Procesamiento del Lenguaje Natural, Revista nº 73, septiembre de 2024, pp. 83-94 recibido 11-03-2024 revisado 10-05-2024 aceptado 16-05-2024

Inductive Graph Neural Network for Job-Skill Framework Analysis

Aplicación de Redes Neuronales Basadas en Grafos para el Análisis del Framework Trabajo-Competencias

Hermenegildo Fabregat* **Rus Poves*** **Lucas Alvarez Lacasa**
Federico Retyk **Laura García-Sardiña** **Rabih Zbib**
Avature Machine Learning
{hermenegildo.fabregat, rus.poves, lucas.lacasa,
federico.retyk, laura.garcia, rabih.zbib} @avature.net
* These authors contributed equally to the work

Abstract: The analysis of skills and their relationship to different occupations is an area of special attention within human capital management processes. Nowadays, job specialization has made this increasingly important. In this paper, we address two main tasks: the retrieval of similar jobs and the retrieval of skills related to a given job. We develop a system that combines the encoding of textual information with a graph neural network, thus mitigating the limitations of a system that relies on either of these separately. We present experiments that show that the proposed system is able to take advantage of both the encoded textual information, and the structured relationships between job titles and skills represented by the graph. We also show the robustness of the proposed model in modeling unseen entities by evaluating the model's performance in simulated cold-recommendation scenarios where a percentage of the skills under study are eliminated during training.
Keywords: Graph Neural Networks, Jobs, Skills, Information Retrieval.

Resumen: En la actualidad, dentro de los procesos de gestión del capital humano, la especialización de profesiones ha hecho que el análisis de competencias profesionales y su relación con distintos empleos cobre cada vez más importancia. En este trabajo abordamos la recuperación de puestos de trabajo similares y la recuperación de competencias relacionadas con una ocupación determinada. Analizamos el rendimiento de un sistema basado en *Graph Neural Networks*, el cual trata de mitigar las eventuales limitaciones de sistemas basados únicamente en el procesamiento de información textual. Los resultados obtenidos indican que este tipo de enfoques son capaces de aprovechar eficientemente tanto la información textual como las relaciones entre ocupaciones y competencias explícitamente representadas. Además, se ha evaluado este sistema en un escenario de recomendación en frío donde un porcentaje de las competencias profesionales estudiadas fueron eliminadas del entrenamiento.
Palabras clave: Redes Neuronales Basadas en Grafos, Trabajos, Competencias, Recuperación de Información.

1 Introduction

Due to the rapid and profound advances that technology has been bringing to the labor market recently, understanding the intricate relationships between jobs and competencies has become of vital importance for individual career development as well as organizational success. Many organizations have understood this need, and have therefore promoted analyzing the labor market from a more detailed perspective in order to facilitate the decision-making processes and the optimization of traditional talent management methods. While at the individual level, skills play a key role in determining employability and earning potential, for companies a thorough understanding of human capital can lead to improvements in their productivity and a better success in their internal mobility processes (Störmer et al., 2014;

ISSN 1135-5948 DOI 10.26342/2024-73-6

Manyika et al., 2017).

The considerable dynamism observed in the modern labor markets, together with the inherent heterogeneity of job definitions and job requirements, are key challenges to be overcome in order to effectively conduct automated analyses of the relationships between skills and jobs. This is especially true in the context of emerging labor sectors. Graphs allow us to easily represent the relationships between different types of entities, making them an ideal framework for processing and understanding the complex dependencies between jobs and the different skill sets required to perform those jobs. Graph Neural Networks (GNNs) further exploit the representational capability and the generalization potential offered by neural models, thus allowing us to capture the latent patterns and dependencies in labor markets.

In this paper we explore a data-driven approach to study the relationships between jobs and skills that combines a pre-trained text encoder and a GNN-based model. The encoder makes use of the textual information associated with each node to models useful semantic information related to the job titles and skills. In our experiments, we analyze the benefit of using in-domain representations versus more generic ones. The GNN, on the other side, captures the relationships beyond the entities themselves, including relationships among adjacent entities. We use this combined approach to evaluate two tasks: For a given job, the extraction of similar jobs (job matching), and linked competencies (job-skills matching).

Besides, we also analyze the robustness of the model in the face of the dynamism of a fast evolving environment, like industries where the requirements for certain occupations are constantly changing. For this purpose, we analyze the behavior of the proposed model in a cold recommendation scenario, where skills not seen during training are analyzed through an exhaustive data ablation study. The results show that our approach is able to effectively tackle the tasks addressed, yielding improvements and benefits over comparable models.

