JOBSKAPE: A Framework for Generating Synthetic Job Postings to Enhance Skill Matching

Antoine Magron[*1] **Anna Dai**[*1] **Mike Zhang**[2]
Syrielle Montariol[1] **Antoine Bosselut**[1]
[1]EPFL, Switzerland
[2]IT University of Copenhagen, Denmark
syrielle.montariol@epfl.ch mikejj.zhang@gmail.com

Abstract

Recent approaches in skill matching, employing synthetic training data for classification or similarity model training, have shown promising results, reducing the need for time-consuming and expensive annotations. However, previous synthetic datasets have limitations, such as featuring only one skill per sentence and generally comprising short sentences. In this paper, we introduce JOBSKAPE, a framework to generate synthetic data that tackles these limitations, specifically designed to enhance skill-to-taxonomy matching. Within this framework, we create SKILLSKAPE, a comprehensive open-source synthetic dataset of job postings tailored for skill-matching tasks. We introduce several offline metrics that show that our dataset resembles real-world data. Additionally, we present a multi-step pipeline for skill extraction and matching tasks using large language models (LLMs), benchmarking against known supervised methodologies. We outline that the downstream evaluation results on real-world data can beat baselines, underscoring its efficacy and adaptability. [1]

1 Introduction

In the dynamic modern labor market, understanding job demands at scale is crucial for informed decision-making by policymakers, businesses, and other stakeholders. One way of measuring job market demand lies in *skill matching*: the extraction and alignment of skills from job descriptions to their disambiguated forms (i.e., a knowledge base or taxonomy). This process facilitates the investigation of current labor market dynamics and the quantification of labor market demands, addressing the occupational skill matching problem.

Regardless of their predictive effectiveness, supervised learning methods for skill matching require regularly collecting and annotating up-to-date data (Zhang et al., 2022b), a process that is both expensive and time-consuming. Synthetic data circumvents the need for such costly annotations. However, despite efforts in generating synthetic training data (Clavié and Soulié, 2023; Decorte et al., 2023) and real-world benchmarks (Zhang et al., 2022a; Decorte et al., 2022), challenges like incoherent sentences and over-simplified setups exist in existing datasets. To address these shortcomings, we introduce JOBSKAPE, a framework for generating realistic skill matching datasets that can be used for training and benchmarking.

JOBSKAPE facilitates the creation of diverse labeled textual datasets that align closely with actual job postings, ensuring cleaner and more coherent data. We demonstrate its practical application by generating SKILLSKAPE, a large-scale dataset linking coherent sets of skills to corresponding job descriptions. JOBSKAPE uses generative large language models (LLMs) to curate meaningful skill combinations and generate appropriate job descriptions containing these combinations. A self-refinement step using LLMs (Madaan et al., 2023) ensures label quality in the refined SKILLSKAPE dataset, assessed through offline metrics. Finally, we challenge traditional supervised skill matching methods with an LLM-based, in-context learning (ICL) pipeline, to circumvent re-training the supervised model given new data. We evaluate skill matching performance on our synthetic dataset and real-world annotated data (Decorte et al., 2022), comparing our proposed extraction and matching pipeline with supervised matching models trained on our dataset as well as previous generation attempts (Decorte et al., 2023) in a controlled setting.

Contributions. In this work, we contribute the following: (1) we propose JOBSKAPE, a framework for generating a synthetic dataset of job de-

[*] Equal contribution.

[1] Code and data available at https://github.com/magantoine/JobSkape

Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024), pages 43–58
March 22, 2024 ©2024 Association for Computational Linguistics

scriptions for skill matching with existing skill taxonomies, (2) using our framework, we release a synthetic train and evaluation dataset (SKILLSKAPE) for skill matching, (3) we show that SKILLSKAPE has higher textual quality measured in perplexity and implicitness compared to previous synthetic datasets, (4) lastly, we introduce an ICL-based approach to extract and match skills from job descriptions to a taxonomy and show that this method can outperform supervised baselines on real-world benchmarks.

2 Related Work

Synthetic Data Generation. Traditional synthetic data generation relies on language models, where a generator model is trained on an existing dataset and then employed to generate new data (Mohapatra and Mohapatra, 2022; Kumar et al., 2020). More recent unsupervised methods, such as Wang et al. (2021), leverage pre-trained language models like GPT-3 (Brown et al., 2020) without the need for explicit supervision. Other examples include Ye et al. (2022); Gao et al. (2023), who use carefully designed prompts for data generation. Honovich et al. (2022) generate synthetic instructions for fine-tuning large language models, while Shao et al. (2023) create synthetic demonstrations to enhance the performance of prompting LLMs.

Synthetic Data for Job Postings. In the job market domain, Decorte et al. (2023); Clavié and Soulié (2023) both employ GPT-3.5/4 to generate synthetic training data for skill matching. Specifically, Decorte et al. (2023) prompt GPT-4 to generate ten examples for each ESCO skill, while Clavié and Soulié (2023) use GPT-3.5 to generate 40 examples for each ESCO skill. In this work, we compare our dataset with the one from Decorte et al. (2023), referred to hereafter as the DECORTE dataset.

Skill Matching. Earlier works focus on standardizing skills through matching with taxonomies. For supervised methods, Gnehm et al. (2022) extract skills from Swiss-German job descriptions and match them with the ESCO taxonomy in a two-step process. Zhang et al. (2022b) assume pre-extracted skills and classify spans into their respective taxonomy codes using multiclass classification. Decorte et al. (2022) use distant supervision with the ESCO taxonomy to obtain labels, employ binary classifiers for each ESCO skill and enhance training through negative sampling strategies. Decorte et al. (2023); Clavié and Soulié (2023) employ LLMs for skill matching with ESCO. Decorte et al. (2023) generate a synthetic training set using GPT-3.5 and optimize a bi-encoder through contrastive training for matching. Clavié and Soulié (2023) use a similar approach, generating synthetic training data and employing a linear classifier for each skill with a negative sampling strategy. Additionally, they use sentence embedders (Reimers and Gurevych, 2019) to measure the similarity between extracted skills and ESCO.

3 The JOBSKAPE Framework

Our goal is to create a synthetic dataset comprising job posting sentences associated with lists of skills from a taxonomy that closely aligns with real-world job posting sentences. We initiate the process by generating combinations of skills, derived from a given taxonomy, that are likely to coexist in a job description. Leveraging LLMs and refinement techniques, we produce diverse, realistic, and accurate job description sentences. To evaluate the quality of our synthetic data generation, we define a set of offline metrics and compare the generated sentences with real job postings.

