Knowledge-driven Unsupervised Skills Extraction for Graph-based Talent Matching

Ioannis Konstantinidis
i.konstantinidis@ihu.edu.gr
International Hellenic University
Thessaloniki, Greece

Manolis Maragoudakis
mmarag@ionio.gr
Department of Informatics, Ionian University
Corfu, Greece

Ioannis Magnisalis
i.magnisalis@ihu.edu.gr
International Hellenic University
Thessaloniki, Greece

Christos Berberidis
c.berberidis@ihu.edu.gr
International Hellenic University
Thessaloniki, Greece

Vassilios Peristeras
v.peristeras@ihu.edu.gr
International Hellenic University
Thessaloniki, Greece

ABSTRACT

In human resource management of large organisations, finding the best candidate for a job description requires an extensive examination of a large number of resume profiles. Even with the advent of Deep Information Retrieval and the supported semantic similarity search, identification of relevant skills within profiles requires thorough investigation over several aspects, including educational background, professional experience, achievements, etc. However, these techniques are based on the existence of domain-specific, human-annotated datasets, a laborious task that portrays high cost and a slow labeling progress. In this paper, we propose Resume2Skill-SE, an end-to-end architecture for interpretable skill-based talent matching. The solution consists of two components. The first module uses an unsupervised approach for skills extraction based on state-of-the-art text embeddings and efficient semantic similarity search. The second module creates a profile-skills bipartite graph and uses a proposed ranking formula for similar resume profiles, minimising the effect of potential errors from the skills extraction module. The optimal ranking formula was identified through an intuitive and automated evaluation method for getting relevance scores. The proposed technique delivers promising results while also including an interpretability layer by showing the common skills of a pair of resume profiles.

CCS CONCEPTS

• **Information systems → Information extraction**; **Recommender systems**; *Thesauri*; • **Computing methodologies →** *Topic modeling*; **Information extraction**; **Lexical semantics**.

KEYWORDS

unsupervised skills extraction, natural language processing, graph analytics, search engine

SETN 2022, September 7–9, 2022, Corfu, Greece
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9597-7/22/09…$15.00
https://doi.org/10.1145/3549737.3549769

ACM Reference Format:
Ioannis Konstantinidis, Manolis Maragoudakis, Ioannis Magnisalis, Christos Berberidis, and Vassilios Peristeras. 2022. Knowledge-driven Unsupervised Skills Extraction for Graph-based Talent Matching. In *12th Hellenic Conference on Artificial Intelligence (SETN 2022), September 7–9, 2022, Corfu, Greece.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3549737.3549769

1 INTRODUCTION

According to a study by Gartner, “By 2023, 25% of large enterprises will conduct continuous, rather than periodic, strategic workforce planning processes” indicating the importance of Human Resource (HR) processes [4]. Human Resources (HR) management is defined as a set of policies and management activities for human resources in enterprises, mainly consisting of HR strategies, recruitment, employee management, training and development [15]. With the emergence of digital transformation, large organisations receive a vast number of resumes of job seekers in the form of textual documents. This makes it extremely time-consuming for HR managers to identify the best talents out of a pool of job profiles with experience descriptions. A typical practice in the industry is to utilise full-text search engines like Elasticsearch, where the documents are indexed to be used for fast keyword search and to limit down the number of retrieved profiles for the recruiters to inspect. However, these engines fail to capture the semantic meaning of the text and thus disregard an important number of relevant profiles resulting in an extensive inspection of the resumes by the recruiters that could also lead to bias due to human fatigue [18, 26].

Another typical practice used by the recruiters is to identify a profile exhibiting similar skills to the job description out of the retrieved job applicants and use an automated approach for identifying similar profiles. Towards a better reranking of the results, there have been rapid advances on using machine learning techniques for talent matching [1, 16]. A machine learning model is trained on pairs of candidate profiles to predict their relevance. Yet, this approach requires a large amount of human-annotated data and is difficult to explain the reasoning of the machine learning model prediction due to the unstructured nature of textual data.

To that end, a comprehensible and straightforward way to identify similar profiles is to extract the skills from the experience descriptions and use them as input for resume similarity scoring. To achieve this, a typical method is to train a machine learning model for skills extraction that, however, requires manual labeling which is time and cost-consuming.

