

# Ontology Learning for ESCO: Leveraging LLMs to Navigate Labor Dynamics

Jarno Vrolijk<sup>1,2</sup>, Victor Poslavsky<sup>2</sup>, Thijmen Bijl<sup>2</sup>, Maksim Popov<sup>2</sup>, Rana Mahdavi<sup>2</sup> and Mohammad Shokri<sup>2</sup>

<sup>1</sup>University of Amsterdam, Plantage Muidergracht 12, 1018TV Amsterdam, Netherlands

<sup>2</sup>Randstad, Diemermere 25, 1112TC Diemen, Netherlands

## Abstract

The labor market is a dynamic environment that supports numerous knowledge-driven applications through ontologies, such as ESCO and O\*NET. Maintaining the relevance and accuracy of information within these ontologies and taxonomies is both resource-intensive and time-consuming. In this paper, we propose an ontology learning system that utilizes self-supervised learning, retrieval-augmented generation, and autoregressive language models to identify, classify, and link labor market mentions and entities from raw job postings. Additionally, we demonstrate the language model's ability to discover "alternative labels" and "preferred labels", and perform relation classification.

**Keywords:** Knowledge Graph, Natural Language Processing, Ontology Learning.

## 1. Introduction

Labor market ontologies enable the organization of information about jobs, skills, and qualifications, facilitating communication between job seekers and employers [1]. However, the labor market is a constantly evolving environment influenced by technological advancements, increasing individual choices, and shifting demographics. Consequently, educators, job seekers, and lifelong learners struggle to identify the relevant knowledge, skills, abilities, and competencies needed to distinguish themselves, each with unique objectives. Keeping these individuals and organizations informed about labor market developments in a timely and accurate manner is challenging and requires significant time and resources.

While many knowledge-driven applications, such as ESCO [2] and the O\*NET [3], have proven valuable in addressing some of the challenges within the labor market, they struggle to keep the information in their ontologies and taxonomies relevant and up-to-date [4]. These ontologies provide information about occupations, knowledge, skills, competences, and qualifications. Constructing these systems is complex, and current approaches are inadequate in handling the incomplete and dynamic nature of real-world knowledge graphs (KGs) [5, 6]. These approaches often fail to represent unseen entities, overlook the abundant textual information in ontologies and taxonomies, and are frequently based on ontological commitments that render them task-specific [5].


---

*KBC-LM'24: Knowledge Base Construction from Pre-trained Language Models workshop at ISWC 2024*

© 0000-0003-0409-4924 (J. Vrolijk); 0009-0002-4535-1413 (V. Poslavsky); 0009-0000-9550-2502 (T. Bijl); 0009-0000-1667-3216 (M. Popov); 0009-0003-2330-8470 (R. Mahdavi); 0000-0001-9250-6743 (M. Shokri)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Extensive research has been conducted on the (semi-)automated identification of terms, types, relations, and potential axioms from text, a process known as Ontology Learning (OL) [7, 8, 9, 10, 11]. Traditional methods for semi-automated extraction rely on lexico-syntactic pattern mining and clustering. However, considering the individual stages of OL—i) mention extraction (ME) and term typing (TT), ii) the discovery of hierarchical relationships, and iii) the discovery of non-taxonomic relationships—the recent advancements in large language models (LLMs) offer a cost-effective and scalable solution to OL.

LLMs enable the development of general-purpose and adaptable language models that can be tailored to various natural language processing (NLP) tasks such as classification, generation, and sequence labeling. Adapting LLMs to specific NLP tasks involves two phases: the pre-training phase, to obtain pre-trained language models (PLMs) typically formalized as a cloze-style task (i.e., sequential and/or masked language models), and the downstream phase, which involves fine-tuning the model or prompt tuning [12, 13]. In the downstream phase, KGs are considered in recent research to adapt PLMs to tasks such as Named Entity Recognition (NER), Relation Extraction (RE), Open Information Extraction, Entity Linking (EL), and Relation Linking [14]. To perform these tasks, the PLM is guided by the KG provided by the ontology (i.e., concepts, relations, domain/range constraints) and a set of sentences.

Despite the significant accomplishments, using PLMs remains challenging and error-prone, irrespective of their size. Firstly, the absence of a grounding mechanism complicates the fact-checking of answers, particularly for tasks with an extractive nature, which are prone to hallucination risks. Plus, many business automation workflows demand a high level of accuracy and thus often incorporate human-in-the-loop interactions for auditing and correcting predictions. This process necessitates knowledge about the precise location of the extracted mentions in the text. Besides, disambiguation of the actual terms requires extra domain-specific knowledge (such as soft skills [15]), making these processes as tedious and error-prone as their predecessor equivalents (i.e., knowledge-driven applications using ESCO [2] and O\*NET [3]).

To tackle the aforementioned issues, in this paper, we proposed a framework with OL to extend and maintain ESCO. The primary contribution of this paper is the development and implementation of a system capable of processing online job postings to extract skills, occupation entities, and their corresponding relationships. Furthermore, the system can identify “new entities” that are not yet included in ESCO, flagging them for further examination by a knowledge or ontology engineer. Our methodology addresses multiple core aspects of OL to answer the following research questions:

- **RQ1:** How effective is the proposed system in automated skill mention extraction from online job postings?
- **RQ2:** How effectively is the proposed system classifying non-taxonomic relations between skill and occupation types?
- **RQ3:** Is the proposed system capable of finding existing and/ or new entities that can extend ESCO?