3.1 The Label Space

In this study, we use the European Skills, Competences, Qualifications, and Occupations (ESCO; le Vrang et al., 2014) taxonomy as the label space. ESCO comprises 13,890 competencies categorized into *Skill*, *Knowledge*, and *Attitudes*. Knowledge, according to ESCO, involves assimilating information through learning, encompassing facts, principles, theories, and practices in a specific field of work or study.[2] For example, acquiring proficiency in the Python programming language through learning represents a *knowledge* component, classified as a *hard skill*. Conversely, the application of this knowledge to perform tasks is considered a *skill* component, defined by ESCO as the ability to apply knowledge and use know-how to accomplish tasks and solve problems.[3] For the synthetic sentence generation task at hand, we do not distinguish between skill and knowledge components.

Our synthetic dataset creation framework generates sentences containing multiple skills listed in

[2] https://ec.europa.eu/esco/portal/escopedia/Knowledge

[3] https://ec.europa.eu/esco/portal/escopedia/Skill

the ESCO taxonomy. To reduce the data generation cost (and to facilitate a fair comparison with prior work), we use the same subset of 514 ESCO skills used in SKILLSPAN-M, an annotated set of real-world job postings.

3.2 Formal Approach

Previous efforts (Decorte et al., 2023; Clavié and Soulié, 2023) focused on generating synthetic training sentences with a single skill. In contrast, we advocate for sentences containing multiple relevant skills to resemble sentences from real job postings. We initiate the process by creating combinations of skills, guided by three main conditions:

1. **Varying Lengths of Skill Combination**: Recognizing the heterogeneity in real-world job postings, we incorporate varying numbers of skills per sentence. By doing so, our sentences will show higher diversity similar to real job advertisements.

2. **Semantic Closeness in Skill Pairing**: In real job postings, skills that are mentioned in the same sentence are often related to each other. Aligning with the logical grouping of skills, we construct more realistic and contextually coherent sentences.

3. **Minimum Skill Representation**: While our dataset aims to reflect the real-world frequency of skill occurrences, we also want to ensure that each skill appears enough times for training. This guarantees that even less common skills are adequately represented, creating a more effective training dataset.

To achieve variety, we introduce two distributions, $\mathcal{N}$ (distribution of combination size) and $\mathcal{F}$ (distribution describing skill frequency in job postings, akin to skill popularity). We iteratively process skills $s_i \in \mathcal{S}$, the set of skills in our taxonomy, ensuring each skill has the same minimum number of samples. For each skill, we identify its k nearest neighbors $\{s'_j\}_{j=1}^k$ based on cosine similarity between embeddings obtained from JobBERT, a language model fine-tuned on domain-specific data (Zhang et al., 2022a). Neighbors with a similarity above threshold T are retained, forming the set of nearest neighbors $\mathcal{S}_i$:

$$\mathcal{S}_i = \{s_j : s_j \in \{s'_i\}_{i=1}^k \wedge \text{sim}(s_j, s_i)\}T\}. \quad (1)$$

This set is used for skill combination selection. We draw a sample size n from distribution $\mathcal{N}$, setting the combination size to $min(n, |\mathcal{S}_i|)$. However, sampling skills directly from $\mathcal{S}_i$ is not straightforward. For instance, in analyzing the top nearest neighbors of *SQL*, a frequently occurring skill in job postings, we find *THC Hydra*, which is much less common. To accurately replicate the real-world frequency distribution of skills in our synthetic dataset, we adjust sampling probabilities to reflect actual skill popularity. Hence, we introduce distribution $\mathcal{F}$ to compute the probability of selection over $\mathcal{S}_i$ using softmax:

$$\mathbf{P}(s_j) = \frac{e^{\mathbf{P}_{\mathcal{F}}(s_j)}}{\sum_{l=1}^{k} e^{\mathbf{P}_{\mathcal{F}}(s_l)}}. \quad (2)$$

where $\mathbf{P}_{\mathcal{F}}(s_i)$ is the popularity of the skill s_i. We then select $min(n, |\mathcal{S}_i|)$ skills from $\mathcal{S}_i$ using the computed probability distribution.

For dataset creation, we form skill combinations from our ESCO subset, with $\mathcal{N}$ set to $U(1, 5)$. We employ JobBERT to obtain domain-specific embeddings for job descriptions and skills. The distribution $\mathcal{F}$ is computed as the average of standardized negative perplexities across sentences generated with GPT-2 (Radford et al., 2019). These sentences include variations like "`I want a job that involves {skill}`", "`For my job, I want to learn [to] {skills}`", "`At my job, my main is skill is [to] {skill}`", ensuring grammatical correctness. We set a similarity threshold T to 0.83 to be closer to SkillSpan distribution, using $k = 20$.

3.3 Prompt Tuning for Generation

Given a skill combination, we generate synthetic job description sentences. A candidate for this hypothetical job would need to be proficient to some extent in each of these skills. We use GPT-3.5 as the text generator. We describe two types of generations:

- **Dense**: For a combination of four or less skills we generate a short job description of at most one sentence. This is done to minimize the number of hallucinated skills that could appear when generating a long job description with a small set of skills.
- **Sparse**: For a combination of more than four skills, we generate a job description paragraph containing multiple sentences. The information is more "sparse".

Our prompt follows Clavié and Soulié (2023), it is used to make the mentions of the skills as implicit as possible (i.e., skill does not have an exact string match in the text). We further enhance the diversity by prompting the model to vary the openings of the descriptions and avoid the examples starting with "`We are looking`" or "`We are searching`" (see Appendix A.3.1).

We add additional instructions to the prompt to reduce ambiguity. To each prompt, we add a list of synonyms of each inputted skill that are in the taxonomy and instruct the model to not refer to *SQL* as *MySQL* since *MySQL* is a separate skill in the taxonomy. Each skill is also given along with their respective definitions to give more context to the model and avoid miscomprehension.

3.4 Refinement of SkillSkape

At this step, the dataset comprises exclusively of positive samples, which means that every generated sentence contains at least one associated skill. To train a supervised classifier for real job descriptions, negative samples – sentences containing unknown skills or no skills – are required. To create negative samples with unknown skills, we apply the same generation method as positive samples but draw from a broader pool of skills. This pool includes skills that are not in our selected list but known in the wider skill universe. We also generate negative samples containing no skills to represent sentences in real-life job ads that do not mention skills required from the candidate. To do so, we use two separate prompts to generate (1) sentences describing the company: its reach, domain, location, et cetera and (2) sentences detailing the salary and perks of a job (see Appendix A.3.2). To guide the model in the generation, we provide two demonstrations. For the SkillSkape dataset, we generate 500 negative samples with unknown skills as well as 500 additional negative samples containing no skills.

