IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 25 May 2025, accepted 3 July 2025, date of publication 14 July 2025, date of current version 1 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3589147

RESEARCH ARTICLE

Efficient Text Encoders for Labor Market Analysis

JENS-JORIS DECORTE[1,2], JEROEN VAN HAUTTE[1], CHRIS DEVELDER[2], (Senior Member, IEEE), AND THOMAS DEMEESTER[2]

[1]TechWolf, 9000 Ghent, Belgium
[2]Internet and Data Science Laboratory (IDLab), Ghent University–imec, 9000 Ghent, Belgium

Corresponding author: Jens-Joris Decorte (jensjoris@techwolf.ai)

This work was supported in part by the Flemish Government, through Flanders Innovation and Entrepreneurship (VLAIO) under Project HBC.2020.2893; in part by the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" Program; and in part by TechWolf.

ABSTRACT Labor market analysis relies on extracting insights from job advertisements, which provide valuable yet unstructured information on job titles and corresponding skill requirements. While state-of-the-art methods for skill extraction achieve strong performance, they depend on large language models (LLMs), which are computationally expensive and slow. In this paper, we propose ConTeXT-match, a novel contrastive learning approach with token-level attention that is well-suited for the extreme multi-label classification task of skill classification. ConTeXT-match significantly improves skill extraction efficiency and performance, achieving state-of-the-art results with a lightweight bi-encoder model. To support robust evaluation, we introduce Skill-XL a new benchmark with exhaustive, sentence-level skill annotations that explicitly address the redundancy in the large label space. Finally, we present JobBERT V2, an improved job title normalization model that leverages extracted skills to produce high-quality job title representations. Experiments demonstrate that our models are efficient, accurate, and scalable, making them ideal for large-scale, real-time labor market analysis.

INDEX TERMS Labor market analysis, text encoders, skill extraction, job title normalization.

I. INTRODUCTION

Labor market analysis plays a central role in addressing global workforce challenges such as talent shortages, skill gaps, and fast-changing job requirements driven by technological advancement. Accurate insights into the skills demanded by employers inform a wide range of applications such as workforce planning and policymaking [1], [2]. In this context, job advertisements (job ads) have served as a valuable resource for understanding labor market trends [3]. Job ads are published on a daily basis across industries and regions and contain fine-grained information about job titles and their respective skill requirements. Although rich in information, job ads often use different terminology to refer to occupations and skills. Therefore, robust labor market analysis requires natural language processing (NLP) techniques to identify and normalize the information contained in job ads.

To address this terminology challenge, the European Commission developed ESCO (European Skills, Competences, Qualifications and Occupations),[1] a multilingual classification system that serves as a common language for the labor market. ESCO provides a comprehensive taxonomy of close to 14,000 skills and over 3,000 occupations, enabling consistent analysis and comparison of job market data across time and different regions. This standardized framework is particularly valuable for large-scale labor market analysis, as it allows for the systematic mapping of diverse job descriptions to a common reference point.

To leverage ESCO's standardized taxonomy effectively, two key tasks are essential: job title normalization and skill requirement extraction from job ads — skill extraction in short. The latter involves identifying the skills mentioned in a job advertisement and mapping them to their corresponding ESCO skill definitions, whereas job title normalization addresses the challenge of mapping diverse job titles to

The associate editor coordinating the review of this manuscript and approving it for publication was Binit Lukose.

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

standardized ESCO occupations. Together, these tasks enable robust labor market analysis by transforming unstructured job descriptions into structured, comparable data points.

Research interest in skill extraction has grown steadily over the last decade [3]. Recently, increased parsing capabilities and semantic understanding of large language models (LLMs) have been shown to achieve state-of-the-art results for skill extraction [4], [5]. Job title normalization has also been approached using LLMs, although with less convincing results [6]. However, insightful labor market analysis necessitates a large volume of job ads to be analyzed, making the need for efficient NLP models greater. Lightweight models that achieve high performance without relying on vast computational resources can democratize access to labor market insights, enabling organizations and researchers to process large datasets cost-effectively and at scale.

Benchmarks for skill extraction have traditionally been formalized as span labeling tasks, without linking the identified spans to respective skills in a taxonomy. This lack of normalization in skill extraction prevents robust analysis of aggregate job skill requirements because of e.g. synonyms. A few benchmarks do provide fine-grained skill labels, yet they are either aggregate labels on the full ad level (thus preventing sentence-level evaluation), or they are the result of post-annotation on sequence annotations, limiting the expressiveness of annotating implicit skills. Finally, while large fine-grained skill taxonomies are great for normalizing skill information and robust analysis, their large size often means that a degree of semantic overlap is present between some skill labels, making it more difficult to annotate the ground truth and evaluate the models in a robust manner.

In this work, we address these challenges through the following contributions:

1) We introduce **ConTeXT-match**, a new contrastive learning approach with token-level attention, designed for extreme multi-label text classification. We apply this to the skill extraction task by training a bi-encoder for sentence-level skill extraction. The model outperforms all other skill extraction models with the same number of parameters, owing to a new token-level contrastive loss. With just 109 million parameters, the model effectively closes the performance gap between encoder models and LLM-based skill extraction systems, achieving state-of-the-art results on most metrics. The model is made available online.[2]
2) We develop the skill extraction evaluation with our newly constructed e**X**haustive **L**abels (**Skill-XL**) benchmark: a large manual annotation effort with a unique focus on annotating job ads with exhaustive labels, explicitly coding redundancy among skill labels in the annotations. The dataset contains 111 job ads annotated on a sentence-by-sentence basis with a total of 8,471 skill labels, and is made available online.[3]
3) We produce **JobBERT V2**, a simplified and superior iteration of the earlier JobBERT model [7] for job title normalization, which achieves results on par with those of complex state-of-the-art methods. **JobBERT V2** is available online.[4]

By combining these contributions, we provide a scalable and efficient framework for labor market analysis. In Section II we will lay out the previous work on both skill extraction and job title normalization. The methodologies for **ConTeXT-match**, **Skill-XL** and **JobBERT V2** are described in Sections III, IV and V respectively. Experimental results are detailed in Section VI.

II. RELATED WORK

We review prior research on both skill extraction and job title normalization. This review focuses on some key limitations of existing approaches, particularly regarding computational efficiency and the trade-off between model complexity and performance — challenges we address throughout this work.

A. SKILL EXTRACTION

Skill extraction is a foundational task in labor market analysis that identifies and standardizes skill mentions in unstructured job advertisements. This task is central to HR applications, such as resume screening, job recommendations, and workforce planning. However, skill extraction poses unique challenges owing to the variability of natural language in job ads and the need to handle explicit and implicit skill descriptions.

Early skill extraction methods identified only skill mentions without normalizing them to a common taxonomy [8], [9], [10], [11]. In their simplest form, these methods rely on named entity recognition (NER), either through rule-based matching [8], [9] or through training recurrent neural networks [10]. A significant advancement came with SkillSpan [11], which reframed the task as a more flexible span detection problem, with recent leading methods such as NNOSE [12] and Skill-LLM [13]. However, by leaving out the normalization towards a common taxonomy, their application to robust large-scale market analysis is limited.

To address the normalization challenge, subsequent work approached skill extraction as an extreme multi-label classification (XMLC) problem. These methods map text to predefined skill taxonomies such as ESCO or O*NET [4], [5], [14], [15], [16], inspired by dense-encoder XMLC methods such as LightXML [17], DeepXML [18] and XR-Transformer [19]. In our earlier work [14], we used binary logistic regression classifiers for each ESCO skill, trained on rule-mined data. We also contributed the SkillSpan-ESCO benchmark, enabling the first systematic evaluation of fine-grained skill label predictions.

