Design of Negative Sampling Strategies for Distantly Supervised Skill Extraction

Jens-Joris Decorte[1,2,*], Jeroen Van Hautte[2], Johannes Deleu[1], Chris Develder[1] and Thomas Demeester[1]

[1]Ghent University – imec, 9052 Gent, Belgium
[2]TechWolf, 9000 Gent, Belgium

Abstract

Skills play a central role in the job market and many human resources (HR) processes. In the wake of other digital experiences, today's online job market has candidates expecting to see the right opportunities based on their skill set. Similarly, enterprises increasingly need to use data to guarantee that the skills within their workforce remain future-proof. However, structured information about skills is often missing, and processes building on self- or manager-assessment have shown to struggle with issues around adoption, completeness, and freshness of the resulting data. These challenges can be tackled using automated techniques for skill extraction. Extracting skills is a highly challenging task, given the many thousands of possible skill labels mentioned either explicitly or merely described implicitly and the lack of finely annotated training corpora. Previous work on skill extraction overly simplifies the task to an explicit entity detection task or builds on manually annotated training data that would be infeasible if applied to a complete vocabulary of skills. We propose an end-to-end system for skill extraction, based on distant supervision through literal matching. We propose and evaluate several negative sampling strategies, tuned on a small validation dataset, to improve the generalization of skill extraction towards implicitly mentioned skills, despite the lack of such implicit skills in the distantly supervised data. We observe that using the ESCO taxonomy to select negative examples from related skills yields the biggest improvements, and combining three different strategies in one model further increases the performance, up to 8 percentage points in RP@5. We introduce a manually annotated evaluation benchmark for skill extraction based on the ESCO taxonomy, on which we validate our models. We release the benchmark dataset for research purposes to stimulate further research on the task.

Keywords

Skill Extraction, Information Extraction, Distant Supervision, Extreme Multi-Label Classification

1. Introduction

Skill extraction is an information extraction task that aims to identify all skills mentioned in a text. It is essential for many HR applications, such as resume screening and job recommendation systems. A comparative survey on skill extraction indicates that research interest has steadily grown over the last decade [1]. Traditionally, skill extraction has been approached as finding and disambiguating entities in texts. These methods typically rely on a named entity recognition (NER) component based on phrase-matching or a trained LSTM model [2, 3, 4].

However, skills are often present implicitly as longer sequences of words (which we refer to as spans) or full sentences rather than being mentioned explicitly: over 85% of unique required skills in job ads have been reported never to be explicitly mentioned [5]. Very recently, the work titled SkillSpan has reformulated skill extraction as a more flexible span detection task [6]. The authors released a dataset of job postings with span annotations and trained SpanBERT-based models to detect skill spans as a sequence labeling task. In follow-up work, the authors developed a classification model to link such a span to the corresponding coarse-grained skill group in ESCO [7]. To overcome the difficulty of labeling these spans, the authors relied on weak supervision by automatically selecting labels based on the ESCO search API [7]. Another study manually annotated job ads with soft skills, which were consolidated into a released dataset called FIJO [8]. However, instead of using an exhaustive list of soft skills, they only incorporated four broad labels to decrease the difficulty of the annotation. The skill *extraction* task can also be reduced to binary skill *detection*, again reducing the challenge compared to fine-grained skill extraction [9]. These works follow a more relaxed formulation of skill extraction, but they all suffer from the difficulty of annotating a fine-grained training dataset.

Some work avoids this labeling difficulty completely by using readily available labeled datasets. For example, in [5], an eXtreme Multi-Label Classification (XMLC) model was trained based on a corpus of job ads with

RecSys in HR'22: The 2nd Workshop on Recommender Systems for Human Resources, in conjunction with the 16th ACM Conference on Recommender Systems, September 18–23, 2022, Seattle, USA.
*Corresponding author.
jensjoris@techwolf.ai (Jens-Joris Decorte); jeroen@techwolf.ai (Jeroen Van Hautte); johannes.deleu@ugent.be (Johannes Deleu); chris.develder@ugent.be (Chris Develder); thomas.demeester@ugent.be (Thomas Demeester)
https://www.techwolf.ai (Jens-Joris Decorte); https://www.techwolf.ai (Jeroen Van Hautte)