## 2. Related Works

OL addresses the challenges of knowledge acquisition and representation across various domains [16, 7]. OL can be subdivided into several sub-processes, including the automatic identification and extraction of terms, types, relations, and axioms from text. The study by [17] introduces LLMs for OL using prompt-based learning. This approach leverages PLMs and cloze-style language prompts to achieve promising results in various NLP tasks, such as sentiment classification, knowledge probing, and natural language inference [18].

In their evaluation, [14] assessed two open-source LLMs (Vicuna-13B and Alpaca-LoRA-13B with in-context learning) and a sentence transformer model (SBERT T5-XXL) using the benchmark Text2KGBench [14]. The results indicate high ontological conformance for both Wikidata-TekGen and DBpedia-WebNLG corpora. However, these off-the-shelf LLMs performed poorly on fact extraction, which the authors attribute to a lack of fine-tuning [14]. To bridge the gap between semantic labelling tasks and text generation models for NER, [19] proposed GPT-NER. This method transforms the NER task into a text-generation task and includes a self-verification strategy to mitigate the excessive confidence of LLMs [19]. The results show performance comparable to fully supervised baselines based on BERT.

Additionally, [20] introduced LMDX, a methodology for using LLMs in information extraction, particularly from visually rich documents. Their approach achieved a new state-of-the-art on publicly available benchmarks such as CORD and VRDU [20]. Despite these notable findings, the authors primarily focused on extracting mentions from text while grounding their predictions.

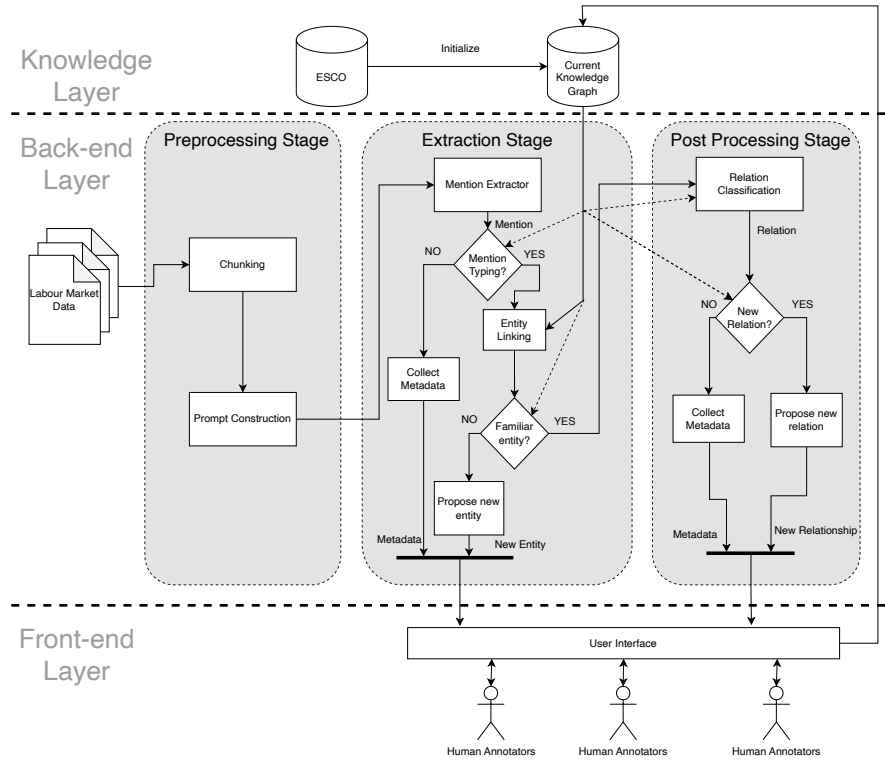
In the labor market, many researchers leverage occupation ontologies to extract relevant information from job posts. The work in [21] adapts a language model from ESCO to extract and classify skill requirements from German-speaking job descriptions. The work in [22] detects skills that are literally or implicitly mentioned in job ads and links them to ESCO. Besides, [23] fine-tuned the Llama model for extracting skills from job advertisements and user profiles. The authors of [24] investigate the zero-shot approach for extracting skills based on ESCO. Despite the aforementioned work, the work in [25] pre-trains a skill-aware language model usable for domain-specific downstream tasks, such as job classification or skill extraction.

There have been several works addressing OL in the labor market domain. The work by [26], proposes NEO, a framework using approximately 2 million online job vacancies for the enrichment of ESCO occupations. NEO identified 49 novel occupations of which 43 were validated by an expert panel [26]. Furthermore, [8] proposed OntoJob, a cost-efficient unsupervised framework that identifies and extracts knowledge, skill, ability, and competence mentions and their corresponding relations using the C-value method and smoothed point-wise mutual information (SPMI). [27] investigate the use of large language models for skill extraction leveraging in-context learning to test two different prompting strategies. While there are parallels to the work presented by us, [27] does not focus on knowledge discovery and relation classification.

## 3. Methodology

To ensure the labor market ontology remains current and accurate, we propose a three-layered framework: the knowledge layer, the back-end layer, and the front-end layer. The knowledge

layer houses the existing knowledge graph, initially based on established labor market graphs such as ESCO. The back-end layer processes online job postings over specified intervals (e.g., weekly or monthly). This layer recommends new labor market mentions and entities, including occupations and skills. The front-end layer serves as the user interface (UI), enabling a human-in-the-loop mechanism via human annotators for updating the knowledge graph. Figure 1 illustrates each layer in updating the labor market’s knowledge graph. Our back-end architecture is divided into three stages: (i) preprocessing, (ii) extraction, and (iii) postprocessing.



