Combined Unsupervised and Contrastive Learning for Multilingual Job Recommendation

Daniel Deniz, Federico Retyk, Laura García-Sardiña, Hermenegildo Fabregat, Luis Gasco and Rabih Zbib

Avature Machine Learning

Abstract

The transformative power of artificial intelligence is revolutionizing the talent acquisition domain. Automatic job recommendation systems are emerging as a key component of this transformation. This study presents a new multilingual job recommendation solution that leverages combined unsupervised and contrastive learning to effectively model the semantic similarity between job titles across 11 languages. Our approach pre-trains a multilingual encoder using unsupervised learning on co-occurrence information of skills and job titles, followed by fine-tuning via contrastive learning on a dataset of similar and dissimilar job pairs based on the European Skills, Competences, Qualifications and Occupations (ESCO) taxonomy. This sequential learning strategy significantly enhances representation quality. Our novel multilingual job title encoder achieves strong ranking results across all languages, with 4.3% improvement in mean Average Precision (mAP) for English compared to previous state-of-the-art monolingual solutions. The proposed method also offers very good cross-lingual capabilities, enabling the ranking of jobs in different languages with improved alignment and uniformity properties in the representation space.

Keywords

Recommender Systems, Deep Learning, Contrastive Learning, Multilingual Semantic Textual Similarity, Job Title Ranking

1. Introduction

In today's fast-paced job market, organizations face an increased competitive pressure to find and retain the best suitable skilled candidates [1]. The introduction of Information and Communication Technologies (ICTs) in Human Resources (HR) processes in recent years, and especially with more recent advancements in artificial intelligence (AI) has helped in alleviating this pressure in time-sensitive recruitment processes, especially those that have relied heavily on human specialists [2]. The integration of AI and machine learning has been transforming the recruitment process by allowing companies to rapidly process and analyze large amounts of candidate data, helping identify top talent with greater precision, and supporting data-driven decisions to optimize and refine recruitment strategies [3, 4]. A key aspect of this transformation is the emergence of automatic job recommendation systems. Job seekers also benefit from these innovations, as they can find better tailored job recommendations within the large amount of job postings available on online platforms, making it easier for them to find and apply to opportunities that suit their skills and their career goals.

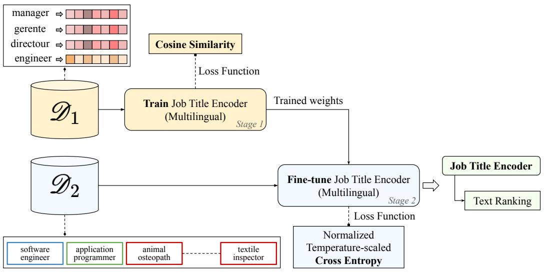

Figure 1: General overview of the proposed two-step method to build a multilingual Job Title Encoder for intelligent job recommendation. The process starts from a pre-trained encoder, and we propose two steps: 1) Unsupervised training: we define target representations ($\mathscr{D}_1$) for every job based on their skills distribution, and then fit the encoder for mapping jobs into these representations. 2) Contrastive training: we use synonyms and relations from ESCO ($\mathscr{D}_2$) to fine-tune the encoder with contrastive learning techniques.

Driven by recent technical advancements in semantic textual similarity (STS), automatic job recommendation systems are increasingly relying on this task to develop matching algorithms that accurately pair job vacancies with candidates based on their skills and experience [5, 6, 7]. Novel STS approaches leverage neural networks to automatically generate vector representations that effectively capture the semantic nuances of text elements, thus significantly enhancing matching performance compared to traditional methods [8].

The methods for training encoders for STS can be broadly divided into two approaches: supervised methods, and unsupervised methods that do not require any

RecSys in HR'24: The 4th Workshop on Recommender Systems for Human Resources, in conjunction with the 18th ACM Conference on Recommender Systems, October 14–18, 2024, Bari, Italy.

machinelearning@avature.net (D. Deniz)

CEUR Workshop Proceedings (CEUR-WS.org)

labeled data. Examples of unsupervised procedures applied to the HR domain were proposed by Zbib et al. [6] and Lavi et al. [9]. They do not rely on human annotations, and instead, they use *noisy* data automatically collected from resumes and job listings. Conversely, supervised methods rely on large amounts of labelled data to train, which can be sometimes hard to obtain in terms of quality and quantity [10]. Contrastive learning has shown very strong results for STS as a technique that helps improve the learned representations and increase the training data efficiency [11, 12, 13].

Our paper builds upon these advances and presents a multi- and cross-lingual intelligent job recommendation solution that relies on STS to measure the similarity between job titles across 11 languages (English, German, French, Spanish, Italian, Dutch, Portuguese, Polish, Japanese, Chinese, and Korean). We introduce a novel two-stage methodology for training an encoder focused on job title semantic similarity for text ranking (see Figure 1).

In the first stage we pre-train a multilingual encoder following an unsupervised approach using *noisy* data automatically collected from job postings and resumes. In the second stage we fine-tune the encoder via contrastive learning on a dataset with pairs of relevant and non-relevant jobs based on the European Skills, Competences, Qualifications and Occupations taxonomy (ESCO) [14, 15]. The ESCO occupations taxonomy is a hierarchically structured standard classification that organizes jobs into groups and concepts based on their similarity in terms of skill level and required specialization. For instance, ESCOXLM-R [16] work has demonstrated the effectiveness of using the ESCO taxonomy for adapting a multilingual language model to the job market domain.

We study the effectiveness of the proposed methodology by training various encoders with different backbone architectures (Transformer-based vs. CNN-based) and original pre-training objectives (semantic-similarity vs. masked language modeling). We find that trained encoders outperform state-of-the-art benchmarks on the job title ranking task. Moreover, the proposed solution shows very good multi- and cross-lingual capabilities, achieving ranking performance results for each language that are remarkably close to English, the language with the best results. Notably, the encoders exhibit strong alignment and uniformity properties in cross-lingual evaluations.

2. Related Methods

Job recommendation systems are a vital component of modern employment platforms that help match job seekers with relevant opportunities. Traditional systems are based on matrix factorization and collaborative filters [17, 18]. However, these approaches suffer from data sparsity and the cold start problem, especially when making recommendations in the case where a user has few interactions with other items [19].