The rest of the paper is organized as follows: Section 2 presents an overview of previous work available in the literature tackling similar problems, both within the domain of human talent management, and more widely. The data explored and the proposed system are described in detail in Section 3. Section 4 presents the experimental results, as well as a comparison with different baselines. Section 5 further reports on a group of experiments to test the robustness of the proposed model in two cold start prediction scenarios. Finally, Section 6 presents the conclusions and future research lines.

2 Related Work

Graph Neural Networks

Graph Neural Networks have experienced a substantial increase in popularity and research interest, mainly due to their remarkable adaptability to a wide range of applications, covering domains as diverse as social networks and biomedical data (Li et al., 2023; Wu et al., 2023).

An early work in this area is Grover and Leskovec (2016), which proposed a method for the generation of node embeddings using a semi-supervised approach. Their strategy focuses on optimizing an n-dimensional feature output space while preserving the inherent neighbor communities of the graph. This is achieved by creating a collection of random walks that explore, for each node, the neighbors up to $2nd$ order. The random walks generated are then used as sequences to train a SkipGram model for learning the structural organization of the nodes (Mikolov et al., 2013).

In contrast to that approach, where the representation space is built from scratch, Zhang et al. (2019) proposed a method to discover a latent representation within an initial feature space. They optimize a transformation $f(x_i)$, where x_i denotes the representation in the original space for a given node. Following Perozzi, Al-Rfou, and Skiena (2014), random walks are generated to capture the structural context, which allows fitting a model for predicting the context associated with each node. The authors claim that the final representation space aggregates both the structural and semantic information of each node.

Hamilton, Ying, and Leskovec (2017) address the challenge of processing new information, a typical limitation of many well-known graph-based models, including those we describe above. They proposed the

GraphSage model, which starts with an initial representation space and trains a set of aggregation functions to capture information about neighboring communities. They handle information about previously unseen nodes by applying the trained aggregation functions on the initial feature space, enhancing the model's ability to model novel data. GraphSage overcomes the limitations associated with the adaptability of graph-based models when faced with new information. In this work we make extensive use of the GraphSage model.

GNNs for Skills Management

The industrial sector and human resource management are areas where GNNs have been well received.

Zhu, Chen, and Wang (2022) proposed the Graph-Community-Enabled (GCE) model that addresses learning and career guidance. This is a recommendation system based on the use of a heterogeneous graph composed of jobs, skills and courses. Its aim is to assist students and employees to achieve their career goals by finding useful training courses and other learning resources. They extracted vocabulary related to skills, jobs and courses from Coursera and various job portals, resulting in a dataset consisting of approximately 1,600 skills, 957 courses and 20,000 job titles. Skills play a key role and serve as a bridge that facilitates the connection between jobs and courses. To improve the generalization capability, they use a community detection algorithm that efficiently aggregates information from different skills. This approach contributes significantly to the efficient linking of job and course data (Rosvall and Bergstrom, 2008).

Unlike our work, where we use information extracted directly from professional resumes, the GCE model explores information extracted through web crawlers applied to job portals and educational platforms. In the context of professional resumes, individuals often list several occupations, listing a wide range of skills. On the other hand, job postings often emphasize job-specific (or company-specific) requirements, which makes the task of generating knowledge graphs simpler. However, the GCE model has limited generalization, since job postings do not usually detail all relevant skills, but only the most important ones for the specific position.

Works like Liu et al. (2021) and Goyal et al. (2023) study the similarities between job titles and skills using GNNs. These works were also based on the analysis of data extracted from job portals. Liu et al. (2021) presented the Multi-Graph Neural Network based Skill Prediction model (MGNSP) to tackle the prediction of relationships between jobs and skills. The MGNSP model uses three different multi-layer neural graph networks to capture multiple types of relationships. One graph represented the relationships between job titles by parsing job description metadata, while another highlighted the interaction of different skills by analyzing co-occurrences, forming a graph that mapped these relationships. The last GNN uses a bipartite graph for representing the direct links between job titles and skills. To help the aggregation of these different knowledge sources, the MGNSP model employs a multi-head attention mechanism (Vaswani et al., 2017).

Along the same lines, Goyal et al. (2023) analyzed the prediction of missing skills in job descriptions from job boards. Using data from about 40,000 job offers and a set of about 2,500 skills, they proposed a multi-label classifier to predict the most likely missing set of skills for a given job description. Their aim was to improve the accuracy of qualification requirements in job postings, thus contributing to the overall improvement of the job search process.