**Figure 1:** Overview of the ontology learning system for the labor market. The solid lines are used during both the training and serving stages. However, the dashed lines are only used for training.

### 3.1. The preprocessing stage

To address input size limitation in LLMs, we split incoming data into manageable segments, allowing the models to better process and understand context. Given document lengths and PLM context window limitations, each job description text is divided into chunks. Then, The prompt construction is used for formatting language model’s inputs and specifying how to generate the output. Carefully designing prompts plays an essential role in specifying exactly what sort of task and output format is expected. In our problem, for the prompt generation stage, we largely adhere to the recommendations by [20]. We take the full set of documents and apply a prompt template to format the document, beginning with “Document:”. Next, we append the task description and schema representation containing the entities to be extracted.

The task description includes hard-coded instructions to guide the PLM in formatting the output according to the schema. We provide the PLM with the following instruction: “*Extract [ENTITY TYPE] from the following document and format the output as a JSON with the following structure: [SCHEMA].*” In line with [20], the schema representation is a structured JSON object, where the keys are the entity types to be extracted, and the values indicate their occurrence (e.g., “1” for a single instance and “[]” for multiple instances). For example, “occupation”: “1”, “skill”: “[]” instructs the PLM to extract a single mention of the entity type “occupation” and multiple mentions of the entity type “skill.” In this context, a prompt template is a predefined format used to structure input for the PLM, and the schema is a blueprint describing the structure and organization of the data elements that need to be extracted.

### 3.2. The extraction stage

Once the input documents are processed in the preprocessing stage, we can begin extracting the labor market mentions specified by the ESCO. Specifically, we focus on identifying the ESCO skill entities, which involves locating skill mentions within the documents. However, our method can be extended with other mention types like occupations or other entity types that might occur in job postings.

We define the mention extraction task to be two-fold, namely: (i) identification and extraction of (sub-)strings in a given job posting, and (ii) indicating the term type of the given mention (i.e., is it a skill or occupation mention). In short, given a set of ESCO types  $\mathcal{T} = \{skill, occupation\}$ , we aim to find the mentions  $m$  of the types  $\mathcal{T}$  from the total set of job posting documents  $\mathcal{D}$ .

Similar to the design by [20], we use a decoder-only model for the extraction. A decoder-only model predicts the next word to generate based on the previous words in the sequence. We use the prompt instruction together with the desired output schema to instruct the PLM on what entity mentions we want it to extract. We instruction-tune task-specific parameters in addition to the pre-trained Mistral parameters on a data mixture containing a variety of (document, schema, extraction) tuples [28, 20]. In contrast to the work by [20], we have a fixed schema during our instruction tuning phase (i.e., we only train on a domain-specific data mixture). Furthermore, all our documents are online job postings.

Since we use a decoder-only model for “completing” the “output” with the correct mentions, and types found in the actual job postings, extraction and typing happen at the same time. However, we do want to separate the two different tasks since it is quite possible for the model to extract mentions that are suitable for extraction but are put into the wrong category, which leads to a mention typing mistake.

Following the earlier instructions on the prompt construction and consequently, the task description and the schema representation, the prompt for the model for the mention extraction and entity linking tasks is as follows:

```
“<s>[INST] Extract {types} from the following document and format the output as a json with
the following structure: {format}
Document: {document} [/INST]
{output}</s>”
```

Then, we collect the mentions that are not being passed from the previous step to report

them to the human annotators as metadata that helps them annotate. Afterwards, given the set of extracted mentions  $\mathcal{M}$ , and the set of ESCO concepts  $\mathcal{C}$ , we aim to map the extracted mentions  $\mathcal{M}$  to the closest entity in the ESCO taxonomy. To achieve this, we define entity linking to provide a many-to-one mapping for all mentions  $m \in \mathcal{M}$  to their corresponding entity  $c \in \mathcal{C}$  as  $f(m) : \mathcal{M} \mapsto \mathcal{C}$ . To create the mapping  $f(\cdot)$ , we propose leveraging a retriever, following the retrieval augmented generation design first proposed by [29], where for each extracted mention, the top five closest entities are chosen. Given the extracted mention, we will retrieve the approximate nearest ESCO entities using the retriever. Note that the index will be entity-specific, meaning that the dense vector representations for ESCO skill entities are in a different index than those for ESCO occupation entities. To map extracted skill mentions to ESCO skills, we make use of both the "PreferredLabel" and the "AlternativeLabels" provided by ESCO for each skill to make it easier for the retriever to retrieve the right skills, if they exist in the taxonomy. Next, we leverage the results from the retriever and combine them with instruction tuning [30]. We use the following prompt for the familiar entity check task:

```
"<s>[INST] Given a skill and options, select the best option that is a semantically exact synonym for the skill. If none of the options is a semantically exact synonym, select 'No Match'.
Skill: {skill}
Options: {options}
Simply answer with the correct option with no explanation. [/INST]
{output}</s>"
```

In contrast to the work by [31], we also consider the "No Match" option to indicate that none of the ESCO entities are a good match for the given mention.

Essentially, the PLM is tasked with flagging the entity as "undiscovered" or mapping it to one of the given ESCO entities provided by the retriever. The retriever acts as a filter to limit the solution space of the matching to the most likely candidates, thus reducing the  $|ESCO - Skills|$ -classification to a 6-class classification problem instead.

After classifying the found mentions in the entity linking, we use mentions that were marked as "undiscovered" and occur frequently, to propose new entities to human annotators. When the frequency of a mention exceeds a set threshold, we propose it as a new entity to the human annotators, who can decide whether to add it as a new entity, add it as a synonym to an existing entity, or not add it to the taxonomy at all if it is irrelevant.