We then apply self-refinement (Madaan et al., 2023), involving feeding the generated sentences back into the same model for feedback. The model is asked to extract skills using the pipeline described in Appendix A.4.1, matching them with the taxonomy. We compare the generated set of skills with the gold set of skills, adding to the gold list all skills that were found in the sentence. We do this because the LLM can extrapolate during sentence generation, thereby adding related skills on top of the original list that was fed to it. The list of skills, along with their associated spans in the sentences, is filtered to include only pairs of skills and spans that have a cosine similarity above a specified threshold. For this refined dataset version, we use JobBERT as a span encoder, and the threshold is empirically set to 0.7 cosine similarity. One of the main reasons for this low similarity is the LLM not reflecting accurately the gold skill during the sentence generation step. In that case, we wish to remove that skill from the gold set of skills associated with the sentence.

Span Extraction. We use GPT-3.5 to label skill sequences in the sentences. Each mention, whether implicit or explicit, is surrounded by `@@` and `##` following Wang et al. (2023). In case the language model fails to label the span, it is asked to self-correct, as outlined in Appendix A.4.2. We showcase two examples extracted from the training set of the refined SkillSkape dataset.

Positive example. This 28-word example, average in length for our dataset, contains three key skills as annotated spans required of the applicant.

> `**Sentence**: The ideal candidate will effectively @@engage with upper-level management##, @@maintain strong communication channels with key stakeholders##, and @@collaborate with peers## to ensure seamless coordination throughout the organization.`
>
> `**Label**: 'liaise with managers', 'communicate with stakeholders', 'liaise with colleagues'`

In the example, the inter-skill similarity is high, showcasing the efficiency of the skill combination selection method.

Negative example. This sentence mentions information about the hiring company instead of the job itself, and therefore, contains no skills.

> `**Sentence**: Embrace a challenging and fulfilling career with us, where your hard work is recognized through a salary range of $80,000 to $90,000, reflecting our appreciation for your contributions.`
>
> `**Label**: NO LABEL`

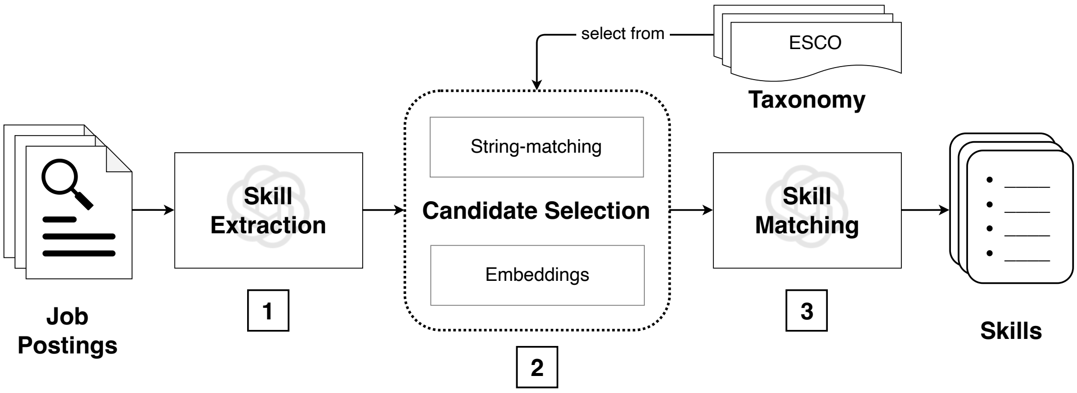

Figure 1: **Three-step Skill Extraction and Matching Pipeline**. We show our in-context learning pipeline for end-to-end skill matching. We use an LLM to extract skills from job ads, then do candidate selection using heuristics, and last, do skill matching with a constrained taxonomy.

Dataset	Split	Avg. # Skills	% UNKs	Avg. # Words	# Samples
SKILLSPAN-M	Dev.	2.0	47.0	15.0	178
	Test	1.9	47.3	16.3	751
DECORTE	Train	1.0	0.2	15.7	5,120
SKILLSKAPE	Train	2.6	7.9	28.2	6,352
	Dev.	2.1	8.3	27.8	1,316
	Test	2.6	8.4	28.1	1,272

Table 1: **Datasets' Statistics.** Average # skills and words refer to the average per sample (job posting sentence(s) and % UNKs refer to the percentage of skill labels are under the unknown UNK label.

3.5 Summary and Comparison

The final version of SKILLSKAPE has 8940 samples, split into training, development and test sets ($\sim$ 70-15-15 split). We provide several descriptive statistics in Table 1. To assess the quality of our generations, we compare the generated dataset SKILLSKAPE with two other datasets from the literature: (1) a manually annotated benchmark, created by Decorte et al. (2022), based on the SKILLSPAN-M(ATCH) dataset (Zhang et al., 2022a), which contains over 14.5K job posting sentences scraped from various sources, and (2) the DECORTE dataset (Decorte et al., 2023), synthetically generated from ESCO using GPT-4. By design, we created SKILLSKAPE to cover the same label space as SKILLSPAN-M, which has only a development and a test set. In that dataset, two labels are used to indicate skills without an adapted label in the taxonomy: UNDERSPECIFIED and LABEL NOT PRESENT. We map these to the UNK label used in SKILLSKAPE. DECORTE associates ten synthetic sentences to each skill in the ESCO taxonomy. It is only used as training data. It covers all of ESCO (13.9K skills), but we restrict it to sentences with skills occurring in SKILLSPAN-M, leading to 5,120 samples (we add 10 random UNK sentences).

4 Experimental Setup

In this section, we introduce several benchmarks for skill matching tasks. We train a supervised multi-label classifier and present an LLM-based approach with in-context learning.

4.1 Supervised Multi-label Classifier

For the supervised baseline, we use a pre-trained BERT$_{base\_uncased}$ model (Devlin et al., 2019) to extract contextualized embeddings from the input text $t = \{w_1, w_2, ..., w_n\}$. These embeddings are then input into a multi-label classifier with a sigmoid activation applied independently to each output logit. Let $y = \{y_1, y_2, ..., y_k\}$ represent binary labels for the k classes. The model predicts the labels using:

$$\hat{y}_i = \sigma(f_i(\text{BERT}(\boldsymbol{t}))), \tag{3}$$

where $f_i(\cdot)$ is a function that maps the output embeddings from BERT to a logit for class i, $\sigma(\cdot)$ is the sigmoid activation, and $\hat{y}_i$ is the predicted probability for class i. The probability threshold can be tuned; we empirically found that 0.2 works well for this task.

We train the BERT$_{base}$ model for 100 epochs with a learning rate of 3×10^{-5} and select the best-performing epoch. We use a batch size of 16 and a maximum sequence length of 128. The model is trained for five different seeds.