In contrast to previous works, our work focuses on the exploration of information obtained from professional resumes. We analyze the potential connections between millions of jobs and a taxonomy covering approximately 5,500 different skills. We evaluate the effectiveness of the proposed system in several scenarios, including the evaluation of its capabilities under cold recommendation cases. This extensive evaluation allows us to gain insights on the system's adaptability and efficacy under different conditions.

3 GNNs for Skills and Job Titles

In this section we start by presenting the retrieval tasks addressed by the proposed model, as well as the datasets used for evaluation. We then present the model along with details on its training.

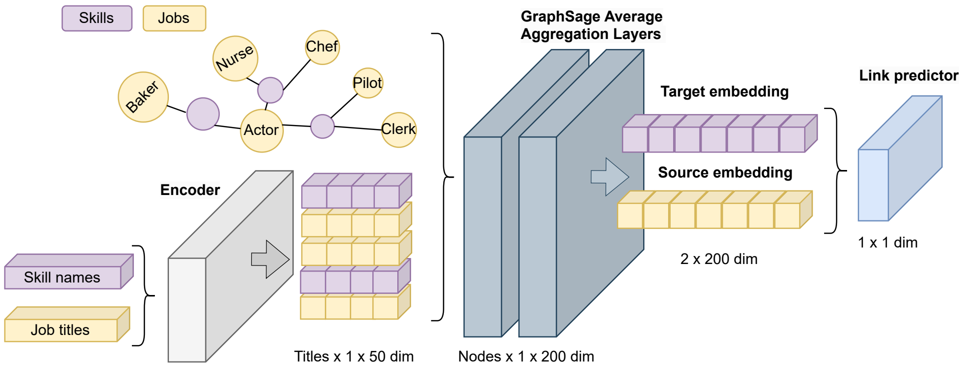

Figure 1: Proposed model. First, the encoder converts the skill names and job titles into embeddings. Those embeddings and the graph information are both input into the GNN layers. Finally, the link prediction component aggregates the information from source and target nodes to predict the scores of potential links.

3.1 Applications & Evaluation Dataset

We investigate two tasks related to the optimization of human capital management processes:

- **Job-Job Matching** explores the similarities between diverse job titles. Given a job title, the task is rank other job titles reflecting the semantic similarity between the query and the retrieved items.
- **Job-Skill Prediction** covers another aspect of human capital management, namely the automatic discovery of skills relevant to a given job title. We also define this task as a ranked retrieval of a set of items, namely the skills.

In both cases no distinction is made between different kinds of similarity relationships, e.g., “required” vs. “preferred”.

For the job title similarity experiments, we used the dataset proposed by Zbib et al. (2022), while we designed a new dataset to explore the similarity between jobs titles and skills. To build this dataset, we manually select a set of job titles covering different jobs and industries.

We retrieved the most frequent skills associated with each job from a large collection of English professional resumes. This initial retrieval is done by matching the literal skills against an in-house skills taxonomy, similar in scope to standard taxonomies like O*NET[1]. For this initial step, we used regular expressions and simple matching rules. Then, the extracted skills are filtered further by two human annotators who select the 25 most representative skills for each job. The final dataset contains 2,535 instances, each of which corresponds to a unique <job_title/skills_set> tuple.

3.2 Proposed Model

The main components of the proposed approach are shown in Figure 1. As can be seen, the model receives on the one hand a bipartite graph representing the relationships between jobs and skills, and on the other hand, the titles associated with each node i.e., skills names and job titles. While the titles are processed by an encoder for the generation of text embeddings, the graph is used by the GraphSage model to exploit the generated embeddings and the relationships explicitly mentioned in the graph. Finally, for each <target_node, source_node> tuple, a neural link predictor is used to estimate the probability of a relationship (link) between them two nodes.

The data used in this work is based on information extracted from professional resumes. We extract job titles and the related skills contained in each document. In other words, for each CV, we generate several tuples containing the job titles and the skills found in same context. We apply the resume parser of Retyk et al. (2023) to process around 22 million resumes from a combination of private and public sources, and retrieve more than 11 million unique job titles

[1]Skills Search. O*NET OnLine, National Center for O*NET Development, www.onetonline.org/skills/. Accessed November 29, 2023

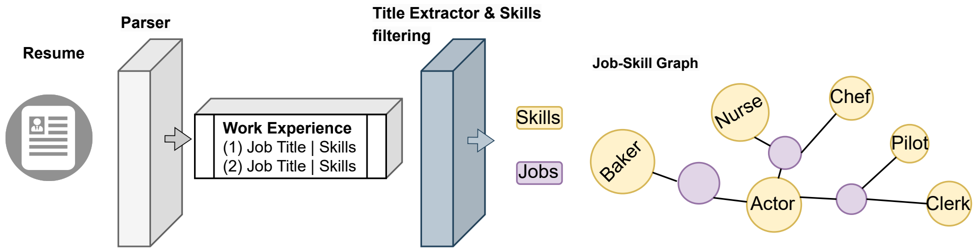

Figure 2: Pre-processing pipeline showing the different steps used for the generation of the Job-Skill graph.

as a result. It is worth noting that the same process could be equally applied to information extracted from job descriptions. Figure 2 shows a view of the pre-process applied to extract the information from resumes and generate the Job-Skills graph.