[2]https://huggingface.co/TechWolf/ConTeXT-Skill-Extraction-base

[3]https://huggingface.co/datasets/TechWolf/Skill-XL

[4]https://huggingface.co/TechWolf/JobBERT-v2

Because of the computational challenges of training thousands of binary classifiers in XMLC tasks, contrastive learning has emerged as an efficient alternative for skill extraction. Contrastive learning builds on the seminal contrastive loss method [20], further popularized by the InfoNCE objective [21], and large-scale instantiations such as SimCLR [22] and CLIP [23]. Applied to skill extraction, this approach trains a bi-encoder architecture that learns to embed both text and skill labels in a shared vector space, thereby enabling efficient similarity-based skill ranking. Its first application to skill extraction [15] used a two-stage pipeline for German job ads: first detecting skill spans, then ranking ESCO skills against these spans, based on a contrastive learning approach. We proposed a significant simplification in an earlier contribution [16] by eliminating the span detection step and directly learning to rank skills against complete sentences, thus enabling the capture of both explicit and implicit skill mentions.

Recent advances in skill extraction have been driven by large language models (LLMs) that offer two distinct advantages. First, their strong language understanding capabilities enable the automatic generation of high-quality training data for skill extraction [4], [16], thereby addressing the data scarcity challenge. Second, LLMs excel at few-shot learning, leading to new extraction approaches that leverage prompting. For instance, [4] demonstrated success with a two-stage approach that first retrieves candidate skills and then uses LLM prompting for final selection. This approach was further refined by [5], which introduced methods to automatically optimize prompts, achieving state-of-the-art results. However, these LLM-based methods face practical limitations: they incur significant computational costs and latency compared with local encoder-based approaches, making them less suitable for large-scale applications requiring real-time processing.

The contrastive learning method to skill extraction [16] has the smallest, yet still very considerable performance gap compared to LLM-based systems. We hypothesize that this at least partially stems from ranking predictions solely at the sentence level, neglecting valuable token-level information. By incorporating token-level information into the skill extraction method, we aim to both improve recall and enhance the interpretability of predictions, while remaining much more efficient than LLM-based systems.

B. JOB TITLE NORMALIZATION

Job title normalization has traditionally been considered as a (semi-)supervised learning problem. A first effort by LinkedIn defined a ''couple dozen'' standard job categories and performed classification based on common key phrases [24]. A more elaborate taxonomy of over 4k job titles was used in **Carotene**, using a hierarchical cascade model for job title classification [25]. The same task was later solved by **DeepCarotene**, using an end-to-end deep convolutional neural network [26]. These methods have a disadvantage as they require extensive annotation of training examples for each job title class.

To overcome this, we introduced **JobBERT** [7], where we pioneered job title normalization as a ranking task, indicating similarity of job titles based on a specialized representation learning method. Specifically, we developed a BERT-based job title representation encoder trained on noisy skill requirements from job ads. This method does not require annotating training data, but instead relies on the assumption that good representations of job titles can be achieved by learning to predict the probability of skill requirements conditioned on the job ad title. This method was adapted in the **Doc2VecSkill** work [27] through separate stages for learning representations for skills and then for job titles. Later, **VacancySBERT** [28] simplified the approach by training a siamese network with contrastive learning and in-batch negatives to draw job title representations closer to the representation of concatenated skill sets from job ads, where skills are extracted using a non-disclosed proprietary algorithm. The most performant approach to job title normalization is the so-called **Job Description Aggregation Network**, which omits the explit skill data requirement and instead learns good job title representations by learning to match them to representations of corresponding job ad descriptions [29].

We build on the insights of previous work, bringing together a new method of contrasting job titles and corresponding skill sets, with full transparency of the skill extraction method as also introduced in this work. Our JobBERT v2 approach is most similar to VacancySBERT but considers the asymmetry of matching job titles to skill sets.

III. SKILL EXTRACTION METHODOLOGY

Our skill extraction approach aims to identify relevant skills in job ad sentences with minimal annotation requirements, following the philosophy of [16]. Instead of fully labeled text spans, we require only pairs of a sentence and an associated skill label. This relaxed data requirement enables scalable training by avoiding expensive, exhaustive span-level annotations. We train a bi-encoder with contrastive learning based on this training data. The key innovation that we introduce here is **Con**trastive **T**oken-level **eX**plainable **T**ext **match**ing (**ConTeXT-match**). It is an adaptation of contrastive text representation learning in which we remove the information bottleneck of averaging the sentence representations, instead allowing label-dependent token-level attention. Thus, the model can better attend to the parts of a sentence that are most indicative of a given skill, providing both accurate retrieval and interpretable attribution.

A. TOKEN-LEVEL CONTRASTIVE LEARNING

Our model architecture is a bi-encoder transformer network that outputs contextual representations of both sentences and skills. For a given sentence x, all skills s are ranked based on a metric match(x, s). Rather than using the cosine similarity of the representations averaged across the tokens, we introduce

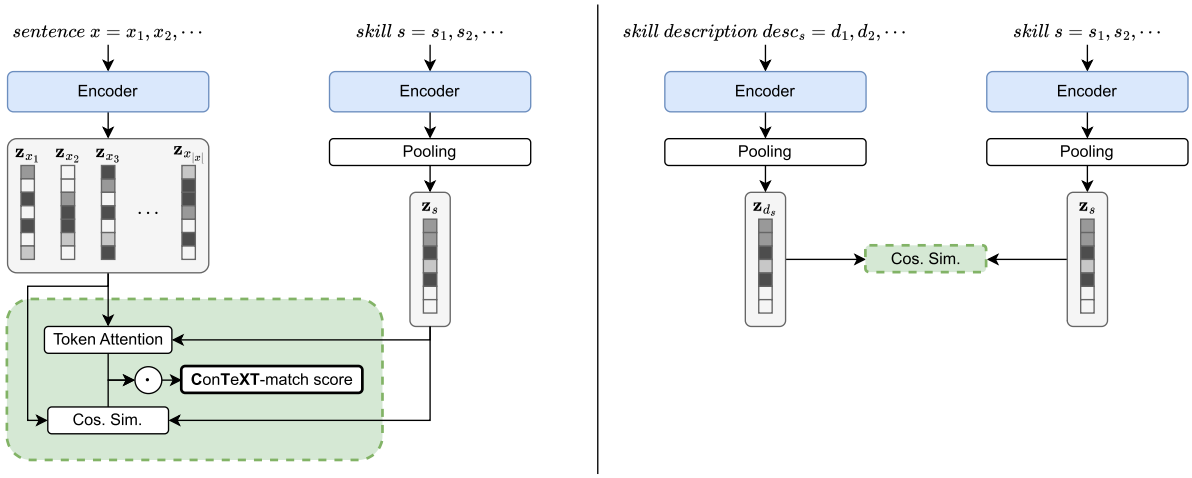

FIGURE 1. Overview of the multi-task setup proposed for our skill extraction training procedure. The bi-encoder produces token embeddings for sentences *x* and averaged embeddings for skills *s* and their descriptions $desc_s$. The ConTeXT-match mechanism is applied to produce the matching scores between sentence and skill (left). For the skill descriptions, simple cosine similarity is used instead of the ConTeXT-match scores (right).

label-dependent token-level similarity aggregation through **ConTeXT-match**. This method calculates the cosine similarity between the averaged skill representation and each token representation of the sentence. The final match score is defined as a weighted average of the token similarities through a simple multiplicative attention mechanism.