In this paper, we propose Resume2Skill-SE (Search Engine), an unsupervised approach for skills extraction that (a) leverages the BERT architecture and Siamese Networks for mapping the descriptions into a vector representation and (b) external knowledge from the ESCO classification of skills and occupations (see Section 3) using the Faiss algorithm for scalable and efficient skill search. Furthermore, the architecture uses the matched skills to model a profile-skills bipartite graph that allows calculating the similarity score between resumes based on different formulas. Finally, we evaluate our method using an automated approach for obtaining relevance scores.

The remaining paper is arranged as follows; Section 2 outlines the related work on unsupervised skills extraction and on graph-based profile matching. Section 3 provides a background for the ESCO classification used as an external knowledge base. Section 4 describes the proposed method. In Section 5, we present the experimental setup and the results. Section 6 provides a discussion of the findings and limitations of the study. Finally, Section 7 provides the conclusions of the paper and future work.

2 RELATED WORK

2.1 Skills extraction and matching

Skills extraction can be considered an Information Extraction task. Initial solutions employed TF-IDF scoring for matching LinkedIn profiles to Wikipedia articles and graph [10]. In [27], the authors introduced a framework for skills extraction and matching on big data. However, this pipeline indicates the need for manual semantic annotation. A hybrid approach for skills extraction was presented in [7]. This work extracts possible skills by using Named Entity Recognition (NER), Part of Speech (PoS) tagging and matching to external sources. The NER module extracts a set of keywords, entities and concepts. The PoS module uses predefined rules to identify candidate skills. The third module calculates the similarity score of each word/phrase to external sources like Wikipedia and skill dictionaries. The outputs are aggregated using a weighted formula for calculating the relevance score of a skill to a description. In [8], the authors address the task of skill extraction as a multi-label classification problem using Convolutional Neural Networks. Despite the effectiveness of the method, it requires the existence of a large-scale annotated dataset, which is based on extensive manual labour.

On the other hand, skills extraction and matching can also be considered an Entity Linking task, where the extracted skills from the text are linked to an existing knowledge base. To that end, a novel approach is introduced in [2] that creates a subspace of all the mentions in the text and makes the assumption that the matched entity will lie in the same subspace as the rest of the mentions. However, this approach also assumes the existence of an effective entity recognition solution that is not available for specific tasks like skills extraction.

Much of the literature pays particular attention to information extraction of research topics from articles. In [16], the authors introduced the Smart Topic Miner (STM), divided into three phases: topic extraction, topic selection and tag inference. The topic extractor receives the author keywords and maps them to topics of an earlier version of the Computer Science Ontology (CSO), a large-scale and automatically generated ontology of research topics that is utilised by many research studies [23]. In topic selection, STM uses a greedy set-covering algorithm to reduce the number of topics to the top-k relevant ones, where k is given by the user. Finally, in the tag inference phase, the selected topics are inferred to the Springer Nature Classification (SNC) by utilising existing mappings between CSO and SNC. However, this approach focuses on author keywords without considering more topics from the abstract or the main text. Another granular framework for research topic extraction is the CSO classifier [22]. The authors presented a hybrid method that combines a syntactic, semantic and post-processing modules. The syntactic module maps n-grams in the text to concepts in CSO by calculating the Levenshtein distance [12]. The semantic module consists of a series of steps with the use of word embeddings with Word2Vec, entity extraction using heuristics, concept filtering and ranking with cosine similarity, term frequency and diversity. Then, CSO classifier uses the elbow method to further filter out extracted concepts based on their tail distribution [24]. In the post-processing module, the topics are enriched using the CSO superTopicOf relationships. A recent version of the CSO classifier integrated an outlier detection module that identifies topics by calculating graph and embeddings similarity matrices [21]. The matrices use the Djikstra algorithm and cosine similarity, respectively, allowing to detect disconnected topics as outliers.

2.2 Talent Matching

The methods for automated talent matching to job descriptions can be classified into semantic- and graph-based. Semantic approaches typically employ Natural Language Processing (NLP) and text mining techniques for information retrieval. To that end, the work in [20] proposes two methods for classification and resume recommendation for a job description. The first approach leverages TF-IDF and cosine similarity for retrieving the most relevant resumes, while the second approach trains a machine learning model to identify the experience domain. However, focusing only on the domain misses important profile-specific details.

Over recent years, there has also been an interest in deep learning for semantic search. The authors in [18] proposed a novel hierarchical deep neural network architecture that leverages Recurrent Neural Networks (RNN) for textual sequence representation and the attention mechanism for identifying the most important skills and requirements to be considered for calculating the similarity between resume experience descriptions and job requirements. To further improve the performance, the authors integrated a topic-aware attention mechanism by initially extracting topics from text through a Latent Dirichlet Allocation model. Furthermore, an important contribution of this work is the fact that introduces the interpretability of the results due to the use of attention that can identify what are the most important items in the classification at a word and a sentence semantic level. A simple yet effective approach for profile matching is presented in [13], where the resume profile and jobs are embedded in a Deep Siamese Network for semantic searching. However, this method lacks in terms of interpretability as it is not able to provide information on the reasoning behind a decision.