Since many job titles only differ by modifiers that occur infrequently, such as location, department or working schedule, we apply an extractive summarization process to remove such modifiers and combine information from the resulting data items. More specifically, we apply a trained sequence tagger model to distinguish between relevant and non-relevant tokens within a given job. The deep learning model uses a classical BiLSTM+CRF architecture, consisting of a stack of two bidirectional recurrent neural networks (64, tanh), followed by a CRF. To pre-process the input titles we use a custom SentencePiece model trained with millions of job titles (Kudo and Richardson, 2018), with 50-dimensional embeddings. The model was trained with a small dataset generated automatically, using a large set of regular expressions design to eliminate noisy modifiers. The generated dataset was then partially reviewed by two annotators. We trained the model using as input the raw text of each job title and as output the IO (Inside, Out) annotation indicating whether a token is relevant or not. This model was trained for 40 epochs, using Adam optimizer and a learning rate of 1e-3.

After applying the normalization step to the full corpus, the 100,000 most frequent job titles were then selected to build the graph.

For skills we use an in-house taxonomy that covers more than 5,000 unique skills, and includes several alternate names for each skill.

After normalizing and filtering the retrieved job titles and skills, we aggregate the information based on job titles, and merge the skills and counts associated with all instances of the same job title. Considering job objects as documents, and associated skills as corresponding vocabulary, we use the TF-IDF of the skills as weights to the links between the job title nodes and the skill nodes in the graph. For each job title, only the top 50 associated skills sorted by TF-IDF are kept. Figure 2 illustrates the Encoder+GraphSage model, which is the main components of the proposed pipeline.

Initial Graph Representation

We exploit the semantic features contained in the textual information of the nodes in the graph (for both job titles and skills) by representing each node's text using embeddings generated from the Multilingual Universal Sentence Encoder (mUSE) (Yang et al., 2020). In the proposed method, these textual embeddings are used as initial features for the graph nodes. However, for comparison, we also explore the performance on the Job-Job Matching task obtained from using the mUSE embeddings as final representations for ranking. We use two variations: the original mUSE representation, as well a mUSE model retrained with in-domain data. To re-train the model, we used the approach proposed by Zbib et al. (2022). While the original mUSE generates embeddings of 512 dimensions, the refined version produces a representation of 50 dimensions only, by adding an activation-free Dense recurrent layer to the original model.

The encoder was trained using a multilingual dataset of job titles and associated skills similar to the one we use for constructing the graph (see above).

We note the parallelism between the jobs-skills graph on one hand and the jobs-skills representations used to fine-tune the encoder

(Zbib et al., 2022). However, a crucial difference is that the graph approach exploits the structural information contained in the relationship between the jobs and skills explicitly. The advantage of this is shown in the improved performance of the graph on the Job-Job Matching task. As importantly, the graph enables the task of Job-Skill Prediction, and the cold-start scnenario of modeling unseen skills.

Graph Neural Network and Model Training

The GNN is the backbone of this work's proposed method. We choose to use GraphSage (Hamilton, Ying, and Leskovec, 2017) for this component, because of its reported outstanding generalization capability, which allows us to easily incorporate new graphs and unseen nodes. We motivate this choice by the model's versatility and adaptability, which make it ideally suited to the dynamic challenges posed by evolving datasets and new graph structures.

For each variant in our experiments, we train the model during 80 epochs, using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 1e-3. The model configuration follows the guidelines provided by the GraphSage paper, where for each node the graph structure is explored for up to 2 hops (denoted by $K = 2$). This exploration is done with a random sampling strategy, incorporating neighborhood sizes of $S_1 = 20$ and $S_2 = 10$ for the first and second hop respectively. As mentioned earlier, we weight the relevance of each edge using the TF-IDF calculated with the co-occurrence of jobs (source) and skills (target). This sampling strategy captures relevant contextual information within the specified neighborhood size.

The GNN model generates a 200-dimensional embedding space for each node. This embedding space serves as the basis for downstream processing, where we pipe the dot product for any given tuple of nodes (source, target) into a multi-layer perceptron (named Link Predictor). The Link Predictor plays a crucial role in assessing and ranking the relevance of potential relationships between any pair of nodes.