4.2 In-context Learning with LLMs

We leverage an LLM to match skills in synthetic job posting sentences to the ESCO taxonomy. This pipeline has three steps, visualized in Figure 1: 1) **skill extraction** from the sentence, 2) **candidate selection** from the taxonomy, and 3) **skill matching** to the list of candidates. Here, we first extract relevant skills using LLM-prompting, pre-select viable candidates from our taxonomy, and then match the skills to candidates in the taxonomy through LLM-prompting again. We adopt a three-step approach to overcome the limited context window of LLMs, specifically 4K for `GPT-3.5-turbo` (OpenAI, 2023), which makes feeding large taxonomies directly to the model impractical.

(1) Skill Extraction. For each job posting, the LLM identifies key skills and tasks within the job ad while omitting irrelevant information. The LLM is directed to respond by repeating the sentence and tagging the skills by surrounding them with `@@` and `##`, following Wang et al. (2023), as shown in our prompt below.

> **System**: You are an expert human resource manager. You need to analyse skills in a job posting.
>
> **Instruction**: You are an expert human resource manager. You are given an extract from a job description. Highlight all the skills, competencies and tasks that are required from the candidate applying for the job, by surrounding them with tags @@ and ##. Make sure you don't highlight job titles, nor elements related to the company and not to the job itself. Make sure to rewrite the sentence with all the tags.
>
> {Demonstrations}
>
> Sentence: {Sentence}
>
> Answer:

We provide seven demonstrations in a few-shot setting to assist the model in understanding the task and following the instructions. To select few-shot examples, we use kNN retrieval from a training set composed of sentences along with their spans and labels (Liu et al., 2022). The closest samples from our dataset are selected as few-shot examples (see Appendix A). Finally, We process the output by extracting the tagged sections as skills.

(2) Candidate Selection. Matching extracted skills with skills defined in the taxonomy is crucial. Each skill in the taxonomy is associated with a tiered structure of names, each providing different levels of detail, and a definition. To provide richer context to the model, we concatenate the most detailed (or granular) name of the skill with its definition. We use two methods for pre-selecting viable candidates from the taxonomy for each skill:

- **`rule-based`**: Through string matching, we seek full or approximate matches of the extracted skill within the taxonomy. If the exact string of the extracted skill is present in the name or definition of a skill in the taxonomy, it is considered a good candidate for a match. We randomly select five entries if more than five candidates are found. If the exact strings do not match, we calculate the `token_set_ratio` using TheFuzz,[4] a similarity score based on Levenshtein's distance (Levenshtein et al., 1966). The top five candidates with the highest scores are chosen.
- **`embedding-based`**: Using a pre-trained language model (JobBERT; Zhang et al., 2022a), we compare the extracted skills with taxonomy entries. We obtain the contextualized embeddings of the extracted skill by embedding the sentences and averaging the vector representation of the tokens of the extracted skill. These embeddings are then compared to the representation of each skill in the ESCO taxonomy if the extracted skill is a substring of the sentences it was extracted from. Otherwise, the embedding of the extracted skill itself is compared to the skills in the taxonomy. The top five most similar candidates are selected based on cosine similarity.

While effective, the rule-based method may miss synonyms and context. On the other hand, the embedding-based method addresses the limitations of the rule-based method but risks selecting contextually similar yet factually dissimilar candidates (e.g., software vs. hardware). Therefore, we adopt a hybrid approach, retaining candidates from both rule-based and embedding-based methods.

(3) Skill Matching. The final step involves matching extracted skills to one of the selected candidate skills. We present the LLM with formatted candidates as options, and request the best

[4] `https://github.com/seatgeek/thefuzz`

	Perplexity (↓)	S2SIM (↑)	Explicitness (%, ↓)
SkillSpan-M	178.2	0.662	5.0
Decorte	65.1	0.739	22.4
SkillSkape	44.3	0.744	6.9

Table 2: **Offline Metrics.** We show the offline metrics as described in Subsection 4.3. (↑) indicates higher the better, (↓) indicates lower is better.

match, resembling a ranking task. The model outputs the most fitting option as a matched skill or provides no match if none are found. To assist the model without overloading the prompt, we provide a one-shot example in the following format:

```
Sentence: {generated sentence}
Skill: {extracted span}
A: {candidate 1}
...
J: {candidate 10}
Answer: {selected candidate}
```

The full prompt with one-shot demontration can be found in Appendix A.2.

To conduct the experiments for in-context learning with LLM, we retrieve demonstrations from the training set to provide examples for both the extraction and matching steps. We conduct an ablation study on SkillSpan's validation set to select the best number of shots for both tasks. Experiments are described in Appendix B, in Table 4, and Figure 2. The matching step is performed with 10 candidates using the mixed setting (5 embedding candidates and 5 string matching candidates). The best setting uses 7 demonstrations for the extraction step and one demonstration for the matching step. Matching step demonstrations have a large number of tokens due to the list of candidates along with their definitions, which can explain the decreased performance associated with adding more demonstrations.

4.3 Offline Quality Metrics

We design a set of metrics to evaluate the quality and diversity of the data at hand. Our intention is not to mirror metrics of SkillSpan-M, which is untidy by nature of scraped data, but to produce high-quality training data for downstream skill matching tasks.

1. First, we consider **Perplexity**, i.e., how realistic the data is from the point of view of a language model. We compute the perplexity of each of the sentences using GPT-2 (Radford et al., 2019), where lower is better.
2. Second, we consider **Skill-Sentence Similarity (S2SIM)**, the average cosine similarity between a skill and the associated sentence. The higher this metric, the closer the generated sentence will be semantically close to the associated skills. We aim to maximize this metric. The embeddings are computed using JobBERT, and BERT model fine-tuned on English job postings with the masked language modeling objective.
3. Finally, we measure **Explicitness** by counting the number of entities that appear exactly in the sample, using string matching.

Table 2 shows offline metrics for SkillSpan-M, Decorte, and SkillSkape. SkillSkape has a lower perplexity, and outperforms SkillSpan and Decorte in terms of S2SIM. The main reason for SkillSpan-M's low skill-sentence similarity is its noisiness, leading to sentences often being cut mid-way and lacking coherence. Around 7% of SkillSkape skills are fully explicit (the label can be found exactly in the sentence), much closer to SkillSpan-M than Decorte. A higher explicitness leads to an easier task; a skill matching model needs to be trained on enough implicit examples to allow it to generalize to implicit skills.

Overall, SkillSkape demonstrates similar statistics in perplexity and S2SIM characteristics as Decorte. However, it notably exhibits a significant (3×) enhancement in implicitly representing skills within each sentence.

5 Results and Analysis

To assess the label refinement method (Section 3.4), we apply it to the development set of the SkillSpan-M benchmark that has annotated skills and associated spans. 40% of our extracted spans match exactly with the annotated span. 60% of our extracted spans are either a perfect match or contain the annotated span. In general, extracted spans have a Jaccard similarity of 62% with the annotated spans.