Specifically, given a sentence x and skill s, they are tokenized into $x_1, x_2, \ldots, x_{|x|}$ and $s_1, s_2, \ldots, s_{|s|}$ respectively. The bi-encoder computes the contextual token representations for the sentence and the skill independently (through full weight sharing), represented by $\mathbf{z}_{x_i}$ and $\mathbf{z}_{s_i}$. We represent the skill s by averaging the embeddings of its tokens:

$$\mathbf{z}_s = \frac{1}{|s|} \sum_{t=1}^{|s|} \mathbf{z}_{s_t}. \tag{1}$$

The attention mechanism operates on the averaged skill embedding $\mathbf{z}_s$ and individual sentence token embeddings $\mathbf{z}_{x_i}$. The cosine similarity between the skill embedding and token x_i is given by:

$$\cos(x_i, s) = \frac{\mathbf{z}_{x_i} \cdot \mathbf{z}_s}{\|\mathbf{z}_{x_i}\| \cdot \|\mathbf{z}_s\|}. \tag{2}$$

Finally, the match between the sentence and the skill is obtained as the weighted average of the token similarities, as follows:

$$\text{match}(x, s) = \sum_{j=1}^{|x|} \alpha_j \cdot \cos(x_j, s), \tag{3}$$

where the weights α_j sum to one, as defined by

$$\alpha_j = \frac{\exp\left(\mathbf{z}_{x_j} \cdot \mathbf{z}_s\right)}{\sum_{k=1}^{|x|} \exp\left(\mathbf{z}_{x_k} \cdot \mathbf{z}_s\right)}. \tag{4}$$

This token-level attention mechanism eliminates the information bottleneck that is otherwise imposed by averaging the sentence representation into a fixed-length embedding, instead allowing skill matching to dynamically attend to the relevant parts of the sentence. Second, it allows for native attribution and visualization of which parts in the text are matched to a certain skill. Skills are ranked by relevance with respect to a sentence x by descending match (x, s).

Our training method requires a pairwise dataset of sentences and corresponding skills. Formally, the training dataset $\mathcal{D}$ consists of pairs (x, s) where x is a sentence from a job ad and s is a corresponding skill label. The training procedure is based on contrastive learning with the InfoNCE framework as introduced in [21] relying on in-batch negatives, and adapted to its symmetric variation. The loss for a given pair (x, s) in a batch of size B is defined as:

$$\mathcal{L}_{x,s} = \frac{\mathcal{L}_{x,s}^{\text{forward}} + \mathcal{L}_{x,s}^{\text{backward}}}{2}, \tag{5}$$

where

$$\mathcal{L}_{x,s}^{\text{forward}} = -\log \frac{\exp(\text{match}(x, s) \cdot \text{scale})}{\sum_{k=1}^{B} \exp(\text{match}(x, s_k) \cdot \text{scale})}, \tag{6}$$

and

$$\mathcal{L}_{x,s}^{\text{backward}} = -\log \frac{\exp(\text{match}(x, s) \cdot \text{scale})}{\sum_{k=1}^{B} \exp(\text{match}(x_k, s) \cdot \text{scale})}. \tag{7}$$

Here, the scale hyperparameter controls the entropy of the softmax function. We rely on the gradient caching technique from [30] to effectively scale up the **ConTeXT-match** method to very large batch sizes, by splitting them into micro-batches of size 512.

Finally, we make use of rich meta-data that is often present in skill taxonomies. We specifically make use of skill descriptions as present in e.g. ESCO, because the assumption is that these descriptions should aid in generalization of the final model. The contrastive task of matching skills with their corresponding description makes use of the same symmetrical InfoNCE loss with the exception of using simple

cosine similarity between the averaged representations of both instead of the **ConTeXT-match** scores. The complete setup is illustrated in Fig. 1. During training, batches are always constructed from pairs of one task only, and the tasks are sampled proportionally to their respective training data set sizes. We note that more information from skill taxonomies can be incorporated into this multi-task setup, and we show the impact of some of these choices in Appendix A.

B. CALIBRATION AND REDUNDANCY FILTERING

The trained model produces a ranked list of skill predictions, which requires the calibration of a threshold τ that we select for maximal F1 score. In addition to retaining relevant skills through a calibrated threshold, we are also concerned about the *redundancy* of the returned labels, as large label sets are often not mutually exclusive, and excessive redundancy in the returned labels can degrade user experience. An example of such redundant skills is ''machine learning'' and ''utilise machine learning'' which are two separate skills present in ESCO. To handle semantic overlap among the predicted skills, we introduce a redundancy filtering step using the **ConTeXT-match**'s token-level attention mechanism. We assume that when two semantically overlapping labels are matched to a sentence, their corresponding token-level attention scores should follow a similar pattern. From the skills that meet threshold τ, we retain only those that have the highest dot product with at least one token in the input sequence. The dot product between the skill representation and the token representation is used instead of the attention scores α, as we empirically find them to work best. This may be because of the normalizing effect of the softmax function, which foregoes the comparability of the scores across multiple labels. The template tokens of the model (*beginning of sentence* and *end of sentence*) are not considered in this selection mechanism.

IV. SKILL-XL BENCHMARK DEVELOPMENT

We refer to [31] for an in-depth survey of the existing benchmarks for skill extraction. Most benchmarks are tailored to span labeling without linking the mentions to a standardized fine-grained classification of skills. There are two exceptions: [32] retrieves job advertisements from a government platform[5] where the employers also added fine-grained skill labels. The labels are not linked to the relevant sentences in the job ad, making them unsuitable for evaluation at the sentence level. Second, in our previous work [14], we published fine-grained ESCO skill labels linked on top of the span labeling benchmark ''SkillSpan'' by [11]. While this benchmark, referred to as **SkillSpan-ESCO**, allows sentence-level evaluation, it does not take into account the redundancy of the large number of skills in ESCO. In other words, the SkillSpan-ESCO data has been annotated with just one representative ESCO skill per skill in the sentence, even when multiple correct ESCO skills apply. This method of annotation leads to an imperfect evaluation of skill extraction.

Based on these insights, we set out to develop **Skill-XL** as the first benchmark for skill extraction that has both fine-grained and exhaustive ESCO skill annotation on the sentence level, with equally informative labels for a sentence being clustered into groups. We perform the annotation for full job ads such that **Skill-XL** can also be used for document-level evaluation.

A team of 12 artificial intelligence experts annotated up to 12 full job ads each, all sampled from the TechWolf data lake, posted in the United States in January 2024. Because the distribution of roles for which job ads are written is skewed, we sample ads for annotation in two different ways. Each expert received a set of five randomly sampled job ads, as well as five *diverse* job ads that were better spread out over the support of the data distribution. The selection of these diverse job ads was based on a proprietary job ad feature representation that was subject to FAst Diversity Subsampling [33]. Finally, each expert was asked to annotate two more job ads which were selected from the other annotators' data, as to allow for measuring inter-annotator agreement.

The average job ad contained 49 sentences, of which an average of 23 were identified as relevant by a proprietary segmentation model. Annotation of one job ad took around 30 minutes on average to annotate for one annotator. We make use of the clustered annotations to define the relevant metrics for the **Skill-XL** benchmark. The precision of skill extraction is defined as the percentage of predicted skill labels that are present in a reference skill cluster. Recall on the other hand is now calculated as the percentage of annotated skill clusters that are represented by at least one label in the predictions. Finally, the balanced optimization of precision and recall is expressed as the F1 score and calculated as the harmonic mean of precision and recall.

In order to measure inter-annotator agreement, we used the job ads where two or more annotators provided annotations. Conventional agreement metrics such as Cohen's kappa are less suitable for multi-label classification tasks [34]. We therefore revert to the averaged value of the F1 scores obtained by in turn evaluating each annotator's labels with another one's labels as reference. Doing so results in an average F1 agreement score of 0.4395, which we consider high given the large number of labels. Where multiple experts annotated the same job ad, we select the annotations from the expert with the highest average F1 agreement score as a simple proxy for the quality of their annotations. Finally, all job ad titles and sentences were anonymized by masking job-related sensitive and personal data regarding organization, location, contact, and name, following [35].