4 Experimental Setup

The tasks addressed in this paper were defined in Section 3.1 as ranking tasks. In this section, we present the details of the experimentation carried out, as well as the results obtained in terms of metrics commonly used for similar tasks: MAP and Precision@K. While with Precision@K we limit the analysis to the number of relevant items in the first K positions, the MAP metric models the quality of the ranking in global terms by considering as preferable a ranking that places the relevant items in the first positions. We report on results obtained from the proposed approach and compare them with different baselines based on both Random Walks and Sentence Encoders.

Sentence Encoder Baseline. The use of the mUSE encoder is a key component in the generation of embeddings that serve as input (i.e., initial features) to the proposed GNN model. To gauge the effectiveness and added benefits resulting from the integration of GraphSage as a new component, a meticulous evaluation of the performance of the mUSE-generated embeddings is important. For this purpose, we compare the performance results before and after the tuning process, versus the results obtained by the proposed GNN model. The aim is to provide a comprehensive understanding of the combined impact of these encoding strategies on the model's overall performance.

Random Walk SkipGram Baseline. We also study the performance of another baseline obtained using a SkipGram model trained on representations generated by Random Walks. We generate training sentences using the 50 top skills for each respective job title sorted by TF-IDF. A corpus of 1.5 million sequences, each consisting of 200 elements, was produced by interleaving the job titles and their associated skills. For the generation of random walks we use the method proposed by Sajjad, Docherty, and Tyshetskiy (2019), in which the conditional likelihood of each node is taken into account. This algorithm enriches the learning process by incorporating contextual dependencies, thus improving the overall quality of the generated representations. We use these generated sequences to train a Word2Vec model over 4 epochs with the goal of generating a 50-dimensional representation space for each job and skill (additional parameters = {negative_sampling: 5, window_size: 5}).

Model	MAP	P@5	P@10	P@20
Proposed model - mUSE	0.697	0.7923	0.6798	0.524
Proposed model - FT mUSE	**0.7329**	**0.8269**	**0.7144**	**0.5433**
mUSE	0.3913	0.5404	0.451	0.3399
FT mUSE	0.6904	0.7942	0.6788	0.5293
Random Walk - SkipGram	0.7021	0.7923	0.674	0.537

Table 1: Results on retrieval of Links between Job titles. The best results are shown in bold. {FT: Fine-tuned - P: Precision - MAP: Mean Average Precision}

Model	Evaluation	MAP	P@5	P@10	P@20
Proposed model - mUSE	Cosine Sim.	0.1932	0.3503	0.3181	0.2759
	Link Predictor	0.2506	0.4511	0.3903	0.327
Proposed model - FT mUSE	Cosine Sim.	0.2782	0.4813	0.4282	0.355
	Link Predictor	**0.2908**	0.4945	0.4412	**0.3693**
mUSE	Cosine Sim.	0.0805	0.2743	0.199	0.1392
FT mUSE	Cosine Sim.	0.2737	**0.5347**	**0.4478**	0.3558
Random Walk - SkipGram	Cosine Sim.	0.2402	0.3997	0.3639	0.3159

Table 2: Results on link retrieval between Job titles and Skills. The best results are shown in bold. {FT: Fine-tuned - P: Precision - MAP: Mean Average Precision}

.

4.1 Job-Job Matching: Results

Table 1 shows the results for the retrieval of similar (or related) occupations. The jobs are ranked according to the cosine similarity between their embeddings and that of the query job title. We note first that the performance of the GraphSage-based model results very large improvements over using the mUSE-based representations. The fine-tuned encoder achieves similar results to those of the GraphSage model using the original mUSE model. Although both models are conceptually different, in both cases representations of the job titles are generated taking into account the related skills. However, some relationships between nodes that are not directly connected are considered more accurately by the proposed model which combines both the fine-tuned encoder and GraphSage. We see this by the further improvements obtained from combining both over both the GraphSage and the fine-tuned mUSE models.

We also note that results based on the embeddings generated by Random Walks and SkipGram are slightly better than those corresponding to the re-trained mUSE encoder. It could be considered as an alternative to the encoder used to represent the information of each graph node based on the results of this task. However, it is important to note that, unlike the encoder approach, the Random Walks+SkipGram approach cannot easily generalize to new job titles that are not seen in the training data, as the embeddings would have to be generated statically for each job title.

4.2 Job-Skill Prediction: Results

The results obtained from the analysis of the relationships between job titles and skills are detailed in Table 2. We evaluate the performance of the proposed GNN model on the Job-Skill Prediction task using two methods: similarity between generated embeddings (cosine similarity) and the link predictor score.

