Hardware-effective Approaches for Skill Extraction in Job Offers and Resumes

Laura Vásquez-Rodríguez[1,*], Bertrand Audrin[2], Samuel Michel[1], Samuele Galli[3], Julneth Rogenhofer[1], Jacopo Negro Cusa[3] and Lonneke van der Plas[1,4,5]

[1]Idiap Research Institute, Switzerland
[2]EHL Hospitality Business School, HES-SO, University of Applied Sciences and Arts Western Switzerland, Switzerland
[3]Arca24.com SA, Switzerland
[4]Institute of Linguistics and Language Technology, University of Malta, Malta
[5]Università della Svizzera Italiana, Switzerland

Abstract

Recent work on the automatic extraction of skills has mainly focused on job offers and not resumes while using state-of-the-art resource-intensive methods and considerable amounts of annotated data. However, in real-life industrial contexts, the computational resources and the annotated data available can be limited, especially for resumes. In this paper, we present our experiments that use hardware-effective methods and circumvent the need for large amounts of annotated data. We experiment with various methods that vary in hardware requirements and complexity. We evaluate these systems both on public and commercial data, using gold-standard for evaluation. We find that standalone rule-based and semantic model performance on the skill extraction task is limited and variable between job offers and resumes. However, neural models can perform competitively and be more stable, even when using small datasets, with an improvement of ∼30%. We present our experiments using minimal hardware, mostly CPU-based with less than 8 GB of RAM for rule-based and semantic methods and using GPUs for neural models with a maximum memory usage for both CPU and GPU of 24 GB, with less than 25 minutes of training time.

Keywords

Human Resources, Skill Extraction, Recruiting, Natural Language Processing

1. Introduction

With the development of professional social networking sites and job boards, job offers get more and more applicants. On average, an online job can get 250 applications [1], making manual handling of applications and selection of candidates no longer possible or practical. Partial automation of the talent acquisition process thus seems to be imperative, leading to cost reduction and increased efficiency [2].

The pre-screening phase is one of the most likely to be automated to filter through large volumes of resumes. This involves recognizing different types of skills, determining their proficiency (e.g., C2 level in English), and classifying and weighting skills according to their category (e.g., hard versus soft skills) in relationship with a specific job offer.

In this work, we focus on the first step of this process: extracting skills from job offers and resumes, starting with hard skills. To do so, organizations rely on "Applicant Tracking Systems (ATS)" that offer a first filter through the resumes. Most HR professionals do not necessarily have a clear understanding of how these ATS work, and even less of the amount of computational resources that they require. However, this topic is paramount not only for the large volumes of data involved in the process of recruiting but also, for the cost that increases with each new application. Our motivation is the development of incremental and hybrid approaches that should be adapted to the needs of every organization.

This paper is developed within the context of the SEM24 project, which seeks the introduction of NLP as a means of guidance and support to HR specialists. In this context, we performed a bottom-up assessment of skill extraction methods, solely focused on hard skills as the first project stage, considering scenarios with limited annotated data and computational resources. We hypothesize that with an incremental approach to the skill extraction task, it is possible to find hardware-effective hybrid methods, which are not only competitive in cost but also in performance. We enumerate our contributions as follows:[1]

- A comparative evaluation including rule-based, semantic, and neural models for detecting hard skills from job offers with publicly available and labeled datasets and from industry-owned resumes, where datasets tend to be more restricted.
- A manual analysis of systems outputs, highlighting the differences between the proposed systems, and systematically categorizing models' mismatches explaining human-system discrepancies beyond to what test sets can measure together with automatic metrics.
- An analysis of hardware requirements, gathering insightful information on the trade-off between the performance and resources needed for the skill extraction task.

2. Related Work

The skill extraction task has already been explored for about a decade, and yet, still not truly solved [3]. Traditionally,

[1]We will release our code on GitHub: https://github.com/idiap/hw_effective_skill_extraction

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.
*Corresponding author.
laura.vasquez@idiap.ch (L. Vásquez-Rodríguez); bertrand.audrin@ehl.ch (B. Audrin); samuel.michel@idiap.ch (S. Michel); samuele.galli92@gmail.com (S. Galli); julneth.rogenhofer@ehl.ch (J. Rogenhofer); j.negrocusa@gmail.com (J. N. Cusa); lonneke.vanderplas@usi.ch (L. v. d. Plas)
0000-0002-7313-905X (L. Vásquez-Rodríguez); 0000-0003-2510-0924 (B. Audrin); 0009-0003-4542-7083 (S. Michel); 0009-0000-1709-5585 (S. Galli); 0000-0002-9917-3811 (J. Rogenhofer); 0009-0004-4403-1957 (J. N. Cusa); 0000-0002-2871-2574 (L. v. d. Plas)

rule-based methods using a taxonomy have been predominant, either by using simple rules [4], or more formally, by recognizing entities in the text using Named Entity Recognition (NER) [5]. Taxonomies have also been involved in the classification of resumes according to predefined jobs or skills [6] and the search of similar terms in a weakly supervised manner [7].

With the recent progress of neural networks, there has been an increased interest in the automation of talent acquisition systems using more advanced machine-learning methods in NLP. However, these contributions are scattered across multiple tasks, domains, and languages. At the document level, Bhola et al. [8] proposed the classification of job offers with a list of relevant skills. Further, they also explored the prediction of missing skills jointly with graph neural networks [9]. More commonly, the skill extraction task has been addressed as the identification and labeling of spans (i.e., sequences of words in texts) and in different languages such as English [10, 11], Danish [12] and German [13]. The transformer architecture has also been explored, where Zhang et al. [14] proposed fine-tuning existing multilingual Large Language Models (LLMs) enriched with taxonomies such as ESCO.[2] More recently, papers based on instruct-based models without further training have proposed novel avenues for NER skill extraction [15].

Data privacy regulations introduce another relevant challenge. Resumes are considered personal data, and they can potentially represent ethical issues if they are not handled correctly. Therefore, public datasets of resumes are scarce and scattered, while job offers are found in multiple languages. In English, Green et al. [11] performed the extraction and annotation of skills and their proficiency from UK job boards in multiple domains such as IT and finance. This domain has also been explored in other languages such as German [16] and French [17]. Also, datasets have been proposed for detecting scams in online jobs [18] and for the identification of privacy-related entities (e.g., names, emails) in job postings [19]. However, skills extraction from resumes is as important as skills extraction from job offers in an industrial setting, or arguably even more important, because numbers are larger and automation is key. To mitigate these limitations, the research community has also focused on the synthetic generation of resumes [20] and job offers [21], avoiding the difficulty of handling users' privacy and scarcity of data.

In this work, we challenge the assumption that computing resources and datasets are readily available, assessing possible and feasible scenarios using low-cost hardware (e.g., CPU) and limited annotated data for skills extraction. Further, we demonstrate the performance of the skill extraction task with our proposed minimal settings, on a wide spectrum of methods increasingly transitioning in method complexity, resource consumption, and effectiveness. To our knowledge, no work has explored the skill extraction task from a resource-effective perspective.

3. Methodology

We define our proposed task in Section 3.1. We investigate the extraction of hard skills in both scenarios from publicly available labeled job offers and resumes, which are more restricted in access while using methods that require minimal GPU. Next, we detail the data collection process in Section 3.2, the selected extraction methods in Section 3.3, and finally, human evaluation of the system outputs are described in Section 3.4.

3.1. Tasks Definition

This paper explores extracting hard skills from job offers and resumes. We selected the NER task [22] to detect skills and occupations as entities. The main difference between the traditional NER and our approach is that we will perform the extraction of text spans (i.e., a contiguous sequence of words), whose length can vary greatly between entities and, in some cases, it may not represent a single concept (e.g., "collaborating with multiple teams").