Recent advancements in this field leverage techniques from machine learning, natural language processing (NLP), and AI. The incorporation of content-based embedding strategies in recommendation systems has led to the development of efficient and reliable item retrieval methods, offering significant benefits in terms of generalization, scalability, and cold start capabilities [20].

2.1. Content-based Job Recommender Systems

In the existing literature, numerous studies on embedding-based recommender systems are based on Deep Learning techniques. For instance, Zhao et al. [20] proposed a Convolutional Neural Network (CNN) with an attention layer as a key component in their recommendation system to encode job titles, skills, and other context-aware elements into a dense-vector representation for ranking candidates with job vacancies.

In [7], authors propose a method to train a job title encoder in which they fine-tune a pre-trained language model with an auxiliary classification loss, using job-skill co-occurrence statistics extracted from vacancies. Zbib et al. [6] first build dense-vector representations for jobs based on their distribution of skills (also extracted from vacancies), which are expected to encode the semantics of the job. Then, they train the encoder to map job title surface forms to such representations. At inference time, the encoder only sees the surface form of (potentially new) jobs, and cosine similarity is used to measure the semantic distance between encoded occupations. These approaches are based on the premise that skills are essential to understand the meaning of a job, and therefore, semantically similar job titles should exhibit similar skill profiles.

Similarly, [21] develops a novel approach to train an encoder derived from BERT [22] to link skills with occupations by leveraging co-occurrence information via a contrastive learning method.

2.2. Contrastive Learning methods for Semantic Textual Similarity

In recent years, contrastive learning has emerged as a powerful technique for learning high-quality sentence embeddings. This technique was initially popularized in computer vision [23]. It focuses on learning representations by distinguishing positive pairs (similar data points) against negative pairs (dissimilar data points). Models are trained to bring the representations of positive pairs

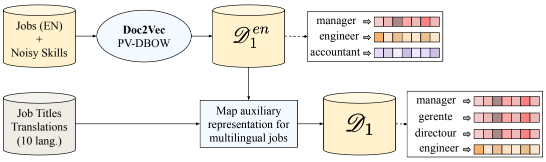

Figure 2: Creation of Multilingual Dataset from Noisy Skills ($\mathscr{D}_1$) for Unsupervised training of Job Title Encoder. The building process for $\mathscr{D}_1^{en}$ is similar to the one depicted in [6].

closer together while pushing the representations of negative pairs farther apart in the embedding space [24, 25].

Contrastive learning techniques can help alleviate the anisotropy problem and improve representation performance in STS tasks [26, 11, 13]. The anisotropy problem refers to the issue where the vector representations do not uniformly occupy the embedding space. Instead, they might cluster in a narrow cone of the space, leading to an uneven distribution that diminishes the model's ability to distinguish between different data points [27, 28].

The adaptation of contrastive learning to NLP has led to the development of several solutions in the context of STS for measuring how closely the meaning of two sentences align. Works in this area often rely on pre-trained language models, such as BERT [22] or RoBERTa [29], as the backbone for generating initial sentence embeddings [30]. As in SimCSE [11] or ConSERT [31], contrastive learning is then applied to fine-tune these embeddings, improving their ability to capture semantic similarity, in STS tasks.

The design of the loss function is crucial in contrastive learning, and has undergone a significant evolution from Siamese pair loss [32], triplet loss [33], to alternatives with multiple negatives such as N-pair loss function [34], or NT-XENT (Normalized Temperature-scaled Cross Entropy) [23], which can lead to improved representation learning [11].

In this work, we leverage the relationship between jobs and skills to build auxiliary representations that later guide the training of the multi- and cross-lingual job title encoder. Then, we fine-tune the encoder using a contrastive learning loss with multiple negatives to improve matching and text ranking between candidates and job vacancies.

3. Our Approach

In this work, we propose a novel approach for training a semantic textual similarity solution for intelligent job recommendation. Then, we describe how we build an encoder model for multilingual job title ranking that operates across 11 languages.

Our main contribution is the two-step training procedure depicted in Figure 1. In Step 1, similarly to the process presented in [6], we use doc2vec [35] and the skills distribution for every job title to create auxiliary dense-vector representations. These are in turn used to train an encoder that maps the surface form of each job title to its reference doc2vec representation. Since auxiliary representations encode language-agnostic semantics, we expose the encoder to job surface forms in various languages, all mapped to the same point in the representation space, resulting in a multi- and cross-lingual encoder. Then, in Step 2, we use contrastive learning to fine-tune the multilingual encoder obtained in the previous step. In this case, we use pairs of similar and dissimilar jobs derived from the ESCO taxonomy.

Below we describe how the data used in the different stages of this approach was prepared. Then, we describe the procedure for training the neural network in each stage to build the **Multilingual Job Title Encoder** for automatic job title ranking.

3.1. Data Preparation

In the first place, we build the different datasets that will take part in the training of the multilingual encoder. These datasets have different origins, structure, and properties.

We construct $\mathscr{D}_1$, a multilingual dataset of job positions by linking the surface form of job occupations to a dense-vector representation. These representations are based on the distribution of *noisy* skills related to each job title collected from a collection of job postings and the work experience sections of anonymized resumes. In this dataset we include data from 11 different languages: English (**en**), German (**de**), Spanish (**es**), French (**fr**), Italian (**it**), Japanese (**ja**), Korean (**ko**), Dutch (**nl**), Portuguese (**pt**), Polish (**pl**), and Chinese (**zh**). Bear in mind that we do not require any manual annotation for building this dataset (unsupervised).

The other dataset is $\mathscr{D}_2$, which we compile as a collection of multilingual pairs of similar and dissimilar job titles, based on relations extracted from the ESCO taxonomy. This dataset includes the same languages as the ones described before, except for the three Asian languages (Japanese, Korean, and Chinese) since they are not included in ESCO. Our experiments show that the contrastive learning step benefits these languages through cross-lingual transfer despite not being included in the training data of that stage.

3.1.1. Job Title Similarity from Noisy Skills Dataset

The $\mathscr{D}_1$ dataset described above is used to learn the semantic relations between job occupations based on the co-occurrence distribution of skills for each job title. The knowledge provided by the skills is transferred to an encoder, enabling us to capture the semantic similarity between these job occupations based on their surface forms.