Resume profile matching has also been addressed using a graph representation either of profiles and skills or through user logs. An approach that utilises collaborative filtering and considers the problem as a recommendation engine is presented in [25]. However, this is highly based on the existence of user logs on previous interactions with other job descriptions indicating the preference of a job seeker, while, if no logs can be found, the recommendation engine can only retrieve job descriptions using centrality algorithms, like PageRank, without addressing the specificity of each resume. Similarly, in [5], the authors proposed the use of a skills-job bipartite graph where relationships between skills are represented with weights indicating the cosine similarity score, while job occupations are connected to skills with their Revealed Comparative Advantage score. However, this approach is not able to scale up on a larger number of resume profiles and assumes the existence of correctly extracted skills.

The work in [17] presented a framework that uses the CSO Classifier for skill extraction and uses an algorithm of graph edit distance that integrates greedy assignment and Hausdorff matching. Similarly, the authors in [7] introduced a greedy maximal matching approach and calculate an affinity score for profile-to-job recommendation.

3 ESCO

ESCO is a multi-lingual classification of European Skills, Competences, Qualifications and Occupations [11]. It covers three different pillars: skills/competencies, occupations and qualifications that are related to the EU labour market and training in order to be matched by employment services to jobs Europe wide. ESCO describes 3,000 occupations and 13,000 skills and competencies and displays different qualifications in 26 languages making resume and job vacancies more transparent. In ESCO, concepts are represented as subclasses of SKOS concepts [14].

The ESCO Skills pillar can provide a rich knowledge base that can trigger other NLP tasks like skills extraction. It presents clear descriptions and labels of the skills while also providing information regarding the reuse level (sector-specific, occupation-specific, cross-sector and transversal). Furthermore, it provides relationships between skills indicating related essential and optional ones and whether a skill is connected to another of a narrower or broader level. Similarly, in the ESCO Occupations pillar, each occupation is connected to related essential and optional skills making it a valuable asset for identifying the required competence for job vacancies. According to a 2021 study on ESCO indicating the high coverage of this knowledge base, it was concluded that the ESCO classification can successfully capture the skills of Industry 4.0 [3]. Furthermore, ESCO provides a structure of fine-grained knowledge exposing more contextual information about each skill [6].

4 THE RESUME2SKILL-SE APPROACH

Resume2Skil-SE consists of two components: the skill extractor and the graph-based mechanism for finding profiles that are similar to a given profile. Figure 1 illustrates the architecture of Resume2Skill-SE.

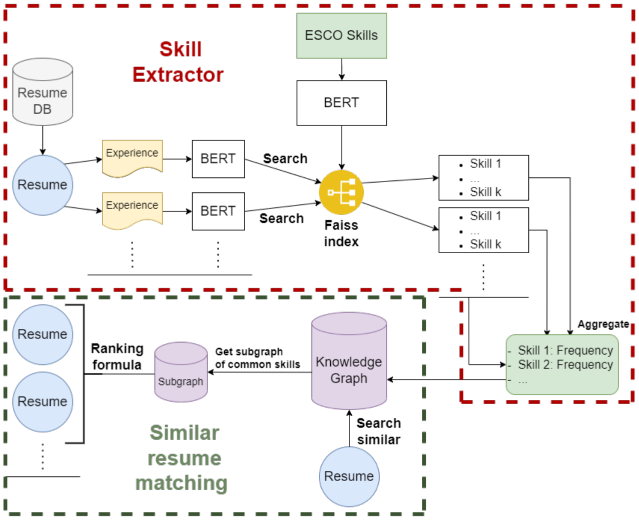

Figure 1: Resume2Skill Architecture

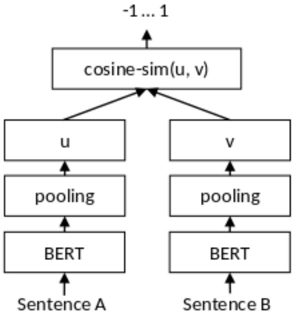

Figure 2: The SBERT architecture for sentence similarity

4.1 Skill Extractor

The scope of the skill extractor is to extract skills from the profile experience descriptions and match them to the ESCO classification. Let $P = p_1, p_2, \ldots, p_n$ be a set of resume profiles, where n is the total number of profiles. Each profile $p_i \in P$ consists of a set of experience descriptions $d_{i,j}$ so that $D_i = \{d_{i,1}, d_{i,2}, \ldots, d_{i,k_i}\}$, where k_i is the number of experiences of resume p_i.

