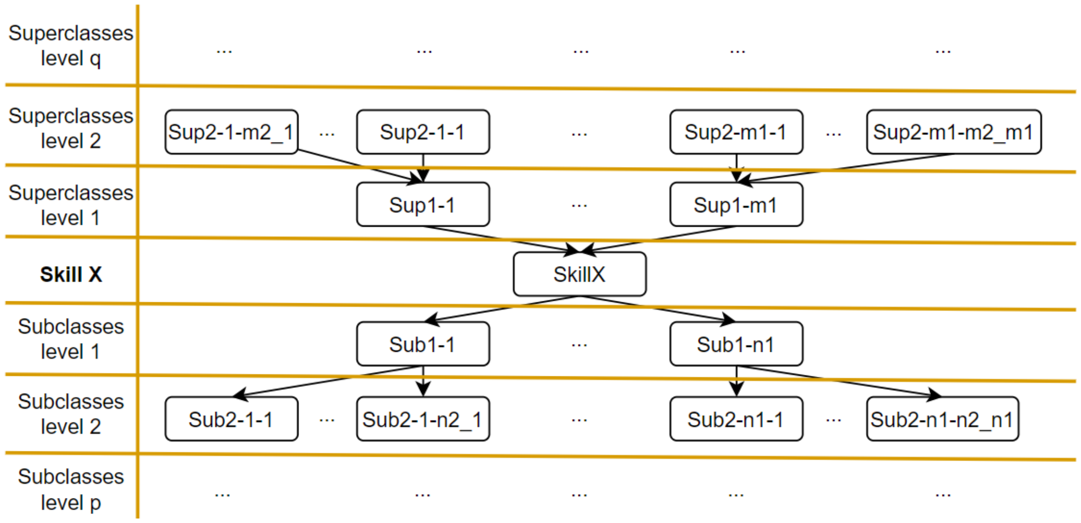


Fig. 2. Subclasses and superclasses of $SkillX$.

```
QSub_1("skill X") = SELECT DISTINCT(STR(?y1))
{?x1 a owl: Class. ?x2 a owl:Class. ?x1 rdfs:subClassOf ?x2.
?x1 rdfs:label ?y1. ?x2 rdfs:label "skill X".}
```

Similarly, *QSub_1("name of skill")*, *QSub_2("name of skill")*,... are defined and they are queries for finding subskills level 1, 2,... of an arbitrary skill.

The same scenario is applied to find superclasses (superskills) of a random skill. Let *QSup_2("name of skill")* denote the query in SPARQL for finding superskills level 2 of an arbitrary skill. *QSup_2("skill X")* can be shown as:

```
QSup_2("skill X") = SELECT DISTINCT(STR(?y3))
{?x1 a owl: Class. ?x2 a owl:Class. ?x3 a owl:Class.
?x1 rdfs:subClassOf ?x2. ?x2 rdfs:subClassOf ?x3. ?x3 rdfs:label ?y3.
?x1 rdfs:label "skill X".}
```

Consequently, *QSup_1("name of skill")*, *QSup_2("name of skill")*,... are queries for finding superskills level 1, 2,... of an arbitrary skill.

Algorithm 5 computes all superskills and subskills of every skill in a CV and adds them to the corresponding levels, from 1 to 3, as the *skillsOntology* built from UK skills taxonomy has only 4 levels. Using the returned result, $pSemantic$ between a skill in a JD and a skill in a CV can be easily calculated. For example, if a skill in a JD appears in Sub2 of a CV, $pSemantic$ between them equals 1/3.

For arbitrary JD_i, skill scores between JD_i and different CVs are calculated. Given $S_i^J = \{S_{i,1}^J, S_{i,2}^J, \ldots, S_{i,p_i}^J\}$ is a set of p_i required skills in JD_i. $CV = \{CV_1, CV_2, \ldots, CV_n\}$ denotes a set of n CVs. For each CV_j $(j = \overline{1,n})$, $S_j^C = \{S_{j,1}^C, S_{j,2}^C, \ldots, S_{j,q_j}^C\}$ is a set of q_j skills listed in CV_j. The skill score between JD_i and CV_j is computed as follows:

$$pScore(JD_i, CV_j) = \frac{1}{p_i} \sum_{k=1}^{p_i} \max_{1 \leqslant l \leqslant q_j} pSemantic(S_{i,k}^J, S_{j,l}^C). \tag{8}$$