The process of creating this dataset consists of two steps. In the first one, we start from a collection of job titles and associated *noisy* skills for each occupation. We consider these skills as *noisy* since they are retrieved in an automated way by simple string matching from English job postings and anonymized resumes. The dataset, sourced from proprietary real data, covers a broad range of industries, including manufacturing, healthcare, tech, finance, or banking, among others. Next, we follow a similar procedure to that presented in Zbib et al. [6], using PV-DBOW Doc2vec to create a dataset with pairs of job titles and fixed-length vectors (512 dimensions). These vectors are the auxiliary representations that will be used as synthetic targets in the first training stage. We refer to the resulting dataset as $\mathscr{D}_1^{en}$.

As shown in Figure 2, the next step is to expand the pairs of job titles and fixed-length vectors to the other languages. To do this, we translate a subset of the English job titles (starting with around 2.3 Million) to the target languages using Machine Translation[1]. Afterwards, we include in the previous dataset new pairs with the surface lexical form of the resulting translations, but with the same vector representation as the corresponding source English.

The same job title in different languages will have the same auxiliary representation (Eg. *manager* (en), *gerente* (es), and *directeur* (fr) will share the same doc2vec vector). By this, we aim to foster the multi- and cross-lingual capabilities of the proposed solution for text retrieval in Human Resource applications based on job title semantic similarity. We refer to this multilingual dataset as $\mathscr{D}_1$. The final dataset has 14.5 Million English job titles and an average of 2.3 Million job occupations per each of the other languages.

3.1.2. Similarity Pairs from ESCO Dataset

Secondly, we prepare the $\mathscr{D}_2$ dataset used for further fine-tuning the encoder using multilingual pairs (English, German, French, Spanish, Italian, Dutch, Portuguese, and Polish) of similar and dissimilar job titles, aiming to extract relations between job occupations that can help improve the matching performance for automatic job recommendations.

We build this collection of pairs using the ESCO Taxonomy (version 1.2.0). The ESCO occupations taxonomy is built on ISCO-08 [36], a four-level hierarchically structured classification that can be used to classify jobs into 436 unit groups. These groups are the most detailed ones of the classification structure, and are aggregated into minor, sub-major and major groups, based on their similarity in terms of the skill specialization and skill level required for the job. ESCO therefore includes occupations located at level 5 and lower of this classification structure. All occupation concepts contain one preferred label and a number of alternative labels.

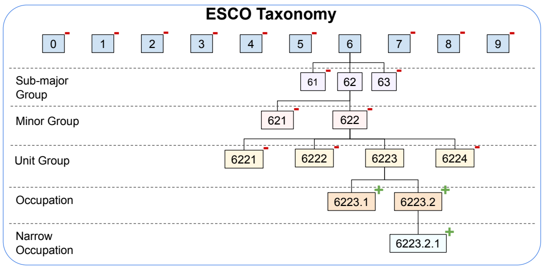

Figure 3: ESCO groups for job occupations. For creating the positive pairs, we do all possible combinations of synonyms for each **occupation** concept. Also, we retrieve positive pairs from parent occupations and from other occupations from the same **Unit Group**. Negatives are retrieved from concepts outside of the selected for the positives ones.

We use these hierarchically structured groups of occupation concepts to extract a collection of multilingual (positive and negatives) job title pairs (see Figure 3).

We create the positive pairs by including all possible combinations without repetition of the preferred and alternative labels of each occupation concept. In addition, to ensure lexical diversity among similar jobs, we generate a small subset of positive pairs by retrieving labels from parent occupations (30%) only if the concept is a narrow occupation. Finally, we retrieve more positive pairs from other occupations of the same Unit Group (10%[2]).

Regarding the negative pairs, we retrieve the samples for each occupation concept from labels of a different Major Group (50%), from another Sub-major Group (25%), from another Minor Group (15%), and from another Unit Group (10%).

The process described above is performed for each language independently, and once more to create cross-lingual pairs selecting job occupations from any language. The final dataset includes around 136 K of different job occupations, detailed into around 1 M different positive pairs (anchor + positive elements) and about 128 negative samples per unique job title.

3.2. Training Methodology

Below, we describe the end-to-end process carried out to train the model for multilingual job occupation ranking.

[1] Google Translate

[2] These two percentages are with respect to the number of positive pairs from each occupation concept.

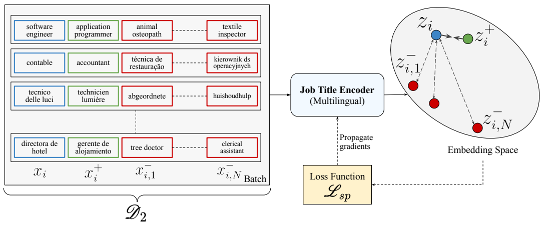

Figure 4: Training of multilingual encoder for Job Title Semantic Similarity with contrastive approach using positive and negative pairs from $\mathscr{D}_2$. Refer to Equation 3 for details about the Loss Function ($\mathscr{L}_{sp}$). N stands for the number of negatives pairs (in red) per anchor element (in blue).

First, we pre-train the multilingual encoder following an unsupervised approach to learn semantic representations for job titles based on vector representations using the job titles and *noisy* skills of $\mathscr{D}_1$ (see §3.1.1). Second, we fine-tune the encoder following a contrastive learning approach using the collection of similar and dissimilar pairs of job positions extracted into $\mathscr{D}_2$ (see §3.1.2).

3.2.1. Unsupervised training from Noisy Skills representations

First, we train the multilingual encoder to align the output representation for each job occupation with the doc2vec auxiliary representation generated in $\mathscr{D}_1$. With this aim, we train the encoder receiving as inputs pairs of job title (k_i) and fixed-length auxiliary embedding (v_i) from $\mathscr{D}_1$. In Equation 1 we show the cosine similarity function used for training the model based on the representation generated from distribution of *noisy* skills related to every job, where h_i refers to the output of the model after processing the surface text form k_i, and B to the batch size.

$$\mathscr{L}_{unsup} = -\frac{1}{B}\sum_{i=1}^{B} sim(h_i, v_i) \tag{1}$$

3.2.2. Contrastive Learning from Similarity Pairs

In the second stage, we fine-tune the multilingual encoder following a contrastive method using a NT-Xent loss function without in-batch negatives (see Equation 3). This function uses several negative pairs for each anchor sample, which helps the model to better discriminate between similar and dissimilar examples, and provides more stability during training (see Figure 4).

