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

Antoine Magron[*▲] **Anna Dai**[*▲] **Mike Zhang**[⊘]
Syrielle Montariol[▲] **Antoine Bosselut**[▲]
[▲]EPFL, Switzerland
[⊘]IT University of Copenhagen, Denmark
{first.lastname}@epfl.ch mikz@itu.dk

Abstract

Recent approaches in skill-to-surface-form matching, employing synthetic training data for classification or similarity model training, have shown promising results, eliminating the need for time-consuming and expensive annotation. However, previous datasets have limitations, such as featuring only one skill per sentence and generally comprising short sentences. This paper introduces JobSkape, a framework to generate synthetic data that resembles real-world job postings, 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 our dataset is more diverse, realistic, and follows a higher quality based on similarities. Additionally, we present a multi-step pipeline utilizing large language models (LLMs), benchmarking against supervised methodologies. We outline that the performances are comparable and that each method can be used for different use cases.

1 Introduction

In the dynamic modern labor market, understanding job advertisement demands at scale is crucial for informed decision-making by policymakers, businesses, and stakeholders. The foundation of this understanding lies in *skill matching*: the extraction and alignment of skills from job descriptions to their disambiguated surface 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 require regularly collecting and annotating up-to-date data (Zhang et al., 2022b), a process that is both expensive and time-consuming. By generating synthetic data, we can circumvent the need for costly annotations, while still retaining the benefits of supervised learning approaches. 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 persist, including incoherence in sentences from existing datasets and artificial setups, such as one skill per sentence. To address these shortcomings, we introduce JobSkape, a framework for realistic skill matching dataset generation that can be used for training and benchmarking.

JobSkape facilitates the creation of diverse, realistic labeled textual datasets that align closely with actual job posting sentences, ensuring cleaner and more coherent data. We showcase its practical application in the labor market by generating SkillSkape, a large-scale dataset linking coherent sets of skills to corresponding job descriptions. JobSkape utilizes generative large language models (LLMs) to curate meaningful skill combinations and generate appropriate job descriptions. A self-refinement step using LLMs (Madaan et al., 2023) ensures label quality in the refined SkillSkape dataset, assessed through offline metrics.

Moreover, we challenge traditional skill matching methods with an LLM-based, in-context learning (ICL) pipeline. We evaluate skill matching performance on our dataset, comparing our proposed pipeline with supervised matching models trained on our dataset in a controlled setting.

Contributions. In this work, we contribute the following: (1) First, we propose JobSkape, a framework for generating a synthetic dataset for skill matching with existing skill taxonomies. (2) Using the framework, we release a synthetic train and evaluation dataset (SkillSkape) for skill matching. (3) Our analysis shows that SkillSkape follows a higher diversity and quality compared to previous synthetic datasets. (4) Lastly, we introduce an LLM-based approach and evaluate against several supervised baselines and for skill matching performance with the ESCO taxonomy.

2 Related Work

2.1 General 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.

2.2 Synthetic Data in the Job Market Domain

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).

2.3 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

Here, we describe the framework that we develop to generate synthetic job posting sentences with annotated skills. To assess the quality of our generations, we compare the generated dataset SkillSkape with two other datasets from the literature. First, a manually annotated benchmark, created by Decorte et al. (2022), based on the SkillSpanM(atch) dataset (Zhang et al., 2022a), which contains over 14.5K job posting sentences scraped from various sources. These sentences are annotated with spans corresponding to specific skills, and these spans have subsequently been manually linked to ESCO. Second, the Decorte dataset (Decorte et al., 2023), synthetically generated from ESCO using GPT-4.

In this work, 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.[1] 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

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

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.[2] For the synthetic sentence generation task at hand, we do not distinguish between skill and knowledge components.

Our synthetic dataset creation framework generates sentences related to multiple skills listed in the ESCO taxonomy. To reduce the data generation cost, we use a subset of ESCO, focusing on the list of 511 skills used in SKILLSPAN-M.

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 skills. We initiate the process by creating combinations of skills, guided by three main conditions: (1) varying skill combination lengths for increased diversity, (2) ensuring semantic closeness of skills within combinations to reflect realistic job posting sentences, and (3) representing all skills a minimal number of times in the 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 a pre-trained language model fine-tuned on domain-specific data. 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 sample n from $\mathcal{N}$, and the combination size is $min(n, |\mathcal{S}_i|)$. However, directly sampling from $\mathcal{S}_i$ is non-trivial, as demonstrated by the combination of **SQL** and **THC Hydra**, each in the other's k-nearest neighbor but with different frequencies. We introduce distribution $\mathcal{F}$ to compute the probability of selection over $\mathcal{S}_i$ using softmax:

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

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

A temperature t_c controls the probability distribution, influencing diversity. Larger t_c values result in more diverse pairings, while approaching 0 selects only popular skills. 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 $\mathcal{N} \sim U(1, 10)$. We employ the pre-trained language model JobBERT (Decorte et al., 2021) to obtain meaningful embeddings for job descriptions and skills. The distribution $\mathcal{F}$ is computed as the average of standardized 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, skewing the combination length distribution towards 1, using $k = 20$ and $t_c = 1$.