Algorithm 5 Finding superskills and subskills in 3 levels using SPARQL

```
Input: $S^C_j = \{S^C_{j,1}, S^C_{j,2}, \ldots, S^C_{j,q_j}\}$ including $q_j$ skills in $CV_j$, $skillsOntology$.
Output: Lists of superskills and subskills of $CV_j$ in 3 levels.
Function $sup\_sub(S^C_j, skillsOntology)$:
1:  Initialize empty lists Sup1, Sup2, Sup3, Sub1, Sub2, Sub3
2:  for each $S^C_{j,l}$ in $S^C_j$ do
3:      Sup1_temp = Execute($QSup\_1(S^C_{j,l})$)
4:      Sup2_temp = Execute($QSup\_2(S^C_{j,l})$)
5:      Sup3_temp = Execute($QSup\_3(S^C_{j,l})$)
6:      Sub1_temp = Execute($QSub\_1(S^C_{j,l})$)
7:      Sub2_temp = Execute($QSub\_2(S^C_{j,l})$)
8:      Sub3_temp = Execute($QSub\_3(S^C_{j,l})$)
9:      Add Sup1_temp, Sup2_temp, Sup3_temp, Sub1_temp, Sub2_temp, Sub3_temp to Sup1, Sup2,
        Sup3, Sub1, Sub2, Sub3 respectively
10: end for
11: return Sup1, Sup2, Sup3, Sub1, Sub2, Sub3
```

Algorithm 6 displays the complete version of the proposed semantic similarity to calculate the score between JD_i and CV_j. First, superskills and subskills of every skill in the input CV_j are found and added to specific levels. For each pair of a skill from JD_i, $S^J_{i,k}$, and a skill from CV_j, $S^C_{j,l}$, $pSemantic$ is calculated based on the lists of superskills and subskills in levels. Eventually, $pScore$ between JD_i and CV_j computes the average $pSemantic$ between pairs of skills in JD_i and CV_j.

3.3.2. *Proposed recommendation system*

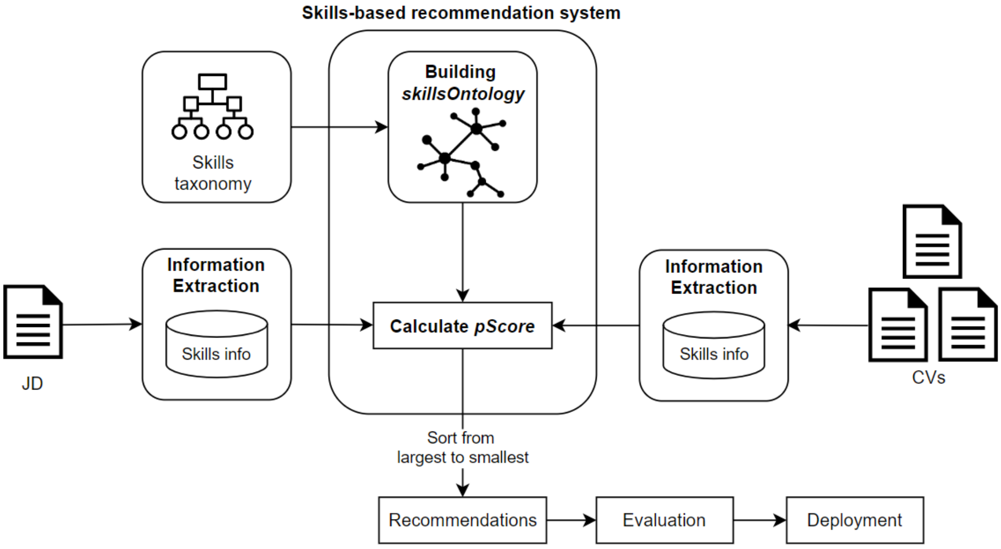


Fig. 3. Proposed skills-based recommendation system.

Algorithm 6 Proposed semantic similarity

```
Input: $S_i^J = \{S_{i,1}^J, S_{i,2}^J, \ldots, S_{i,p_i}^J\}$, $S_j^C = \{S_{j,1}^C, S_{j,2}^C, \ldots, S_{j,q_j}^C\}$, $skillsOntology$.
Output: pScore between $JD_i$ and $CV_j$.
Function $pScore(JD_i, CV_j)$:
score = 0
Sup1, Sup2, Sup3, Sub1, Sub2, Sub3 = $sup\_sub(skillsOntology, S_j^C)$
for each $S_{i,k}^J \in S_i^J$ do
    pSemantic = 0
    if $S_{i,k}^J \in$ Sup3 or $S_{i,k}^J \in$ Sub3 then
        pSemantic = 1/4
    end if
    if $S_{i,k}^J \in$ Sup2 or $S_{i,k}^J \in$ Sub2 then
        pSemantic = 1/3
    end if
    if $S_{i,k}^J \in$ Sup1 or $S_{i,k}^J \in$ Sub1 then
        pSemantic = 1/2
    end if
    if $S_{i,k}^J \in S_j^C$ then
        pSemantic = 1
    end if
    score = score + pSemantic
end for
pScore = score / $p_i$
return pScore
```