5.1 Supervised vs. Few-shot ICL Matching

In Table 3, we show the results of the skill matching task on the SkillSpan-M test set and SkillSkape test set. We compare the performance of supervised and in-context learning methods *trained*

	Supervised		Few-Shot ICL	
↓Train / Test→	SKILLSKAPE	SKILLSPAN-M	SKILLSKAPE	SKILLSPAN-M
DECORTE	28.0 ± 0.8	23.0 ± 0.7	36.8 ± 0.2	26.9 ± 0.5
SKILLSKAPE	**68.0 ± 0.5**	22.2 ± 0.9	**37.6 ± 0.2**	26.9 ± 0.3
Both	67.2 ± 1.0	**26.1 ± 1.2**	**37.6 ± 0.2**	**27.3 ± 0.4**

Table 3: **Supervised and Few-Shot ICL Results**. *Both* indicates the concatenation of DECORTE and SKILLSKAPE. The scores are micro-F_1.

on the DECORTE training set, SKILLSKAPE training set, or the concatenation of both. The supervised approach uses training data to train a supervised multi-label classifier, whereas the few-shot ICL approach uses it as a demonstration pool to retrieve kNN demonstrations. For simplicity, we refer to both training and few-shot learning with demonstrations as *training* in the remainder of this section.

Comparing across supervised and few-shot in-context learning settings, we observe that both supervised and ICL approaches achieve higher performance on both real-world data (SKILLSPAN-M) and our synthetic dataset (SKILLSKAPE) when trained on SKILLSKAPE training set or a combination of SKILLSKAPE and DECORTE training sets. This increase in matching performance is likely due to a higher textual diversity in the SKILLSKAPE dataset. Across both matching approaches, we also observe that training on both SKILLSKAPE and DECORTE consistently achieves the highest test F_1 scores on real-world data. However, the difference in performance is greater for the supervised approach than ICL, highlighting that the ability to generate high-quality data is most impactful for supervised approaches.

Additionally, we observe an interesting result in the large difference between the supervised and ICL performance on the SKILLSKAPE test set, 68.0/67.2 and 37.6 micro-F_1 respectively when trained on SKILLSKAPE or a combination of SKILLSKAPE and DECORTE. We suspect that this difference could largely be due the characteristics of our training and test data. Supervised models tend to perform well when the training and test data follow the same distributions. In contrast, the few-shot ICL method consistently outperforms than the supervised approach on the SKILLSPAN-M test set. Given the minimal tuning required for the ICL method, the ICL approach can be better suited to flexibly handle messy real-world data. These results suggest that, for use cases when we have a sample of annotated data from the same distribution as the data we want to predict, we can combine it with synthetic training data and leverage supervised models. Otherwise, the in-context learning approach is less dependent on the training data.

In Appendix Table 6, we show several qualitative examples of predictions of both the multi-label classifier and LLM. Several noticeable patterns are underprediction for the multi-label classifier and overprediction of the LLM. Additionally, we notice that the predictions of both models are rather close "semantically" to the gold labels, but are deemed incorrect by the evaluation.

In summary, the results underscore the significance of both the quantity and diversity of training data in the development of effective skill matching dataset generators.

5.2 Effect of In-context Demonstrations

We evaluate the sensitivity of our method to the number and candidate selection methodology of in-context learning examples.

Demonstrations. We perform an ablation study on the number of demonstrations for both skill extraction and matching. Results in Appendix Table 4 show that 7 shots for extraction with 1 shot for matching leads to the best performance.

Candidate Selection. Candidate selection using the hybrid method for $n = 5$ candidates from each of the rules- and embedding-based methods (i.e., 10 candidates in Figure 3) presents the best trade-off between performance and computational cost. While we do observe a higher F_1 score as we increase the number of candidates, the increase in performance appears to be marginal while it would more than double the number of input tokens.

Finally, an ablation study on the matching step of the pipeline (see AppendixB.3) shows that directly selecting the top-1 candidate (rule-based) as skill prediction lags behind the performance of using GPT-3.5 as a re-ranker by around 8% F_1.

5.3 Effect of Sentence Length

The sentence length distribution is heavily skewed toward shorter sentences in the DECORTE and SKILLSPAN-M test sets, with 50% of sentences being 13–19 words in DECORTE and 7–20 words in SKILLSPAN-M. In contrast, 50% of the sentences in SKILLSKAPE are between 23–33 words (see Figure 4 in Appendix for a visualization of the length distribution for each dataset).

When splitting the SKILLSPAN-M test set into two equal-sized sets depending on the size of the sequence (less than 12 words, or more than 12 words), training on DECORTE leads to slightly higher performance than SKILLSKAPE for shorter sentences (0.26 vs. 0.24 F_1). For longer sentences, however, SKILLSKAPE reaches an F_1-score of 0.18 while Decorte's F_1 is 0.17.

6 Conclusion

We introduce JOBSKAPE, a general framework for generating synthetic job posting sentences for skill matching. Using our framework, we release SKILLSKAPE a large dataset of synthetic job posting sentences labeled with ESCO skills. Our analysis shows that SKILLSKAPE contains more implicit skills, has longer sentences, and is overall closer to real-world data, compared to alternative synthetic dataset from the literature. Using our dataset, we conducted several skill matching experiments by training a supervised multi-label classifier and using in-context learning with an LLM, and showed that both methods achieved comparable results when evaluated on real-world data (F_1 of 26.1 and 27.3 respectively). Furthermore, we note that the potential applications of JOBSKAPE extend beyond its current scope. Its application in creating synthetic CVs, for instance, can enhance job matching algorithms and facilitate skill-gap analysis in various industries. The framework's adaptability to different skill taxonomies also opens up possibilities for use across multiple sectors. While promising, these extended applications require further exploration to fully assess their impact.

7 Limitations

Closed model. One of the primary limitations comes from our use of Large Language Models (LLMs) that are closed. This restricts our ability to understand, modify, or customize the underlying mechanisms of these models. The closed nature of the LLMs used in our study also limits the transparency, adaptability, and reproducibility of our system.

English only. Our method is limited to processing and understanding English language content. This language-specific focus narrows the scope of our system's applicability, excluding non-English speaking demographics.

Bias inherited from LLMs. Another significant limitation is the potential bias inherited from the LLMs. Since these models are trained on large datasets that may contain biases, there is a risk that our system may inadvertently perpetuate these biases in its generations. This could manifest in various forms, such as gender, cultural, or industry-specific biases, and could affect the fairness and neutrality of the job postings generated. Furthermore, if biased postings are used extensively, they could adversely influence downstream tasks. For example, biased job postings could skew job recommendation algorithms, leading to unfair job suggestions that do not treat all individuals equally. This highlights the need for careful consideration and mitigation of biases in our approach to ensure equitable outcomes in all applications.