We first note that in all cases, using the link predictor scores achieves better results than with the cosine similarity between embeddings. The classifier-based link predictor seems to perform more effectively with borderline links than methods solely based on the analysis of the similarity between embeddings.

Another interesting observation is that fine-tuning the mUSE encoder with job titles data improves the results substantially over using the original mUSE encoder. However, the results of GraphSage with original mUSE are not as good, unlike in the job-job matching case. For this task, GraphSage training alone cannot therefore be considered an alternative to the fine-tuning of the mUSE encoder. This is likely due to the manual heuristic that was used to generate the training graph, which might be too conservative, giving too much importance to potentially noisy relationships, thus affecting the accuracy of the model.

5 *Cold Start: Dealing with Unseen Skills*

Cold start recommendations are a major challenge in the field of recommendation systems. In this scenario, the system is presented with an item for which it has limited or no data. In light of the constant fast-paced evolution of the labor market, and the corresponding landscape of professional competencies, a more realistic assessment of a recommendation system like the one we are presenting here requires an evaluation under "cold start" conditions.

In this section, we present the results of experiments we performed for this scenario, and present a detailed analysis under multiple data ablation settings. We propose two different evaluations, where we first evaluate the behavior of the model when the graph is presented with new skills that are unseen during training, and then we deepen the analysis by removing all information about these new skills.

- **Evaluation Scenario 1**: Systematic elimination of different skill sets during training. For each setting, we evaluate whether the model is able to deal with new graphs where skills not seen during training are included.
- **Evaluation Scenario 2**: Given the first scenario, a random sub-sampling of information related to new skills (edges sub-sampling). Instead of considering all available edges during the test stage, a small sample of edges is randomly selected to be included. This scenario is designed to analyze the amount of information needed to obtain sufficiently good results when dealing with skills not seen during training.

5.1 Evaluation Scenario 1

Since we use a weighted graph in which the associations between job titles and competencies are characterized by the corresponding TF-IDF values, in this case we use an estimation of the weights in order to accurately reproduce a cold start scenario. Thus in this evaluation scenario, to weight the links between job titles and unseen skills, we use the average of the TF-IDF values associated with the links present in the training set. In the extreme where all elements of the test graph were not seen during training, all links will have a homogeneous weighting.

5.1.1 Job-Job Matching: Results

Table 3a outlines the ablation study performed on the dataset, which aims to comprehensively evaluate the performance of the proposed model on the task of recovering similar jobs. At an aggregate level, by looking at the MAP score we note a remarkable robustness of the model against the removal of links and competencies from the training. However, a closer inspection of the results reveals a further deterioration of the quality at the top positions of the generated ranking. This highlights the importance of certain relationships within the training graph which have a disproportionate impact on the model's ability to accurately prioritize and retrieve the most similar positions. On the other hand, the average TF-IDF value of the training edges shows to be a good estimator for weighting unseen links. In many of the explored cases, we did not detect a strong degradation of the results; we even observed improvements in the most extreme cases, where more than 50% of the test set skills were removed from the training.

5.1.2 Job-Skill Prediction: Results

We next discuss the results of the ablation study performed on the skill retrieval task. While Table 3b shows the results obtained using the generated embeddings, Table 3c shows the results obtained using the link predictor as a ranker. Same as in Section 4.2, the best results were obtained using the link predictor. This approach not only yields interesting improvements, but also shows greater robustness to the changes introduced during the processing of unseen skills. Particularly noteworthy are the results observed in scenarios where more than 50% of the test set skills were completely removed from the training. In these cases, the link predictor generates rankings that show consistent performance and an acceptable capability to work with unseen competencies.

Regarding edge weighting, interesting findings have been obtained when dealing with unseen skills. In this case, the use of the average, as opposed to the actual TF-IDF, shows a remarkable improvement. This

Test skills removed from training	# Training edges	MAP (Δ)	P@5 (Δ)	P@30 (Δ)
25%	4.02 M	0.7053 (-0.027)	0.7673 (-0.059)	0.4385 (-0.001)
50%	3.09 M	0.6971 (-0.035)	0.7654 (-0.061)	0.4343 (-0.002)
75%	2.12 M	0.6989 (-0.034)	0.7711 (-0.055)	0.434 (-0.002)
95%	1.35 M	0.691 (-0.041)	0.7558 (-0.071)	0.433 (-0.003)

(a) Job-Job matching using **generated embeddings**.