### 3.3. The postprocessing stage

Conceptualization of the identified and extracted mentions is fourfold, namely; i) mention extraction - identifying and extracting relevant terms, ii) entity linking - mapping the identified and extracted mentions to their corresponding entities, iii) relationship extraction - identification and extraction of the relations between the identified and extracted mentions, and lastly iv) relationship classification - to map the identified and extracted relationships to their domain-specific equivalent. We will primarily focus our attention on the non-taxonomical relations, in particular: i) relationship(s) between mentions  $m \in \mathcal{M}$ , and entities  $c \in \mathcal{C}$  such that for  $v \in \mathcal{V}$ , where  $\mathcal{V}$  is the total set of relation types in our knowledge graph, we look for the triplet

relation  $r = (m, v, c) \in \mathcal{R}$ , and ii) relations between entities such that  $(c_i, v, c_j) \in \mathcal{R}$ , where  $c_i \neq c_j$ . We will refer to relations i) as classifying whether a mention is an alternative label - or synonym - for the entity in question, whereas we refer to ii) as finding out whether a skill entity is essential, optional, or unrelated to an occupation entity.

Since our research focuses on a subset of the entities in ESCO, we will also solely focus on the possible links between these entities. As such, we will mainly look at the "IsOptionalFor", and "IsEssentialFor" labels. In a similar fashion to the work by [31], we will construct a dataset using the relations found in ESCO. Given that there are only three potential options, namely  $\mathcal{S} = \{isOptionalFor, isEssentialFor, notRelated\}$ , we opt for a similar approach to the entity linking discussed earlier, but without a retriever (since there is no need to reduce the number of classes, as was the case with the entity linking task). As such, we will task the PLM to select the relation between a given skill and occupation entity from the set of relations  $\mathcal{S}$ . We use the following prompt for the model for the relation classification task:

```
"<s>[INST] Given a skill and an occupation, tell me how important the skill is for the occupation choosing from the following three options: essential, optional or not important.  
Skill: {skill}  
Occupation: {occupation}  
Simply answer with the correct option with no explanation. [/INST]  
{output}</s>"
```

Instruction training of the Mistral 7B model was done by leveraging the "AlternativeLabel" and "PreferredLabel" data from ESCO in the generation of a train and test set. We used a true "AlternativeLabel" related to the actual PreferredLabel as a positive example and randomly sampled  $K - 1$  non-related "AlternativeLabels" for negative examples. We would then use this dataset and the instruction to train task-specific parameters in addition to the pre-trained Mistral parameters for the entity linking task [28]. For more information on the evaluation and implementation details, we refer to Section 4.

In the current implementation, we only take into consideration two relations, namely: i) "isOptionalSkillFor", and ii) "isEssentialSkillFor". Therefore, the check is relatively straightforward: we see if the incoming entity pair ( $skill, occupation$ ) are related via either i) or ii), or we deem that the skill entity is not important to the occupation at all. If the relationship between the ( $skill, occupation$ ) pair did not exist, we propose a new relation with the given prediction.

Similar to how we suggest new entities to human annotators, we also identify and propose new relations between different entities in the taxonomy. By classifying relations between skill entities and the occupation of the posting, we analyze currently non-existent relations. If a frequently occurring new relation is found, we propose it to human annotators, who can then decide whether to add it to the taxonomy.

## 4. Experiments

This paper aims to leverage ESCO to cost-efficiently optimize and fine-tune PLMs for i) mention extraction and term typing, ii) entity linking and knowledge discovery, and iii) relationship classification. These fine-tuned PLMs, in turn, will help in the construction and maintenance of

the ontology and taxonomy. As described, we propose an evaluation of the full system and the individual PLMs performance on each of these three tasks.

#### 4.1. Dataset

In order to answer our research questions, we propose the three following experiments.

Dataset	Train			Dev			Test		
	ME	EL	RC	ME	EL	RC	ME	EL	RC
# total	635	40,409	1,459,630	-	5,051	250	58	5,052	2,000
# occupations	-	-	3,005	-	-	236	-	-	1,392
# skills	29,837	10,617	13,078	-	1,339	231	2,142	1,332	1,496
# essential	-	-	448,363	-	-	70	-	-	609
# optional	-	-	393,495	-	-	74	-	-	509
# negative	-	-	617,772	-	-	106	-	-	882

**Table 1**

Statistics of the different datasets used in this study. Where we use the following abbreviation for tasks: i) ME, ii) EL, and iii) relation classification (RC).

#### 4.2. Experiment 1: Mention Extraction

In order to extract skill mentions, we make use of the SkillSpan benchmark dataset, which was provided by [32]. In particular, we employ the publicly available HOUSE, and TECH data annotations, which contain, respectively, 90 and 110 job postings. In addition to the SkillSpan dataset, we also make use of approximately 495 manually annotated job postings. This proprietary dataset contains 1,058 chunks, which in total comprise 24,760 skills.

As the data from the SkillSpan dataset and the proprietary dataset used internally were in the BIO-tagging format, it was necessary to transform this data into a list of raw skill mentions, as seen in the texts. It was essential to obtain the full raw skill mentions, as this is the type of output that our extractor will be trained to extract. As our objective is to extract both hard and soft skills, we employ both the "skill" and "knowledge" mentions as annotated in the SkillSpan dataset. Furthermore, while the original data was segmented on a sentence level, we utilize the provided vacancy index to convert the sentences back to their original job posting format.