The resulting **Skill-XL** statistics are shown in table 1. The benchmark provides a valuable resource for evaluating skill extraction models, with its granularity and diversity making it broadly applicable to a range of methods, including

[5]https://www.mycareersfuture.gov.sg/

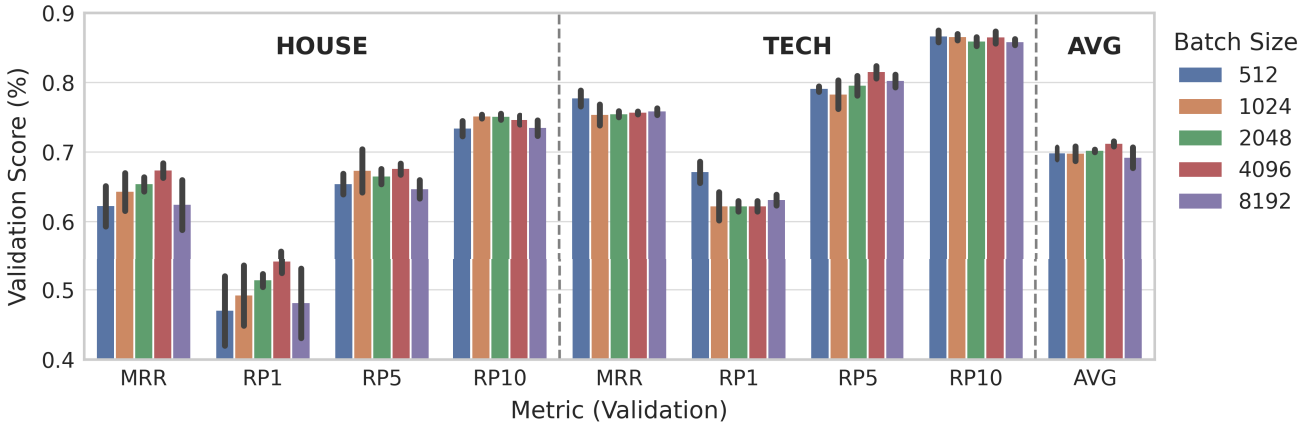

FIGURE 2. Average performance on the SkillSpan-ESCO development set for varying batch sizes. Each bar shows the mean score over three training runs with different random seeds; black whiskers denote one standard deviation.

TABLE 1. Skill-XL statistics. The table presents the number of job ads, sentences, and related statistics across different splits.

	Statistics	RANDOM	UNIQUE	TOTAL
Development	# Job ads	28	27	55
	# Sentences	1,650	1,039	2,689
	# Relevant sentences	894	435	1,329
	# Skill clusters	1,055	721	1,776
	# Skills	2,262	1,695	3,957
Test	# Job ads	28	28	56
	# Sentences	1,428	1,298	2,726
	# Relevant sentences	728	512	1,240
	# Skill clusters	1,103	795	1,898
	# Skills	2,685	1,829	4,514

span-based and sequence-labeling approaches. An annotated job ad example is shown in Appendix B.

V. SKILL-BASED JOB TITLE NORMALIZATION

For job title normalization, we follow the idea of using job-skill information to learn strong job title representations — as originally proposed in JobBERT [7]. Compared to the original study that used literal occurrence to find ESCO skills, we use qualitative skills extracted with our trained skill extraction model. We decide to use a simple contrastive learning strategy with the InfoNCE loss, optimizing a transformer encoder architecture to distinguish the corresponding skill set for a job title from the other skill sets in the batch. Rather than using complete weight-sharing as is common with sentence transformer models, we add an asymmetrical linear layer which accommodates for the different semantic meaning of job titles on the one hand and skill sets on the other.

VI. EXPERIMENTS AND RESULTS

A. SKILL EXTRACTION

For the skill extraction model, we used ESCO (v1.1.0) as the taxonomy of choice because of its prominent position in skill extraction research and its rich meta-data. The taxonomy contains 13,981 unique skills, each with a description. For the job ad sentence training data, we used the synthetic dataset from [16] which contains up to 10 synthetically generated job ad sentences for each skill in ESCO, totalling 138,260 unique sentence-skill pairs. For evaluation, we used both **SkillSpan-ESCO** to compare to the performance of previous methods, as well as **Skill-XL**. The latter is particularly used for the calibration of the final model, as well as for measuring the prediction redundancy.

For both **SkillSpan-ESCO** and **Skill-XL** we used the ranking-based evaluation metric macro-averaged R-Precision@K (RP@K) [36]. Because predictions are made on a sentence-basis, we restrict the evaluation to low values of K. RP@K is defined in (8), where the quantity $\text{Rel}(n, k)$ is a binary indicator of whether the k^{th} ranked label is a correct label for data sample n, and R_n is the number of gold labels for sample n. In addition, we use the mean reciprocal rank (MRR) of the highest ranked correct label as an indicator of the ranking quality.

$$\text{RP@K} = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{\text{Rel}\,(n, k)}{\min\,(K, R_n)} \tag{8}$$

We trained each model for a maximum of one epoch, and build in an early-stopping criterion that measures RP@5 on the SkillSpan-ESCO development set every 10% of the epoch and halts when there has been no increase two consecutive times. A sentence embedding model based on MPNET [37] and further trained on 1B positive pairs was used as initialization.[6] The model contains 109 million parameters and has a default maximum sequence length of 384 tokens which we keep. As our method predicts skills sentence by sentence, this sequence length is sufficiently long for practical purposes. The maximum sentence length observed in the TECH and HOUSE validation sets is 178 tokens, with a median of 16 tokens. The AdamW optimizer is used

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

TABLE 2. Performance comparison of skill extraction methods across the SkillSpan-ESCO test sets (HOUSE, TECH, TECHWOLF). Metrics include Mean Reciprocal Rank (MRR) and recall precision at Top-K (RP@1, RP@5, RP@10). Following previous work, the RP@K scores are reported in percentage points. Average positive training examples per skill are reported in the first column (N). Results of previous methods are the reported scores, and empty cells mean the metric was not reported. The strongest results per metric are shown in bold, and the second strongest are underlined. Whenever our results are significantly ($p < 0.05$) stronger than the second-best method, we indicate this with *.

		HOUSE				TECH				TECHWOLF			
	N	MRR	RP@1	RP@5	RP@10	MRR	RP@1	RP@5	RP@10	MRR	RP@1	RP@5	RP@10
Encoder classifiers													
Decorte *et al.* [14]	365	0.299	–	30.82	38.69	0.339	–	31.71	39.19	–	–	–	–
Clavié *et al.* [4]	40	0.326	27.20	37.60	46.47	0.299	27.16	33.41	39.86	–	–	–	–
Encoder rankers													
Clavié *et al.* [4]	40	0.355	26.44	35.22	43.73	0.405	32.84	49.67	58.66	–	–	–	–
Decorte *et al.* [16]	**10**	0.426	27.10	45.94	53.87	0.521	38.46	54.19	61.52	0.506	37.42	52.64	60.10
ConTeXT-match (Ours)	**10**	**0.530***	38.42	51.09	65.84	**0.632***	**50.99***	**63.98**	**73.99***	**0.562***	**43.15***	**57.69***	**66.08**
LLM-based systems													
Clavié *et al.* [4] GPT3.5	40	0.427	36.92	43.57	51.44	0.488	40.53	52.50	59.75	–	–	–	–
Clavié *et al.* [4] GPT4	40	0.495	**40.70**	53.34	61.02	0.537	46.52	61.50	68.94	–	–	–	
IReRa [5]	n/a	–	–	**56.50**	**66.51**	–	–	59.61	70.23	–	–	57.04	65.17

with WarmupLinear learning rate scheme, learning rate 5e−5 and 10% of the data for the warmup window. The scale hyperparameter is kept at its default value of 20.

We use the augmentation trick that we introduced in [16], which is to either prepend or append (equally split) each job ad sentence with another random job ad sentence during training. The idea behind this augmentation strategy is to overcome the fact that the synthetic job ad sentences tend to focus on just one skill, whereas we want the encoder to be able to effectively model multiple topics in its input. Note that this augmentation technique is only applied to the job ad sentences, not to skills or descriptions.

Contrastive learning with in-batch negatives has generally been shown to benefit significantly from large batch sizes [22]. Therefore, we decided to focus on determining the optimal batch size first, while keeping all other hyper parameters untouched. Fig. 2 shows the results obtained for the SkillSpan-ESCO development set for different batch sizes. For each batch size, we trained three models with different random seeds, and the standard deviation is shown as error bars. Based on this analysis, we selected batch size 4,096 as optimal, given that it obtains the highest average metric values, with lower standard deviation than observed for smaller batch sizes. For further analysis of training decisions, we refer to Appendix A.