We subdivide our task according to the access level (i.e., public or restricted) of HR datasets. In this domain, datasets containing job offers are often publicly available, whereas datasets with resumes are almost non-existent, limiting the development of extraction methods in multiple scenarios (e.g., domains and languages). As the quality and level of access differ between resumes and job offers, we propose the following tasks:

Task 1: Skill Extraction from Job Offers: The extraction of hard skills using annotated data is well explored with various methods. Using human-annotated data ensures a better quality of the outputs; however, this type of data is not available for all languages and domains. An additional advantage of working on job offers is that data can be shared without ethical concerns which favours reproducibility. In this task, we will automatically annotate job offers using a taxonomy, evaluating the results with publicly available datasets.

Task 2: Skill Extraction from Resumes: There is a scarcity of annotated data for the scenario of skill extraction from resumes. Also, the content and the layout are more variable, affecting the accuracy of models that are trained on more standardized content such as job offers. We propose the manual annotation of resumes for training and testing using labeled data and a manual analysis of the outputs to determine the performance of this task.

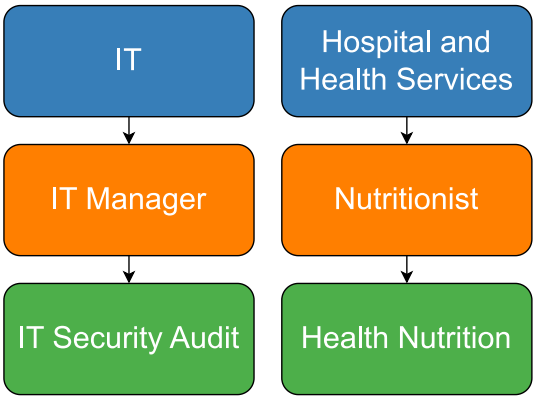

Figure 1: Example of a job taxonomy.

3.2. Data Collection

For our experiments, we tried to satisfy both industrial and academic requirements. We perform the proposed task in

[2] https://esco.ec.europa.eu/en

the best possible yet cost-effective way, but we also make sure that methods and experiments are reproducible and shareable with the research community. To achieve this goal, we use a combination of public and in-house, professionally crafted taxonomies[3] as a knowledge source. Second, we selected public datasets related to job offers and industry-owned resumes for training and testing.[4]

3.2.1. Manually Annotated Taxonomies

Arca24_DB: crafted by in-house professional annotators from our industrial partner of the SEM24 project. This taxonomy has 10,379 entities, including domains, skills, and job occupations, available in 10 languages. We consider that the use of this taxonomy is relevant given that skills are extracted from actual resumes and job offers, curated by HR specialists. Other taxonomies such as ESCO (see below) tend to be more standardized, with a language style dissimilar to the ones in resumes and job offers, representing a challenge for skill extraction task [23]. In this work, we will perform skill extraction from English resumes and job offers. Also, we focused on the selection of skills and occupations, discarding other elements in the hierarchy (Figure 1) such as domain entities, resulting in a total of 10,356 entities.

ESCO_DB: a more comprehensive and publicly available taxonomy of 131,623 entities. To adapt this taxonomy to our use case, and to make evaluations comparable, we downloaded the ESCO dataset (v1.1.1, content classification) for English.[5] We identified all the files and columns relevant to skills and knowledge, creating a single knowledge base with no duplicates.

3.2.2. Manually Annotated Datasets

Green_JOB [11]: a collection of job offers from UK job boards annotated with skills (18,617 entities for training and 908 for testing).[6] It was annotated by crowdsourcing with the definition of the following types of entities: skills, knowledge, occupation, experience, and domain. As some of these entities can be quite similar, we have remapped them into 2 categories as follows: skills, knowledge as hard skills, and experience and domain together with occupations. For our experiments, we have used the train, validation, and test splits published on HuggingFace (HF).[7]

Arca24_CV: we collected a set of 50 resumes in English from multiple domains (i.e., IT, Finance, Sales). The selection of the CVs was a random sample from a larger corpus from *Arca24_CV*. The HR specialists (2 people) from our project performed the annotations of these resumes jointly, discussing possible disambiguation of the results. We selected the following entities for the annotations: hard skills, soft skills, knowledge, language, occupation, domain, and degrees/certifications. Also, as a way of easing the annotation process, we automatically highlighted the dates in the text. Similarly, as in the *Green_JOB* dataset, we also remapped the existing entities to achieve consistency across all datasets and better performance (See Section 4.1). As for degrees/certifications, we mapped them under hard skills. Also, we discarded soft skills and languages in our experiments as this will be explored in future work.[8]

3.3. Methods in a Nutshell

In this section, we give an overview of the implemented methods (i.e., rule, semantic, and neural) for the skill extraction task. Further, we expand on the technical details of our methods in Section 4.2.

3.3.1. Rule-based Methods

The implementation of a rule-based model is straightforward. We define a set of rules that allows the matching of concepts in a knowledge base or taxonomy and word sequences in a text. These concepts should be normalized, as these could be shown in different surface forms (e.g., engineer, engineering) while referring to equally relevant meanings for recruitment purposes.

We performed this task by applying multiple transformations to the concepts in the taxonomy and the word sequences in the texts as follows: 1) Remove punctuation and spaces through word tokenization, 2) Lemmatisation (e.g., better → good), 3) Stemming (e.g., python development → python develop) and 4) Expansion of the search space by considering not only words but also subwords (e.g., "communicate on-line" → "communicate on-line", "communicate", "on-line"). The normalized text from both taxonomy and text are compared to find possible matches. As a result, the same text span may match multiple concepts from the taxonomy. As a final disambiguation step, candidate concepts are selected considering their semantic similarity,[9] Levenshtein distance [24] and skills length.

The main advantage of this approach is that the output is highly predictable, precise, explainable, and controllable because there is a direct match between the taxonomy and the skills extracted. The extracted skills can be mapped to the knowledge base and skills that the system failed to extract can be attributed to the fact that they are missing in the taxonomy. However, keeping a taxonomy up-to-date involves manual labor, which is time-consuming and expensive. There are often problems with the coverage of such hand-built resources, which leads to problems of recall in system output.

Another limitation of purely rule-based systems is that they do not generalize well, as they are tied to fixed concepts. Hence, if a concept is absent in the taxonomy, that particular skill will not be detected. The taxonomy must be updated often, as new jobs and skills are constantly required in the job market [25]. Also, if similar skills are expressed with different words (e.g., reporter, journalist), they will not be detected.

3.3.2. Methods based on Semantic Similarity

In view of the above-mentioned limitations, we do not only attempt to find a direct mapping between skills in the text and concepts in the taxonomy but allow a mapping between

[3] We refer to a taxonomy as a collection of skills and knowledge, hierarchically connected to occupations in different domains, as shown in Figure 1.

[4] In Table 1, we show the statistics of the collected and the automatically annotated datasets, which will be explained further in Section 4.

[5] https://esco.ec.europa.eu/en/use-esco/download

[6] https://www.kaggle.com/datasets/airiddha/trainrev1

[7] https://huggingface.co/datasets/jjzha/green

[8] For consistency, we use the term "Resume" throughout the paper, however, for naming convenience, we use "CV" to name our dataset. Please note that we refer to the same concept in both cases.

[9] Although we use semantic similarity in the final disambiguation step, we consider the model as rule-based since the main selection of skills is limited to the concepts that are explicitly in the taxonomy.

Table 1
We report the statistics for our datasets and taxonomies, including their licensing.