Figure 3 describes the proposed system to recommend CVs for JDs. First, a *skillsOntology* is built from the UK's skills taxonomy, as presented in 3.1. Along with that, skills from a JD and CVs are extracted and pre-processed. Then, pScore, our proposed score, between each pair of JD and CV is calculated following steps in Algorithm 6. All scores are sorted from the largest to the smallest and top *k* corresponding CVs are recommended. Finally, the system goes through the evaluation process by analyzing metrics (MAP@K, NDCG@K) before being deployed officially. For the deployment process, the proposed skills-based recommendation system is integrated into a hiring application. In this phase, extracted skills information from different JDs and CVs is uploaded and stored, and the semantic scores between them are also computed and saved. Based on the semantic similarities between CVs and a specific JD, top *k* suitable CVs are suggested to recruiters.

4. Experiments and Results

4.1. Dataset

The dataset includes JDs and CVs collected in Vietnam, available in https://github.com/anhnguyenthingoc/Ontology-based-skills-matching. In each JD, there is "required_skills", a list of required skills for that position. Each CV includes "skills", a list of skills that an applicant has. There are 1200 pairs of JD-CV and they are divided into 6 datasets. Each JD-CV pair is labelled with 1 if it is a successful pair and

0 otherwise. For each JD, there is only one successful CV and vice versa, for each CV, only one JD is suitable. Note that for each JD in the dataset, there is only one successful CV and nine failed CVs.

4.2. Evaluation metrics

4.2.1. Mean Average Precision@K (MAP@K)

MAP@K calculates the average precision across all JDs. Let $Precision@K_i$ denote the accuracy of top K recommendation CVs for JD_i. $Precision@K_i$ will be either 0 or 1, indicating whether the successful CV is among the top K recommendation CVs. N is denoted by the total number of different JDs. The Formula for MAP@K can be represented as follows:

$$\text{MAP@K} = \frac{1}{N}\sum_{i=1}^{N} Precision@K_i. \tag{9}$$

4.2.2. Normalized Discounted Cumulative Gain@K (NDCG@K)

NDCG@K measures the quality of top K recommendation CVs for a given JD by considering both the relevance of the CVs (successful or unsuccessful CVs) and their positions in the ranked list. NDCG@K is calculated based on Discounted Cumulative Gain@K (DCG@K) and Ideal Discounted Cumulative Gain@K (IDCG@K):

$$DCG@K = \sum_{i=1}^{K} \frac{rel_i}{\log_2(i+1)}, IDCG@K = \sum_{i=1}^{K} \frac{irel_i}{\log_2(i+1)}, NDCG@K = \frac{DCG@K}{IDCG@K}, \tag{10}$$

in which i is the ith position in the list, K is the top K CVs in the ranked list, rel_i is the relevance of the CV at position i in the ranked recommendation list (0 for unsuccessful and 1 for successful CV), $irel_i$ is the relevance of the CV at position i in the ranked actual list (0 for unsuccessful and 1 for successful CV).

4.3. Results

4.3.1. skillsOntology

As stated before, the UK skills taxonomy (Gallagher et al., 2022) is used to create *skillsOntology*.

Figure 4 describes the *skillsOntology* in Protégé, an open-source ontology development platform. Classes in *skillsOntology* are skills in the UK skills taxonomy. Note that in every ontology created in Protégé, there is always a superclass of everything, named $owl : Thing$.

Figure 5 shows the mapping of the UK skills taxonomy to the *skillsOntology*. In Figure 5, from the right to the left are the skills in levels 1, 2, 3, and 4 in the skills taxonomy respectively.

4.3.2. Ranking CVs

After the *skillsOntology* is built, different skills-matching techniques are applied to recommend the top 1, 3, and 5 suitable CVs for each JD in the dataset. Table 3 shows the comparison results between different skills matching techniques[5].

[5] Result conducted in Google Colab with Intel(R) Xeon(R) 2.20GHz and 16GB RAM.