Subset of the Taxonomy. Due to limited resources, we restricted the generation of our synthetic dataset to $\sim$8K samples, with a fraction of the ESCO taxonomy that is also used in the SKILLSPAN-M dataset. Consequently, the multiclass classifier is also trained to classify with a limited set of skills. Scaling up to the full taxonomy might modify the behavior of the supervised classification model, while it should have little to no impact on the ICL skill-matching pipeline.

8 Ethics Statement

In this work, we strictly used publicly available data and generated synthetic datasets, avoiding the use of sensitive or private information. This approach aligns with ethical standards concerning data privacy and security.

However, our system can be used to extract information from personal documents, or be used for sensitive applications in the human resources domain, notably pre-selecting candidates to hire. It shall not be used without the supervision of a human. In this work, we focus on the development of a framework to reduce reliance on real-world annotated data. Extended to resumes, it could allow users to perform the skill extraction and matching task without requiring personal data to be anonymized. Given the limited performance of anonymization tools, generating data following similar distribution would greatly reduce privacy issues for such applications.

Acknowledgements

We thank Jibril Frej (EPFL) for fruitful discussions and feedback on the first version of the paper. MZ is supported by the Independent Research Fund Denmark (DFF) grant 9131-00019B and in parts by ERC Consolidator Grant DIALECT 101043235. We also gratefully acknowledge the support of the Swiss National Science Foundation (No. 215390), Innosuisse (PFFS-21-29), the EPFL Science Seed Fund, the EPFL Center for Imaging, Sony Group Corporation, and the Allen Institute for AI.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Benjamin Clavié and Guillaume Soulié. 2023. Large language models as batteries-included zero-shot ESCO skills matchers. In *Proceedings of the 3rd Workshop on Recommender Systems for Human Resources (RecSys in HR 2023) co-located with the 17th ACM Conference on Recommender Systems (RecSys 2023), Singapore, Singapore, 18th-22nd September 2023*, volume 3490 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Jens-Joris Decorte, Jeroen Van Hautte, Johannes Deleu, Chris Develder, and Thomas Demeester. 2022. Design of negative sampling strategies for distantly supervised skill extraction. *arXiv preprint arXiv:2209.05987*.

Jens-Joris Decorte, Severine Verlinden, Jeroen Van Hautte, Johannes Deleu, Chris Develder, and Thomas Demeester. 2023. Extreme multi-label skill extraction training using large language models. *CoRR*, abs/2307.10778.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiahui Gao, Renjie Pi, Lin Yong, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2023. Self-guided noise-free data generation for efficient zero-shot learning. In *International Conference on Learning Representations (ICLR 2023)*.

Ann-sophie Gnehm, Eva Bühlmann, Helen Buchs, and Simon Clematide. 2022. Fine-grained extraction and classification of skill requirements in German-speaking job ads. In *Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS)*, pages 14–24, Abu Dhabi, UAE. Association for Computational Linguistics.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2022. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Martin le Vrang, Agis Papantoniou, Erika Pauwels, Pieter Fannes, Dominique Vandensteen, and Johan De Smedt. 2014. Esco: Boosting job matching in europe with semantic interoperability. *Computer*, 47(10):57–64.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *CoRR*, abs/2303.17651.

Sovesh Mohapatra and Somesh Mohapatra. 2022. Sentiment is all you need to win US presidential elections. In *Proceedings of the 2nd International Workshop on Natural Language Processing for Digital Humanities*, pages 15–20, Taipei, Taiwan. Association for Computational Linguistics.

OpenAI. 2023. GPT-3.5 Documentation. `https://platform.openai.com/docs/models`.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Synthetic prompting: Generating chain-of-thought demonstrations for large language models. *arXiv preprint arXiv:2302.00618*.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021. Towards zero-label language learning. *arXiv preprint arXiv:2109.09193*.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mike Zhang, Kristian Jensen, Sif Sonniks, and Barbara Plank. 2022a. SkillSpan: Hard and soft skill extraction from English job postings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4962–4984, Seattle, United States. Association for Computational Linguistics.

Mike Zhang, Kristian Nørgaard Jensen, and Barbara Plank. 2022b. Kompetencer: Fine-grained skill classification in Danish job postings via distant supervision and transfer learning. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 436–447, Marseille, France. European Language Resources Association.

A Prompts

A.1 Extraction Demonstrations

In our in-context learning pipeline, we provide seven demonstrations to guide the LLM in performing extractions. Below is one example.

Sentence: we are looking for a team leader with strong communication skills to foster collaboration and information sharing within the team.
Answer: We are looking for a team leader with strong @@communication skills## to foster collaboration and information sharing within the team.

Sentence: the ability to work collaboratively across disciplines is a key criterion for this position.
Answer: @@ability to collaborate across disciplines## is a key criterion for this position.

Sentence: As a Java Senior Software Engineer with experience, you will be a member of a Scrum team.
Answer: As a Java Senior Software Engineer with experience, you will be a member of a Scrum team.

Sentence: In her role as a team leader, she has continuously supported the professional development of her employees.
Answer: In her role as a team leader, she has continuously fostered the professional @@development of her employees##.

Sentence: He is a resilient employee who has been able to set proper priorities and organize tasks thoughtfully during periods of heavy workload.
Answer: He is a resilient employee who has been able to set @@correct priorities and organize tasks thoughtfully## during periods of high workload.

Sentence: Highly qualified, flexible employees from the insurance and IT industry develop them further.
Answer: Highly qualified, flexible employees from the insurance and IT industries continue to develop them.

Sentence: Over the past few years, it has succeeded in continuously

developing itself in a rapidly changing environment.
Answer: Over the past few years, he has succeeded in @@continuously developing## himself in a rapidly changing environment##.

A.2 Matching

A.2.1 Prompt

Here we provide the prompt used to match each extracted skill to one of the pre-selected candidates. The one-shot demonstration used in this prompt is provided in section A.2.2.

System: You are an expert human resource manager. You need to analyse skills in a job posting.

Instruction: You are given a sentence from a job description, and a skill extracted from this sentence. Choose from the list of options the one that best match the skill in the context. Answer with the associated letter.

{Demonstration}

Sentence: {Sentence}
Skills: {Extracted}
A: {Candidate 1}
...
J: {Candidate 10}

Answer:

A.2.2 Demonstration

The demonstration we use in the matching step (Section A.2.1) of our in-context learning pipeline.

Sentence: Understand basic provisions of copyright and privacy.
Skill: Data protection.
Options:
A: "Respect privacy principles"
B: "Understand data protection"
C: "Ensure data protection in aviation operations"
D: "Data protection"
Answer: b, d.