In the following $(x_i, x_i^+, x_{i,1..N}^-)$ represents the tuples that are fed as input to the model during training. x_i refers to the anchor element, x_i^+ to the positive sample, and $x_{i,1..N}^-$ to each negative element. On the other hand, z refers to the output embedding of the model for a particular input, resulting in $(z_i, z_i^+, z_{i,1..N}^-)$. N stands for the number of negatives per anchor element, B is equal to the batch size, and τ is the temperature value used in the function. To compare vectors, we use the cosine similarity (*sim*) function.

$$\delta(a, b) = e^{sim(a,b)/\tau} \tag{2}$$

$$\mathscr{L}_{sp} = -\frac{1}{B}\sum_{i=1}^{B} \log\left(\frac{\delta(z_i, z_i^+)}{\delta(z_i, z_i^+) + \sum_{j=1}^{N}\delta(z_i, z_{i,j}^-)}\right) \tag{3}$$

The temperature τ calibrates the intensity of the energy to push away the representation of the negative pairs with respect to the anchor and positive element. During training, the tuples of inputs are built dynamically from the collection of positive and negative pairs of $\mathscr{D}_2$.

4. Results and Discussion

In this Section, we describe the implementation details and report on results obtained from experiments using the proposed methodology for training a multilingual encoder to develop an intelligent job occupation recommendation solution. First, we introduce the Multilingual Language Models selected to study the proposed approach, these architectures are studied in terms of their efficiency (performance on job title ranking vs. model size).

Next, we do an ablation analysis of the different stages for training the model with the introduced solution to assess the benefits that each stage provides. Finally, we analyze the performance of the trained encoder for cross-lingual job title ranking.

To evaluate the matching performance of the proposed method, we use the English job titles similarity dataset[3] from Zbib et al. [6]. This set includes more than 2,500 job occupations distributed between query and corpus document elements. Each pair of query-document is labeled for binary relevance. We replicate this dataset for the other languages by using Human Translation (**de**, **fr**, **ja**, and **zh**) or Machine Translation (for the remaining languages). These datasets are publicly available at GitHub[4].

We evaluate models using the Mean Average Precision (mAP) of the output of the ranked lists. We use the *trec_eval* software library to compute this metric[5].

Architecture	**Fine-Tuned**	**#Params** (Million)	**Job Title Ranking Eval** (mAP) en	Avg_{EU}	Avg_{AS}	Avg_{ALL}
mUSE-CNN [37]	✗	69	0.4704	0.3832	0.4014	0.3961
P. Multi. MPNet [38]	✗	278	0.4701	0.3801	0.2862	0.3627
XLM-RoBERTa [39]	✗	278	0.1755	0.1141	0.1439	0.1278
Multi. E5 Large [40]	✗	560	0.5316	0.4617	0.3308	0.4324
E5 Mistral 7B Instr. [41]	✗	7111	0.6579	0.5458	0.3818	0.4324
BERT†	✓	110	0.7077	-	-	-
mUSE-CNN (ours)	$\mathscr{D}_1 + \mathscr{D}_2$	69	0.7344	0.6721	0.6688	0.6768
P. Multi. MPNet (ours)	$\mathscr{D}_1 + \mathscr{D}_2$	278	0.7384	**0.6900**	0.6627	0.6870
XLM-RoBERTa (ours)	$\mathscr{D}_1 + \mathscr{D}_2$	278	**0.7386**	0.6894	**0.6751**	**0.6900**

Table 1

Job Title Ranking comparison. Mean Average Precision (mAP) results are shown. The English test set is from [6]. The test sets for the other languages are built by translating them from the English one. Result with † is from our previous work in [6]. We highlight in bold the candidate architecture with the best average results on all languages. In addition, we highlight the best result for each collection of languages. Avg_{EU}={de, es, fr, it, nl, pl, pt}. Avg_{AS}={ja, ko, zh}.

4.1. Text Ranking with Multilingual Encoder

We perform an analysis of state-of-the-art Multilingual Language Models by testing them on text ranking for job title recommendation. In particular, we selected three candidate architectures based on their size in terms of parameters and their zero-shot performance on the ranking task.

It is worth noting that in this paper, our aim is not solely to identify the base model with the best results, but also to study the effectiveness of models in terms of their size and computational cost, with the goal of providing the practitioner with a practical guide for accuracy vs. efficiency trade-offs.

With that in mind, we select two Transformer-based architectures: *XLM-RoBERTa* [39] and *Paraphrase Multilingual MPNet* [38]. Both architectures have around 278 M of parameters and are based on *XLM-RoBERTa base models.* The main difference between them is that XLM-RoBERTa is pre-trained on multilingual masked language modeling objective, while *Paraphrase Multilingual MPNet* is a multilingual model that has been distilled from the monolingual *MPNet v2*[6] model, a sentence embedding model built from MPNet [42] and fine-tuned on sentence pairs.

On the other hand, we choose the Multilingual Universal Sentence Encoder model CNN-based (*mUSE-CNN*) [37]. This model generates good quality sentence embeddings in multiple languages and is very computationally efficient compared to the previous two models, with around only 69M parameters. Despite its smaller size, it offers reasonable zero-shot performance on job ranking.

Table 1 shows the results obtained with our proposed approach using the selected architectures. We start with comparing the evaluation performance with respect to other models without any fine-tuning. We observe that pre-trained models without any additional fine-tuning struggle to match the performance of fine-tuned alternatives in job title similarity ranking. Even the *E5 Mistral 7B Instruct* fails to outperform the fine-tuned BERT model on English despite having 64 times more parameters.

Our method achieves the best results, with **0.7386** mAP (*XLM-RoBERTa*) for English, which is 4.3% higher compared to the current state-of-the-art on the English job title similarity test set with monolingual BERT model from [6]. In addition, we get the best average results for **all** languages with **0.6900** mAP using the *XLM-RoBERTa* architecture.

We observe that the best results for European languages (de, es, fr, it, nl, pl, pt) are obtained with the fine-tuned Transformer-based architectures. Also, the *XLM-RoBERTa* architecture obtains the best results on the Asian languages (ja, ko, zh). We select this architecture as the best candidate for building the multilingual intelligent job recommendation system.

We also find that the efficient *mUSE-CNN* encoder delivers competitive results with significantly lower computational costs, making it an optimal choice for deployment in resource-limited environments. Refer to Appendix A for details on the configuration for training the selected encoder architectures.

4.2. Ablation Analysis

In the ablation analysis we examine the contribution of every stage of the proposed approach to the final ranking performance of job recommendation.