Subsequently, the postings were divided into sections of 384 tokens to ensure that the complete prompt, along with the document and the expected output, would always fit within the context window of the model. To assess the performance of both the base and fine-tuned models, we conducted evaluations using all 58 job postings obtained from the SkillSpan test dataset. To ensure reproducibility, no additional job postings were incorporated into the evaluation set. As the input, we provided the model with complete job postings. Subsequently, the generated output, which consists of the extracted raw skill mentions, was utilized to determine the F1 score of the model in categorizing each token in the vacancy text as either a skill (1) or a non-skill (0) token. The F1 score for the test set is presented in Table 2. For the mention extraction, we train a LoRA of the Mistral-7B model. The model is trained on the job posting chunks for four epochs and a batch size of eight. The ADAM optimizer is quantized with an 8-bit precision, with  $5 * 10^{-5}$  learning rate, 50 warmup steps, and a weight decay of 0.01.



### 4.2.1. Baselines

For the mention extraction, we will utilize the most effective model from the study by [27] as a baseline. This decision is primarily motivated by the fact that both our study and the study by [27] use the evaluation data provided by [32]. Furthermore, [27] considers an extracted entity correct even if it only partially overlaps with the gold span from the annotation. This aligns with our metric, thus facilitating comparison. Additionally, we will compare the results of our model to those discussed in the blog by [33]. The Skills Extractor Library, developed by [33], employs a NER model based on spaCy’s architecture. This model maps the extracted skills to existing taxonomies using semantic similarity. Conducting this comparison will allow us to evaluate the performance of our model relative to their established skill extraction framework. Our last baseline for the mention extraction experiment will be the models developed in the SkillSpan paper [32]. The two models, one for “knowledge” extraction and one for “skill” extraction, are BERT-based token classification models. We combine the results of both the “knowledge” extractor and the “skill” extractor and consider tokens as either not a skill token or a skill token without making any distinction between “knowledge” or “skill” or the beginning tokens (B) and inside (I) tokens.

### 4.3. Experiment 2: Relation Classification

In the second experiment, the objective is to classify relations between skill and occupation pairs. To construct the datasets for this experiment, we make use of all known relations between skills and occupations in the ESCO database. For each skill/occupation relation, up to six variations are included, utilizing the available alternative skill labels in ESCO. This encompasses the *optional* and *essential* relations. Furthermore, for each of the existing relations, five random skill-occupation combinations are sampled from the taxonomy. The aforementioned random combinations will serve as the *not important* relation samples. Table 1 provides an overview of the dataset distribution. We evaluate the system by computing the F1-score on the test set detailed in Table 1.

The training setup is as follows: A LoRA of the Mistral-7B model was trained. The model was trained for 1500 steps with a batch size of 32 and 20 warmup steps. The Adam optimizer, quantized to 8 bits, was employed with a learning rate of  $5 * 10^{-5}$  and a weight decay of 0.01.

### 4.4. Experiment 3: Knowledge Discovery

To create the dataset for the entity classification dataset, we again make use of the ESCO alternative labels for the skills. For each skill, we use up to 4 alternative labels to generate data points. Each data point consists of the alternative label that is treated as the raw skill mention, the preferred label as the correct answer, and the top 5 closest skills from the retriever. With this information, we can fill in the prompt and expected output to train the model. For 40% of the data points we show to the model, we will not include the correct skill as an option, and instead, we show 5 incorrect options for which the target response will be “No Match” to allow for the discovery of new skills or to discard mentions that should not be seen as a skill.

To get insights into the performance of the proposed decoder-only model for i) linking extracted skill mentions to ESCO skill entities, and ii) discovery of new potential ESCO skill

entities, we propose the following experimental setup. First, we will test the performance of the model in the entity linking task by evaluating the F1 score on our test set.

In addition, to know how the model performs on the discovery of new potential ESCO skill entities, we manually annotate 1,237 skill mentions extracted by our earlier stages in two different stages. In the first stage, we manually annotate whether the skill mention matches one of the five (i.e., assigning it the number of the suggestion) proposed suggestions by the retriever or assign it a 6 if it matches none of the suggestions. Next, we filter out all the skill mentions annotated with a 6, and check them for the following cases: i) the extracted mention itself does not describe a skill (e.g., an occupation or organization name etc.), ii) the mention maps to one (or more) existing ESCO skills but these options were not in the 5 suggestions, iii) the model selected one or more good options for the list but the mention also includes additional skills that are not in the top 5, iv) the mention is a proper skill mention, but ESCO does not contain any suitable skills in the current taxonomy, and lastly, there is not enough context in the extracted "mention" to judge (e.g., the mention just states "development").

We train a LoRA of the Mistral-7B model. We train the model for 300 steps with a batch size of 2 and 50 warmup steps. We use a quantized Adam 8-bit optimizer with a learning rate of  $2.5 * 10^{-5}$  and a weight decay of 0.01. Retrieval of the five "closest" skills, is done with dense retrieval. For embedding the skills, we used MixedBread without any additional fine-tuning.

#### 4.4.1. Baselines

To evaluate the performance of our mapper, we will compare its results to those of the mapper introduced by [33]. The Skills Extractor Library developed by [33] utilizes MiniLM to encode skills and map them to the closest ESCO entity. To assess the effectiveness of our approach, we use the same dataset with the ESCO alternative labels but without "No Match" data points. We employ the evaluation method from [33] and present the F1 score in Table 2 for comparative analysis. It is worthless that in this case, the mapper from [33] solves a simpler problem compared to ours since it does not consider "No Match".