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`".

We add another instruction to the prompt to reduce ambiguity. We add to each prompt, a list of synonyms of each inputted skill, that are in the taxonomy. For instance, prompt the model that referring to **SQL** by **MySQL** is forbidden since **MySQL** is a skill by itself. The skills are 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 exclusively comprises positive samples, meaning that all generated sentences have 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. Let Ω be the universe of skills and $\mathcal{S} \subset \Omega$ the selected skill subset. For generating negative samples with unknown skills, the same framework is applied with $S^c = \Omega \backslash \mathcal{S}$ as the input taxonomy. For the SKILLSKAPE dataset, 500 such negative samples are generated. An additional 500 negative samples for SKILLSKAPE are generated, representing sentences found in real job descriptions with no skills. Examples include: (1) A sentence describing the company: its reach, domain, location, et cetera (see Appendix A.3.2). (2) A sentence detailing the salary and perks of a job (see Appendix A.3.2). 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 spans and skills using the pipeline described in Appendix A.4. The retrieved set of skills, along with their associated spans, is filtered to include only trustworthy pairs. This is determined by a cosine similarity above a specified threshold. For this refined dataset version, we use JobBERT as an encoder, and the threshold is empirically set to $T = 0.7$.

Span Extraction. We use GPT-3.5 to label skill sequences in the sentences. Each mention, whether implicit or explicit, is surrounded by `@@` and ##. In case the language model fails to label the span, it is asked to self-correct, as outlined in Appendix A.4. To minimize errors during sentence rewriting, we use a temperature of 0.45.

We showcase two examples extracted from the train set of the refined SKILLSKAPE dataset.

Positive example: (28 words, which is the average length in the dataset)

```
Sentence with annotated spans:
{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.

The following is a negative example, which mentions information about the hiring company instead of the job itself.

```
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}
```

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 8 940 samples, split into training, development and test sets ($\sim$ 70-15-15 split). We provide several descriptive statistics in Table 1, along with information on the two other datasets used in this paper for evaluation: SKILLSPAN-M and DECORTE. 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 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

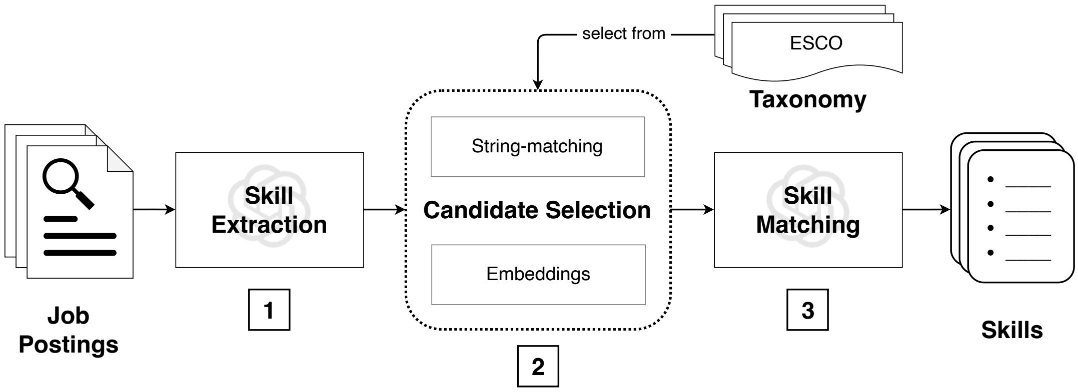

Figure 1: **Three-step Skill Extraction and Matching Pipeline**. We show our in-context learning pipeline for end-to-end skill matching. We use an LLM to extract skills from job ads, then do candidate selection using heuristics, and last, do skill matching with a constrained taxonomy.

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 baselines for skill matching. We present a supervised multi-label classifier and an LLM-based approach with in-context learning.

4.1 Supervised Multi-label Classifier

For the supervised baseline, we use a pre-trained $\text{BERT}_{\text{base}}$ (uncased; Devlin et al., 2019) model 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:

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

where $f_i(\cdot)$ is the 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.[3]

[3] Seed numbers are 276800, 381552, 497646, 624189, and 884832.

4.2 In-context Learning with LLMs

We use 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). Feeding the large taxonomies directly to the model is impractical.

(1) Skill Extraction. For each job posting, we prompt the LLM to identify skills and essential tasks within the job ad while excluding 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). We process this output to extract the tagged sections as skills. We provide seven demonstrations in a few-shot setting to assist the model in understanding the task and following the instructions.

To construct the few shots, we use a training set composed of sentences along with their spans and labels using demonstration retrieval (Liu et al., 2022) with kNN. The closest samples from our dataset are selected to construct the shots:

```
Sentence: {generated sentence}

Answer:{feedback annotated sentence}
```

(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 and a definition. To provide richer context to the model, we concatenate the most granular name and 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 (e.g., A. B. C.) and request the best 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 overwhelming the prompt, we provide a one-shot example with the following format:

```
Sentence: {generated sentence}
Skills: {feedback span}
A: {candidate 1}
...
J: {candidate 10}
Answer: {associated skill}
```

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.

	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.

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.

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

↓Train / Test→	Supervised		Few-Shot ICL	
	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 results are in micro-F_1.

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 in the test set.

Overall, SKILLSKAPE demonstrates similar statistics in perplexity and S2SIM characteristics as DECORTE. However, it notably exhibits a significant ($3\times$) enhancement in explicitly representing skills within each sentence.

5 Results and Analysis

We wish to assess the refinement methods of the labels. To do so we applied it to the development set of the SKILLSPAN-M benchmark that has annotated skill and associated span. 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, the 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* on the DECORTE training set, SKILLSKAPE training set, or both 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.

We highlight some trends from the results. First, comparing across Supervised and few-shot in-context learning approaches, we observe that the yields stronger performance for both approaches on both SKILLSPAN-M and SKILLSKAPE test sets. This is likely due to a higher diversity in the data. Second, across both setups, we observed a consistent pattern in terms of skill matching on SKILLSPAN-M when trained on different datasets. Third, we observe that in-context learning approach consistently achieves higher performance than the supervised approach on the on the SKILLSPAN-M test set. This indicates that that the ICL approach can be more flexible to handle messy real world data.

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-F1 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. We also notice that the LLM tend not to predict any `UNK`, which might contribute to this performance gap. The ICL approach does achieve higher performance on the

SkillSkape test when we train on Decorte. This suggests 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 with generated training data and leverage supervised models. Else, the in-context learning approach is less dependent on the training data.

In Table 6 (Appendix), 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 Ablation Study: Few-Shot ICL

We perform ablation studies to determine the best setting for our in-context learning skill extraction and matching pipeline. We experiment with the number of demonstrations and candidate selection methodology on the SkillSpan-M dev. dataset to drive our parameter choice for the final experiments.

Demonstrations. We perform an ablation study on the number of demonstrations for both skill extraction and matching. Results in Table 4 (Appendix) 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 rules- and embedding-based (i.e., 10 candidates in Figure 3) methods presents the best trade-off between performance and computational costs. 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 our input tokens.

Additionally, 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% (F1-score).

5.3 Focus on the Effect of Sentence Length

Sentence length distribution is heavily skewed towards shorter sentences in Decorte and SkillSpan-M test set, with 50% of sentences in Decorte being between 13–19 words and 7–20 words for SkillSpan-M. In contrast, 50% of the sentences in SkillSkape are between 23–33 words, hence fully disjoint from the other two (see Figure 4 in Appendix).