Test skills removed from training	# Training edges	MAP (Δ)	P@5 (Δ)	P@30 (Δ)
25%	4.02 M	0.272 (-0.006)	0.4742 (-0.007)	0.302 (-0.006)
50%	3.09 M	0.2686 (-0.009)	0.4641 (-0.017)	0.2994 (-0.008)
75%	2.12 M	0.2487 (-0.029)	0.4146 (-0.066)	0.2842 (-0.025)
95%	1.35 M	0.2216 (-0.056)	0.3574 (-0.123)	0.2593 (-0.048)

(b) Job-Skill matching using **generated embeddings**.

Test skills removed from training	# Training edges	MAP (Δ)	P@5 (Δ)	P@30 (Δ)
25%	4.02 M	0.2862 (-0.004)	0.4861 (-0.008)	0.3129 (-0.006)
50%	3.09 M	0.2774 (-0.013)	0.4624 (-0.032)	0.3094 (-0.009)
75%	2.12 M	0.283 (-0.007)	0.468 (-0.026)	0.3119 (-0.007)
95%	1.35 M	0.2693 (-0.021)	0.4216 (-0.072)	0.3101 (-0.008)

(c) Job-Skill matching using **the link predictor**.

Table 3: Cold Start - Evaluation Scenario 1. Results obtained by the proposed model after eliminating edges from the training graph (original # Training edges: 4.96 M ; # Test skills: 3738). The average TF-IDF of the training skills is used as an estimate to weight links of unseen skills. The difference in scores with the ideal scenario, where all skills are known, is shown in parentheses.

Test skills removed from training	Sampled Oracle edges	MAP (Δ)	P@5 (Δ)	P@30 (Δ)
25%	10 edges per removed skill	0.2596 (-0.018)	0.4472 (-0.034)	0.3684 (0.06)
25%	1 edge per removed skill	0.2213 (-0.056)	0.4137 (-0.067)	0.2676 (-0.04)
50%	10 edges per removed skill	0.2499 (-0.028)	0.4328 (-0.048)	0.3584 (0.05)
50%	1 edge per removed skill	0.1744 (-0.103)	0.3660 (-0.115)	0.2914 (-0.016)
95%	10 edges per removed skill	0.2052 (-0.073)	0.3741 (-0.107)	0.2480 (-0.06)
95%	1 edge per removed skill	0.0121 (-0.266)	0.0405 (-0.440)	0.0383 (-0.269)

Table 4: Cold Start - Evaluation Scenario 2: Job-Skill matching task. Results obtained by the proposed model after eliminating edges from the training graph. During the test and for each skill not seen during training, a set of related edges is randomly sampled. The results shown are obtained using the **generated embeddings**. The difference from the ideal scenario where all skills are known is shown in parentheses.

improvement may be due to the fact that certain skills may not be sufficiently represented in the processed professional resumes, so we may be generating edge weights that may underestimate the relevance of certain skills. By using the average, a more balanced representation of the skills relevance is achieved, which helps to increase the overall performance.

5.2 Evaluation Scenario 2

While the results obtained during the Evaluation Scenario 1 showed the robustness of the model when faced with new populated graphs at test time, this scenario carries a more complex evaluation that measures the effect of the amount of information on the correct prediction of skills unseen in the training data.

Table 4 shows the corresponding results, where the oracle information (i.e., the full data) of skills not included during training was randomly sub-sampled. We remove 25%, 50% and 95% of the test skills from the training data. With up to 50% of the test skills removed, and with only 10 randomly sampled edges for each unseen skill, the proposed model shows a negligible degradation of less than 0.1 MAP in comparison with the ideal scenario. However, a more pronounced performance drop can be observed when the more challenging condition of using only one edge per unseen skill.

In short, while the results indicate that a certain amount of information is needed to adequately process new skills, that amount is significantly smaller than the information used for modeling the well-known skills during training.

6 Conclusions

This paper presents a novel approach based on Graph Neural Networks to address a set of tasks related to human capital management. The proposed approach combines the advantages of representing job titles and skills in an embedding space using an encoder, together with the capabilities of GNNs for the explicit representation of the structure of relationships between jobs and skill. In that sense, this work is an extension of previous work where only textual information was used. Our experiments showed that combining textual information and latent relationships between the skills and job titles (through the encoder) together with explicit relationships between these entities (through the GNN) results in better performance than either approach by itself.

We also evaluated the robustness of the model against perturbations to a complete knowledge graph based built with all the training information available. The model shows a high generalization capacity with a limited level of degradation even in the extreme case where up to 95% of test skills are ablated from the training. We also examined the effect of removing information of link weights and showed that there are no strong dependencies that make it difficult to deal with unseen nodes. In summary, our experiments demonstrate the usefulness and adaptability of this mode, which is able to represent complex relationships between job titles and skills in an efficient way.