Data	Type	License	Train	Valid	Test	Unit	Avg.Tokens
Green_JOB	Dataset	CC-BY-4.0	8,669	964	335	Sentences	23.03
Arca24_CV		Restricted	1,564	195	196		24.02
ESCO_DB	Taxonomy	Public	131,623	-	-	Entities	-
Arca24_DB		Restricted	10,379	-	-		-

semantically similar terms as well. We use a pre-trained token-based embedding model to encode the n-grams of the original text (i.e., job offers and resumes) and the concepts in the taxonomy. Then, we calculate the similarity of each pair to determine how close they are to each other. However, the longest skills in the taxonomy can range up to eight words in size, this leads to unreliable results as most of the skills are smaller.

To obtain a better threshold for the skill search and avoid unnecessary comparisons, we estimated the maximum number of tokens per skill. To achieve this, we calculated the average size of a skill in the taxonomy, resulting in a range of one to four words. When multiple words are present in the taxonomy or in the text, word vector values are averaged, meaning word matching does not depend on the token order.[10] We benefit from this approach, as smaller groups of tokens will have less sparse representations. Finally, we perform a selection of skills using a threshold for the semantic similarity and Levenshtein distance, similarly as performed in the rule-based system.[11] We proposed two approaches for matching the text and concepts in the taxonomy: 1) comparison of all the possible n-grams of the text and taxonomy (full) and 2) comparison of those n-grams in the text that have not been paired yet with any skill. Hence, repetitive comparisons of already detected texts are avoided (reduced). The comparison of the taxonomy with all the possible text spans is highly resource-consuming. Hence, we selected the second approach which is faster acknowledging the trade-off that we could get suboptimal results given that the first match is not necessarily the best.

3.3.3. Supervised Machine Learning Methods

We proposed a neural, supervised setting where the models are more likely to learn concepts and generalize better. Although the semantic similarity methods can provide a means of generalization between similar concepts, there is no real learning so that the model can perform the skill extraction task in similar domains or different languages without having an explicit example for every case. Therefore, we selected a supervised scenario where we could fine-tuned multiple models which has no previous knowledge about the task as BERT-base [26], but also, further models that has domain knowledge in the field of HR such as JobBERT [10] and ESCOXLM-R [14]. We comment further on the technical details of these models in Section 4.2.

3.3.4. Neural Methods (Instruct-based)

Instruct-based models are also capable of performing the skill extraction task with a NER approach. These models are characterized to have strong inference and proficient text generation. However, the text generation is less structured and unpredictable than the previous methods. Nguyen et al. [15] modeled NER-style skill extraction using GPT-3.5-turbo[12] model. The privacy-sensitive data we are working with does not allow us to work with such models. Results from this work are also not directly comparable to our results because they are limited to the F1 scores for the Green dataset, without details on individual precision and recall for skills and occupations. In future work, we will experiment with open-source instruct-based models that allow us to work on resumes without privacy concerns.

3.4. Evaluation Methods

We performed a NER-based evaluation to assess the quality of our task. We used the Inside-Outside-Beginning (IOB) format [27], highlighting the text's entities and the gold reference using a set of predefined labels. We adapted all systems outputs to this format to have a comparable evaluation. The detected skills in job offers and resumes will be identified as entities and compared against a gold standard (i.e., human-annotated data). We assessed our files using the *nerevaluate* Python Package[13] to determine the precision, recall, and F1 score of the results for the labeled datasets in the exact and partial evaluation. For the neural models, we used the seqeval[14] library during training for its suitable integration with HF. However, for consistency, we report our results using the nervaluate-based evaluation with all model predictions exported in IOB format.

For our use case in the HR domain, identifying an entity type is challenging as each dataset has a different set of entity types (e.g., skills, knowledge, domains versus hard and soft skills). Also, it is difficult for the automatic methods to differentiate between similar categories (e.g., skills and knowledge). To mitigate this scenario, we focused solely on hard skills (e.g., Python, Inventory Management) and occupations (e.g., Software Engineer, Sales Assistant), and mapped other entities to these categories, as explained in Section 4.1. Finally, we selected the *exact-match* schema for detecting the skills, as proposed by Segura-Bedmar et al. [28]. In this schema, credit is given to detected entities regardless of the type. We also considered reporting our results into a *strict* evaluation, where credits depend both on the type and the entity. However, as for a job-resume match the entity type is not imperative, we focused only on the exact metric. Furthermore, we consider a *partial-match* evaluation; where there is also credit given for those entities that are not extracted in full. This allows us to understand which entities could potentially be detected when the evaluation is extended beyond the defined entity boundaries. Finally, we performed a human evaluation of the neural outputs, which we will detail in Section 4.3.

[10] https://spacy.io/usage/linguistic-features/#similarity-expectations
[11] In the rule-based system, we use the semantic similarity as a final step to disambiguate matches for the same text span. The skill extraction task is completely done by using rules to match the taxonomy.
[12] gpt-3.5-turbo-instruct
[13] https://pypi.org/project/nervaluate/
[14] https://github.com/chakki-works/seqeval

Table 2
Results for the Human Evaluation

Dataset	Model	Categories						
		0	1	2	3	4	5	6
Green_JOB	bert-based-cased	47.69%	10.77%	10.77%	9.23%	3.08%	15.38%	3.08%
Green_JOB	escoxlmr_skill_extraction	44.26%	6.56%	9.84%	4.92%	14.75%	13.11%	6.56%
Arca24_CV	bert-based-cased	57.53%	21.92%	5.48%	2.74%	5.48%	6.85%	0.00%
Arca24_CV	escoxlmr_skill_extraction	56.94%	27.78%	0.00%	1.39%	2.78%	4.17%	6.94%

4. Experiments

In this section, we describe the datasets used, the implementation details of our selected methods (i.e., rule-based, semantic, and neural), the preprocessing steps for the data, and the evaluation. Finally, we present the training details of our models in Section 4.2.3.

4.1. Data

To perform our experiments on the *Green_JOB* dataset [29], we selected the split published in HF,[15] which has been used in previous work as well. The main difference between this dataset and the one originally published [29], is the redistribution of the training set to create a development set, while the test set remains unchanged. We performed the skill extraction task per sentence, as in the original format provided in this dataset. We also kept the original tokenization given by the dataset. Concerning the mapping of entities, we mapped knowledge and qualifications together to hard skills entities and experience to occupations entities.

For the *Arca24_CV*, we annotated the dataset using Doccano.[16] We migrated the entities exported in JSONL into the IOB format, merging spacy spans[17] and doc[18] with customized alignment[19] functions, as the original methods[20] could not align properly all the annotated entities, showing tokenization discrepancies with unmapped entities. This dataset has resumes that are longer than the model input size, hence, we divided the texts into smaller extracts. Due to the variability of resume formats, sentence boundaries are not always present in the text, therefore, a more structured approach was required. To be consistent with the sentence-level job offers dataset, we subdivided the *Arca24_CV* texts based on the average sentence length in the Green dataset, resulting in 23 tokens per text, split by whitespace. We also included the restriction that no labeled entity (e.g., B-Skill, I-Skill) should be split between sentences, hence texts could be slightly longer.[21]

4.2. Models

In this section, we discuss the implementation and technical details of our rule-based, semantic similarity, and neural models.

4.2.1. Rule-based and Semantic

For the rule-based implementation, we adapted the existing open-source tool SkillNER.[22] Also, because multiple candidates can match the same text span, we carried out the final selection using semantic similarity with Spacy English model.[23] In the case that the word vectors were not available, we used Levenshtein distance from NLTK package[24] as an alternative. As a knowledge base for skills search, we used *Arca24_DB* and *ESCO_DB* as taxonomies.