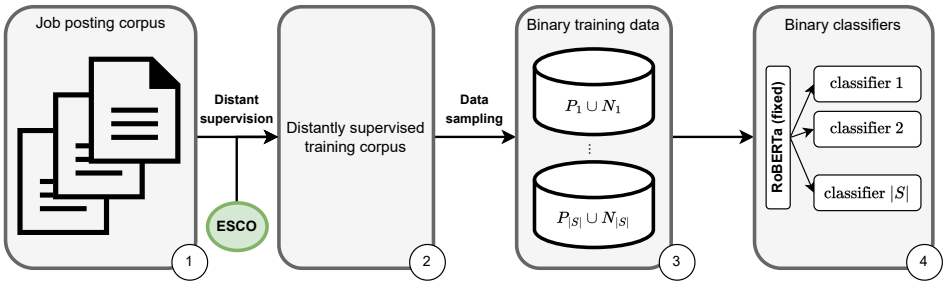


Figure 1: Overview of our method. Using the ESCO skill taxonomy, the distantly supervised training corpus ② is created from our job posting corpus ①. Based on the negative sampling strategy, the positive data is combined with negative examples ③ and finally a classifier is trained for each skill ④.

attached skills provided by an online job ads platform. However, the authors reported that for that corpus, at least 40% of the vacancies missed 20% of explicitly stated skills in their labels. Recent work [10] successfully reconstructed the BERT-XLMC approach on Dutch vacancy texts using the Dutch RobBERT model [11]. The training dataset used for this work is however based on the output of an existing commercial skill extraction solution.

We propose a new end-to-end approach to fine-grained skill extraction that does not rely on a large hand-labeled training corpus. Instead, we ease the requirements on the training data such that it can be automatically collected through distant supervision. We cast the multi-label skill classification task into independent binary classification problems, with skills labeled on the sentence level, to encompass both explicit and implicit skill descriptions. To the best of our knowledge, our work is the first one to tackle fine-grained skill extraction in such a flexible distant supervision setup. Our distant supervision training set contains few false positives, due to the literal matching of known skills, which is a task with low ambiguity. However, we expect many false negatives, for skills not literally mentioned. This is quantified in Section 4.1. We investigate to what extent the distantly supervised training set can be leveraged at maximum effectiveness to train a fine-grained skill extraction system. To that end, we design a number of negative sampling strategies that can be used to tune the extraction model training process on a small annotated development set, covering only a fraction of all potential skills (0.2%, to be precise, in our experimental setting). Finally, in order to stimulate research on automated skill extraction, and to facilitate the comparison of future models with our results, we release[1] our development and test data, which is constructed on top of the "SkillSpan" dataset [6], adding annotations with the ESCO [12] skill labels.

[1]https://github.com/jensjorisdecorte/Skill-Extraction-benchmark

2. Related Work

Multi-label classification datasets often have a skewed label distribution, with many labels occurring only a few times or even being completely absent in the training data. Some works have focused on improving the few-shot and zero-shot classification performance of multi-label text classification on these rare or unseen labels. Typically, the information in structured label graphs (such as label descriptions or relations) or word embeddings are used as an input to the system in order to generalize to unseen labels [13, 14]. However, these methods still rely on a large labeled training dataset to work. In the absence of any supervision, [15] uses a novel self-supervision training objective to train a dense sentence representation model that is used to assign labels based on cosine similarity in the learned space. Yin *et al.* [16] propose an entailment approach to zero-short text classification, where the input text is called the premise, and a hypothesis is constructed for each label using the template "the text is about *label*". The premise and hypothesis are concatenated before being presented to a BERT-based model for prediction, making this method slow at inference for large label spaces.

Multi-label classification datasets not only suffer from the rare label problem, also many labels are just missing, since they are usually only partially labeled: instances without labels thus may either be truly negative, or positive but not identified as such during labeling. The "Single Positive Labels" scenario is an extreme case of missing labels, where only one positive label is available for each training instance [17]. Research on this topic is limited, and typically focuses on designing custom loss functions [18] or online estimation of the missing labels during training [17]. This line of work is closely related to "Positive-Unlabeled" (PU) binary classification, which is typically also tackled using custom loss functions [19].