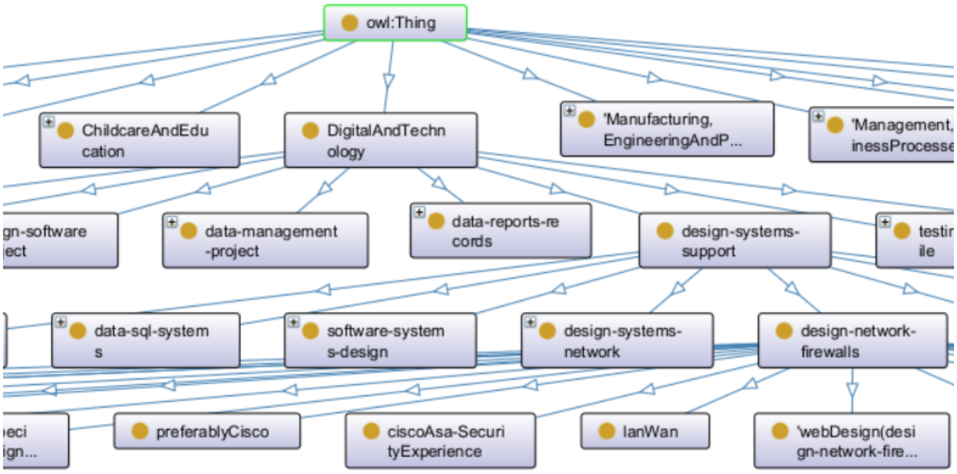


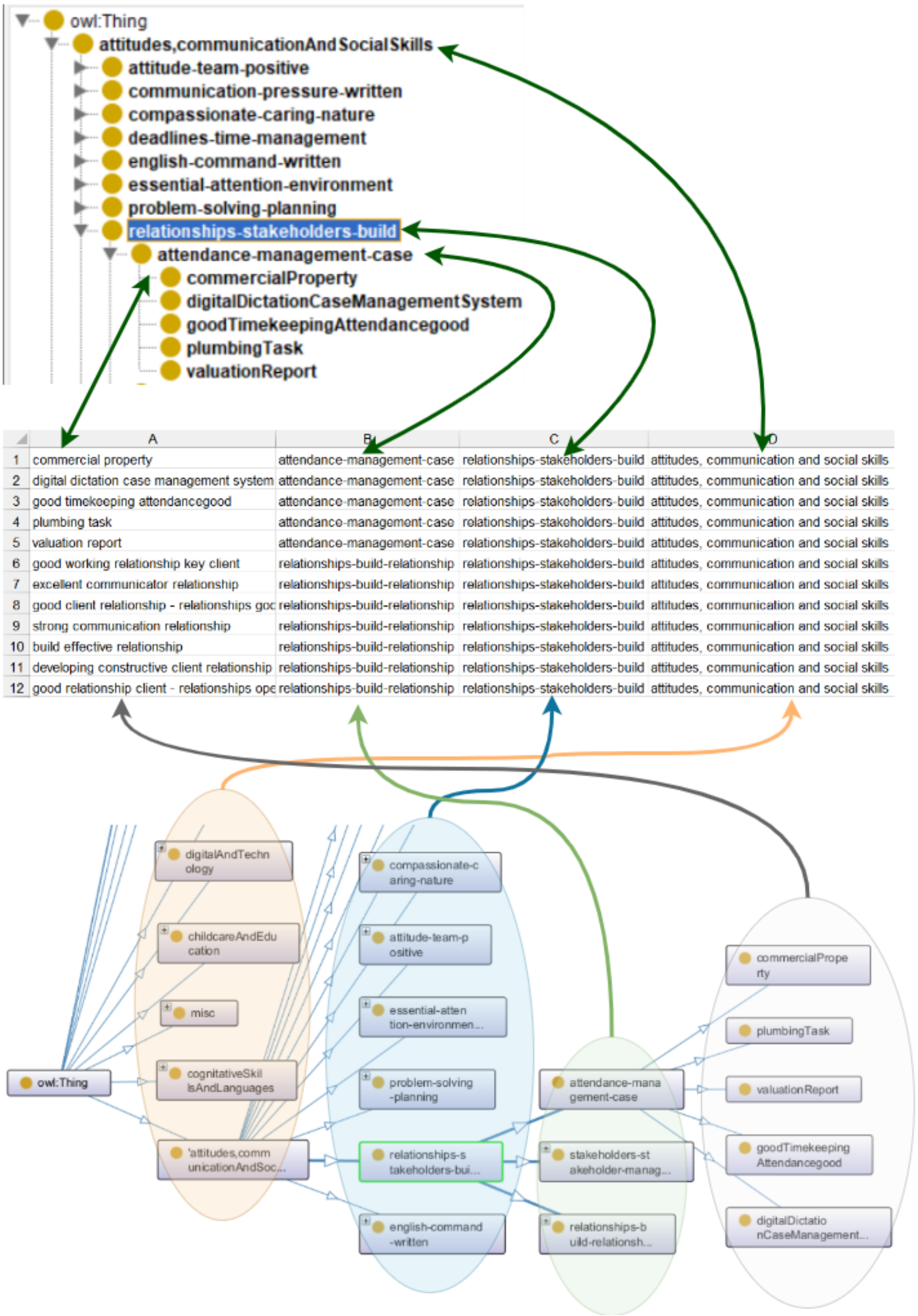
Fig. 4. Example of *skillsOntology* in Protégé.