The final model results on the test set of SkillSpan-ESCO are shown in table 2. A detailed runtime and cost comparison with the recent IReRa pipeline is provided in Appendix C. The results demonstrate that **ConTeXT-match** outperforms all other encoder-based methods, and even outperforms the LLM-based systems on most metrics.

Finally, we calibrate the model predictions using the total **Skill-XL** development set. The optimal τ was searched between 0 and 1, with steps of 0.01. The value $\tau = 0.53$ yields the highest F1 score of 0.4100, with an average redundancy of 27.32%. We formally define the redundancy of a prediction as the highest fraction of true positives that can be left out while still having the same clusters represented by at least one label in the predictions. When applying the intelligent filtering, in which the predictions are filtered based on their token-level attention, the optimal threshold was found to be 0.48, and an F1 score of 0.4389 was reached with a significantly reduced redundancy of only 13.46%.

The calibrated model achieves an F1 score of 0.407 and 0.401 on the **Skill-XL** RANDOM and UNIQUE test sets respectively, with a corresponding prediction redundancy of 16.88% and 15.38%. These numbers serve as the first baseline results for our new **Skill-XL** benchmark.

B. JOB TITLE NORMALIZATION

To train **JobBERT V2**, we start from the same base sentence embedding model (based on MPNET [37]) as we did for the skill extraction model.[7] We instantiate the asymmetrical linear layer to project the original 768-dimensional representations to a 1,024-dimensional space. To train **JobBERT V2**, we first enriched a large number of job ads with ESCO skills using the final skill extraction model. We randomly selected 100,000 job ads per month (from January 2020 to December 2024) posted in the United States. Note that we again applied the proprietary algorithm to filter out irrelevant sentences. After removing those ads that had fewer than five unique skills tagged, we ended up with a dataset of 5,579,240 enriched job ads, each consisting of a job ad title and the extracted set of skills. The skills of a job ad are shuffled and combined into one comma-separated text. Because of the limited context window of the model (512 tokens), we decided to limit the number of skills in a job ad to 25, which we achieve through random selection when necessary. A resulting pair in the training dataset may then appear as follows:

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

- Title: *Oracle Apps CRM Technical Consultant*
- Skills: *think analytically, Oracle Data Integrator, data models, computer technology, customer relationship management, use ICT systems*

TABLE 3. Performance comparison on the original JobBERT job title normalization benchmark. Metrics include Mean Reciprocal Rank (MRR) and recall at Top-K (RP@5, RP@10). Results of previous methods are the reported scores. The recall is reported in percentage points.

Model	MRR	R@5	R@10
JobBERT V1 [7]	0.309	38.65	46.04
Doc2VecSkill [27]	0.341	45.95	54.00
JD Aggregation Network [29]	0.387	49.24	57.22
JobBERT V2 (Ours)	**0.390**	**50.08**	**58.47**

We trained the model for one full epoch, with the InfoNCE loss function, batch size of 2 048, scale of 20 and learning rate 5e−5, and linear learning rate decay without warm-up. The resulting performance is shown in table 3, from which can be seen that the model outperforms all previous baselines even with minimal hyperparameter tuning. This demonstrates the power of the qualitative (skills) training data and the simple yet effective training objective we proposed or **JobBERT V2**.

TABLE 4. Token attention scores for each skill predicted, for three different example sentences. Higher attention scores are visualized as a stronger background color for the corresponding token.

(a) Lead the group in charge of cost and risk management objectives										
cost management										
lead	the	group	in	charge	of	cost	and	risk	management	objectives
lead a team										
lead	the	group	in	charge	of	cost	and	risk	management	objectives
risk management										
lead	the	group	in	charge	of	cost	and	risk	management	objectives

(b) You will write software in Java, Python and C++										
C++										
you	will	write	software	in	java	,	python	and	c	+	+
authoring software										
you	will	write	software	in	java	,	python	and	c	+	+
Java (computer programming)										
you	will	write	software	in	java	,	python	and	c	+	+
Python (computer programming)										
you	will	write	software	in	java	,	python	and	c	+	+

(c) Responsible for diagnosing, repairing, and maintaining cars									
diagnose problems with vehicles									
responsible	for	dia	gno	sing	,	repairing	,	and	maintaining	cars
carry out repair of vehicles									
responsible	for	dia	gno	sing	,	repairing	,	and	maintaining	cars
maintain vehicle service									
responsible	for	dia	gno	sing	,	repairing	,	and	maintaining	cars

C. VISUALIZATION OF SKILL EXTRACTION

We can visualize the token-level attention scores of **ConTeXT-match** to explain based on what tokens a skill was predicted. Three sentences, alongside their predicted skill tags and respective token-level explanation are shown in table 4. An alignment study between these token-level explanations and human rationales is reported in Appendix D.

VII. CONCLUSION

In this work, we have introduced a scalable and efficient approach to labor market analysis by tackling two of its main tasks being skill extraction and job title normalization. We proposed **ConTeXT-match**, a method that enhances accuracy and explainability of skill extraction, achieving state-of-the-art results with competitive performance and significantly lower computational cost compared to the LLM-based systems evaluated. While this work focuses on skill extraction, **ConTeXT-match** is a general method that can be applied to other extreme multi-label classification tasks, offering potential gains in performance across various domains.

Secondly, we train **JobBERT V2**, our second version of JobBERT which also achieves the best performance for the job title normalization task using a simple yet effective training method, made possible by leveraging large-scale high-quality skill data generated by our skill extraction model.

Finally, we have developed and released **Skill-XL**, our comprehensive skill extraction benchmark with sentence-level annotations that explicitly address redundancy in large label spaces. This benchmark serves as a new evaluation framework that allows for measuring both the accuracy and usefulness of skill extraction methods.

The efficiency of our skill extraction and job title normalization model unlocks the potential for large-scale, real-time labor market analysis in practical applications, which is something that can be further explored in future research. Extending the methods and results to non-English languages remains an important topic for future work. By open-sourcing both our models and the new benchmark, we aim to enable further research and innovation in this field.

APPENDIX A
SKILL EXTRACTION ABLATIONS

Different adaptations to the skill extraction training mechanism were compared against the final model. Their impact is assessed on both development sets of SkillSpan-ESCO in table 5, and we only report RP5 for conciseness, selected for its lowest observed standard deviations.

In intervention (a), we replace the **ConTeXT-match** mechanism with a cosine similarity score between averaged skill and sentence representations, as used in [16]. The large drop in performance signifies the strength of **ConTeXT-match**.

TABLE 5. Performance comparison of ablation interventions. Significantly worse or better results are indicated with ↓ or ↑ ($p < 0.05$).

Intervention	House RP@5	Tech RP@5
-	67.568 ± 0.779	81.489 ± 0.882
Training objective		
(a) No ConTeXT	58.989 ± 1.352 ↓	74.341 ± 0.105 ↓
(b) No augmentation	58.097 ± 1.375 ↓	77.044 ± 0.588 ↓
(c) Asymm. loss	65.638 ± 0.284 ↓	81.637 ± 1.008
Training tasks		
(d) No descriptions	66.658 ± 1.057	80.896 ± 1.557
(e) Add synonyms	67.960 ± 1.232	75.600 ± 0.839 ↓

Intervention (b) leaves out the prepend/append augmentation during training, which also leads to a considerable drop in performance. In intervention (c), the symmetrical loss $\mathcal{L}_{x,s}$ is replaced by only the forward loss $\mathcal{L}_{x,s}^{\text{forward}}$. While this incurs a significant loss on the HOUSE development set, the performance on the TECH set remains comparable.