Typically in a distant supervision setup, the labeling function is followed by a filtering step that aims to reduce the number of false positives in the labels [20]. However, we find that the number of false positives produced by the distant supervision step is low in our case of literal skill mentions. This has been shown previously by [21] where literal skill mentions have been successfully used as distant supervision for the task of job title representation learning. Rather than focusing on a filtering step, we draw inspiration from the idea of "hard negative examples" in representation learning to improve the learning process. In contrastive learning, hard negative examples refer to samples that are difficult to distinguish from an anchor point [22]. This approach improves the discriminative abilities and downstream performance of unsupervised representation learning methods. We adapt this idea to the multi-label classification setup, by oversampling negative examples from related labels. More details on this approach are contained in the following section.

3. Skill Extraction Approach

We approach the task of skill extraction as a sentence-level multi-label classification task. A high-level overview of the method is shown in Fig. 1. Our method uses distant supervision based on the ESCO skill taxonomy to automatically assign (partial) skill labels for a given set of sentences from the HR domain (in particular, mined from vacancies). Negative sampling strategies are used to combine 'positive sentences' for a given skill (i.e., sentences labeled with that skill during the distant supervision step) with sentences not containing that skill (referred to as 'negative sentences'). Finally, a binary classifier f_s is trained for each skill s, based on the constructed positive and negative sentences for that skill. It consists of a logistic regression classifier on top of a (frozen) representation for the sentences, as described in more detail below.

Distantly supervised training set: Given a set S of skills and a background corpus of sentences D, for each skill $s \in S$, a set P_s of positive sentences is collected from D through distant supervision. In particular, we use the ESCO [12] skills taxonomy as the set of classification labels. The set P_s of positive sentences for each skill s, consists of those sentences in D that literally mention the skill s or any of its alternative forms, as provided in the taxonomy. This assumes that there are no ambiguous skill names, which holds in most cases as skill names tend to be specific. The positive labels are very precise, due to the distant supervision process based on literal matches with the highly specific ESCO skill names. However, this means potentially many skills remain unlabeled, i.e., the training data is prone to false negatives. After the distant supervision step, on average 365 sentences were labeled per skill (for the set of 13,891 ESCO skills). This dataset follows a long tail distribution, with 75.1% of skills occurring in only ten or fewer sentences.

Skill extraction model: The model architecture is depicted in Fig. 1. We use a frozen pre-trained RoBERTa [23] model with mean pooling to transform input sentences into fixed-length contextual representations, before presenting them for classification. The classification is performed by separate binary text classification models f_s, each generating an independent prediction value for their respective skill label s. In contrast to a typical multi-label model, we optimize each classification model separately on a different corresponding dataset, instead of training all weights together.

Training with negative sampling: P_s serves as positive training data for classifier f_s, and negative examples are sampled from the union of all positive sentence datasets of all other skills. The basic mechanism for sampling negatives is uniform sampling from this union. However, following the ideas in representation learning [22], we hypothesize that sentences from related skills are more *informative*, harder to distinguish from the positive sentences (i.e., closer to the decision boundary), and could thus improve the learning process. As such, a fraction of the negative examples are sampled specifically from sentences that are labeled with a related (but different) label to skill s. We refer to these sentences as "hard" negative samples. Our negative sampling strategy is thus defined by two important factors. First, the fraction of uniformly sampled negatives versus the hard negative samples is important. Secondly, how we define whether two skills are related is crucial to the learning process. We introduce three different strategies for selecting the related skills in Section 3.1.

Inference and evaluation: The final model is used to rank the relevance of all skills for a given sentence. Similar to [14], we use the macro-averaged R-Precision@K (RP@K) metric to evaluate the performance of the method. Since predictions are made on a sentence-basis, we restrict the evaluation to low values of K. RP@K is defined in (1), where the quantity $Rel(n,k)$ 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. More information on the evaluation is presented in Section 4.1.