We also use the ESCO classification as a knowledge base to match the skills from the profile experience descriptions. Let $S = \{s_1, s_2, \ldots, s_m\}$ be the set of ESCO skill descriptions.

First, we calculate the embeddings for the ESCO skill descriptions and the profile experience descriptions to convert them to a numeric representation with semantic meaning. We use the Siamese BERT-Network architecture for sentence embeddings presented in [19]. The architecture uses MPNet as a base for the BERT module and it has been trained on 1 billion pairs of sentences/short paragraphs originating from multiple data sources [26]. Figure 2 depicts the architecture of MPNet.

```
Data:
$P$: a set of resume profiles
$D_i$: a set of experience descriptions for resume $p_i \in P$
S: a set of skill descriptions from ESCO classification
Result:
$P_s$: a set of skills and respective weights for each profile $p_i \in P$
foreach $s_i \in S$ do
    $e_{si} \leftarrow$ SentenceBERT($s_i$);
end
$I \leftarrow$ FaissIndex($E_s$)
foreach $p_i \in P$ do
    $P_{si} \leftarrow \{s_1 : 0, \dots, s_m : 0\}$;
    foreach $d_{ij} \in D_i$ do
        $e_{dij} \leftarrow$ SentenceBERT($d_{ij}$);
        $S_{rel} \leftarrow$ FaissIndexSearch($I, e_{dij}$);
        foreach $d_{ij} \in D_i$ do
            $P_{si}[s_k] \leftarrow P_{si}[s_k] + 1$);
        end
    end
end
$P_s \leftarrow \{P_{s1}, \dots, P_{sn}\}$);
```

Algorithm 1: Skill Extractor

Using this architecture, we calculate the embeddings $E_Di = \{e_{di,1}, e_{di,2}, \dots, e_{di,k_i}\}$ and $E_S = \{e_{s1}, e_{s2}, \dots, e_{sm}\}$ for each profile $p_i \in P$ and the ESCO skill descriptions s_j , respectively.

Having obtained the ESCO skill embeddings E_S, we can use them for semantic search. To do this, we use the Faiss algorithm, which is a novel state-of-the-art method for scalable and efficient semantic similarity search on high-dimensional text embeddings [9]. Faiss allows indexing the embeddings E_S and is able to scale on billions of vectors leveraging GPUs for efficient similarity search on these embeddings.

The experience embeddings E_Di for each profile p_i are fed to the semantic search engine in order to retrieve the top-k relevant skills to each profile experience description. In our experiments, we retrieved the top-10 relevant skills, but this hyperparameter can be optimised depending on the length of each textual description. In such a manner, we are able to extract the skills from textual descriptions and match them to the ESCO classification for data harmonisation. As one skill can occur in more than one experience descriptions, we need to aggregate the skills for each profile. To do this, we count the number of occurrences of a skill being in the top-10 relevant to each experience description. As a result, for each profile $p_i \in P$, we obtain a set of skills with their corresponding weights based on the number of occurrences. The reasoning behind this is that when a skill appears in multiple experience descriptions then there is higher confidence that the resume includes the respective skill. The output of this process is a bipartite graph of resume profiles and skills as nodes and the relationships between profiles and skills represent the number of experience descriptions a skill appears in a profile.

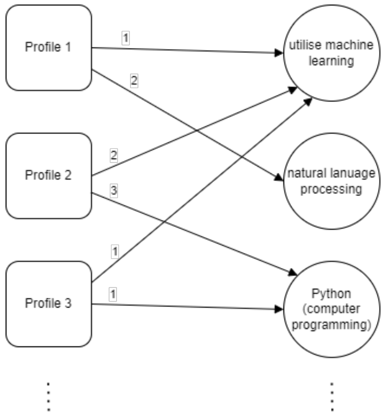

Figure 3: A generated bipartite graph of profiles and skills with weights representing the number of occurrences of a skill in experiences

4.2 Graph-based Similar Resume Matching

Using the ESCO skills extraction from the previous subsection, we obtain a more structured representation of the data as a bipartite graph of profiles and skills. A typical example of this graph is depicted in Figure 3.