We also perform some analysis on the influence of the additional training task. Specifically, we drop the description-skill matching task in intervention (d) and observe a moderate drop in performance. Finally, we consider using the skill synonyms provided in ESCO to create a third contrastive training task of skill synonym matching. This third task uses the same InfoNCE objective as the description-skill matching task. As seen in intervention (e), this incurs a significant drop in performance on the TECH set and a small but insignificant performance increase on the HOUSE set. The TECH set contains mostly technical jobs, leading to more hard skills (like specific programming languages) and less soft skills [11]. Inspection of the ESCO synonyms provided for hard skills reveals some inaccuracies like listing ''Live Script'' as a synonym for ''JavaScript'', which is related but synonymous. These inaccuracies are a likely cause for the performance drop on the TECH set.

APPENDIX B
SKILL-XL EXAMPLE

Table 6 contains part of an annotated job ad from the **Skill-XL** development set. Each sentence is shown in a separate row, and skills in the same annotation cluster are separated by a semicolon.

APPENDIX C
ConTeXT-match COMPARISON WITH IRERAS

To compare **ConTeXT-match** with a modern LLM-based system in terms of cost and quality, we evaluated it against **IReRa** [5], a three-stage pipeline that combines a local LLaMA-2-13B model,[8] a dense retriever and a re-ranking step using OpenAI's GPT-4 accessed through API. We create predictions on the combined HOUSE and TECH validation sets (136 unique sentences) and compare both the efficiency of the process and the quality of the predictions.

A. EFFICIENCY COMPARISON

Both systems were deployed on Google Cloud Platform (GCP) using on-demand resources.[9] The Llama model uses float16 weights (stored on 2 bytes), so it requires a minimum of 26 GB VRAM to use locally. **IReRa** was run on an `a2-highgpu-1g` virtual machine (1×A100-40 GB, 12 vCPU, 85 GB RAM) billed at $4.09 per hour, plus GPT-4 API usage at April 2025 prices ($10 per million input tokens, $30 per million output tokens). In contrast, **ConTeXT-match** was evaluated on an inexpensive `e2-medium` virtual machine (2 vCPU, 4 GB RAM) billed at $0.10 per hour. All virtual machine prices include a 500GB boot disk and are based on April 2025 pricing.

Both systems processed the combined 136 validation sentences of the HOUSE and TECH benchmarks. **ConTeXT-match** ran with a batch size of 8, averaging 1.1 seconds per batch – about 7 sentences per second. The entire run finished in 19 seconds and cost just $0.00053. The maximum memory usage ran up to 2.2 GB. IReRa completed in 14 minutes and 44 seconds, or roughly 6.5 seconds per sentence, which comes down to $1.00 for compute time. Because the current IReRa implementation does not support batching, the GPU usage is not optimized, so the $1.00 is an upper bound, and the throughput a lowerbound. OpenAI usage added $3.06, totaling $4.06 for the IReRa processing. These experiments show a cost reduction with a factor of 1/7660 when using **ConTeXT-match** compared to IReRa, as well as strongly reduced compute and memory requirements. These results confirm that ConTeXT-match delivers state-of-the-art skill-extraction quality *without the prohibitive costs and specialised infrastructure* associated with modern LLM pipelines, making large-scale, affordable deployment feasible.

B. QUALITATIVE COMPARISON

To understand *how* both systems differ, we manually inspected their outputs for the 136 validation sentences and grouped the discrepancies into four recurrent patterns. Throughout the discussion, predictions from **ConTeXT-match** are shown in blue, while those from IReRa are shown in violet. For IReRa we only consider the top–5 ranked labels, mirroring a realistic downstream cut-off.

1) PRECISION VERSUS RECALL

ConTeXT-match exhibits a deliberate high-precision bias: it almost never proposes a skill that is not textually or semantically supported. The flip side is occasional under-extraction. For the sentence *"Work on a mix of front-end, back-end and cloud technologies."* the model yields {cloud technologies}, omitting the equally explicit *front-end* and *back-end*. IReRa, in contrast, returns {cloud technologies, implement front-end website design, design cloud architecture, manage cloud data and storage, cloud security and compliance}. It has a higher coverage, but three of the five predictions

[8] https://huggingface.co/meta-llama/Llama-2-13b-chat-hf

[9] https://cloud.google.com/compute/all-pricing

TABLE 6. Annotated example job Ad. Relevant sentences are annotated with skill clusters and specific skills. Non-relevant sentences are grayed out.

Sentence	Skills
Job Title:	-
Bigdata Architect (Data Modeling / Data Architect)	data models; create data models; design database scheme
Location:	-
location (Onsite)	-
Expertise in Hive, Big Data environments (Hadoop preferred), HBase, Spark (Scala preferred but python is also ok)	analyse big data; data mining
	Hadoop
	SPARK
	Scala
	Python (computer programming)
Should have experience in scaling an application	distributed computing; decentralized application frameworks
Extensive experience in Data Modeling	data models; create data models; design database scheme
Extensive experience in Data Architect	information architecture; manage ICT data architecture
Should have strong Performance (compute and I/O) background	ICT performance analysis methods; advise on efficiency improvements
Should have experience in UNIX shell scripting, Jenkins configurations	use scripting programming
	Jenkins (tools for software configuration management)
Perform Code Reviews and understand the stack	conduct ICT code review
	provide technical expertise; consult technical resources
Follow best practices while coding	implement ICT coding conventions; ensure adherence to organisational ICT standards
Good Knowledge in Machine Learning and Data Science algorithms	machine learning; utilise machine learning; ML (computer programming)
	algorithms; data analytics
Thanks & Regards,	-
name	-
Senior Executive	-
- International Consulting	-
organization	-

('design cloud architecture', 'manage cloud data and storage', 'cloud security and compliance') have no lexical anchor in the sentence and would count as false positives.

2) LEXICAL STABILITY VERSUS SYNONYM PROLIFERATION

ConTeXT-match's redundancy-filtering outputs a single canonical form per skill, making the list compact and deduplicated across the corpus. IReRa frequently emits several near-identical variants, inflating the candidate set. For a simple requirement such as *"Fluent in written and spoken English"*, **ConTeXT-match** returns {English}, whereas IReRa offers all of {English, write English, understand written English, understand spoken English, communication}. While this boosts recall for loosely coupled downstream ontologies, it pushes extra work to any consumer that must cluster the five labels back to one concept.

3) HALLUCINATED ABSTRACTIONS VERSUS LITERAL TOKENS

Driven by GPT-4 re-ranking, IReRa is prone to infer abstract or industry-specific abilities that are only tangentially related to the source sentence. From the prompt *"Are you ready to work in a dynamic and international team?"* it proposes collaborate on international energy projects and develop international cooperation strategies. Conversely, **ConTeXT-match** stays close to surface evidence, yielding the literal

work in an international environment. While hallucinated abstractions may reveal latent needs, they harm precision in settings where textual faithfulness is mandatory.

4) ROBUSTNESS TO NOISY TOKENS VERSUS TOKEN ARTEFACTS

ConTeXT-match occasionally seems less robust in case of an abundance of punctuation, abbreviations or parentheses. In the phrase *"HTML and CSS (LESS, SCSS, PostCSS)"* it emits the skill JSSS. IReRa correctly outputs LESS and SCSS.

C. SUMMARY

Qualitatively, **ConTeXT-match** fails mainly by omission or minor token glitches, whereas IReRa fails by over-generation and synonym duplication. For high-precision use-cases (e.g., automated profile completion) our model's conservative stance and ten-thousand-fold lower cost are decisive advantages. When maximum recall is the overriding goal, IReRa's expansive, noisier outputs may be preferable, provided that post-hoc clustering and human validation steps are affordable.