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

	Siblings	Levenshtein	Embedding
disarm land mine	ensure flock safety protect important clients signal for explosion deal with challenging people	find land mines search for land mines identify land mines dismantle machines	repair mine machinery handle mining plant waste management of mine ventilation construct road base
Haskell	DevOps XQuery Windows Phone SPARK	add smell upsell sink wells speak well	PostgreSQL Erlang JavaScript C++
manage musical staff	discharge employees manage volunteers supervise nursing staff guide staff	manage musical groups manage musical events manage musicians manage educational staff	manage agricultural staff manage staff manage dental staff manage educational staff

Table 1
Examples of related skill labels for the three different selection strategies. The "siblings" examples are in no particular order as they form a set of siblings, rather than an ordered list.

3.1. Negative Sampling Strategies

Rather than randomly sampling negative examples for training each binary skill classifier, we assume that sampling more *informative* negatives will likely lead to a more efficient training procedure. Instead of sampling hard negative sentences directly, we first identify related (yet different) skills, and then sample sentences with those labels. We introduce three different strategies for identifying such related skills, which we analyze through the experiments defined in Section 4. The considered sets of related skills, given a particular skill s are obtained as follows:

- **Siblings**: all skills that share a parent concept with s, as indicated by the "broader concepts" field in ESCO.
- **Levenshtein**: The top 100 skills closest to s, according to their Levenshtein distance.
- **Embedding**: The top 100 skills closest to s in terms of cosine similarity with their mean-pooled RoBERTa-encoded skill name representations.

For each of the negative sampling strategies, some example ESCO skills with their related labels according to the strategy are shown in table 1.

4. Experimental setup

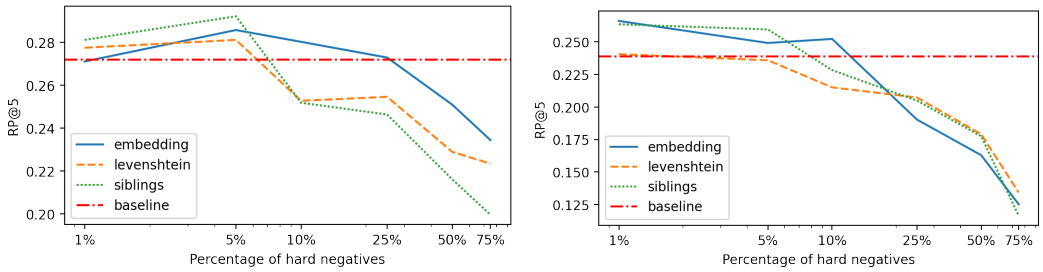
4.1. Evaluation

While hand-labeling a training dataset for skill extraction is infeasible (given the huge number of skills, e.g., over 13k in ESCO), we argue that with reasonable manual work, it is possible to construct a benchmark that can be used to compare the performance of different models. We build upon the test set of the *SkillSpan* dataset from [6], which contains job posting sentences annotated with skill spans. We manually annotate each span in SkillSpan with its corresponding ESCO skill (if it exists). This span-based multi-class annotation is less complex than annotating complete sentences with multiple labels. The process is performed on the test sets of the publicly released subsets *TECH* and *HOUSE*. Details on the annotation guidelines can be found in Appendix A. The annotation effort results in fine-grained ESCO skill labels for 64.5% of the spans. We split this dataset into a validation and test set using a 20%/80% split. The validation set contains 165 unique skill labels, and over 80% of the unique skill labels in the test set never occur in the validation set. A more detailed breakdown of the number of spans and annotations is shown in table 2.

	TECH		*HOUSE*	
	val	test	val	test
# sentences	470	1882	243	973
# spans	262	1024	191	786
# spans with ESCO label	152	644	131	532

Table 2
Benchmark dataset statistics on the number of sentences, spans, and ESCO labeled spans, for both the *TECH* and *HOUSE* partitions of the SkillSpan dataset. Numbers of the validation and test split are indicated in the table.