For the semantic approach, we compared similarities between the text and the taxonomy, using the same Spacy models as in the rule-based model. Because semantic similarity may propose completely unrelated skills for our use case, we included the requirement that the candidates should have a minimal Levenshtein distance to the concepts in the taxonomy. The thresholds we used are 0.5 and 0.7. We selected this lower-bound as it still shows relevant candidates relevant to skills. As for the upper bound, we observed it shows a more precise selection of candidates with minimal false positives. Similarly, as in the rule-based system, we used the same taxonomy as a reference. Concerning the models, we used the large English Spacy model,[25] which is optimized to run in CPU.

4.2.2. Supervised Machine Learning Methods

Previous systems have the limitation that they rely on a given taxonomy, where concepts are completely isolated with no context. Hence, we experimented with the Green and the *Arca24_CV* dataset to establish supervised baselines using the available splits. For the latter, we split the dataset using the Datasets library[26] from HuggingFace (HF) [30], resulting in train, validation and test (80/10/10) splits. We comment further on the technical details of the proposed models.

We considered both the cased and uncased version of BERT-base [26], a baseline model for the skill exaction task. We selected this model as it is the base of well-established fine-tuned models for this task. Also, in this scenario, the model has no previous knowledge of the NER and the skill extraction task in the domain of HR. Previously, this 110M of parameters model was trained for the tasks of Masked Language Modeling (MLM) and Next Sentence Prediction (NSP).

Further, we selected JobBERT [10], a model based on the uncased version of BERT-based, with a domain adaptive pre-training on ∼3.2M sentences from job postings. These

[15] https://huggingface.co/datasets/jjzha/green
[16] https://github.com/doccano/doccano
[17] https://spacy.io/api/span
[18] https://spacy.io/api/doc
[19] https://spacy.io/api/doc#char_span
[20] https://spacy.io/api/top-level/#gold
[21] For clarity, we will also refer in our experiments to *Green_JOB* as "Jobs (Green)", and to *Arca24_CV* as "Resumes (Ours)".
[22] https://github.com/AnasAito/SkillNER
[23] https://spacy.io/models/en
[24] https://www.nltk.org/api/nltk.metrics.distance.html
[25] https://spacy.io/models/en#en_core_web_lg
[26] https://huggingface.co/docs/datasets/v2.21.0/en/package_reference/main_classes#datasets.Dataset.train_test_split

Table 3
Performance of the neural models in the detection of Hard Skills (Skill) and Occupations (Occ). We report the Precision (p), Recall (r), and F1-Score (f1) for the exact and partial evaluation (Eval.) schema.

Train/Test	Eval.	Model	Skill_p	Skill_r	Skill_f1	Occ_p	Occ_r	Occ_f1	All_p	All_r	All_f1
Jobs (Green)	exact	bert-base-cased	**0.415**	0.401	**0.408**	**0.633**	0.631	0.631	**0.465**	0.453	**0.459**
		bert-base-uncased	0.39	0.388	0.389	0.632	**0.639**	**0.635**	0.445	0.445	0.445
		jobbert_skill_extraction	0.406	0.398	0.402	0.576	0.584	0.58	0.445	0.44	0.443
		jobbert_knowledge_extraction	0.376	0.383	0.379	0.617	0.616	0.616	0.429	0.435	0.432
		escoxlmr_skill_extraction	0.402	**0.408**	0.404	0.632	0.629	0.63	0.453	**0.457**	0.455
		escoxlmr_knowledge_extraction	0.394	0.402	0.398	0.601	0.629	0.614	0.442	0.453	0.447
Resumes (Ours)	exact	bert-base-cased	**0.496**	**0.503**	**0.499**	**0.545**	0.605	0.571	**0.504**	**0.518**	**0.51**
		bert-base-uncased	0.461	0.485	0.469	0.449	0.595	0.511	0.457	0.502	0.476
		jobbert_skill_extraction	0.456	0.449	0.451	0.489	0.551	0.517	0.462	0.464	0.462
		jobbert_knowledge_extraction	0.457	0.434	0.445	0.502	0.566	0.531	0.464	0.454	0.459
		escoxlmr_skill_extraction	0.429	0.428	0.428	0.537	**0.657**	**0.591**	0.448	0.462	0.454
		escoxlmr_knowledge_extraction	0.444	0.439	0.439	0.517	0.652	0.576	0.456	0.471	0.462
Jobs (Green)	partial	bert-base-cased	0.645	0.624	0.634	**0.744**	0.742	0.743	**0.668**	0.65	0.659
		bert-base-uncased	0.631	0.628	0.629	0.745	**0.753**	**0.749**	0.657	**0.656**	0.656
		jobbert_skill_extraction	0.639	0.626	0.633	0.694	0.705	0.699	0.652	0.644	0.648
		jobbert_knowledge_extraction	0.608	0.618	0.613	0.725	0.724	0.724	0.634	0.642	0.638
		escoxlmr_skill_extraction	**0.633**	**0.643**	**0.638**	0.736	0.732	0.734	0.656	0.663	**0.66**
		escoxlmr_knowledge_extraction	0.626	0.638	0.632	0.715	0.748	0.73	0.646	0.663	0.654
Resumes (Ours)	partial	bert-base-cased	**0.578**	**0.587**	**0.582**	**0.652**	0.725	0.684	0.59	**0.607**	**0.598**
		bert-base-uncased	0.548	0.576	0.557	0.574	0.763	0.654	0.551	0.604	0.573
		jobbert_skill_extraction	0.565	0.554	0.558	0.612	0.69	0.647	**0.572**	0.575	0.573
		jobbert_knowledge_extraction	0.55	0.523	0.536	0.633	0.715	0.67	0.564	0.552	0.558
		escoxlmr_skill_extraction	0.544	0.542	0.542	0.648	**0.793**	**0.713**	0.562	0.579	0.57
		escoxlmr_knowledge_extraction	0.548	0.54	0.542	0.627	**0.793**	0.699	0.561	0.578	0.568

models were trained for the skill extraction task, distinguishing between skills and knowledge. Finally, we select a large model in comparison to the selected efficient baselines, ESCOXLM-R [14], a 559M parameters state-of-the-art model for skill extraction, based on XLM-RoBERTa large model. This model also performed domain-adaptive pretraining using the available concepts in the ESCO taxonomy 27 languages. As in the previous model, there is a skill and knowledge variant, which we also fined-tuned in the NER task.

With respect to the implementation details of these models, we used HF and Pytorch Lightning libraries.[27] Also, we used the models published in HF for the BERT-based (cased,[28] uncased),[29] and JobBERT (skill[30] and knowledge).[31] For the implementation of ESCOXLM-R model, we also fine-tuned the models available for knowledge[32] and skills.[33] We fine-tuned all the models using the available train and validation splits from the selected job offers and resumes in Section 3.2. Next, we tested the performance of the skill extraction task using the held-out test split.

4.2.3. Training Details

In Table 5 we include the time and hardware resources consumed for our experiments. For the neural models, we experimented with multiple hyper-parameters as suggested by Zhang et al. [14], including the batch size of 8, 16, 32 and learning rate of $1e^{-4}$, $1e^{-5}$ and $5e^{-5}$. We found our best setting, by using a learning rate of $5e^{-5}$, batch size of 16, and training the models for 10 epochs. In particular, for *escoxlmr_knowledge_extraction* and *escoxlmr_skill_extraction* we used a learning rate of $5e^{-5}$ for the *Green_JOB* dataset and $1e^{-5}$ for the *Arca_CV* dataset, which showed a more stable and better-performing setting for these large models. We also seeded our experiments for reproducible results and selected the experiments with the best-performing result on the validation set. For our neural experiments, we report our average results for all the seeded experiments using the following randomly selected values: 42, 31, 22, 57, 37.