This representation can facilitate fast, scalable and effective retrieval of similar profiles to a source profile overcoming the limitations of the traditional full-text-based search engines. As a result, more context is included in the profiles leading to a more comprehensible approach to explaining why a profile is considered relevant by providing a list of similar skills.

However, due to the unsupervised manner of the skills extraction methodology, the process can be error-prone, leading to false positives as skill matches. Therefore, careful consideration needs to be taken on the ranking formula for relevant profiles in order to minimize the effect of false positives.

There are various approaches to handling this issue. One way would be to consider the common number of skills:

$$simScore(p_i, p_j) = |P_{si} \cap P_{sj}| \quad (1)$$

However, this approach does not consider the weight of the relationships indicating the importance of skill to each profile. To that end, an alternative formula would be to calculate the sum of the source and target profile weights:

$$simScore(p_i, p_j) = \sum_k P_{si}[s_k] + \sum_p P_{sj}[s_p] \quad (2)$$

Yet, this formula fails at combining the weights of the source profile p_i and the target profile p_j.

An improvement to this can be to use the following formula by considering the dot product:

$$simScore(p_i, p_j) = \overline{P}_{si} \circ \overline{P}_{sj} = \sum_{k,t} P_{si}[s_k] \cdot P_{sj}[s_t], s_k = s_t \quad (3)$$

In this way, we consider a weighted sum of the target profile weights based on the source profile weights. As a result, if a skill

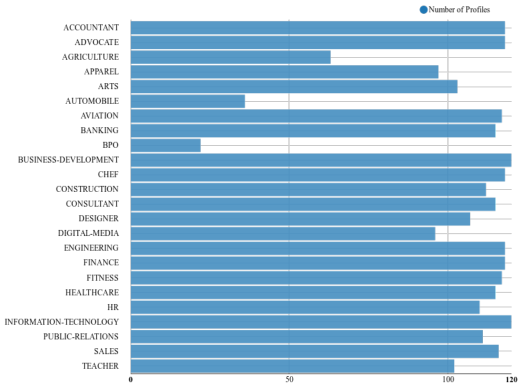

Figure 4: Number of resume profiles per experience type

weight of a source profile p_i is high and the target profile has also a high weight for the same skill, it will lead to a high similarity score. However, this approach can cause a bias towards a limited number of skills and also towards more general skills that occur many times, like management skills and can include more false positive matches.

Considering all the above-mentioned aspects, we propose the following formula:

$$simScore(p_i, p_j) = \frac{|P_{si} \cap P_{sj}|^2}{|P_{si}| \cdot |P_{sj}|} = \frac{|P_{si} \cap P_{sj}|}{|P_{si}|} \cdot \frac{|P_{si} \cap P_{sj}|}{|P_{sj}|} \quad (4)$$

With this ranking score, we consider two profiles to be similar if they have a high number of common skills but at the same time the source p_i and the target p_j profiles have total skills close to the number of common skills. The notion behind this is that the more identical skills two profiles have, the more similar they are. Furthermore, this method minimises the effect of false positive skill matches, because, if two resumes have similar experiences, then they are going to have similar false positives.

We argue that the skills extraction module should not be used independently as the profile matching module tackles the issue of the false positives in the extracted skills.

5 EXPERIMENTAL SETUP AND RESULTS

For our experiments and as a proof of concept, we used a resume dataset from Kaggle, which contains a collection of approximately 2,400 resumes in PDF format that were gathered from livecareer.com, a platform for job seekers. Figure 4 illustrates the number of profiles per experience type. We can observe there is a uniform distribution of the job profiles in relation to the experience types indicating the domain-agnostic character of the dataset.

For calculating the description embeddings, we employed the *all-mpnet-base-v2* model that is a general purpose pre-trained model on more than 1 billion text pairs from various datasets and achieved overall state-of-the-art results over 14 different tasks on different domains. The output of this model resulted in textual embeddings of 768 dimensions.

Then, we used the Faiss algorithm to index the ESCO skill descriptions. We opted for the IndexIVFFlat index of Faiss that improves search performance by segmenting the dataset into pieces using Voronoi cells in the d-dimensional space. The index was trained on the ESCO skills vectors and we opted for 100 cells as recommended by the creators of this algorithm. As mentioned in the proposed method, for each query embedding of experience description, we extracted the top-10 relevant skills based on the similarity with the skill descriptions, in order to have a high recall of the retrieved skills.