First, we carry out the pre-training of the selected architectures solely with the **first training stage** (see §3.2.1) on $\mathscr{D}_1$. In Table 2 we observe how the most simple architecture *mUSE-CNN* achieves very competitive results compared to the Transformer-based alternatives for all languages. In addition, the performance in English (**0.7028**) is comparable with the results presented in Zbib et al. [6] (**0.7077**).

Next, we do an additional ablation by training the architecture directly on the **second training stage** (see §3.2.2) using only contrastive learning on similar and dissimilar pairs of jobs from $\mathscr{D}_2$. The results show that when applying only contrastive learning for training, we can improve the mAP compared to not applying any fine-tuning. However, the results stay significantly lower than those obtained in the previous stage.

Also, note the difference for ranking performance in Korean for the *Paraphrase Multilingual MPNet* architecture. This is because this architecture without fine-tuning offers very poor results for this language compared to the other alternatives (with mAP values below 0.05).

[3] https://github.com/rabihzbib/jobtitlesimilarity_dataset

[4] https://github.com/Avature/jobtitlesimilarity-dataset

[5] https://github.com/usnistgov/trec_eval

[6] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Architecture	Fine-Tuning		Job Title Ranking Eval (mAP)										
	Unsup.	Contr.	en	de	es	fr	it	nl	pl	pt	ja	ko	zh
mUSE-CNN	$\mathscr{D}_1$	✗	0.7028	0.6291	0.6653	0.6189	0.6427	0.6520	0.6464	0.6563	0.6464	0.6271	0.6519
P. Multi. MPNet	$\mathscr{D}_1$	✗	0.6738	0.6215	0.6486	0.6069	0.6261	0.6383	0.6393	0.6476	0.6244	0.5725	0.6329
XLM-RoBERTa	$\mathscr{D}_1$	✗	0.6941	0.6362	0.6603	0.6211	0.6301	0.6485	0.6428	0.6559	0.6256	0.6260	0.6409
mUSE-CNN	✗	$\mathscr{D}_2$	0.5326	0.4135	0.4860	0.4608	0.4847	0.4215	0.4226	0.4779	0.4506	0.4102	0.4938
P. Multi. MPNet	✗	$\mathscr{D}_2$	0.5695	0.4509	0.4975	0.4969	0.4989	0.4763	0.4722	0.4878	0.3595	0.0483	0.5048
XLM-RoBERTa	✗	$\mathscr{D}_2$	0.4955	0.3740	0.4359	0.4067	0.4057	0.3920	0.3840	0.4213	0.4047	0.3909	0.4693
mUSE-CNN	$\mathscr{D}_1$	$\mathscr{D}_2$	0.7344	0.6629	0.6771	0.6540	0.6831	0.6774	0.6674	0.6827	0.6742	0.6533	0.6788
P. Multi. MPNet	$\mathscr{D}_1$	$\mathscr{D}_2$	0.7384	0.6764	**0.7105**	0.6712	**0.6957**	**0.6892**	0.6820	**0.7053**	**0.6778**	0.6243	**0.6860**
XLM-RoBERTa	$\mathscr{D}_1$	$\mathscr{D}_2$	**0.7386**	**0.6814**	0.7015	**0.6794**	0.6850	0.6832	**0.6918**	0.7038	0.6762	**0.6709**	0.6783

Table 2
Ablation Study and Evaluation. We evaluate the mAP for each language based on the methodology followed for training the selected architectures. First, we present the results of training the models solely with the first stage using unsupervision on $\mathscr{D}_1$. Next, we examine the ranking performance when using only contrastive learning on similar and dissimilar pairs from $\mathscr{D}_2$. Finally, we provide the numerical results per language after following the complete procedure outlined in this work.

Finally, the **combination of both stages sequentially** leads to the best mAP values. For the *mUSE-CNN*, we observe an average improvement of 4.29% with respect to results from Stage 1. Also, improvements of 9.01% and 7.18% are observed for the *Paraphrase Multilingual MPNet v2* and *XLM-RoBERTa* architectures respectively. We remark that, after this stage, Transformer-based architectures surpass the efficient *mUSE-CNN* for job title ranking for up to 1.3 points on average.

To evaluate the statistical significance of the improvements achieved by the two-stage approach compared to applying only the first-stage method, we employed the Wilcoxon signed-rank test [43]. The results indicate that the difference in performance between these methods is statistically significant at a significance level of 0.05 for all variants.

It is relevant to note that the contrastive learning stage brings gains for ranking in European languages (which are included in the dataset used for training - $\mathscr{D}_2$). However, all Asian languages also benefit from this training procedure, as the cross-lingual capabilities of these architectures (see §4.3) can contribute in improving ranking on languages not seen during this training stage.

In short, training the models for semantic-similarity based on the skills of each job occupation leads to a reasonable solution for multilingual job title ranking. In addition, fine-tuning this alternative using relations of job titles from the ESCO Taxonomy furthers adjust the embedding space to improve the ranking performance of the final solution on each language. This shows that both stages of our proposed method are beneficial.

4.3. Cross-lingual Evaluation

Finally, we conducted a cross-lingual evaluation of the proposed method on the selected architectures. For this purpose, we created several datasets where the language of the query elements differed from that of the corpus. In particular, we studied the ranking performance when English was fixed as one of the languages and assessed the cross-lingual capabilities with respect to German and Chinese.

Architecture	Fine-tuned	Cross-lingual Ranking Eval (mAP)			
		en-de	de-en	en-zh	zh-en
mUSE-CNN	✗	0.3628	0.3768	0.4228	0.4325
P. Multi. MPNet	✗	0.4033	0.3807	0.4450	0.4399
XLM-RoBERTa	✗	0.0346	0.0405	0.0425	0.0414
mUSE-CNN	$\mathscr{D}_1 + \mathscr{D}_2$	0.6846	0.7020	0.6854	0.7202
P. Multi. MPNet	$\mathscr{D}_1 + \mathscr{D}_2$	**0.7059**	0.7046	**0.7032**	0.7202
XLM-RoBERTa	$\mathscr{D}_1 + \mathscr{D}_2$	0.6995	**0.7106**	0.6921	**0.7226**

Table 3
Cross-lingual Evaluation. Each column represents a test set indicating the language of the queries and corpus elements respectively. For example, in the *en-de* set, the queries are in English and corpus documents in German.