APPENDIX D
INTERPRETABILITY STUDY: HUMAN ALIGNMENT OF conTeXT-MATCH ATTENTION

In addition to quantitative accuracy, an important requirement for industrial skill extraction systems is *explainability*. Because **ConTeXT-match** produces token-level attention weights for every predicted skill, we performed a small-scale user study to verify whether those weights correspond to the tokens that humans regard as most diagnostic. We drew a random sample of 50 sentences from the development split of **Skill-XL**. Two independent annotators from the team were shown the sentence and a corresponding ground-truth skill cluster. Annotators highlighted all spans of words they considered an *explanation word*, i.e., a word or phrase that justifies the presence of the given skill. The instructions emphasized that noncontiguous selections were allowed and that they should abstain if no token qualified. Each annotator worked in isolation. We measured rank correlation between the human binary vector $\mathbf{h} \in \{0,1\}^{|x|}$ and the **ConTeXT-match** attention scores $\alpha \in [0,1]^{|x|}$ using Spearman's ρ, computed per sentence and then averaged. The template tokens of the model (*beginning of sentence* and *end of sentence*) are not included in this annotation nor correlation measurement. Where an annotator labeled no token, that sentence was skipped for that annotator. The Spearman rank correlation (ρ) between the model's attention and Annotator 1 was 0.56, while for Annotator 2 it was 0.50. The spearman correlation between the binary annotations of both annotators was 0.60. These results indicate that the model's attention correlates positively with both annotators (mean $\rho \approx 0.53$), approaching human-human consistency.

Table 7 and 8 visualize the human annotations next to the **ConTeXT-match** attention scores for two specific examples.

TABLE 7. Comparison of model attention scores (α) and human annotation, for the sentence *"Interact with different people of all ages and backgrounds"* and skill *"communicate with elderly groups"*. The attention scores are indicated by the background intensity. Binary human annotations are shown in bold.

α	Annotator 1	Annotator 2
interact	**interact**	**interact**
with	**with**	**with**
different	**different**	different
people	**people**	**people**
of	**of**	**of**
all	**all**	**all**
ages	**ages**	**ages**
and	and	and
backgrounds	backgrounds	backgrounds

TABLE 8. Comparison of model attention scores (α) and human annotation, for the sentence *"You will also maintain strong relationships with business partners to help shape requirements and provide both analytical and technical support."* and skill *"build business relationships"*. The attention scores are indicated by the background intensity. Binary human annotations are shown in bold.

α	Annotator 1	Annotator 2
you	you	you
will	will	will
also	also	also
maintain	**maintain**	**maintain**
strong	**strong**	**strong**
relationships	**relationships**	**relationships**
with	**with**	**with**
business	**business**	**business**
partners	**partners**	**partners**
to	to	to
help	help	help
shape	shape	shape
requirements	requirements	requirements
and	and	and
provide	provide	provide
both	both	both
analytical	analytical	analytical
and	and	and
technical	technical	technical
support	support	support
.	.	.

The human annotations in table 7 show that the skill is attributed to the large majority of words in the sentence. Correspondingly, the attention scores are spread out across the tokens.

In contrast, table 8 shows a case where the skill is clearly explained by one confined part of the sentence. The model's attention scores center around the same words, with a particularly sharp signal on the token ''relationships''.

These results demonstrate that the token-level explanations produced by **ConTeXT-match** are both informative and, to a large extent, intuitively plausible to human experts. Deeper analysis into potential improvements can be done in future

work by extending the number of samples and annotators in this exercise.

REFERENCES

[1] I. Rahhal, I. Kassou, and M. Ghogho, "Data science for job market analysis: A survey on applications and techniques," *Expert Syst. Appl.*, vol. 251, Oct. 2024, Art. no. 124101.

[2] W. V. Cunningham and P. Villaseñor, "Employer voices, employer demands, and implications for public skills development policy connecting the labor and education sectors," *World Bank Res. Observer*, vol. 31, no. 1, pp. 102–134, Feb. 2016.

[3] I. Khaouja, I. Kassou, and M. Ghogho, "A survey on skill identification from online job ads," *IEEE Access*, vol. 9, pp. 118134–118153, 2021.

[4] B. Clavié and G. Soulié, "Large language models as batteries-included zero-shot ESCO skills matchers," in *Proc. 3rd Workshop Recommender Syst. Human Resour. (RecSys HR) 17th ACM Conf. Recommender Syst. (RecSys)*, vol. 3490, M. Kaya, T. Bogers, D. Graus, C. Johnson, and J.-J- Decorte, Eds., 2023, pp. 1–10.

[5] K. D'Oosterlinck, O. Khattab, F. Remy, T. Demeester, C. Develder, and C. Potts, "In-context learning for extreme multi-label classification," 2024, *arXiv:2401.12178*.

[6] P. Safikhani, H. Avetisyan, D. Föste-Eggers, and D. Broneske, "Automated occupation coding with hierarchical features: A data-centric approach to classification with pre-trained language models," *Discover Artif. Intell.*, vol. 3, no. 1, p. 6, Feb. 2023.

[7] J. J. Decorte, J. Van Hautte, T. Demeester, and C. Develder, "JobBERT: Understanding job titles through skills," in *Proc. FEAST, ECML-PKDD Workshop*, 2021, p. 9.

[8] M. Zhao, F. Javed, F. Jacob, and M. McNair, "SKILL: A system for skill identification and normalization," in *Proc. AAAI Conf. Artif. Intell.*, Jan. 2015, vol. 29, no. 2, pp. 4012–4017.

[9] S. Jia, X. Liu, P. Zhao, C. Liu, L. Sun, and T. Peng, "Representation of job-skill in artificial intelligence with knowledge graph analysis," in *Proc. IEEE Symp. Product Compliance Eng.-Asia (ISPCE-CN)*, Dec. 2018, pp. 1–6.

[10] L. Sayfullina, E. Malmi, and J. Kannala, "Learning representations for soft skill matching," in *Proc. Int. Conf. Anal. Images, Social Netw. Texts*, W. M. P. van der Aalst, V. Batagelj, G. Glavaš, D. I. Ignatov, M. Khachay, S. O. Kuznetsov, O. Koltsova, I. A. Lomazova, N. Loukachevitch, A. Napoli, A. Panchenko, P. M. Pardalos, M. Pelillo, and A. V. Savchenko, Eds., Jan. 2018, pp. 141–152.

[11] M. Zhang, K. Jensen, S. Sonniks, and B. Plank, "SkillSpan: Hard and soft skill extraction from English job postings," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Language Technol.*, M. Carpuat, M.-C. de Marneffe, and I. V. M. Ruiz, Eds., 2022, pp. 4962–4984.

[12] M. Zhang, R. van der Goot, M.-Y. Kan, and B. Plank, "NNOSE: Nearest neighbor occupational skill extraction," in *Proc. 18th Conf. Eur. Chapter Assoc. Comput. Linguistics Long Papers*, vol. 1, S. Julian's, Ed., 2024, pp. 589–608.

[13] A. Herandi, Y. Li, Z. Liu, X. Hu, and X. Cai, "Skill-LLM: Repurposing general-purpose LLMs for skill extraction," 2024, *arXiv:2410.12052*.

[14] J.-J. Decorte, J. Van Hautte, J. Deleu, C. Develder, and T. Demeester, "Design of negative sampling strategies for distantly supervised skill extraction," 2022, *arXiv:2209.05987*.

[15] A.-S. Gnehm, E. Bühlmann, H. Buchs, and S. Clematide, "Fine-grained extraction and classification of skill requirements in german-speaking job ads," in *Proc. 5th Workshop Natural Language Process. Comput. Social Sci. (NLP+CSS)*, D. Bamman, D. Hovy, D. Jurgens, K. Keith, B. O'Connor, and S. Volkova, Eds., 2022, pp. 14–24.

[16] J.-J. Decorte, S. Verlinden, J. V. Hautte, J. Deleu, C. Develder, and T. Demeester, "Extreme multi-label skill extraction training using large language models," in *Proc. AI4HR PES, Int. Workshop AI Human Resour. Public Employment Services*, Jan. 2023, pp. 1–10.

[17] T. Jiang, D. Wang, L. Sun, H. Yang, Z. Zhao, and F. Zhuang, "LightXML: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, May 2021, pp. 7987–7994.