Table 3

Overall recommendation result with the best results bold. The proposed model is superior performance on average from 1% to 5 %.

Metrics	Average(cScore) $\pm\sigma$	Average(sScore) $\pm\sigma$	Average(pScore) $\pm\sigma$
Average time (s)	**7.4874** $\pm$ **0.693**	93.1796 $\pm$ 5.5297	14.588 $\pm$ 1.886
MAP@1	0.6667 $\pm$ 0.0258	0.7167 $\pm$ 0.0258	**0.7667** $\pm$ **0.0408**
MAP@3	0.791 $\pm$ 0.0636	0.875 $\pm$ 0.0274	**0.9167** $\pm$ **0.0258**
MAP@5	0.8417 $\pm$ 0.0376	0.95 $\pm$ 0.0204	**0.9667** $\pm$ **0.0258**
NDCG@3	0.8933 $\pm$ 0.0118	0.9009 $\pm$ 0.0167	**0.9174** $\pm$ **0.0148**
NDCG@5	0.8596 $\pm$ 0.0167	0.8659 $\pm$ 0.0203	**0.892** $\pm$ **0.0187**

The table provided in Table 3 presents a summary of the overall recommendation outcomes, with the most successful metrics emphasized in bold. Despite pScore's not faster running speed compared to cScore, it is only one-fifth of sScore's speed and produces the most optimal outcomes. The metrics consist of the mean values of cScore, sScore, and pScore, each accompanied by their respective standard deviations. Additionally, the precision metrics (MAP@1, MAP@3, and MAP@5) and Normalized Discounted Cumulative Gain (NDCG@3, NDCG@5) further confirm the model's exceptional quality. The precision metric, specifically MAP@5, demonstrates exceptional performance with a value of **0.9667** $\pm$ **0.0258**, making it the top-performing precision metric. Furthermore, Figure 6 clearly illustrates the outcomes of the suggested model in contrast to other competitive models, thus highlighting its robust performance. Overall, the results of the proposed model exhibit exceptional performance across multiple criteria, demonstrating an average enhancement ranging from **1% to 5%** compared to alternative models.


owl:Thing
attitudes,communicationAndSocialSkills
attitude-team-positive
communication-pressure-written
compassionate-caring-nature
deadlines-time-management
english-command-written
essential-attention-environment
problem-solving-planning
relationships-stakeholders-build
attendance-management-case
commercialProperty
digitalDictationCaseManagementSystem
goodTimekeepingAttendancegood
plumbingTask
valuationReport
digitalAndTechnology
childcareAndEducation
misc
cognitiveSkillsAndLanguages
'attitudes,communicationAndSoc...
owl:Thing
compassionate-caring-nature
attitude-team-positive
essential-attention-environmen...
problem-solving-planning
relationships-stakeholders-bui...
english-command-written
attendance-management-case
stakeholders-stakeholder-manag...
relationships-build-relationsh...
commercialProperty
plumbingTask
valuationReport
goodTimekeepingAttendancegood
digitalDictationCaseManagement...

	A	B	C	D
1	commercial property	attendance-management-case	relationships-stakeholders-build	attitudes, communication and social skills
2	digital dictation case management system	attendance-management-case	relationships-stakeholders-build	attitudes, communication and social skills
3	good timekeeping attendancegood	attendance-management-case	relationships-stakeholders-build	attitudes, communication and social skills
4	plumbing task	attendance-management-case	relationships-stakeholders-build	attitudes, communication and social skills
5	valuation report	attendance-management-case	relationships-stakeholders-build	attitudes, communication and social skills
6	good working relationship key client	relationships-build-relationship	relationships-stakeholders-build	attitudes, communication and social skills
7	excellent communicator relationship	relationships-build-relationship	relationships-stakeholders-build	attitudes, communication and social skills
8	good client relationship - relationships goc	relationships-build-relationship	relationships-stakeholders-build	attitudes, communication and social skills
9	strong communication relationship	relationships-build-relationship	relationships-stakeholders-build	attitudes, communication and social skills
10	build effective relationship	relationships-build-relationship	relationships-stakeholders-build	attitudes, communication and social skills
11	developing constructive client relationship	relationships-build-relationship	relationships-stakeholders-build	attitudes, communication and social skills
12	good relationship client - relationships ope	relationships-build-relationship	relationships-stakeholders-build	attitudes, communication and social skills

Fig. 5. Mapping of the UK skills taxonomy to the *skillsOntology*.