## 5. Results

### 5.1. Experiment 1: Mention Extraction

Table 2 demonstrates that the Mistral 7B model (the Base model) has an F1 score of 0.28 for skill mention extraction across 58 job postings. Following the instruction tuning of a Low-Rank Adaptation (LoRA) model on 635 manually annotated job postings (denoted as Base + ME), evaluation on the same set of 58 job postings results in an F1 score of 0.54. This indicates that instruction tuning on self-supervised labor market ontology data enhances performance by approximately 0.26. We also conducted a comprehensive comparison by evaluating three existing methods on the same dataset. Specifically, the model proposed by [27] yielded an F1 score of 0.46, while the baseline model discussed in [33] achieved one of 0.27. Notably, the model developed by [32] outperformed our Base + ME model, achieving an F1 score of 0.80 on the evaluation dataset. These results suggest that our model's performance is comparable to, but not in all cases superior to, other existing methodologies on this task.

Model ↓ / Task →	ME	RC	EL
[27]*	0.46	-	-
[33]*	0.27	-	0.57
[32]*	<b>0.80</b>	-	-
Base	0.28	0.54	0.30
Base + ME	0.54	-	-
Base + RC	-	<b>0.66</b>	-
Base + EL	-	-	<b>0.67</b>

**Table 2**

F1 scores for three experiments evaluating the effect of fine-tuning LoRA weights on task-specific domain data, in comparison to existing methods. Column ME shows the performance on the mention extraction and term typing task (Experiment 1). Column RC compares the performance on the relation classification task (Experiment 2). Column EL presents the results for Experiment 3, which measures how well the models contribute to knowledge discovery through entity linking.

## 5.2. Experiment 2: Relation Classification

Results from Table 2 show that the Base model scored a F1 score of 0.54. The instruction fine-tuned LoRA model, the so-called “Base + RC”, scored an F1 of 0.66. Results include incorporation of negative examples as dictated in Table 1, and shuffling the options into a random ordering. In total, we see that instruction tuning leads to an approximate performance increase of 0.12.

## 5.3. Experiment 3: Knowledge Discovery

The third experiment was essentially twofold. Firstly, we report the results of evaluating the parsed skill mentions from the job posting test set. Table 2 shows us that the “Base + EL” model scored an F1 score of 0.67, while the Base model only scored an F1 score of 0.30. As such, instruction tuning the Mistral Base leads to an increased performance of approximately 0.37. Besides, we compared our method with the baseline model described in [33], which yielded an F1 score of 0.57. In this case, “Base + EL” model showed a better result, despite the fact that it solved a more complex problem, not only identifying the most fitting ESCO skills, but also indicating potentially new ones.

Results for the manual annotation of the mentions marked as “new entities” showed an F1 score of 0.41. In the cases where the extracted mention is not a valid skill mention or there is not enough context, the model shows an F1 score of 0.42. Lastly, in the case where the extracted mention can map to an existing ESCO skill but this ESCO skill was not included as one of the 5 options that the model could select from, the model obtains an F1 score of only 0.16.

Manual annotation of the 1,237 extracted skill mentions showed that 704 of those mentions could not be linked to one of the five provided suggestions of the retriever. Careful examination of the 704 skill mentions that could not be linked to the five provided suggestions showed that 94 had an existing ESCO entity, but the retriever failed to select the appropriate entity for the suggestion. From the 704 total skill mentions 282 were manually annotated to be a potential “New Skill”. Additionally, 136 were not actual skill types, and 160 lacked the context to make a valid prediction. In total, the model extracted a total of 253 skill mentions that were annotated as a potential new skill to be reviewed by human annotators for addition to the ESCO taxonomy. A few examples of these mentions are: “ReactJS”, “AWS”, and “Docker”.

## 6. Discussion

Our paper explored three different research questions. To address **RQ1**, we instruction-tuned a LoRA model to the Mistral Base on a total of 635 job postings. According to 2, the "Base + ME" model outperforms the "Base" model in extracting the skill mentions by scoring an F1 of 0.54 compared to 0.28. Accordingly, although the "Base" model struggles with the extraction of the skill mentions, decoder-only architectures can be instruction-tuned to extract and format skill mentions from raw job posting texts. Plus, on the ME task, the "Base + ME" model outperforms the best-performing model from [27] by 0.8. While it is difficult to credit this difference solely to the proposed system (i.e., this would require more details on the actual differences between gpt-3.5 turbo and the Mistral Base model), we believe that it demonstrates the competitiveness of our proposed system with the state-of-the-art. We still see that the BERT-based models from [32] outperform our proposed method. However their proposed system has a major drawback that it requires BIO-tagged data on the token level to train, which is labour-intensive to obtain compared to just having to obtain a list of mentions in the text like we need for our approach.

To answer **RQ2**, we performed the relation classification which determines if a skill entity is "optional", "essential", or "not important" to be added ESCO. The results indicate that the autoregressive model is capable of learning the relation classification task via self-supervised instruction tuning from ESCO. However, we only trained the Base + RC model on 32,000 examples due to time constraints. Thus, there exist opportunities to improve the current model's performance by training with more examples.

To the best of our knowledge, there is no other study that looks at the relation classification between ESCO skill, and occupation entities using autoregressive models as proposed in this work. However, the work by [31] can be regarded as very similar. [31] perform entity classification, and relation classification at once, therefore, we can't use their F1 scores for direct comparison. Having said that, the "Base + RC" models' performance appears to be on par with the F1 score of 0.51, outperforming by 0.15 on the slightly different task. [31] has a model that predicts both the types of the subject and object and the predicate while constraining the possible labels to a predefined set (i.e., choose between skill and occupation for the entity type). On the other hand, the "Base + RC" model only predicts the predicate.