[18] K. Dahiya, D. Saini, A. Mittal, A. Shaw, K. Dave, A. Soni, H. Jain, S. Agarwal, and M. Varma, "DeepXML: A deep extreme multi-label learning framework applied to short text documents," in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, Mar. 2021, pp. 31–39.

[19] J. Zhang, W.-C. Chang, H. Yu, and I. S. Dhillon, "Fast multi-resolution transformer fine-tuning for extreme multi-label text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 7267–7280.

[20] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 1735–1742.

[21] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[22] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2020, pp. 1597–1607.

[23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2021, pp. 8748–8763.

[24] R. Bekkerman and M. Gavish, "High-precision phrase-based document classification on a modern scale," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2011, pp. 231–239.

[25] F. Javed, Q. Luo, M. McNair, F. Jacob, M. Zhao, and T. S. Kang, "Carotene: A job title classification system for the online recruitment domain," in *Proc. IEEE 1st Int. Conf. Big Data Comput. Service Appl.*, Mar. 2015, pp. 286–293.

[26] J. Wang, K. Abdelfatah, M. Korayem, and J. Balaji, "DeepCarotene -job title classification with multi-stream convolutional neural network," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 1953–1961.

[27] R. Zbib, L. L. Alvarez, F. Retyk, R. Poves, J. Aizpuru, H. Fabregat, V. Simkus, and E. García-Casademont, "Learning job titles similarity from noisy skill labels," in *Proc. FEAST ECML-PKDD Workshop*, Jan. 2022, pp. 1–13.

[28] M. Y. Bocharova, E. V. Malakhov, and V. I. Mezhuyev, "VacancySBERT: The approach for representation of titles and skills for semantic similarity search in the recruitment domain," *Appl. Aspects Inf. Technol.*, vol. 6, no. 1, pp. 52–59, Apr. 2023.

[29] N. Laosaengpha, T. Tativannarat, C. Piansaddhayanon, A. Rutherford, and E. Chuangsuwanich, "Learning job title representation from job description aggregation network," in *Proc. Findings Assoc. Comput. Linguistics*, Bangkok, Thailand, L.-W. Ku, A. Martins, and V. Srikumar, Eds., 2024, pp. 1319–1329.

[30] L. Gao, Y. Zhang, J. Han, and J. Callan, "Scaling deep contrastive learning batch size under memory limited setup," in *Proc. 6th Workshop Represent. Learn. NLP (RepL4NLP)*, A. Rogers, I. Calixto, I. Vulić, N. Saphra, N. Kassner, O.-M. Camburu, T. Bansal, and V. Shwartz, Eds., 2021, pp. 316–321.

[31] E. Senger, M. Zhang, R. V. D. Goot, and B. Plank, "Deep learning-based computational job market analysis: A survey on skill extraction and classification from job postings," in *Proc. 1st Workshop Natural Language Process. Human Resour. (NLP4HR)*, E. Hruschka, T. Lake, N. Otani, and T. Mitchell, Eds., Jan. 2024, pp. 1–15.

[32] A. Bhola, K. Halder, A. Prasad, and M.-Y. Kan, "Retrieving skills from job descriptions: A language model based extreme multi-label classification framework," in *Proc. 28th Int. Conf. Comput. Linguistics*, Barcelona, Spain, D. Scott, N. Bel, and C. Zong, Eds., 2020, pp. 5832–5842.

[33] B. Shang, D. W. Apley, and S. Mehrotra, "Diversity subsampling: Custom subsamples from large data sets," *Informs J. Data Sci.*, vol. 2, no. 2, pp. 161–182, Oct. 2023.

[34] M. Marchal, M. Scholman, F. Yung, and V. Demberg, "Establishing annotation quality in multi-label annotations," in *Proc. 29th Int. Conf. Comput. Linguistics*, N. Calzolari, C.-R. Huang, H. Kim, J. Pustejovsky, L. Wanner, K.-S. Choi, P.-M. Ryu, H.-H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, and S.-H. Na, Eds., Oct. 2022, pp. 3659–3668.

[35] K. N. Jensen, M. Zhang, and B. Plank, "De-identification of privacy-related entities in job postings," in *Proc. 23rd Nordic Conf. Comput. Linguistics (NoDaLiDa)*, Jan. 2021, pp. 210–221.

[36] I. Chalkidis, E. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "Extreme multi-label legal text classification: A case study in," in *Proc. Natural Legal Language Process. Workshop*, N. Aletras, E. Ash, L. Barrett, D. Chen, A. Meyers, D. Preotiuc-Pietro, D. Rosenberg, and A. Stent, Eds., 2019, pp. 78–87.

[37] K. Song, X. Tan, T. Qin, J. Lu, and T. Liu, "MPNet: Masked and permuted pre-training for language understanding," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, Jan. 2020, pp. 16857–16867.

JENS-JORIS DECORTE received the M.S. degree in engineering from Ghent University, Belgium, in 2020. He is currently pursuing the Ph.D. degree with the Internet Technology and Data Science Laboratory (IDLab), Ghent University–imec, in collaboration with TechWolf. His supervisors are Prof. Chris Develder and Prof. Thomas Demeester. His research focuses on AI for HR applications, including skill extraction and skill-related tasks, using large-scale text encoding and contrastive learning methods. He has been part of TechWolf, since 2019, where he contributed to the development of the company's AI systems and led the AI Team. In addition to his research and industry work, he is a Lecturer in ICT evening education with Hogeschool Gent. His research interests include natural language processing, skills intelligence, and machine learning.

JEROEN VAN HAUTTE received the M.Sc. degree in computer science engineering from Ghent University, Belgium, and the M.Sc. degree in advanced computer science from the University of Cambridge. He is currently the Co-Founder and the Chief Technology Officer of TechWolf, one of Europe's fastest-growing AI companies, specializing in skills intelligence. His AI research at the University of Cambridge led to a novel approach to capturing workforce skills. His work in artificial intelligence and entrepreneurship has been widely recognized, including being named to the Forbes 30 Under 30 list and being named a Technology Pioneer by the World Economic Forum. At TechWolf, he oversees the product and engineering organization, as well as research and innovation.

CHRIS DEVELDER (Senior Member, IEEE) received the M.Sc. degree in computer science engineering and the Ph.D. degree in electrical engineering from Ghent University, Ghent, Belgium, in 1999 and 2003, respectively. He was a Research Visitor with the University of California at Davis, Davis, CA, USA, from July 2007 to October 2007, and Columbia University, New York, NY, USA, from January 2013 to June 2015. He is currently an Associate Professor with the Research Group IDLab, Department of Information Technology (INTEC), Ghent University–imec. He was/is involved in various national and European research projects, such as FP7 Increase, FP7 C-DAX, H2020 CPN, H2020 Bright, H2020 BIGG, H2020 RENergetic, and H2020 BD4NRG. He also (co-)leads two research teams within the IDLab, such as the UGent-T2K on converting text to knowledge, NLP, mostly information extraction using machine learning, and the UGent-AI4E on artificial intelligence for energy applications, smart grid. He has co-authored over 200 refereed publications in international conferences and journals. He is a fellow of the Research Foundation, FWO. He is a Senior Member of ACM and a member of ACL.

THOMAS DEMEESTER received the master's degree in electrical engineering, in 2005, and the Ph.D. degree in computational electromagnetics, with a grant from the Research Foundation, Flanders (FWO-Vlaanderen), in 2009. He is currently an Associate Professor with IDLab, Department of Information Technology, Ghent University-imec, Belgium. He has been involved in a series of national and international projects in the area of NLP, and co-authored around 150 peer-reviewed contributions in international journals and conferences. His research interests then shifted to information retrieval (with a research stay at the University of Twente, The Netherlands, in 2011), natural language processing (NLP), and machine learning (with a stay at University College London, U.K., in 2016), and more recently to Neuro-Symbolic AI, with applications in AI for clinical and pre-clinical research. He is a member of AAAI and ELLIS.