Table 3 shows the cross-lingual ranking performance of our solution. We significantly improve the ranking performance for German and Chinese with respect to English compared to the architectures without any fine-tuning. For example, for the encoder based on *XLM-RoBERTa* after fine-tuning, we improve mAP when testing using Chinese queries and English corpus samples from only 0.0414 for up to 0.7226. Our cross-lingual analysis reveals improved performance on non-English languages compared to monolingual results in Table 2. This can be attributed to two key factors: the better representation quality for English job titles and the encoder's ability to align representations of job occupations across languages.

In Figure 5, we investigate the representation quality of encoders trained on cross-lingual occupation pairs, consisting of pairs of the same job title in two different languages. For simplicity, we focus on a subset of languages and on two of the presented encoders. To evaluate this, we employ the **alignment** ($\mathscr{L}_{align}$) and **uniformity** ($\mathscr{L}_{uniform}$) metrics introduced by Wang et al. [26]. The **alignment** metric evaluates how close the features from positive pairs (i.e., same occupation in different languages) are positioned in the feature space. On the other hand, the **uniformity** metric measures how

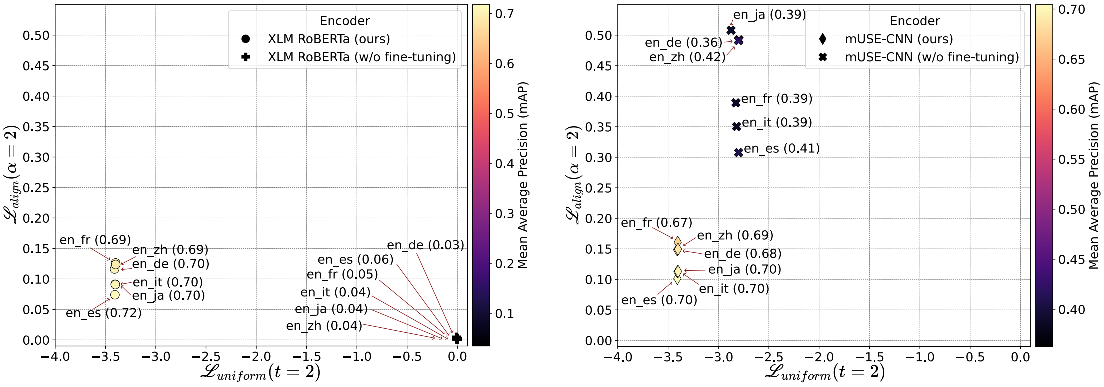

Figure 5: Left: Analysis of cross-lingual capabilities and performance of the *XLM-RoBERTa* trained encoder. Right: Analysis of cross-lingual capabilities and performance of *mUSE-CNN* trained encoder. We measure the **alignment** ($\mathscr{L}_{align}$) and **uniformity** ($\mathscr{L}_{uniform}$) metrics introduced in [26] to analyze the cross-lingual properties of the encoder. Each data point represents a collection of cross-lingual pairs, in which the first element is the English job title, and the second is its translation. We want to examine the representation quality of occupations in different languages. Note that, for both **alignment** and **uniformity**, **lower numbers are better**. We observe that better alignment and uniformity values (bottom-left corner) leads to better ranking results.

evenly the normalized features are distributed over the unit hypersphere, allowing for the analysis of features density. Refer to Wang et al. [26] for a detailed definition of these metrics.

A good encoder is one with very good alignment (features of the same occupation in different languages are very close), and good uniformity (otherwise, the learned representations would occupy a narrow cone in the vector space, which severely limits their expressiveness [44]). For both metrics, **lower** numbers are better, which is consistent with the literature.

Figure 5-left shows that the *XLM-RoBERTa* without any fine tuning (pre-trained on masked language modeling) offers very good alignment. However, the uniformity is very poor, which means that the vector space is quite compressed and limited when encoding job occupations. However, after our fine-tuning, the uniformity is significantly improved from 0 to around -3.4 at the expense of slightly worse alignment, which results in an encoder that is very competitive at recommending cross-lingual jobs, offering mAP values for up to 0.72 points.

Figure 5-right shows the same analysis for the *mUSE-CNN* encoder. In this case, the model without fine-tuning offers reasonable uniformity with low alignment. Both properties are significantly improved after our training procedure, resulting in an encoder that is better at ranking cross-lingual occupations. Note, too, that when comparing with the *XLM-RoBERTa* trained model, better alignment values lead to better mAP values.

5. Conclusions

In this paper we presented a methodology for developing a solution for intelligent multilingual job occupation recommendation, resulting in a job title encoder model for text ranking based on semantic similarity across 11 languages.

This method optimizes the neural network architectures in two steps. First, a pre-trained multilingual language model is adapted as an encoder to obtain the vector representation of every job occupation by learning their similarity based on the distribution of skills linked to each job. This stage provides large insights to the trained model about the semantic similarity of the different multilingual occupations, resulting in a good solution for job title ranking.

Secondly, we take advantage of the ESCO taxonomy and its hierarchically structured classification of jobs based on their similarity in terms of skill level and skill specialization, to extract similar and dissimilar pairs of job occupations. This information together with contrastive learning techniques helps to further optimize the embedding representation obtained with the models, and can improve the average mAP for ranking for up to 9% with respect to only carrying out the first training stage.

To the best of our knowledge, the contrastive learning stage introduced in this study contributes to surpass job title ranking performance in English by 4.3% over state-of-the-art monolingual approaches. Finally, we observe that the presented solution exhibits remarkable cross-lingual capabilities, enabling the ranking of jobs in different languages, offering good alignment and uniformity properties.

Our work presents limitations that open promising directions for future research. In particular, our solution does not leverage contextual information, which might be essential for a comprehensive understanding of job requirements. As a result, it may struggle to provide more tailored job recommendations, particularly for job seekers with diverse backgrounds and experiences. Furthermore, the lack of contextual information may lead to a narrow focus on job titles, which may not accurately reflect the complexities of modern job markets. Including information such as skills, job descriptions, career path, or education could significantly enhance the solution's ability to provide personalized and relevant job recommendations.

In future work, we aim to address these limitations by focusing on a key direction. We will study the potential benefits of incorporating skills and job descriptions into the encoder, developing novel techniques to represent and combine these contextual features, with the focus in improving the overall quality of the job recommendation system.