Lastly, to answer **RQ3**, in the knowledge discovery experiment, we consider two different experiments. Results from the first experiment help us figure out whether instruction tuning the "Base" model with self-supervised data from ESCO would increase the performance on the entity-linking task. The "Base + EL" model scores approximately 0.37 above the "Base" model, demonstrating the effectivity of self-supervised instruction tuning using ESCO. Additionally, having a more complex task, which includes, in addition to entity linking, the indication of not familiar entities, the "Base + EL" model outperforms the method proposed in [33] by 0.1. This demonstrates that our approach is highly competitive with other methods in entity linking.

The second experiment grants us insight into the ability of the decoder-only model to augment and enrich the ESCO with skill mention suggestions for human annotators. The "Base + EL" model suggested 1,054 skill mentions with no matching ESCO entity, whereas manual annotation by 6 human domain experts revealed only 704. However, there was no further manual annotation of the 1,054 mentions predicted to have "No Match" and the quality of the suggestions provided by the retriever. We believe that similarly to the 704 mentions that

received annotation, there will be a proportion where the provided suggestions by the retriever were wrong (i.e., 94 out of the 704 for which we provided manual annotation had existing ESCO skill entities that were not suggested as an option by the retriever).

Overall, the “Base + EL” models are capable of selecting mentions that have no match in ESCO. However, there is a need for post-processing steps to filter out “false positives” *inter alia*; the extracted mention is not a “skill”, lacks the full context to make a prediction, and was wrongly classified as “No Match” due to missing suggestion. For this study, post-processing was done manually, leading to flagging 253 extracted skill mentions as a potential addition to ESCO.

## 7. Implications and Limitations

The results from this paper demonstrate the effectiveness and adaptability of using LoRA and decoder-only architectures for ontology learning in the labor market setting. Results indicate that the generation of self-supervised datasets, combined with instruction tuning, leads to impressive performance gains on skill mention extraction, relation classification (i.e., on skill and occupation entity pairs), and lastly, discovery of skill mentions that could potentially extend the labor market ontology/ taxonomy. We believe the unification of LLMs and ontologies can aid in the lacking abilities of knowledge persistence in LLMs. Since the labor market is a constantly changing environment, editing knowledge without re-training the whole LLM is of utmost importance. The proposed system provides an intuitive way to enrich and maintain ontologies (not just labor market specific), while at the same time leveraging the knowledge of the ontology to keep the models up-to-date by creating self-supervised training sets.

Furthermore, we believe that our results demonstrated the potential strength of using a proposed system to augment and enrich existing labor market ontologies and/ or taxonomies (i.e., ESCO, the O\*NET, etc.). In particular, our results show the pivotal role of the retriever in easing the construction of self-supervised data that the decoder-only model easily leverages via instruction tuning. Additionally, our models demonstrate utility in alleviating the time- and resource constraints in human annotation by training “smaller” language models to assist (i.e., models that fit on a single GPU).

The current study has some limitations to be considered. Firstly, for entity linking, we tried leveraging the skill attribute “AlternativeLabel” as provided by ESCO. However, we did so under the assumption that the listed alternative labels are in a way synonyms to the “PreferredLabel”. This assumption does not necessarily hold in reality. Secondly, the current study did not experiment with hyperparameter tuning during model training and evaluation. There is considerable room for improvement of the models via hyperparameter tuning.

Thirdly, another limitation in the entity-linking experiment is that we do not consider the full context of the job posting when linking the entity. For example, the extracted mention “engineering”, gets five valid suggestions, namely; “software engineering”, “packaging engineering”, and “power engineering”. Since there is no context, none of the five options is more valid than the other. Finally, our current implementation does not incorporate any knowledge-grounding methodologies. During the study, we experimented with the incorporation of index values to ground the extractions by similarly checking the index to the work by [20]. However, Mistral 7B seemed to have trouble with the provided index values.

## 8. Conclusion

We introduce an OL system to upgrade the existing labor market ontologies. We analyzed the job postings to extract new entities such as occupations and skills. We propose a framework to recommend entity linking between skills and occupations. To evaluate the performance, we designed multiple experiments to address our research questions about the performance of skill extraction, non-taxonomical relationship retrieval, and knowledge discovery.

As future work, we add “post-processing” filters to better distinguish the different types of non-matches would be valuable. This potentially saves the human annotator from sifting through mentions that are amongst other things; not skill types, too vague, and/ or present in ESCO. Furthermore, there is considerable room for improvement on the hyperparameter settings used in training the models in this paper, we consider this one of the easiest avenues for improvement of the results. Lastly, we would be very interested in testing out the current system on a variety of different types that are currently not part of ESCO. For example, extracting wage information, educational requirements, benefits, requirements about work experience, etc.