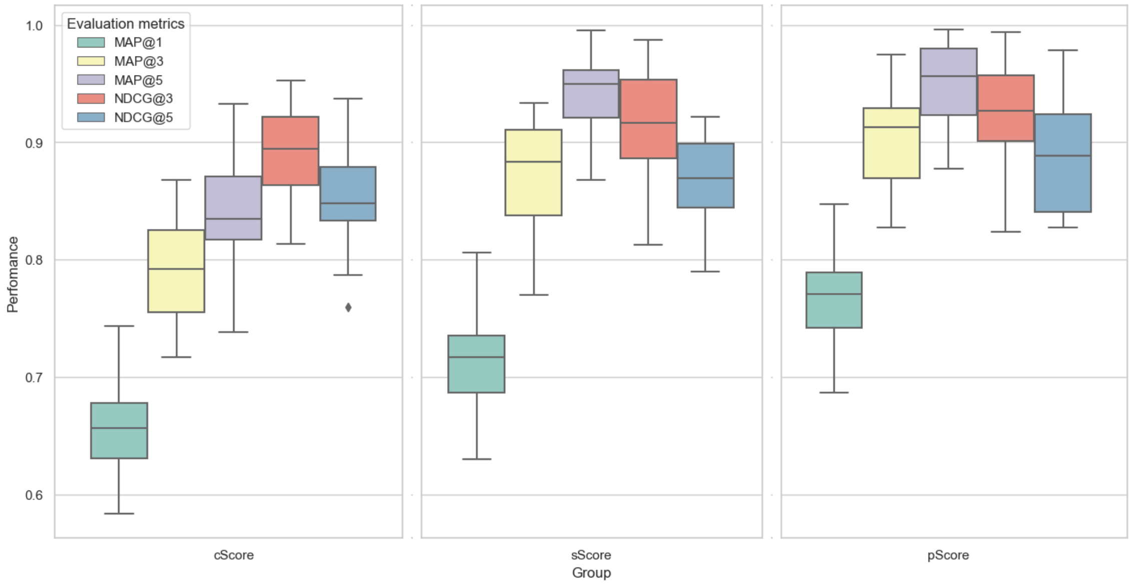


Fig. 6. Results of the proposed model and other competitive models.

5. Conclusion

5.1. Main contributions

Our main contribution is an ontology-based skills-matching system for Job Descriptions (JDs) and Curriculum Vitae (CVs), in which we (i) build a skills ontology and (ii) suggest a semantic skills-matching utilizing SPARQL. This approach has clear benefits, particularly its ability to provide a semantic comprehension of deep knowledge skills that surpass superficial skill keywords. The inherent flexibility of ontology-based skills matching enables the incorporation of a wide range of job requirements and candidate qualities. Moreover, the system's ability to reason improves the accuracy of matching, leading to more precise and contextually aware outcomes. The outcomes of the proposed model demonstrate outstanding performance across various criteria, showcasing an average improvement ranging from **1% to 5%** compared to other models. As we explore the challenges of job matching, ontology-based skills matching presents itself as a possible solution, providing a comprehensive and intelligent approach that is in line with the changing demands of recruitment.

As the results from the proposed system are promising, it should be implemented for future practical use. To work in practice, the proposed system utilizes RDF stores by Apache Jena Fuseki, specifically designed for efficiently storing and querying large RDF graphs, ensuring the system's scalability in the real world. In reality, there are a huge number of JDs and CVs, therefore, it is necessary to have a better configuration server than the configuration in the experiment to meet the actual needs. We suggest using a hardware server equipped with an Intel Core i9 processor with 16 cores, 32GB of RAM, and a 1TB SSD hard drive. Also, a Kubernetes service should be created for container orchestration and load balancing.

5.2. Limitations and Future work

Within this context, the approach offers outstanding advantages, particularly its ability to offer a deep comprehension of skills that go beyond conventional keyword searching. Therefore, the ontology-based skills matching system results in more precise and contextually aware results. However, to improve the robustness of the ontology-based skills matching system, it is crucial to conduct future work to address the following issues.