References

[1] S. Böhm, O. Linnyk, J. Kohl, T. Weber, I. Teetz, K. Bandurka, M. Kersting, Analysing gender bias in IT job postings: A pre-study based on samples from the german job market, in: S. Laumer, J. L. Quesenberry, D. Joseph, C. Maier, D. Beimborn, S. C. Srivastava (Eds.), SIGMIS-CPR 2020, Computers and People Research Conference, ACM, 2020, pp. 72–80.

[2] Z. Tasheva, V. Karpovich, Transformation of recruitment process through implementation of ai solutions, Journal of Management and Economics (2024).

[3] P. V. Yadav, U. S. Kollimath, T. V. Chavan, D. T. Pisal, S. A. Giramkar, S. M. Swamy, Impact of artificial intelligence (ai) in talent acquisition process: A study with reference to it industry, 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE) (2023) 885–889.

[4] T. Madhavi, A. Kaveri, The impact of artificial intelligence in recruitment and selection processes in IT companies, in: 16th International Conference on Electronics, Computers and Artificial Intelligence, ECAI, IEEE, 2024, pp. 1–5.

[5] J. Yao, Y. Xu, J. Gao, A study of reciprocal job recommendation for college graduates integrating semantic keyword matching and social networking, Applied Sciences (2023). URL: https://api.semanticscholar.org/CorpusID:265236145.

[6] R. Zbib, L. L. Alvarez, F. Retyk, R. Poves, J. Aizpuru, H. Fabregat, V. Simkus, E. G. Casademont, Learning job titles similarity from noisy skill labels, 2022. arXiv:2207.00494.

[7] J. Decorte, J. V. Hautte, T. Demeester, C. Develder, Jobbert: Understanding job titles through skills, CoRR abs/2109.09605 (2021). arXiv:2109.09605.

[8] D. Chandrasekaran, V. Mago, Evolution of semantic similarity - A survey, ACM Comput. Surv. 54 (2022) 41:1–41:37.

[9] D. Lavi, V. Medentsiy, D. Graus, consultantbert: Fine-tuned siamese sentence-bert for matching jobs and job seekers, in: M. Kaya, T. Bogers, D. Graus, K. Verbert, F. Gutiérrez (Eds.), Proceedings of the Workshop on Recommender Systems for Human Resources (RecSys in HR), volume 2967 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021.

[10] P. Neculoiu, M. Versteegh, M. Rotaru, Learning text similarity with siamese recurrent networks, in: P. Blunsom, K. Cho, S. B. Cohen, E. Grefenstette, K. M. Hermann, L. Rimell, J. Weston, S. W. Yih (Eds.), Proceedings of the 1st Workshop on Representation Learning for NLP, Association for Computational Linguistics, 2016, pp. 148–157.

[11] T. Gao, X. Yao, D. Chen, Simcse: Simple contrastive learning of sentence embeddings, in: M. Moens, X. Huang, L. Specia, S. W. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, 2021, pp. 6894–6910.

[12] H. Qiu, W. Ding, P. Chen, Contrastive learning of sentence representations, in: S. Bandyopadhyay, S. L. Devi, P. Bhattacharyya (Eds.), Proceedings of the 18th International Conference on Natural Language Processing ICON, NLP Association of India (NLPAI), 2021, pp. 277–283.

[13] L. Xu, H. Xie, Z. Li, F. L. Wang, W. Wang, Q. Li, Contrastive learning models for sentence representations, ACM Trans. Intell. Syst. Technol. 14 (2023) 67:1–67:34.

[14] M. le Vrang, A. Papantoniou, E. Pauwels, P. Fannes, D. Vandensteen, J. D. Smedt, ESCO: boosting job matching in europe with semantic interoperability, Computer 47 (2014) 57–64.

[15] E. Commission, ESCO (European Skills, Competences, Qualifications and Occupations taxonomy), 2024. URL: https://esco.ec.europa.eu/select-language?destination=/node/1, accessed: 2024-07-08.

[16] M. Zhang, R. van der Goot, B. Plank, ESCOXLM-R: Multilingual taxonomy-driven pre-training for the job market domain, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 11871–11890.

[17] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl,

Item-based collaborative filtering recommendation algorithms, in: V. Y. Shen, N. Saito, M. R. Lyu, M. E. Zurko (Eds.), Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001, ACM, 2001, pp. 285–295.

[18] R. Patel, S. K. Vishwakarma, An efficient approach for job recommendation system based on collaborative filtering, in: M. Tuba, S. Akashe, A. Joshi (Eds.), ICT Systems and Sustainability, Springer Singapore, Singapore, 2020, pp. 169–176.

[19] I. Obadic, G. Madjarov, I. Dimitrovski, D. Gjorgjevikj, Addressing item-cold start problem in recommendation systems using model based approach and deep learning, in: D. Trajanov, V. Bakeva (Eds.), ICT Innovations 2017 - Data-Driven Innovation. 9th International Conference, volume 778 of *Communications in Computer and Information Science*, Springer, 2017, pp. 176–185.

[20] J. Zhao, J. Wang, M. Sigdel, B. Zhang, P. Hoang, M. Liu, M. Korayem, Embedding-based recommender system for job to candidate matching on scale, CoRR abs/2107.00221 (2021). arXiv:2107.00221.

[21] M. Bocharova, E. Malakhov, V. Mezhuyev, Vacancysbert: the approach for representation of titles and skills for semantic similarity search in the recruitment domain, CoRR abs/2307.16638 (2023). arXiv:2307.16638.

[22] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT), Association for Computational Linguistics, 2019, pp. 4171–4186.

[23] T. Chen, S. Kornblith, M. Norouzi, G. E. Hinton, A simple framework for contrastive learning of visual representations, in: Proceedings of the 37th International Conference on Machine Learning, ICML, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 1597–1607.

[24] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR, IEEE Computer Society, 2006, pp. 1735–1742.

[25] M. Ye, X. Zhang, P. C. Yuen, S. Chang, Unsupervised embedding learning via invariant and spreading instance feature, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Computer Vision Foundation / IEEE, 2019, pp. 6210–6219.

[26] T. Wang, P. Isola, Understanding contrastive representation learning through alignment and uniformity on the hypersphere, in: Proceedings of the 37th International Conference on Machine Learning, ICML, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 9929–9939.

[27] J. Gao, D. He, X. Tan, T. Qin, L. Wang, T. Liu, Representation degeneration problem in training natural language generation models, in: 7th International Conference on Learning Representations, ICLR, 2019.