## References

- [1] J. D. Smedt, M. le Vrang, A. Papantoniou, *Esco: Towards a semantic web for the european labor market*, in: LDOW@WWW, 2015.
- [2] E. Commission, *Esco handbook european skills, competences, qualifications and occupations*, Publications Office of the EU (2019).
- [3] National Center for O\*NET Development., *O\*net online.*, 2024. Online; accessed 4-June-2024.
- [4] J. Djumalieva, C. Sleeman, *An open and data-driven taxonomy of skills extracted from online job adverts*, in: *Developing skills in a changing world of work*, Rainer Hampp Verlag, 2018, pp. 425–454.
- [5] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, *Unifying large language models and knowledge graphs: A roadmap*, *IEEE Transactions on Knowledge and Data Engineering* (2024).
- [6] E. M. Sibarani, S. Scerri, C. Morales, S. Auer, D. Collarana, *Ontology-guided job market demand analysis: a cross-sectional study for the data science field*, in: *Proceedings of the 13th international conference on semantic systems*, 2017, pp. 25–32.
- [7] P. Buitelaar, P. Cimiano, B. Magnini, *Ontology learning from text: methods, evaluation and applications*, volume 123, IOS press, 2005.
- [8] J. Vrolijk, S. T. Mol, C. Weber, M. Tavakoli, G. Kismihók, M. Pelucchi, *Ontojob: Automated ontology learning from labor market data*, in: *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, IEEE, 2022, pp. 195–200.
- [9] K. Frantzi, S. Ananiadou, H. Mima, *Automatic recognition of multi-word terms: the c-value/nc-value method*, *International journal on digital libraries* 3 (2000) 115–130.
- [10] S. Roller, D. Kiela, M. Nickel, *Hearst patterns revisited: Automatic hypernym detection from large text corpora*, arXiv preprint arXiv:1806.03191 (2018).

- [11] H. Mousavi, D. Kerr, M. Iseli, C. Zaniolo, Harvesting domain specific ontologies from text, in: 2014 IEEE International Conference on Semantic Computing, IEEE, 2014, pp. 211–218.
- [12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), *Advances in Neural Information Processing Systems*, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901.
- [13] T. Schick, H. Schütze, It’s not just size that matters: Small language models are also few-shot learners, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Online, 2021, pp. 2339–2352.
- [14] N. Mihindikulasooriya, S. Tiwari, C. F. Enguix, K. Lata, Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text, in: *International Semantic Web Conference*, Springer, 2023, pp. 247–265.
- [15] D. Beauchemin, J. Laumonier, Y. L. Ster, M. Yassine, "fijo": a french insurance soft skill detection dataset, 2022. [arXiv:2204.05208](https://arxiv.org/abs/2204.05208).
- [16] A. Konys, Knowledge repository of ontology learning tools from text, *Procedia Computer Science* 159 (2019) 1614–1628. *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES2019*.
- [17] H. Babaei Giglou, J. D’Souza, S. Auer, Llms4ol: Large language models for ontology learning, in: *International Semantic Web Conference*, Springer, 2023, pp. 408–427.
- [18] N. Ding, Y. Chen, X. Han, G. Xu, P. Xie, H.-T. Zheng, Z. Liu, J. Li, H.-G. Kim, Prompt-learning for fine-grained entity typing, *arXiv preprint arXiv:2108.10604* (2021).
- [19] S. Wang, X. Sun, X. Li, R. Ouyang, F. Wu, T. Zhang, J. Li, G. Wang, Gpt-ner: Named entity recognition via large language models, *arXiv preprint arXiv:2304.10428* (2023).
- [20] V. Perot, K. Kang, F. Luisier, G. Su, X. Sun, R. S. Boppana, Z. Wang, J. Mu, H. Zhang, N. Hua, Lmdx: Language model-based document information extraction and localization, *arXiv preprint arXiv:2309.10952* (2023).
- [21] 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. doi:10.18653/v1/2022.nlpccs-1.2.
- [22] J.-J. Decorte, S. Verlinden, J. V. Haute, J. Deleu, C. Davelder, T. Demeester, Extreme multi-label skill extraction training using large language models, 2023. [arXiv:2307.10778](https://arxiv.org/abs/2307.10778).
- [23] N. Li, B. Kang, T. D. Bie, Skillgpt: a restful api service for skill extraction and standardization using a large language model, 2023. [arXiv:2304.11060](https://arxiv.org/abs/2304.11060).
- [24] B. Clavié, G. Soulié, Large language models as batteries-included zero-shot esco skills matchers, 2023. [arXiv:2307.03539](https://arxiv.org/abs/2307.03539).

- [25] C. Fang, C. Qin, Q. Zhang, K. Yao, J. Zhang, H. Zhu, F. Zhuang, H. Xiong, *Recruitpro: A pretrained language model with skill-aware prompt learning for intelligent recruitment*, in: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '23*, ACM, 2023.
- [26] A. Giabelli, L. Malandri, F. Mercurio, M. Mezzanzanica, A. Seveso, *Neo: A tool for taxonomy enrichment with new emerging occupations*, in: *International Semantic Web Conference*, Springer, 2020, pp. 568–584.
- [27] K. C. Nguyen, M. Zhang, S. Montariol, A. Bosselut, *Rethinking skill extraction in the job market domain using large language models*, *arXiv preprint arXiv:2402.03832* (2024).
- [28] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *Lora: Low-rank adaptation of large language models*, *arXiv preprint arXiv:2106.09685* (2021).
- [29] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., *Retrieval-augmented generation for knowledge-intensive nlp tasks*, *Advances in Neural Information Processing Systems* 33 (2020) 9459–9474.
- [30] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, Q. V. Le, *Finetuned language models are zero-shot learners*, *arXiv preprint arXiv:2109.01652* (2021).
- [31] J. Vrolijk, D. Graus, *Enhancing plm performance on labour market tasks via instruction-based finetuning and prompt-tuning with rules*, *arXiv preprint arXiv:2308.16770* (2023).
- [32] M. Zhang, K. N. Jensen, S. D. Sonniks, B. Plank, *Skillspan: Hard and soft skill extraction from english job postings*, *arXiv preprint arXiv:2204.12811* (2022).
- [33] E. Gallagher, I. Kerle, C. Sleeman, J. Vines, *The skills extractor library*, 2023. Accessed: 2024-06-06.