A.3 Generation of dataset

A.3.1 Positive samples

We use this prompt to generate samples containing ESCO skills.

System: You are the leading AI Writer at a large, multinational HR agency. You are considered as the world's best expert at expressing required skills and knowledge in a variety of clear ways. You are particularly proficient with the ESCO Occupation and Skills framework. As you are widely lauded for your job posting writing ability, you will assist the user in all job-posting, job requirements and occupational skills related tasks.

Instruction: You work in collaboration with ESCO to gather rigid standards for job postings. Given a list of ESCO skills and knowledges, you're asked to produce a single example of exactly one sentence that could be found in a job ad and refer to all skill or knowledge component. Ensure that your sentence is well written and could be found in real job advertisement. Use a variety of styles. You're trying to provide a representative sample of the many, many ways real job postings would evoke skills. All the skills in : {skillList} must be integrated. A candidate should have different degrees of expertise in all the given skills. This degree should be specified for each skills in the sentence. You must not include any skills in ESCO that were not given to you. Try to be as implicit as possible when mentionning the skill. Try not to use the exact skill string {wordsToAvoid}. Avoid explicitly using the wording of this extra information in your examples. Your sentence must not start with

'We are seeking', 'We are looking' or 'We are searching'. Generate stricly only one example.

A.3.2 Negative samples

We use two different prompts to generate negative samples: This first prompt generates negative samples that describe the company.

> **System:** You are the leading AI Writer at a large, multinational HR agency. You are considered as the world's best expert at writing introductions of job posting.
>
> **Instruction:** You are the leading AI Writer at a large, multinational HR agency. You are considered as the world's best expert at writing introductions of job posting. You should write {nExamples} examples of the first line of the job posting. It should consists in introducing the company, its localization, the number of employees, and any information relevant to a future candidates who wants to learn about the company. The description should be concise, specify the potential growth of the company and a domain of action. You shouldn't mentoin anything about the actual job, no skills required for the candidate and shouldn't mention the candidate at all. You should mention a wide range of company field, size, and localization in each of the examples.

This second prompt generates sentences detailing the salary and perks of a job.

> **System:** You are the leading AI Writer at a large, multinational HR agency. You are considered as the world's best expert at specifying administrative information in job posting.
>
> **Instruction:** You are the leading AI Writer at a large, multinational HR agency. You are considered as the world's best expert at specifying administrative information in job posting. You should produce {nExamples} descriptions of the salary and the perks a candidate to a certain job would have. You shouldn't mention the actual job and the candidate itself. You could add diversity by varying the salary and the perks. You must write a salary range between 40k and 100k according to the job in half of your generation.

A.4 Refinement of dataset

A.4.1 Initial prompt

> **System:** You are an expert human resource manager. You need to analyse skills in a job posting.
>
> **Instruction:** You are an expert human resource manager. You are given an extract from a job description and a skill coming from ESCO. Highlight all the parts of the job description that relates to the given skill, by surrounding them with tags '@@' at the beginning and '##' at the end. You should rewrite the entire sentence. The highlighted parts should precisely talk about the given skills and only this skills. The higlighted parts must precisely be about the given skills. Do not highlight parts not related to it. The sentence should be rewritten perfectly, using the same exact same words. You must highlight at least one part in the sentence that you will rewrite. The highlighted part should be as short as possible.

A.4.2 Refining shots

In case of incorrectly bound annotations :

> In your response, you highlighted some parts using @@ at the beginning and @@ at the end.

```
Please use @@ at the beginning
of the parts and ## at the
end of the part you want to
highlight. Annotate the previous
sentence, but with the correct
highlighting.
```

When there is a lack of annotations :

```
In your response, you highlighted
nothing.  Please annotate the
previous sentence, and highlight
at least one part linked to the
skill.
```

B Ablation studies - Few-Shot ICL

B.1 Demonstrations

To conduct the experiments for the In-context Learning with LLM, we will use the demonstrations retrieval from the training set to provide few shots for both the extraction and the matching. We need to determine the number of demonstration to use for both parts. For this purpose we conduct an ablation study on SKILLSPAN-M 's validation test trying different configuration of number of shots. We try the following experiments :

- *baseline* : Same shots for all the sentences A.1 A.2.2
- M_1 : 1 demonstration for the matching part, baseline shot for extraction
- E_5 : 5 demonstration for the extraction part, baseline shot for matching
- E_7 : 7 demonstration for the extraction part, baseline shot for matching
- E_{10} : 10 demonstration for extraction, baseline shot for matching
- E_7M_1 : 1 demonstration for the matching part and 7 for the extraction part
- E_7M_3 : 3 demonstrations for the matching part and 7 for the extraction part

Given the stats in Table 4, displayed on Figure 2 we see the adding the demonstration retrieval for the extraction part yields a significative improvement on the recall. We will run the subsequent experiments with 7 demonstrations for the extraction part and one demonstration for the matching part.

	Recall	Precision	F_1
baseline	0.260	0.303	0.280
E_5	0.279	0.296	0.287
E_7	0.282	0.301	0.291
E_{10}	0.282	0.298	0.289
M_1	0.267	**0.305**	0.284
E_7M_1	**0.289**	0.298	**0.2934**
$E_{10}M_3$	0.283	0.293	0.288

Table 4: Ablation study for In-context Learning: Selecting optimal number of demonstrations for extraction and matching with GPT-3.5