4.3. Human Evaluation

To understand the quality of our neural models, we performed the analysis of 120 random sentences in total for the selected models: *bert-based-cased* (best) and *escoxlmr_skill_extraction* (larger) on the *green* and *Arca24_CV* dataset. Following Nguyen et al. [15], we classified the entities in each sentence into 5 error categories: 1) Skill definition misalignment, when the system extracts a career-related term that is not a skill; 2) Wrong extraction, where the system extracts an entity that is completely unrelated to any skill; 3) Conjoined skills, when two skills appear in a single text span, e.g. develop reporting software and statistical software, but the systems sees it as one; 4) Extended span, where the system selected entities that are longer than the ground truth; 5) Incorrect annotations, where the human annotation is not precise; and 6) Others. The category "Others" included scenarios such as the incomplete detection of skills in comparison with the ground truth. We also included an additional category for the correct entities as well (Category 0). The evaluation was done by a domain expert on the skill extraction task. Finally, we report the percentages for each category in our error analysis as shown in Table 2.

[27] https://lightning.ai/docs/pytorch/stable/
[28] https://huggingface.co/google-bert/bert-base-cased
[29] https://huggingface.co/google-bert/bert-base-uncased
[30] https://huggingface.co/jjzha/jobbert_skill_extraction
[31] https://huggingface.co/jjzha/jobbert_knowledge_extraction
[32] https://huggingface.co/jjzha/escoxlmr_knowledge_extraction
[33] https://huggingface.co/jjzha/escoxlmr_skill_extraction

Table 4
Performance of the rule-based and semantic models in the detection of Hard Skills (Skill) and Occupations (Occ) using the selected taxonomies (DB). We report the Precision (p), Recall (r), and F1-Score (f1) for the exact and partial evaluation (Eval.).

DB	Test	Eval.	Model	Skill_p	Skill_r	Skill_f1	Occ_p	Occ_r	Occ_f1	All_p	All_r	All_f1
Arca24	Jobs (Green)	exact	Rule-based	0.276	**0.1**	**0.147**	0.488	0.199	0.283	0.337	**0.125**	**0.183**
			Semantic_09_05	0.135	0.038	0.06	0.554	**0.218**	**0.313**	0.273	0.085	0.129
			Semantic_09_07	**0.361**	0.022	0.041	**0.605**	0.123	0.205	**0.494**	0.048	0.087
ESCO			Rule-based	**0.155**	**0.159**	**0.157**	0.295	**0.341**	**0.316**	0.195	**0.207**	**0.2**
			Semantic_09_05	0.081	0.051	0.063	0.41	0.204	0.272	0.152	0.091	0.114
			Semantic_09_07	0.145	0.017	0.03	**0.604**	0.137	0.224	**0.333**	0.048	0.084
Arca24	Resumes (Ours)	exact	Rule-based	**0.151**	**0.154**	**0.152**	**0.478**	**0.289**	**0.361**	**0.187**	**0.177**	**0.182**
			Semantic_09_05	0.071	0.06	0.065	0.273	0.158	0.2	0.097	0.077	0.086
			Semantic_09_07	0.143	0.044	0.067	0.286	0.105	0.154	0.171	0.055	0.083
ESCO			Rule-based	**0.102**	**0.253**	**0.145**	0.077	**0.289**	0.122	0.096	**0.259**	**0.14**
			Semantic_09_05	0.067	0.104	0.082	0.097	0.158	0.12	0.072	0.114	0.088
			Semantic_09_07	0.064	0.038	0.048	**0.294**	0.132	**0.182**	0.095	0.055	0.069
Arca24	Jobs (Green)	partial	Rule-based	0.507	**0.183**	**0.269**	0.744	0.303	0.431	0.574	**0.214**	**0.312**
			Semantic_09_05	0.397	0.112	0.175	0.777	**0.306**	**0.439**	0.522	0.162	0.248
			Semantic_09_07	**0.611**	0.037	0.069	**0.802**	0.164	0.272	**0.715**	0.069	0.127
ESCO			Rule-based	0.363	**0.374**	**0.369**	0.445	**0.514**	**0.477**	0.386	**0.41**	**0.398**
			Semantic_09_05	0.374	0.238	0.291	0.643	0.32	0.427	0.432	0.259	0.324
			Semantic_09_07	**0.471**	0.054	0.097	**0.75**	0.171	0.278	**0.585**	0.084	0.147
Arca24	Resumes (Ours)	partial	Rule-based	0.261	**0.266**	**0.264**	**0.739**	**0.447**	**0.557**	0.313	**0.298**	**0.305**
			Semantic_09_05	0.224	0.19	0.205	0.636	0.368	0.467	0.276	0.22	0.245
			Semantic_09_07	**0.286**	0.088	0.134	0.643	0.237	0.346	**0.357**	0.114	0.172
ESCO			Rule-based	0.195	**0.484**	**0.278**	0.151	**0.566**	0.239	0.185	**0.498**	**0.269**
			Semantic_09_05	0.191	0.297	0.232	0.234	0.382	**0.29**	**0.199**	0.311	0.242
			Semantic_09_07	**0.197**	0.118	0.148	**0.441**	0.197	0.273	0.23	0.132	0.168

5. Results

We present our results for the rule-based and semantic systems in Table 4. For the exact evaluation, we report the highest values for the detection of skills in job offers. Overall, *Semantic_09_07* models are highly precise, especially for the detection of occupations. However, in exchange for precision, they would have a minimal recall in comparison to the *Rule-based* and *Semantic_09_05*. In terms of the F1-score, the rule-based systems are the more balanced when it comes to considering both precision and recall. The taxonomy selected, also impacts the output of the skill detection task, while ESCO-based results tend to be less precise, these have a higher recall due to the size of this resource. Concerning the evaluation of resumes, a similar trend is shown in all the systems, but with lower values given the nature of these texts. We will discuss in detail these issues in Section 6.

Further, we report our results on the neural systems in Table 3. Given that these models are mostly based on similar BERT models, they are not divergent between them. However, we can observe differences in whether the models consider the difference between cases (i.e., small case or upper case) or not. For both resumes and job offers, surprisingly, the *bert-based-cased* model showed the best performance although originally there was no knowledge of the task or domain. As a goal to compare with a resource-consuming scenario, we also run our datasets using the *escoxlmr* models, which showed to have comparable performance to previous models. We analyze the effect of these differences in Tables 6 and 7.

Additionally, we comment on our human evaluation. Given that the results in Table 3 are quite close, we considered it relevant to perform an analysis of the neural system output. We present the results of our domain expert evaluation in Table 2. While around 50% of the samples were correct, we report a diverse distribution in the rest of the error categories. There is a fair share of annotations that show misalignment between what humans and systems consider to be a skill, followed by extended spans (when the system extracts a longer span than the skill itself), as well as incorrect annotations (when the system extracts an entity but is not demonstrated in the ground truth). Next, there are also a few conjoined skills, when the system extracts two skills at once (e.g., develop reporting software and statistical software).

Finally, we propose the analysis of this work from a hardware-effective perspective, presenting our resource consumption analysis of all our systems runs in Table 5.

6. Discussion

For rule-based systems, the main limitation is the dependency of the systems on a manually built taxonomy. Results are shown to be more precise in the detection of concepts, which means that the concepts that are detected are very likely to be true. However, we found that entities are disconnected from their context, as the main goal is to match a rule in the taxonomy. For example, we can find generic sentences about organizations present in job offers that are identified as skills (i.e., "The company specializes in marketing, PR, Account Handling"), although they do not directly refer to skills. If the same sentence were to appear in a resume, this would indeed be considered a skill. With respect to the taxonomies selected, we also expected an incremental difference with large datasets such as ESCO, however, when

Table 5
We report the hardware resources used for our experiments. For CPU memory usage, we used the *seff* command. For GPU usage, we monitored resource consumption using *nvitop*. Finally, we report the total time (train+test) for each case.