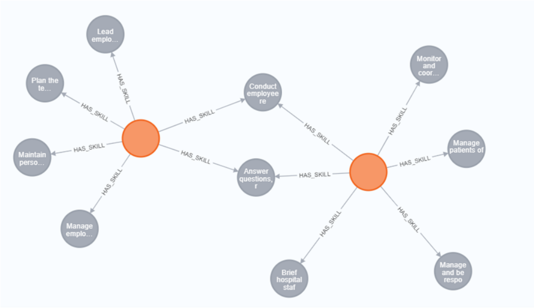

Figure 5: Resume-skills bipartite graph in Neo4j

The extracted skills were injected into a Neo4j graph database constituting a bipartite graph of resume profiles and skills. An indicative screenshot of the graph is depicted in Figure 6. Each resume profile had on average 40 relationships to skills, while each skill contained on average 10 relationships to resume profiles. As a result, even with a graph database of 2,400 job profiles, the graph exhibited high connectivity.

Having obtained a structured representation of the initially unstructured data, we experimented with the different similarity score formulas between resume profiles.

To avoid extensive manual labour for evaluating the results, we employed an automated assessment approach. More specifically, for each inspected pair of profiles, we calculate a similarity score to be considered as ground truth, by using the following formula:

$$simScoreReal(p_i, p_j) = mean\left\{max\left\{\frac{\overline{e}_k \circ \overline{e}_t}{|\overline{e}_k| \cdot \overline{e}_t}\right\} : k = 1...m\right\} \quad (5)$$

where $\overline{e}_k$ and $\overline{e}_k$ are the embeddings of the k-th and t-th experience description for the resume profiles p_i and p_j, respectively and n, m are the number of experience descriptions for each profile, respectively. Intuitively, for each experience description, this formula considers the maximum cosine similarity between two resume profiles and, for each resume profile, we calculate the average of these scores to obtain an aggregated score per profile. Therefore, we retrieve the best matches between resumes and, subsequently, two profiles are considered most similar if all the experience descriptions demonstrate high relevance score. This approach is powerful for small samples, but it is not possible to scale up on large datasets and this is the reason for incorporating it only for assessment purposes. For the final similar profile search evaluation, we use the standard metrics for evaluating information retrieval systems: the Mean Average Precision (MAP) and the Average Discounted Cumulative Gain (ADCG) using the following formulas:

Table 1: MAP and ADCG scores for different ranking methods

topk	Domain	Ranking formula	MAP	ADCG
top5	cross-domain	formula1	0.2184	2.1076
		formula2	0.216	2.0874
		formula3	0.2738	2.4549
		formula4	0.297	2.5379
	domain-specific	formula1	0.2156	2.1026
		formula2	0.2129	2.0773
		formula3	0.2552	2.3735
		formula4	0.276	2.4564
top10	cross-domain	formula1	0.2343	2.6452
		formula2	0.2333	2.6261
		formula3	**0.2968**	**3.0991**
		formula4	0.3291	3.2361
	domain-specific	formula1	0.229	2.5071
		formula2	0.2284	2.4908
		formula3	0.2807	2.8943
		formula4	0.3027	2.9888

$$MAP = \frac{1}{|P_{source}|}\left(\sum_{p_i}\frac{1}{|P_{target}|}\sum_{q=1}^{|P_{target}|} rel(d_r)\frac{\sum_{k=1}^{r} rel(d_k)}{r}\right) \tag{6}$$

$$ADCG = \frac{1}{|P_{target}|}\sum_{q=1}^{|P_{target}|}\sum_{i=1}^{k}\frac{rel(i)}{log_2(i+1)} \tag{7}$$

Where P_{source} is the set of source resume profiles that we want to identify relevant profiles, P_{target} is the set of the top-10 profiles ordered by relevance score and rel is the relevance score between a source resume profile and a target profile based on the formula (NUM) and range between 0 and 1.

In our experiments, we compared various parameters for skills extraction and different approaches for ranking the most relevant resume profiles. We distinguished between different values for the top-k parameter for extracting the most relevant ESCO skills to the experience descriptions. Furthermore, we compared the ranking score formulas, as mentioned above, and we also considered two cases of only domain-specific skills or not. The code and experiments can be found on GitHub.

Table 1 illustrates the comparative results of the different methods with the MAP and ADCG scores, respectively. Formula 1-4 refer to the ranking score formulas (1)-(4). The results indicated that formula (4) outperformed the other approaches while using a larger number of extracted skills (top-10) improved the performance. Furthermore, it is evident that considering only the domain-specific skills did not result in higher MAP and ADCG scores.