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 higher performance than SkillSkape for shorter sentences (0.26 vs. 0.24 F_1). For longer sentences, on the contrary, SkillSkape performs better with an F_1 of 0.18 compared to Decorte's F_1 of 0.17.

6 Conclusion

We introduce JobSkape, a general framework for generating synthetic job posting sentences for skill matching. Given this framework, we also release SkillSkape a large-scale synthetic job posting sentence train and evaluation set. Our analysis shows that SkillSkape follows a strong similar trend with real job posting data. In addition, for skill matching, we conducted several experiments with a supervised multi-label classifier and in-context learning with an LLM. We showed that both methods are comparable and each can be used for suitable applications.

Furthermore, the potential applications of JobSkape extend beyond its current scope. It could serve as a foundation for generating synthetic CVs, offering a broader spectrum of applications in the field of employment and skills assessment. The adaptability of our framework is highlighted by its compatibility with diverse taxonomies, making it a valuable tool across various domains.

7 Limitations

Closed source model. One of the primary limitations comes from our use of Large Language Models (LLMs) that are closed source. This restricts our ability to understand, modify, or customize the underlying mechanisms of these models. The closed-source 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.

Subset of the Taxonomy. Due to limited resources, we restricted the generation of our synthetic dataset to ∼8K samples, with a fraction of the ESCO taxonomy that is also used in the SKILLSPAN-M dataset. Hence the multi-class 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.

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, Thomas Demeester, and Chris Develder. 2021. Jobbert: Understanding job titles through skills. *CoRR*, abs/2109.09605.

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 prompt

A.1.1 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. 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.

sentence : {sentence}

A.1.2 Shots

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

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.

Sentence: {sentence}
Skills: {span}
A: {candidate 1}
...
J: {candidate 10}

A.2.2 Shots

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

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

Samples with no linked labels :

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.

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

Incorrect bound annotation :

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.

Lack of annotation :

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.1.2 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

	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

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.

B.2

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

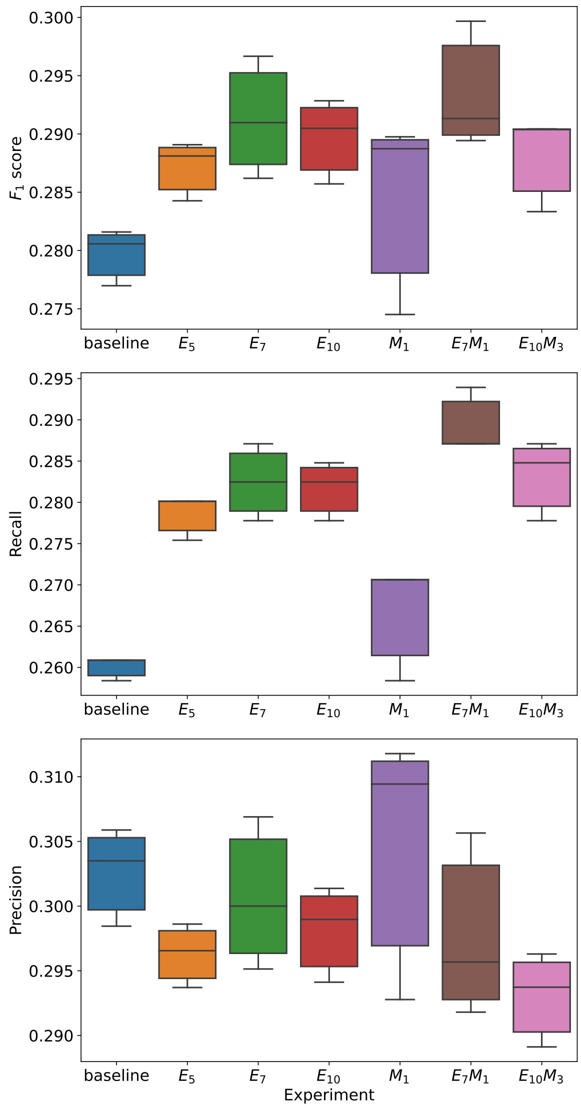

Figure 2: Ablation study for In-context Learning

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 re-ranker by around 8%.

C Qualitative Analysis

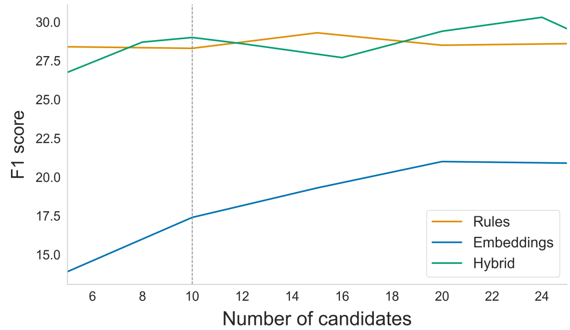

Figure 3: Rule-based, embedding-based, and hybrid candidate selection methods to select n candidates. Note, since the hybrid method takes the union of rule-based and embedding-based methods, $n = 5$ using the hybrid method would approximate $n \times 2$ actual number of actual candidates selected

# 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.

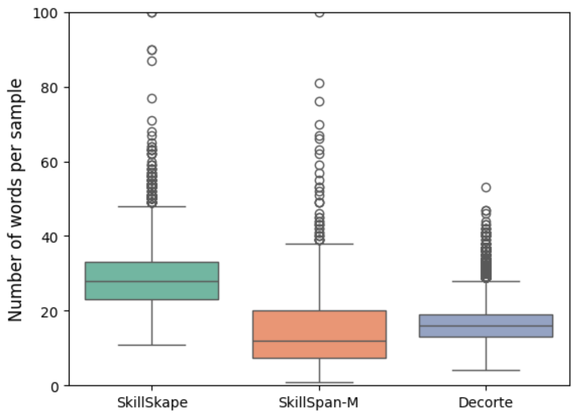

Figure 4: Sentence length distribution in the three datasets. SKILLSKAPE has much longer sentences. DECORTE has very short sentences and low length variance.

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. What is noticeable is the lower number of skills the supervised classifier predicts (threshold of 0.15). Usually, the predictions are viable and are close to the gold label, but evaluation penalizes them.