DB / Train	Test set	Model	Total Time	CPU (RAM)	GPU (RAM)	Hardware
Arca24_DB	Jobs (Green)	Rule-based	3.95m	1.26 GB	-	AMD EPYC 7742 64-Core Processor, 8GB RAM (CPU)
		Semantic_09_05	52.15m	3.70 GB	-	
		Semantic_09_07	52.15m	3.68 GB	-	
ESCO_DB		Rule-based	51.87m	1.82 GB	-	
		Semantic_09_05	8.86h	4.22 GB	-	
		Semantic_09_07	8.77h	4.24 GB	-	
Arca24_DB	Resumes (Ours)	Rule-based	2.40m	1.25 GB	-	
		Semantic_09_05	27.92m	3.64 GB	-	
		Semantic_09_07	28.10m	3.05 GB	-	
ESCO_DB		Rule-based	30.62m	1.78 GB	-	
		Semantic_09_05	4.83h	4.06 GB	-	
		Semantic_09_07	4.76h	4.12 GB	-	
Green_JOB	Jobs (Green)	bert-base-cased	6.92m	5.35 GB	7.3 GB	NVIDIA GeForce RTX 3090, 24 GB RAM (GPU/CPU)
		bert-base-uncased	6.81m	7.40 GB	6.8 GB	
		jobbert_skill_extraction	6.89m	6.45 GB	8.7 GB	
		jobbert_knowledge_extraction	6.47m	5.38 GB	8.7 GB	
		escoxlmr_skill_extraction	22.16m	23.94 GB	19.5 GB	
		escoxlmr_knowledge_extraction	21.72m	23.94 GB	19.5 GB	
Arca24_CV	Resumes (Ours)	bert-base-cased	1.27m	7.33 GB	3.72 GB	
		bert-base-uncased	1.52m	5.57 GB	3.55 GB	
		jobbert_skill_extraction	1.70m	5.34 GB	4.04 GB	
		jobbert_knowledge_extraction	1.44m	5.33 GB	4.04 GB	
		escoxlmr_skill_extraction	4.79m	19.27 GB	12.79 GB	
		escoxlmr_knowledge_extraction	4.69m	17.74 GB	12.79 GB	

a strict rule-based approach is followed, it is impossible to match a given concept to similar ones that have different contexts and styles in terms of the language used between the taxonomy, job offers, and resumes. Still, we can see cases, such as in Tables 6 and 7, when *Rule-based* using *ESCO* shows competitive results to the neural systems.

In contrast, semantic systems may result in a more flexible extraction of skills regarding the surface properties of text and the taxonomy, but this would affect significantly the precision of the skills selected. For end-customers, precision is highly important as customers' and HR specialists' trust depends more on whether the system is correct or not. Hence, an automated solution would be seen as unreliable compared to manually selecting candidates. Driven by the significance of precision, we proposed the use of more strict thresholds, such as in the *Semantic_09_07*. In this manner, we obtained concepts that are quite close to the ones existing in the taxonomy, however, it resulted in a very low recall. From an inclusive perspective, recall is very relevant, as we would like to give equal opportunity and importance to all candidates. Therefore, the importance of balancing the expectations of the system, without neglecting or showing bias towards any candidate.

Rule-based and semantic systems have the advantage that the origin of the detection is highly traceable to a taxonomy, and the reason for a success or failure is straightforward. While this has been less relevant in the past, today is an important challenge [31] which is also materialized in existing regulations, where high-risk AI-based applications, such as recruiting algorithms, will consider explainability as an essential feature.[34] These approaches are also inexpensive, with the limitation that they could be potentially not scalable as their performance would be bound to the size of a given taxonomy. This could be the case of using a large taxonomy such as ESCO with methods such as *semantic_09_05* and *semantic_09_07* (See Table 5), which represents a base worst case of processing time. However, it is still possible to include optimizations such as the parallelization of search algorithms, the inclusion of more flexible and efficient database solutions, and the simplification of taxonomies with non-relevant terms. Overall, this is not a trivial task, however, it is worth exploring alternatives before choosing resource-consuming solutions.

Regarding neural-based systems, we observed a significant increase for all the proposed scenarios compared to the previous rule-based and semantic systems. Also, these models achieved equivalent performance for the exact evaluation in both job offers and resumes, considering their overall average (i.e., *all_f1*). Interestingly, for the partial evaluation, there is a boost in the job offer metrics. We hypothesize that this could be due to the proximity of the prediction to the true label, but yet, is not as precise as the exact annotation. Job offers tend to be more standardized, while resumes can express the same skill differently. Also, there is more noise in resumes, as data is often available in PDF format and layouts can vary greatly between candidates. This poses a challenge to the skill extraction task in settings using NER, where it is expected to find a bounded entity within the existing text. Also, these methods discard possible skills that are not explicitly in the text, which is a typical characteristic of soft skills.

We consider that the main challenge has been the selection of quality data. We not only account for the variability of the resumes and the noise of data parsing but also, the restricted access of resumes. While we partnered with a company that can provide these data, it is mostly unlabeled for

[34] https://eur-lex.europa.eu/eli/reg/2024/1689/oj

supervised settings. Also, quality test sets are scarce which represents a time-consuming job and challenging task for annotators to achieve accurate labeling of skills in multiple domains. For our use case, we selected the Green dataset as a state-of-the-art human-annotated resource. This dataset is publicly available, which allows the reproducibility of the experiments. Nevertheless, this dataset represents a collection of sentences from job offers from the UK, showing a limited domain and context to evaluate our systems.

Further, to support the quality analysis of our data, we comment on the manual analysis. We consider that the major challenge of skill extraction is the subjective nature of this task, supported by most of the annotations in the first category. We also found it interesting that although we scored similarly on the results, the distribution of the errors was significantly different. There is also a minority related to skills that need to be inferred, but we also acknowledge that these datasets are mostly dedicated to explicit skill extraction. Also, the detection of entities within rigid boundaries in tasks such as NER can be difficult, which was clear in the errors reported for Category 4.

Finally, we discuss the main topic and motivation of our work: hardware-effective skill extraction methods. While the NLP hype pushes toward large-scale LLMs, these are still under consideration for real-user cases when it comes to cost and explainability. Within the HR domain, companies may process thousands of resumes per week, hence, representing an increased cost directly associated with the data volume. Also, customers would be required to understand the reason for selection which is hard to track in non-deterministic systems. In our work, we show the possibilities of systems that can rely solely upon CPUs or within limited GPU training time. We believe it is important to evaluate the benefit between these methods and LLMs, including the budget that needs to be invested. For example, the *escoxlm-based* models represent 5X the size of the *bert-based* models and more than 3X in resource consumption, however, the benefit in detection is small and similar to the original baselines. We would like to stress with our work, that it is valuable to evaluate a wide spectrum of solutions that not only rely on large models and datasets. We understand that some LLMs could be proficient in many HR-related tasks, but we should consider hybrid approaches that not only benefit in precision, and cost-effective but also explainability for fair and inclusive AI-based recruiting.

7. Conclusions and Future Work

In this paper, we have proposed three different skill extraction methods, for the detection of hard skills, using multiple configurations and datasets, including resumes and job offers. We have exposed the boundaries of rule-based, semantic, and neural methods using minimal hardware, including CPU-based architectures with less than 8GB of RAM and/or minimal GPU training in less than 25 minutes and 24GB RAM. Also, we have analyzed the performance of these systems by using supervised baselines, showing that the latter can improve $\sim$30% with minimal data for the taxonomy-based ones.