In order to verify our hypothesis that the distant supervision labeling leads to quite precise positive labels, at the cost of many false negatives, we validated the distant supervision labeling of the test set against the manual annotations. The automatically assigned labels are indeed rather precise (overall precision of 79%), but at the cost of low coverage (i.e., a recall of 14.6%).



(a) *TECH* (b) *HOUSE*

Figure 2: Evaluating the effect of the fraction of hard negatives used during training, for each of the three strategies (siblings, levenshtein and embedding-based similarity with the considered positive skill) separately. The baseline model performance without hard negative sampling is shown by the horizontal red line. Metrics are reported on the validation sets.

4.2. Experiments

The sentences used for training are collected from a large proprietary corpus of public job postings. This dataset has been collected from different public job boards and contains a large number of English job postings. ESCO is used for the distant supervision step: a skill label is assigned when the skill itself, or one of its alternative forms provided by ESCO, is literally mentioned in a sentence. For each skill classifier f_s, a maximum of one thousand positive sentences is retained. The amount of negative examples per positive example is set to 10. We train a baseline classifier without hard negative sampling. In this case, all negative examples are sampled uniformly from the other positive corpora. To investigate the optimal hard negative sampling procedure, we conduct a hyper-parameter search for the fraction of negatives sampled using the three strategies (sibling, levenshtein, embedding) versus uniform sampling. Based on the performance on the validation sets, we decide on an optimal value for this percentage. Finally, we report the contribution of each of the negative sampling strategies when combined. This is reported based on performance on the unseen test set, and contributions of the strategies are shown through ablations, by leaving one strategy out at a time. We refer to Appendix B for more details on the training procedure.

5. Results and Discussion

The results of the hyper-parameter search for each of the negative sampling strategies are shown in Fig. 2. From these results, it is clear that the different strategies have different effects on the model performance. Most notably, we find that the optimal fraction of hard negative sampling is no higher than 5% for any strategy. This is in line with previous findings on hard negative sampling [22]. Sampling large amounts of hard negatives even has a large negative impact on the performance of the model. Secondly, the "levenshtein" strategy brings the least improvements out of all three strategies.

Finally, we trained a model that combines all strategies. Based on the results of the above hyper-parameter search, we chose 5% as an optimal value for the fraction of negatives sampled through the combined hard negative strategies. To assess the impact of each of the strategies within this combination, we trained three more models in which each of the three strategies is left out respectively. The performance of these final models is shown in table 3. The combination of all three strategies yielded the overall best model. This model has large performance gains across the MRR and RP@K metrics for both the *TECH* and *HOUSE* dataset.

Leaving out the "Levenshtein" strategy has a relatively low impact on the performance. This might be understood by looking at the examples in table 1: string similarity surfaces unrelated skills, for example for proper nouns such as *Haskel*. This could partially explain the relatively low utility of this negative sampling strategy. On the other hand, leaving out the "siblings" strategy takes away the largest part of the performance improvements. This strategy makes use of the hierarchy defined in the ESCO taxonomy, and thus is a reliable method for selecting informative hard negatives. The effect of the "embedding" strategy is comparable to the "siblings" strategy and thus proves a good alternative in case a hierarchy such as the one in ESCO is not available.

6. Conclusion and Future Work

We propose an end-to-end approach to skill extraction using distant supervision. The method is able to make fine-grained skill predictions (using 13,891 skills from ESCO) for a given input sentence. We introduce the

	TECH			*HOUSE*		
	MRR	RP@5	RP@10	MRR	RP@5	RP@10
Baseline classifier	0.246	23.65	33.71	0.255	26.66	34.19
Classifier$_{neg}$	0.326	**31.71**	39.09	**0.299**	**30.82**	**38.69**
Classifier$_{neg}$ without embeddings	0.323	31.43	**39.19**	0.298	29.09	37.70
Classifier$_{neg}$ without Levenshtein	**0.339**	31.11	38.55	0.298	30.14	37.22
Classifier$_{neg}$ without siblings	0.303	30.57	37.07	0.281	29.20	35.91

Table 3
Evaluation metrics of final skill extraction models on the *TECH* and *HOUSE* test sets. Reported metrics are mean reciprocal rank (MRR), R-Precision at 5 and at 10 (RP@5, RP@10).