Firstly, it is essential to develop a standardization system for skills extracted from JDs and CVs. Usually, this information lacks organization and is presented in different levels of complexity. For instance, educational qualifications in a CV can provide insight into the skills that a person acquired. Additionally, a skill might have multiple ways of being described (synonyms) in CVs and JDs. Therefore, skill standardization is vital to find all implicit and explicit skills and use unified skill names.

Secondly, we intend to research a more dynamic model which utilizes machine learning and deep learning to build the skills ontology automatically. Existing skills taxonomies are static and require manual updates, making them slow to adapt to evolving job markets. Thus, the research direction will enable the integration of new skills by utilizing real-time job market data, such as data from platforms like LinkedIn. This ensures that the skills acquired are aligned with the evolving demands of the job market in various countries. While relying less on manual work, the dynamic ontology will still capture the core skill relationships.

Finally, skill proficiency and weights for each skill should be included. Certain skills may be indispensable requirements for the job (with significant importance), whereas others may be regarded as less crucial (with lower weight) or capable of being applied from prior experience. Moreover, knowing the desired level of proficiency for each skill (e.g., beginner, advanced) is vital. This allows employers or the recommendation system to evaluate whether a candidate possesses the requisite skills at a sufficiently advanced level. In general, incorporating both skill weights and proficiency levels creates a clearer picture of a candidate's suitability for the role.

Acknowledgements

This work was supported by the Vietnam Ministry of Education and Training [grant number B2023-BKA-07].

References

Alani, H. & Brewster, C. (2005). Ontology ranking based on the analysis of concept structures. In *Proceedings of the 3rd International Conference on Knowledge Capture*. K-CAP '05 (pp. 51–58–). New York, NY, USA: Association for Computing Machinery. doi:10.1145/1088622.1088633.

Alfattni, G., Peek, N. & Nenadic, G. (2021). Attention-based bidirectional long short-term memory networks for extracting temporal relationships from clinical discharge summaries. *Journal of Biomedical Informatics*, *123*, 103915. doi:10.1016/j.jbi.2021.103915.

Balachander, Y. & Moh, T.-S. (2018). Ontology Based Similarity for Information Technology Skills. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (pp. 302–305). doi:10.1109/ASONAM.2018.8508726.

Bird, S. (2006). NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*. COLING-ACL '06 (pp. 69–72–). USA: Association for Computational Linguistics. doi:10.3115/1225403.1225421.

Cabrera-Diego, L.A., El-Bèze, M., Torres-Moreno, J.-M. & Durette, B. (2019). Ranking résumés automatically using only résumés: A method free of job offers. *Expert Systems with Applications*, *123*, 91–107. doi:10.1016/j.eswa.2018.12.054.

Cabrera-Diego, L., Durette, B., Lafon, M., Torres-Moreno, J. & El-Bèze, M. (2015). How Can We Measure the Similarity Between Résumés of Selected Candidates for a Job? *Proceedings of the International Conference on Data Mining (DMIN)*, 99–106.

Chen, J. & Saad, Y. (2009). *Divide and Conquer Strategies for Effective Information Retrieval*. In *Proceedings of the 2009 SIAM International Conference on Data Mining (SDM)* (pp. 449–460). doi:10.1137/1.9781611972795.39.

Derous, E. & De Fruyt, F. (2016). Developments in Recruitment and Selection Research. *International Journal of Selection and Assessment*, *24*(1), 1–3. doi:https://doi.org/10.1111/ijsa.12123.

Dinh, P.-H. (2021). A novel approach based on Grasshopper optimization algorithm for medical image fusion. *Expert Systems with Applications*, *171*, 114576. doi:10.1016/j.eswa.2021.114576.

Embley, D.W. (2004). Toward semantic understanding: an approach based on information extraction ontologies. In *Proceedings of the 15th Australasian Database Conference - Volume 27*. ADC '04 (pp. 3–12–). AUS: Australian Computer Society, Inc..

Fallahnejad, Z. & Beigy, H. (2022). Attention-based skill translation models for expert finding. *Expert Systems with Applications*, *193*, 116433. doi:https://doi.org/10.1016/j.eswa.2021.116433.

Fareri, S., Melluso, N., Chiarello, F. & Fantoni, G. (2021). SkillNER: Mining and mapping soft skills from any text. *Expert Systems with Applications*, *184*, 115544. doi:https://doi.org/10.1016/j.eswa.2021.115544.

Francis C. Fernández-Reyes, S.S. (2019). CV Retrieval System based on job description matching using hybrid word embeddings. *Computer Speech & Language*, *56*, 73–79. doi:10.1016/j.csl.2019.01.003.