6 CONCLUSION

In this paper, we proposed an unsupervised approach for skills extraction and subsequently for identifying the most relevant resume profiles to a source profile. To that end, we incorporated an existing knowledge base, the ESCO classification, that represents a large number of skills from various domains for matching experience descriptions to the skills. We employed Siamese Networks using BERT for text representation and trained on diverse datasets from different domains for the task of text similarity. Subsequently, we proposed and experimented with different ranking formulas for profile matching that allowed minimisation of possible errors exposed by the skill extractor. We evaluated our approach using an intuitive automated assessment through sentence similarity that can only be applied to small samples and thus only for evaluation. The experiments indicated that allowing extraction of more skills to increase recall, resulted in better performance, while considering only domain-specific skills led to information loss without improving the results.

We argue that this method should be used mainly for the purpose of graph-based matching as an indicator for finding similar profiles and not as a clean knowledge base of profile skills, due to possible false matches in skill extraction. For future work, we aim to improve this aspect by incorporating larger datasets from LinkedIn and fine-tuning the Siamese Network on the ESCO skill descriptions as well as on a pool of job descriptions. This would lead to a more accurate and robust representation of the text embeddings for semantic search. Furthermore, we plan to employ key phrase extraction approaches to increase recall of skills matching and show where the skills are located within the text. Finally, we aim to extend our evaluation through human assessment and through extensive comparison to the CSO Classifier.

ACKNOWLEDGMENTS

This study was funded by the EU project DE4A (Digital Europe for All) Horizon 2020 —the Framework Programme for Research and Innovation (2014-2020) under Grant Agreement: 870635.

We would like to show our gratitude to Dr. Efstratios Kontopoulos for sharing his pearls of wisdom with us during the course of this research.

REFERENCES

[1] Edward Tristram Albert. 2019. AI in talent acquisition: a review of AI-applications used in recruitment and selection. *Strategic HR Review* 18, 5 (2019), 215–221. https://doi.org/10.1108/shr-04-2019-0024

[2] Akhil Arora, Alberto Garcia-Duran, and Robert West. 2021. Low-Rank Subspaces for Unsupervised Entity Linking. (2021), 8037–8054. https://doi.org/10.18653/v1/2021.emnlp-main.634 arXiv:2104.08737

[3] Filippo Chiarello, Gualtiero Fantoni, Terence Hogarth, Vito Giordano, Liga Baltina, and Irene Spada. 2021. Towards ESCO 4.0 – Is the European classification of skills in line with Industry 4.0? A text mining approach. *Technological Forecasting and Social Change* 173 (2021), 121177. https://doi.org/10.1016/j.techfore.2021.121177

[4] Alan D. Duncan. 2021. *100 Data and Analytics Predictions Through 2025.* Technical Report. https://www.gartner.com/en/doc/100-data-and-analytics-predictions-through-2025

[5] Anna Giabelli, Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, and Andrea Seveso. 2021. Skills2Job: A recommender system that encodes job offer embeddings on graph databases. *Applied Soft Computing* 101 (2021), 107049. https://doi.org/10.1016/j.asoc.2020.107049

[6] Lino González, Elena García-Barriocanal, and Miguel Angel Sicilia. 2021. Entity Linking as a Population Mechanism for Skill Ontologies: Evaluating the Use of ESCO and Wikidata. *Communications in Computer and Information Science* 1355 CCIS (2021), 116–122. https://doi.org/10.1007/978-3-030-71903-6_12

[7] Akshay Gugnani and Hemant Misra. 2020. Implicit skills extraction using document embedding and its use in job recommendation. *Proceedings of the 32nd Innovative Applications of Artificial Intelligence Conference, IAAI 2020* (2020), 13286–13293.

[8] Kameni Florentin Flambeau Jiechieu and Norbert Tsopze. 2020. Skills prediction based on multi-label resume classification using CNN with model predictions explanation. *Neural Computing and Applications* 33, 10 (2020), 5069–5087. https://doi.org/10.1007/s00521-020-05302-x

[9] Jeff Johnson, Matthijs Douze, and Herve Jegou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2021), 535–547. https://doi.org/10.1109/TBDATA.2019.2921572 arXiv:1702.08734

[10] Ilkka Kivimäki, Alexander Panchenko, Adrien Dessy, Dries Verdegem, Pascal Francq, Cédrick Fairon, Hugues Bersini, and Marco Saerens. 2020. A graph-based approach to skill extraction from text. *Proceedings of TextGraphs@EMNLP 2013: The 8th Workshop on Graph-Based Methods for Natural Language Processing* October (2020), 79–87.