A limitation of this work is the difficulty of annotating a test set for the job-skill matching task comprehensively, where all the potentially relevant skills for each job title are examined. In future work we will try to increase the coverage of this test set, not limiting it to 25 skills per job title.

Skills are of great value for understanding the relationships between job titles, but are not the only concepts useful for the modeling of human capital. Other types of entities such as educational background fill gaps that cannot be addressed by the study of skills alone. Also, since some skills are strictly related to the industry and not so much to the job title, their modeling could benefit from the integration of industry-related information as a new type of node. Finally, we also plan to extend this exploration to cover tasks such as career path modeling and skill gap retrieval. We hope thus to be able to extend and improve the semantic representations of concepts related to human capital.

References

Goyal, N., J. S. Kalra, C. Sharma, R. Mutharaju, N. Sachdeva, and P. Kumaraguru. 2023. JobXMLC: EXtreme Multi-Label Classification of Job Skills with Graph Neural Networks. In A. Vlachos and I. Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2136–2146. Association for Computational Linguistics.

Grover, A. and J. Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 855–864. ACM.

Hamilton, W. L., Z. Ying, and J. Leskovec. 2017. Inductive Representation Learning on Large Graphs. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034.

Kingma, D. P. and J. Ba. 2015. Adam: A Method for Stochastic Optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA,* *USA, May 7-9, 2015, Conference Track Proceedings.*

Kudo, T. and J. Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In E. Blanco and W. Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November. Association for Computational Linguistics.

Li, X., L. Sun, M. Ling, and Y. Peng. 2023. A survey of graph neural network based recommendation in social networks. *Neurocomputing*, 549:126441.

Liu, L., W. Zhang, J. Liu, W. Shi, and Y. Huang. 2021. Learning Multi-Graph Neural Network for Data-Driven Job Skill Prediction. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pages 1–8. IEEE.

Manyika, J., S. Lund, M. Chui, J. Bughin, J. Woetzel, P. Batra, R. Ko, and S. Sanghvi. 2017. Jobs lost, jobs gained: What the future of work will mean for jobs, skills, and wages.

Mikolov, T., K. Chen, G. Corrado, and J. Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In Y. Bengio and Y. LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings.*

Perozzi, B., R. Al-Rfou, and S. Skiena. 2014. Deepwalk: online learning of social representations. In S. A. Macskassy, C. Perlich, J. Leskovec, W. Wang, and R. Ghani, editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 701–710. ACM.

Retyk, F., H. Fabregat, J. Aizpuru, M. Taglio, and R. Zbib. 2023. Résumé Parsing as Hierarchical Sequence Labeling: An Empirical Study. In M. Kaya, T. Bogers, D. Graus, C. Johnson, and J. Decorte, editors, *Proceedings of the 3rd Workshop on Recommender Systems for Human Resources (RecSys in HR 2023) co-located with the 17th ACM Conference on Recommender Systems (RecSys 2023), Singapore, Singapore, 18th-22nd September 2023*, volume 3490 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Rosvall, M. and C. T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, January.

Sajjad, H. P., A. Docherty, and Y. Tyshetskiy. 2019. Efficient Representation Learning Using Random Walks for Dynamic Graphs. *CoRR*, abs/1901.01346.

Störmer, E., C. Patscha, J. Prendergast, C. Daheim, and M. Rhisiart. 2014. *The Future of Work: Jobs and Skills in 2030.* UK Commission for Employment and Skills, February.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Wu, Y., Y. Chen, Z. Yin, W. Ding, and I. King. 2023. A survey on graph embedding techniques for biomedical data: Methods and applications. *Information Fusion*, 100:101909.

Yang, Y., D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Ábrego, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil. 2020. Multilingual Universal Sentence Encoder for Semantic Retrieval. In A. Celikyilmaz and T. Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 87–94. Association for Computational Linguistics.

Zbib, R., L. L. Alvarez, F. Retyk, R. Poves, J. Aizpuru, H. Fabregat, V. Simkus, and E. G. Casademont. 2022. Learning Job

Titles Similarity from Noisy Skill Labels. *CoRR*, abs/2207.00494.

Zhang, D., J. Yin, X. Zhu, and C. Zhang. 2019. Attributed network embedding via subspace discovery. *Data Mining and Knowledge Discovery*, 33(6):1953–1980.

Zhu, G., Y. Chen, and S. Wang. 2022. Graph-Community-Enabled Personalized Course-Job Recommendations with Cross-Domain Data Integration. *Sustainability*, 14(12).