Gallagher, E., Kerle, I., Sleeman, C. & Richardson, G. (2022). A New Approach to Building a Skills Taxonomy. Economic Statistics Centre of Excellence (ESCoE) Technical Reports ESCOE-TR-16, Economic Statistics Centre of Excellence (ESCoE). https://ideas.repec.org/p/nsr/escoet/escoe-tr-16.html.

Grimm, S., Abecker, A., Völker, J. & Studer, R. (2011). *Ontologies and the Semantic Web*. In *Handbook of Semantic Web Technologies* (pp. 507–579). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-92913-0_13.

Guo, S., Alamudun, F. & Hammond, T. (2016). RésuMatcher: A personalized résumé-job matching system. *Expert Systems with Applications*, *60*, 169–182. doi:10.1016/j.eswa.2016.04.013.

Huang, Y., Liu, D.-R. & Lee, S.-J. (2023). Talent recommendation based on attentive deep neural network and implicit relationships of resumes. *Information Processing & Management*, *60*(4), 103357. doi:10.1016/j.ipm.2023.103357.

Jiang, J.-Y., Zhang, M., Li, C., Bendersky, M., Golbandi, N. & Najork, M. (2019). Semantic Text Matching for Long-Form Documents. In *The World Wide Web Conference*. WWW '19 (pp. 795–806–). New York, NY, USA: Association for Computing Machinery. doi:10.1145/3308558.3313707.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *Proceedings of Conference on Empirical Methods in Natural Language Processing*. doi:10.3115/v1/D14-1181.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. & Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Lopez, F. (1999). Overview Of Methodologies For Building Ontologies. In *International Joint Conference on Artificial Intelligence*. doi:10.1017/S0269888902000462.

Lv, Z. & Peng, R. (2021). A novel periodic learning ontology matching model based on interactive grasshopper optimization algorithm. *Knowledge-Based Systems*, *228*, 107239. doi:10.1016/j.knosys.2021.107239.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'13 (pp. 3111–3119–). Red Hook, NY, USA: Curran Associates Inc..

Najjar, A., Amro, B. & Macedo, M. (2022). An Intelligent DSS For Recruitment: Resumes Screening And Applicants Ranking. *Informatica*, *45*, 617–623. doi:10.31449/inf.v45i4.3356.

Negi, Y.S. & Kumar, S. (2014). A Comparative Analysis of Keyword- and Semantic-Based Search Engines. In D.P. Mohapatra and S. Patnaik (Eds.), *Intelligent Computing, Networking, and Informatics* (pp. 727–736). New Delhi: Springer India.

Ntioudis, D., Masa, P., Karakostas, A., Meditskos, G., Vrochidis, S. & Kompatsiaris, I. (2022). Ontology-Based Personalized Job Recommendation Framework for Migrants and Refugees. *Big Data and Cognitive Computing*, *6*(4). doi:10.3390/bdcc6040120.

Rao, C.R. (1964). The Use and Interpretation of Principal Component Analysis in Applied Research. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, *26*(4), 329–358.

Sharma, A. & Kumar, S. (2023). Ontology-based semantic retrieval of documents using Word2vec model. *Data & Knowledge Engineering*, *144*, 102110. doi:10.1016/j.datak.2022.102110.

Steve Harris, E.P. Andy Seaborne (2013). SPARQL 1.1 Query Language. In *W3C Recommendation*.

Tang, X., Feng, Z., Xiao, Y., Wang, M., Ye, T., Zhou, Y., Meng, J., Zhang, B. & Zhang, D. (2023). Construction and application of an ontology-based domain-specific knowledge graph for petroleum exploration and development. *Geoscience Frontiers*, *14*(5), 101426. doi:10.1016/j.gsf.2022.101426.

Wang, Z., Wei, W., Xu, C., Xu, J. & Mao, X.-L. (2022). Person-job fit estimation from candidate profile and related recruitment history with co-attention neural networks. *Neurocomputing*, *501*, 14–24. doi:10.1016/j.neucom.2022.06.012.

Zaouga, W., Arfa Rabai, L.B. & Alalyani, W.R. (2019). Towards an Ontology Based-Approach for Human Resource Management. *Procedia Computer Science*, *151*, 417–424. The 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019). doi:10.1016/j.procs.2019.04.057.

Zhao, G. & Meersman, R. (2005). Architecting Ontology for Scalability and Versatility. In R. Meersman and Z. Tari (Eds.), *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE* (pp. 1605–1614). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/11575801_42.