[11] Martin le Vrang, Agis Papantoniou, Erika Pauwels, Pieter Fannes, Dominique Vandensteen, and Johan De Smedt. 2014. ESCO: Boosting Job Matching in Europe with Semantic Interoperability. *Computer* 47, 10 (2014), 57–64. https://doi.org/10.1109/MC.2014.283

[12] Vladimir I Levenshtein and Others. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. Soviet Union, 707–710.

[13] Saket Maheshwary and Hemant Misra. 2018. Matching Resumes to Jobs via Deep Siamese Network. *The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018* (2018), 87–88. https://doi.org/10.1145/3184558.3186942

[14] Alistair Miles, Brian Matthews, Michael Wilson, and Dan Brickley. 2005. SKOS core: simple knowledge organisation for the web. In *International conference on dublin core and metadata applications*. 3–10.

[15] Raymond Noe, John Hollenbeck, Barry Gerhart, and Patrick Wright. 2006. *Human Resources Management: Gaining a Competitive Advantage, Tenth Global Edition.* McGraw-Hill Education New York, MA, New York.

[16] Francesco Osborne, Angelo Salatino, Aliaksandr Birukou, and Enrico Motta. 2016. Automatic classification of springer nature proceedings with smart topic miner. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9982 LNCS (2016), 383–399. https://doi.org/10.1007/978-3-319-46547-0_33

[17] Tung T. Phan, Vinh Q. Pham, Hien D. Nguyen, Anh T. Huynh, Dung A. Tran, and Vuong T. Pham. 2021. Ontology-Based Resume Searching System for Job Applicants in Information Technology. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12798 LNAI (2021), 261–273. https://doi.org/10.1007/978-3-030-79457-6_23

[18] Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Chao Ma, Enhong Chen, and Hui Xiong. 2020. An Enhanced Neural Network Approach to Person-Job Fit in Talent Recruitment. *ACM Transactions on Information Systems* 38, 2 (2020). https://doi.org/10.1145/3376927

[19] Nils Reimers and Iryna Gurevych. 2020. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference* (2020), 3982–3992. https://doi.org/10.18653/v1/d19-1410 arXiv:1908.10084

[20] Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, and Rocky Bhatia. 2020. A Machine Learning approach for automation of Resume Recommendation system. *Procedia Computer Science* 167, 2019 (2020), 2318–2327. https://doi.org/10.1016/j.procs.2020.03.284

[21] Angelo Salatino, Francesco Osborne, and Enrico Motta. 2021. CSO Classifier 3.0: a scalable unsupervised method for classifying documents in terms of research topics. *International Journal on Digital Libraries* (2021). https://doi.org/10.1007/s00799-021-00305-y

[22] Angelo A. Salatino, Francesco Osborne, Thiviyan Thanapalasingam, and Enrico Motta. 2019. *The CSO Classifier: Ontology-Driven Detection of Research Topics in Scholarly Articles*. Vol. 11799 LNCS. Springer International Publishing. 296–311 pages. https://doi.org/10.1007/978-3-030-30760-8_26

[23] Angelo A. Salatino, Thiviyan Thanapalasingam, Andrea Mannocci, Aliaksandr Birukou, Francesco Osborne, and Enrico Motta. 2020. The computer science ontology: A comprehensive automatically-generated taxonomy of research areas. *Data Intelligence* 2, 3 (2020), 379–416. https://doi.org/10.1162/dint_a_00055

[24] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. 2011. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*. IEEE, 166–171.

[25] Walid Shalaby, Bahaa Eddin Alaila, Mohammed Korayem, Layla Pournajaf, Khalifeh Aljadda, Shannon Quinn, and Wlodek Zadrozny. 2017. Help me find a job: A graph-based approach for job recommendation at scale. *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017* 2018-Janua (2017), 1544–1553. https://doi.org/10.1109/BigData.2017.8258088 arXiv:1801.00377

[26] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. NeurIPS (2020), 1–14. arXiv:2004.09297 http://arxiv.org/abs/2004.09297

[27] Damian A. Tamburri, Willem Jan Van Den Heuvel, and Martin Garriga. 2020. DataOps for Societal Intelligence: A Data Pipeline for Labor Market Skills Extraction and Matching. *Proceedings - 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science, IRI 2020* (2020), 391–394. https://doi.org/10.1109/IRI49571.2020.00063 arXiv:2104.01966