In future work, we aim to experiment with more complex architectures, such as instruct-based LLMs and more annotated datasets by professional experts in different languages. The presented state-of-the-art models show that these are quite competitive, however, they tend to be less explainable to a non-technical audience (e.g. business and HR professionals) due to their indeterministic nature.

Our current experiments represent a starting point to fully understand the correct approach for the skill extraction task, where resource optimization and explainability are important. Further, we will also expand our experiments in a multilingual setting, including French, Italian, Portuguese, Spanish, Italian, and German. Next, we will continue with an essential part of this task which is the inclusion and analysis of soft skills, which truly represent the human aspect of recruiting. Also, we would investigate how to cast the skill extraction task as a sentence classification task, because NER is tightly linked particular literal text span.

Ethics Statement

Developing NLP algorithms in the HR domain is challenging due to the relatively scarce availability of public datasets and evaluation scenarios. For our experiments, we have used public datasets to ensure the reproducibility of our methods. However, we also include industry-proprietary datasets and resumes with sensitive information. We acknowledge that we have acquired the corresponding licenses and data consents for managing this information. However, due to the nature of these resources and the ethical considerations that come with them, we are not able to share them publicly.

Limitations

The availability of public datasets for the task of skills extraction is limited, where there are small samples per domain and languages. As a start, we have relied on available resources for English, however, these are not large and some are specific (e.g. detection of hard skills in UK-based job offers). Similarly, public resumes are mostly unavailable, which limits the possibility of benchmarking our methods with data that we are allowed to share with no privacy concerns. We acknowledge that larger deep-learning architectures, including instruct-based models, could show a better performance for the proposed task; however, it is important to consider that the availability of GPUs is not always a given. Finally, developing a taxonomy as a knowledge base is essential to build an explainable system that users trust. Although depending solely on a list of concepts shows limited performance, it can be complemented with additional neural methods for each scenario. With respect to the hardware resources, we understand that the memory consumption could also be affected by external factors outside of our control (e.g., jobs running in the same CPU/GPU nodes). However, our relative estimation shows in perspective the resource usage with different architectures and data sizes, which is enough for the proposal of resource-saving strategies.

Acknowledgments

We would like to thank Alexandre Nanchen for his feedback on this paper. We also thank Ewan Roche for his support in enabling the efficiency calculations. Finally, we gratefully acknowledge the support from Innosuisse (grant 104.069 IP-ICT).

Table 6
Error analysis on system outputs, we highlight relevant entities detected by the systems using the Green test set.

Model	Taxonomy / Train	Example
Reference	-	The *Test Consultant* *Automation Test Analyst* will ideally be confident with *Selenium* and good experience of *web-based testing* *HTML* and *Javascript*
Rule-based	Arca24_ DB	The Test Consultant Automation Test Analyst will ideally be confident with Selenium and good experience of web-based testing HTML and *Javascript*
	ESCO	The Test *Consultant* Automation *Test Analyst* will ideally be *confident* with *Selenium* and good experience of web-based testing HTML and *Javascript*
Semantic	Arca24_ DB / 09_05	The Test Consultant *Automation Test Analyst* will ideally be confident with Selenium and good experience of web-based *testing HTML* and *Javascript*
	Arca24_DB / 09_07	The Test Consultant Automation Test Analyst will ideally be confident with Selenium and good experience of web-based testing *HTML* and *Javascript*
	ESCO_DB / 09_05	The Test Consultant *Automation Test Analyst* will ideally be confident with Selenium and good experience of web-based testing HTML and Javascript
	ESCO_DB / 09_07	The Test Consultant Automation Test Analyst will ideally be confident with Selenium and good experience of web-based testing HTML and Javascript
bert-base-cased	Green	The *Test Consultant* / *Automation Test Analyst* will ideally be confident with *Selenium* and have good experience of *web-based testing* *HTML* and *Javascript*
bert-base-uncased	Green	the *test consultant* *automation test analyst* will ideally be confident with *selenium* and good experience of *web-based testing* *html* and *javascript*
jobbert_ knowledge_ extraction	Green	The *Test Consultant Automation / Test Analyst* will ideally be confident with *Selenium* and good experience of *web-based testing*, *HTML* and *Javascript*.
escoxlmr_ skill_ extraction	Green	The *Test Consultant* *Automation Test Analyst* will ideally be confident with *Selenium* and good experience of *web-based testing* *HTML* and *Javascript*.
escoxlmr_ knowledge_ extraction	Green	The *Test Consultant* *Automation Test Analyst* will ideally be confident with *Selenium* and good experience of *web-based testing* *HTML* and *Javascript*.

Table 7
Error analysis on system outputs, we highlight relevant entities detected by the systems using the *Arca24_CV* test set.

Model	Taxonomy / Train	Example
Reference	-	drafting of bug fix reports, *project virtualization* to optimize the testing process, *project documentation*, *reporting development*, uml design training.
Rule-based	Arca24_ DB	drafting of bug fix reports, project virtualization to optimize the testing process, project documentation, reporting development, uml *design* training.
	ESCO	drafting of bug fix *reports*, project virtualization to optimize the testing *process, project* documentation, *reporting* development, *uml design training.*
Semantic	Arca24_DB / 09_05	drafting of bug fix reports, project virtualization to optimize the testing process, project documentation, *reporting development*, uml design training.
	Arca24_DB / 09_07	drafting of bug fix reports, project virtualization to optimize the testing process, project documentation, *reporting development*, uml design training.
	ESCO / 09_05	*drafting of* bug fix reports, project virtualization to *optimize the testing process*, *project documentation*, *reporting development*, uml design training.
	ESCO / 09_07	drafting of bug fix reports, project virtualization to optimize the testing process, project documentation, reporting development, uml design training.
bert-base_ cased	Resumes (Ours)	*drafting of bug fix reports*, project virtualization to optimize the testing process, project documentation, *reporting development*, *uml design training.*
bert-base- uncased	Resumes (Ours)	*drafting* of *bug fix reports*, project virtualization to optimize the testing process, project documentation, *reporting development*, uml *design training.*
jobbert_ skill_ extraction	Resumes (Ours)	drafting of bug *fix* reports, *project* virtualization to optimize the testing process, project *documentation*, *reporting* development, uml design *training*.
jobbert_ knowledge_ extraction	Resumes (Ours)	*drafting* of bug fix reports, project virtualization to optimize the testing process, project documentation, *reporting* development, uml *design training.*
escoxlmr_ skill_ extraction	Resumes (Ours)	*drafting of bug fix reports*, *project virtualization* to optimize the testing process, project documentation, *reporting development*, *uml design training.*
escoxlmr_ knowledge_ extraction	Resumes (Ours)	drafting of *bug fix reports*, *project virtualization* to optimize the *testing* process, *project* documentation, *reporting development*, *uml design training.*

References

[1] J. S. Black, P. v. Esch, AI-enabled recruiting: What is it and how should a manager use it?, Business Horizons 63 (2020) 215–226. URL: https://www.sciencedirect.com/science/article/pii/S0007681319301612. doi:https://doi.org/10.1016/j.bushor.2019.12.001.

[2] R. D. Johnson, D. L. Stone, K. M. Lukaszewski, The benefits of eHRM and AI for talent acquisition, Journal of Tourism Futures 7 (2021) 40–52. URL: https://doi.org/10.1108/JTF-02-2020-0013. doi:10.1108/JTF-02-2020-0013, publisher: Emerald Publishing Limited.

[3] E. Senger, M. Zhang, R. van der Goot, B. Plank, Deep learning-based computational job market analysis: A survey on skill extraction and classification from job postings, in: E. Hruschka, T. Lake, N. Otani, T. Mitchell (Eds.), Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024), Association for Computational Linguistics, St. Julian's, Malta, 2024, pp. 1–15. URL: https://aclanthology.org/2024.nlp4hr-1.1.