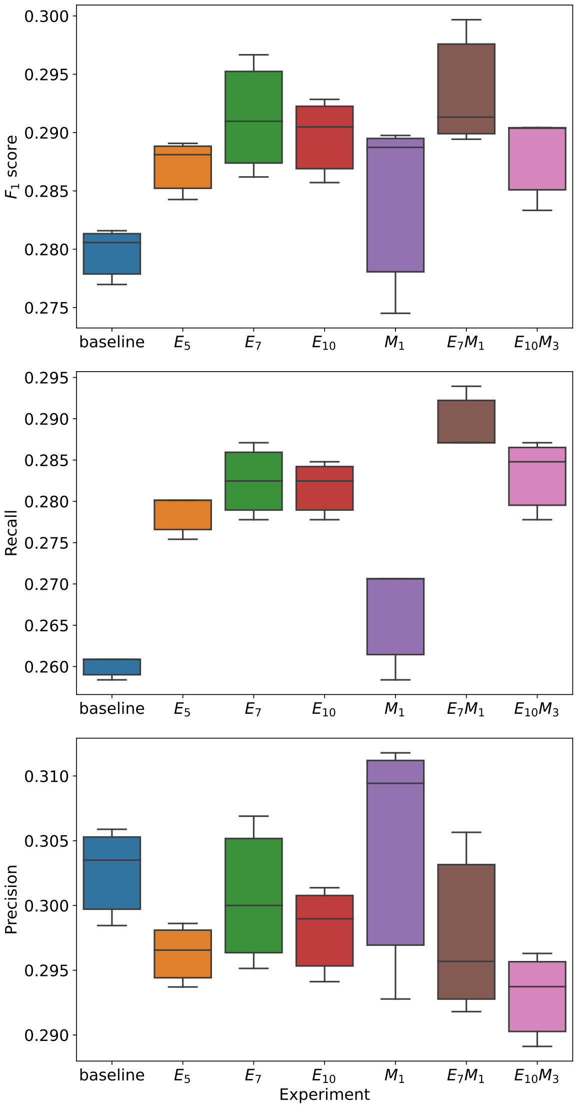

Figure 2: Ablation study for In-context Learning

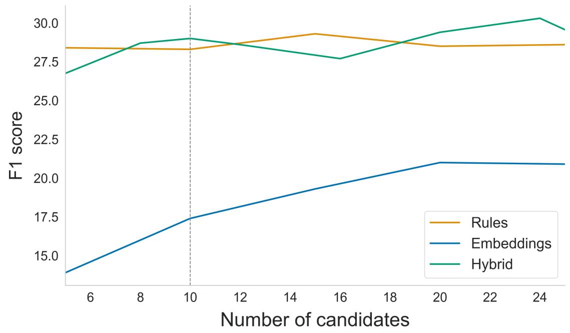

Figure 3: Rule-based, embedding-based, and hybrid candidate selection methods to select n candidates. Note, since the hybrid method takes the union of rule-based and embedding-based methods, $n = 5$ using the hybrid method would approximate $n \times 2$ actual number of actual candidates selected

# of Candidates	Precision	Recall	F_1
1	24.2	16.9	19.9
2	17.1	23.5	19.8

Table 5: Ablation study of the matching step: Performance of the ICL pipeline when taking only the top 1 or 2 candidates using the rule-based selection methods.

B.2 Candidate Selection

Figure 3 shows shows the F_1 scores of the ICL when we vary the number of candidates selected using the rule-based, embedding-based, and hybrid candidate selection methods, holding other elements constant. Looking at our results, we elect to use 10 candidates ($n = 5$) with the hybrid method.

While further increasing the candidates can increase matching performance slightly, we find that providing too many candidates can lead to a noticeable increase in inference time.

B.3 Matching Step

We conduct an ablation study on the matching. We remove the matching step from the pipeline and we only extract the spans from the inputted sentences and use rule-based to find matches. We focus on the rule-based method that yields the best results when extracting a small amount of candidates. Table 5 shows that the top-selected candidates are behind the performance of using GPT-3.5 as a reranker by around 8%. Therefore, we continue our experiments using the full three-step pipeline.

C Qualitative Analysis

We present various qualitative examples of predictions from the test set of SKILLSKAPE in Table 6. These examples include outcomes from both the supervised multi-label classifier and the in-context learning results using GPT-4. A key observation is the relatively lower number of skills predicted by the supervised classifier, which operates with a threshold of 0.15. Generally, these predictions are feasible and align closely with the gold standard label. However, it should be noted that the evaluation process tends to penalize these predictions for their limited scope.

D Other Summary Statistics on SKILLSKAPE

D.0.1 Skill Groups

We show in Table 7 the skill groups and counts of skills in each ESCO skill group that is included in the label spaced used for the SKILLSKAPE dataset.

D.0.2 Sentence Length

Looking at Figure 4, we can see that SKILLSKAPE has longer sentences and contains more variation in sentence length than DECORTE. The distribution of SKILLSKAPE resembles more that of real-world data (SKILLSPAN-M).

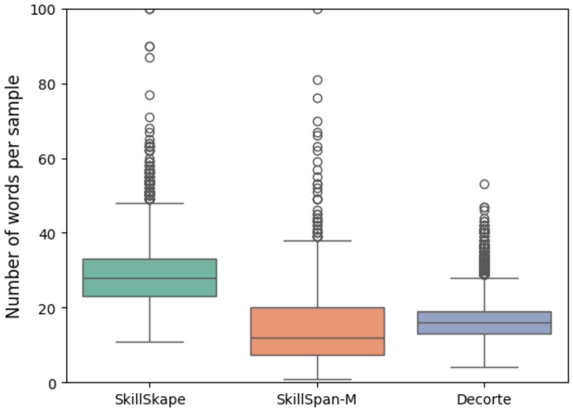

Figure 4: Sentence length distribution in the three datasets. SKILLSKAPE has much longer sentences. DECORTE has very short sentences and low length variance.

Sentence	Multi-label Classifier	In-context Learning	Gold
(1) Seeking a highly skilled individual with extensive expertise in overseeing and optimizing the operation and maintenance of various technical components and systems on board maritime vessels.	shipping industry	overseeing and optimizing the operation and maintenance of various technical components and systems on board maritime vessels	manage vessel engines and systems
(2) As an integral part of our team, the ideal candidate should possess a deep understanding of coordinating the alignment and seamless interaction of various system components, while executing rigorous testing and implementing an overarching strategy for the integration of ICT systems	ICT system integration, define integration strategy	coordinating the alignment and seamless interaction of various system components, rigorous testing, integration of ICT systems	ICT system integration, define integration strategy, define software architecture, manage ICT data architecture
(3) Ability to effectively adapt to changing circumstances while maintaining a vigilant attitude, maintaining composure in challenging situations, and efficiently managing workload and responsibilities.	handle stressful situations	effectively adapt to changing circumstances, vigilant attitude, composure, efficiently managing workload and responsibilities	exercise patience, adjust priorities, stay alert
(4) Are you an experienced professional with a proven track record in designing and implementing comprehensive technology testing frameworks, ensuring the seamless integration of software applications and systems?	develop ICT test suite, execute software tests	designing and implementing comprehensive technology testing frameworks, seamless integration of software applications and systems	develop ICT test suite

Table 6: We show several qualitative examples of predictions on the test set of SKILLSKAPE using the supervised multi-label classifier and in-context learning results with GPT-4.

Skill Group	skill count
agriculture, forestry, fisheries and veterinary	4
arts and humanities	8
assisting and caring	13
business, administration and law	40
communication, collaboration and creativity	111
constructing	3
education	3
engineering, manufacturing and construction	22
generic programmes and qualifications	6
handling and moving	15
health and welfare	7
information and communication technologies (icts)	71
information skills	57
management skills	65
natural sciences, mathematics and statistics	10
services	5
social sciences, journalism and information	1
working with computers	35
working with machinery and specialised equipment	14
TOTAL	514

Table 7: ESCO skill groups present in SKILLSKAPE dataset