[28] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, L. Li, On the sentence embeddings from pre-trained language models, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, 2020, pp. 9119–9130.

[29] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). arXiv:1907.11692.

[30] X. Sun, Y. Meng, X. Ao, F. Wu, T. Zhang, J. Li, C. Fan, Sentence similarity based on contexts, Trans. Assoc. Comput. Linguistics 10 (2022) 573–588. doi:10.1162/TACL_A_00477.

[31] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, W. Xu, Consert: A contrastive framework for self-supervised sentence representation transfer, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, 2021, pp. 5065–5075.

[32] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, 2005, pp. 539–546. URL: https://doi.org/10.1109/CVPR.2005.202.

[33] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE Computer Society, 2015, pp. 815–823.

[34] K. Sohn, Improved deep metric learning with multi-class n-pair loss objective, in: D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, 2016, pp. 1849–1857.

[35] Q. V. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31th International Conference on Machine Learning, ICML, volume 32 of *JMLR Workshop and Conference Proceedings*, JMLR.org, 2014, pp. 1188–1196.

[36] International Labour Organization, International Standard Classification of Occupations 2008 (ISCO-08): Structure, group definitions and correspondence tables, 2024. URL: https://www.ilo.org/media/352556/download, Accessed: 2024-7-30.

[37] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Ábrego, S. Yuan, C. Tar, Y. Sung, B. Strope, R. Kurzweil, Multilingual universal sentence encoder for semantic retrieval, in: A. Celikyilmaz, T. Wen (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL, Association for Computational Linguistics, 2020, pp. 87–94.

[38] N. Reimers, I. Gurevych, Making monolingual sentence embeddings multilingual using knowledge distillation, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, 2020, pp. 4512–4525.

[39] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL, Association for Computational Linguistics, 2020, pp. 8440–8451.

[40] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Multilingual E5 text embeddings: A technical report, CoRR abs/2402.05672 (2024). arXiv:2402.05672.

[41] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Improving text embeddings with large language models, CoRR abs/2401.00368 (2024). arXiv:2401.00368.

[42] K. Song, X. Tan, T. Qin, J. Lu, T. Liu, Mpnet: Masked and permuted pre-training for language understanding, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS, 2020.

[43] J. Demsar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[44] K. Ethayarajh, How contextual are contextualized word representations? comparing the geometry of bert, elmo, and GPT-2 embeddings, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, 2019, pp. 55–65.

[45] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations, ICLR, 2019.

A. Training Configuration

In this appendix, we provide information about the selected encoders used for doing the analysis of the proposed methodology for multilingual job recommendation. We also describe details about the hyperparameters and training configuration for each architecture.

- **mUSE-CNN**: Multilingual Universal Sentence Encoder CNN [37]. This model is based on Convolutional Neural Networks for efficient inference at the cost of reduced accuracy. This model is trained using 16 languages on multiple tasks: question-answer prediction, translation ranking, and natural language inference. The architecture is implemented in Tensorflow and is publicly available at Kaggle[7]. This model has 69 M parameters.
- **XLM-RoBERTa** [39]: This architecture is a Transformer-based model pre-trained on 100 languages using masked language modeling. We use the model implemented in Tensorflow available at Kaggle[8]. This model has 278 M parameters.
- **Paraphrase Multilingual MPNet** [38]: This model is a multilingual encoder that has been distilled from the monolingual *MPNet v2*[9] model, a sentence embedding model built from MPNet [42] and fine-tuned on a dataset with 1 billion English sentence pairs. The model backbone is *XLM-RoBERTa*, and it is distilled to mimic the vector-dense representation of English sentences on +50 languages. This architecture is implemented in PyTorch and is publicly available at HuggingFace[10]. This model has 278 M parameters.

The training configuration differs between CNN-based vs. Transformer-based encoders, due to the different nature of their backbone architecture. Also, training for Transformer-based architecture is different given that one of the architectures is trained on masked language modeling, and the other on sentence similarity. These configurations are the result of preliminary experiments in which we compared the impact of the different hyperparameters for every encoder model.

A common element shared by all variants is the use of the AdamW [45] optimizer, which employs a learning rate decay mechanism that gradually reduces the learning rate to zero.

Architecture	Batch Size	# Steps	Learning rate
mUSE-CNN	150	3,200 K	$1x10^{-4}$
P. Multi. MPNet	64	500 K	$3x10^{-5}$
XLM-RoBERTa	64	1,700 K	$3x10^{-5}$

Table 4
Training configuration on Stage 1 with unsupervised method (§3.2.1)

Architecture	Batch Size	# Steps	Learning rate	N	τ
mUSE-CNN	128	28 K	$1x10^{-6}$	64	0.25
P. Multi. MPNet	16	28 K	$5x10^{-7}$	16	0.20
XLM-RoBERTa	4	112 K	$5x10^{-7}$	16	0.10

Table 5
Training configuration on Stage 2 with contrastive learning method (§3.2.2). N stands for the number of negatives per anchor element, and τ represents the temperature value that obtained best results.

In Table 4, we show the training hyperparameters used for training the encoders on Stage 1, unsupervised training. *mUSE-CNN* was trained for a larger number of steps, higher batch size and learning rate due to its reduced number of parameters and CNN-based architecture. On the other hand, for Transformer-based encoders, we use a smaller learning rate. Note the difference in terms of training steps between *Paraphrase Multilingual MPNet* and *XLM-RoBERTa*. This is due to the training objective of the original pre-trained model (sentence similarity vs. masked language modeling).

Finally, in Table 5, we show the training configuration for each encoder with the contrastive learning method (Stage 2). Due to its reduced size in terms of parameters, for *mUSE-CNN* encoder, we use a bigger batch size and larger number of negatives per anchor element compared to Transformer-based architectures. As for Transformer-based solutions, due to the limitation in terms of the Tensorflow implementation, we have to reduce the batch size of *XLM-RoBERTa* encoder, consequently, we increase the number of training steps. Lastly, regarding the temperature (τ), we do a grid search with the following values $\tau = \{0.01, 0.05, 0.1, 0.2, 0.25, 0.3\}$.

[7] https://www.kaggle.com/models/google/universal-sentence-encoder/tensorFlow2/multilingual

[8] https://www.kaggle.com/models/kaggle/xlm-roberta/tensorFlow2/multi-cased-l-12-h-768-a-12

[9] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[10] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2