[4] K. Bothmer, T. Schlippe, Skill scanner: Connecting and supporting employers, job seekers and educational institutions with an ai-based recommendation system, in: Innovative Approaches to Technology-Enhanced Learning for the Workplace and Higher Education, Springer International Publishing, Cham, 2023, pp. 69–80.

[5] M. Zhao, F. Javed, F. Jacob, M. McNair, Skill: A system for skill identification and normalization, Proceedings of the AAAI Conference on Artificial Intelligence 29 (2015) 4012–4017. URL: https://ojs.aaai.org/index.php/AAAI/article/view/19064. doi:10.1609/aaai.v29i2.19064.

[6] D. A. Tamburri, W.-J. V. D. Heuvel, M. Garriga, Dataops for societal intelligence: a data pipeline for labor market skills extraction and matching, in: 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI), 2020, pp. 391–394. doi:10.1109/IRI49571.2020.00063.

[7] M. Zhang, K. N. Jensen, R. van der Goot, B. Plank, Skill extraction from job postings using weak supervision, in: RecSys in HR'22: The 2nd Workshop on Recommender Systems for Human Resources, in conjunction with the 16th ACM Conference on Recommender Systems, September 18–23, 2022, Seattle, USA., CEUR Workshop Proceedings, 2022.

[8] A. Bhola, K. Halder, A. Prasad, M.-Y. Kan, Retrieving skills from job descriptions: A language model based extreme multi-label classification framework, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 5832–5842. URL: https://aclanthology.org/2020.coling-main.513. doi:10.18653/v1/2020.coling-main.513.

[9] N. Goyal, J. Kalra, C. Sharma, R. Mutharaju, N. Sachdeva, P. Kumaraguru, JobXMLC: EXtreme multi-label classification of job skills with graph neural networks, in: A. Vlachos, I. Augenstein (Eds.), Findings of the Association for Computational Linguistics: EACL 2023, Association for Computational Linguistics, Dubrovnik, Croatia, 2023, pp. 2181–2191. URL: https://aclanthology.org/2023.findings-eacl.163. doi:10.18653/v1/2023.findings-eacl.163.

[10] M. Zhang, K. Jensen, S. Sonniks, B. Plank, SkillSpan: Hard and soft skill extraction from English job postings, in: M. Carpuat, M.-C. de Marneffe, I. V. Meza Ruiz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 4962–4984. URL: https://aclanthology.org/2022.naacl-main.366. doi:10.18653/v1/2022.naacl-main.366.

[11] T. Green, D. Maynard, C. Lin, Development of a benchmark corpus to support entity recognition in job descriptions, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, S. Piperidis (Eds.), Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 1201–1208. URL: https://aclanthology.org/2022.lrec-1.128.

[12] M. Zhang, K. N. Jensen, B. Plank, Kompetencer: Fine-grained skill classification in Danish job postings via distant supervision and transfer learning, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, S. Piperidis (Eds.), Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 436–447. URL: https://aclanthology.org/2022.lrec-1.46.

[13] A.-s. Gnehm, E. Bühlmann, H. Buchs, S. Clematide, Fine-grained extraction and classification of skill requirements in German-speaking job ads, in: D. Bamman, D. Hovy, D. Jurgens, K. Keith, B. O'Connor, S. Volkova (Eds.), Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS), Association for Computational Linguistics, Abu Dhabi, UAE, 2022, pp. 14–24. URL: https://aclanthology.org/2022.nlpcss-1.2. doi:10.18653/v1/2022.nlpcss-1.2.

[14] 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 (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 11871–11890. URL: https://aclanthology.org/2023.acl-long.662. doi:10.18653/v1/2023.acl-long.662.

[15] K. Nguyen, M. Zhang, S. Montariol, A. Bosselut, Rethinking skill extraction in the job market domain using large language models, in: E. Hruschka, T. Lake, N. Otani, T. Mitchell (Eds.), Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024), Association for Computational Linguistics, St. Julian's, Malta, 2024, pp. 27–42. URL: https://aclanthology.org/2024.nlp4hr-1.3.

[16] A.-S. Gnehm, E. Bühlmann, S. Clematide, Evaluation of transfer learning and domain adaptation for analyzing German-speaking job advertisements, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, S. Piperidis (Eds.), Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 3892–3901. URL: https://aclanthology.org/2022.lrec-1.414.

[17] D. Beauchemin, J. Laumonier, Y. L. Ster, M. Yassine, "fijo": a french insurance soft skill detection dataset, 2022. URL: https://arxiv.org/abs/2204.05208. arXiv:2204.05208.

[18] S. Vidros, C. Kolias, G. Kambourakis, L. Akoglu, Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset, Future Internet 9 (2017). URL: https://www.mdpi.com/1999-5903/9/1/6. doi:10.3390/fi9010006.

[19] K. N. Jensen, M. Zhang, B. Plank, De-identification

of privacy-related entities in job postings, in: S. Dobnik, L. Øvrelid (Eds.), Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa), Linköping University Electronic Press, Sweden, Reykjavik, Iceland (Online), 2021, pp. 210–221. URL: https://aclanthology.org/2021.nodalida-main.21.

[20] P. Skondras, P. Zervas, G. Tzimas, Generating synthetic resume data with large language models for enhanced job description classification, Future Internet 15 (2023). URL: https://www.mdpi.com/1999-5903/15/11/363. doi:10.3390/fi15110363.

[21] A. Magron, A. Dai, M. Zhang, S. Montariol, A. Bosselut, JobSkape: A framework for generating synthetic job postings to enhance skill matching, in: E. Hruschka, T. Lake, N. Otani, T. Mitchell (Eds.), Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024), Association for Computational Linguistics, St. Julian's, Malta, 2024, pp. 43–58. URL: https://aclanthology.org/2024.nlp4hr-1.4.

[22] J. Li, A. Sun, J. Han, C. Li, A survey on deep learning for named entity recognition, IEEE Transactions on Knowledge and Data Engineering 34 (2022) 50–70. doi:10.1109/TKDE.2020.2981314.

[23] S. Neutel, M. H. de Boer, Towards automatic ontology alignment using bert., in: AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering, 2021, pp. 1–12.

[24] V. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics Doklady 10 (1966).

[25] Cedefop, Challenging digital myths – First findings from Cedefop's second European skills and jobs survey, Publications Office of the European Union, 2022. doi:doi/10.2801/818285.

[26] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[27] L. Ramshaw, M. Marcus, Text chunking using transformation-based learning, in: Third Workshop on Very Large Corpora, 1995. URL: https://aclanthology.org/W95-0107.

[28] I. Segura-Bedmar, P. Martínez, M. Herrero-Zazo, SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013), in: S. Manandhar, D. Yuret (Eds.), Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 341–350. URL: https://aclanthology.org/S13-2056.

[29] T. A. Green, D. Maynard, C. Lin, Development of a benchmark corpus to support entity recognition in job descriptions, in: Proceedings of the 13th Conference on Language Resources and Evaluation, 2022, pp. 1201–1208. URL: http://www.lrec-conf.org/proceedings/lrec2022/pdf/2022.lrec-1.128.pdf.

[30] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Huggingface's transformers: State-of-the-art natural language processing, 2020. URL: https://arxiv.org/abs/1910.03771. arXiv:1910.03771.

[31] Y. Mashayekhi, N. Li, B. Kang, J. Lijffijt, T. De Bie, A challenge-based survey of e-recruitment recommendation systems, ACM Comput. Surv. 56 (2024). URL: https://doi.org/10.1145/3659942. doi:10.1145/3659942.