idea of hard negative sampling through related labels in a multi-label classification setup and propose three different strategies to select these related labels. We investigate the impact of each of the strategies, and found that all three strategies combined yield the highest increase on top of a baseline model without hard negative sampling. Both the distant supervision and the hard negative sampling are designed to work well without manual labeling, which makes the whole method very flexible. To the best of our knowledge, we are the first to design such a system for skill extraction, and we improve on prior work by providing methods that have relaxed the requirements from ground-truth data and that have the ability to make very fine-grained skill predictions. Finally, we release our hand-labeled test and validation dataset for skill extraction to stimulate further research on the task.

Future work could entail a more extensive investigation of other hyper-parameters, such as the number of negatives per positive sentence (k), which was fixed to 10 in this work. Secondly, more performance gains could be made if the RoBERTa weights were fine-tuned during training, but this requires changes in the training setup which should be carefully investigated. Lastly, it could be interesting to investigate how limited manual labor can maximally improve the performance of the method even further with techniques such as active learning.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback. This project was funded by the Flemish Government, through Flanders Innovation & Entrepreneurship (VLAIO, project HBC.2020.2893).

References

[1] I. Khaouja, I. Kassou, M. Ghogho, A survey on skill identification from online job ads, IEEE Access 9 (2021) 118134–118153.

[2] M. Zhao, F. Javed, F. Jacob, M. McNair, Skill: A system for skill identification and normalization, in: Twenty-Seventh IAAI Conference, 2015.

[3] L. Sayfullina, E. Malmi, J. Kannala, Learning representations for soft skill matching, in: International conference on analysis of images, social networks and texts, Springer, 2018, pp. 141–152.

[4] S. Jia, X. Liu, P. Zhao, C. Liu, L. Sun, T. Peng, Representation of job-skill in artificial intelligence with knowledge graph analysis, in: 2018 IEEE symposium on product compliance engineering-asia (ISPCE-CN), IEEE, 2018, pp. 1–6.

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

[6] M. Zhang, K. N. Jensen, S. D. Sonniks, B. Plank, Skillspan: Hard and soft skill extraction from English job postings, arXiv preprint arXiv:2204.12811 (2022).

[7] M. Zhang, K. N. Jensen, B. Plank, Kompetencer: Fine-grained skill classification in Danish job postings via distant supervision and transfer learning, arXiv preprint arXiv:2205.01381 (2022).

[8] D. Beauchemin, J. Laumonier, Y. L. Ster, M. Yassine, "FIJO": a French insurance soft skill detection dataset, arXiv preprint arXiv:2204.05208 (2022).

[9] D. A. Tamburri, W.-J. Van Den Heuvel, M. Garriga, DataOps for societal intelligence: A data pipeline for labor market skills extraction and matching, in: 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI), IEEE, 2020, pp. 391–394.

[10] N. Vermeer, V. Provatorova, D. Graus, T. Rajapakse, S. Mesbah, Using RobBERT and eXtreme multi-

label classification to extract implicit and explicit skills from Dutch job descriptions (2022).
[11] P. Delobelle, T. Winters, B. Berendt, RobBERT: a Dutch Roberta-based language model, arXiv preprint arXiv:2001.06286 (2020).
[12] ESCO, European skills, competences, qualifications and occupations, EC Directorate E (2017).
[13] J. Lu, L. Du, M. Liu, J. Dipnall, Multi-label few/zero-shot learning with knowledge aggregated from multiple label graphs, arXiv preprint arXiv:2010.07459 (2020).
[14] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, Extreme multi-label legal text classification: A case study in EU legislation, arXiv preprint arXiv:1905.10892 (2019).
[15] Y. Xiong, W.-C. Chang, C.-J. Hsieh, H.-F. Yu, I. Dhillon, Extreme zero-shot learning for extreme text classification, arXiv preprint arXiv:2112.08652 (2021).
[16] W. Yin, J. Hay, D. Roth, Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach, arXiv preprint arXiv:1909.00161 (2019).
[17] E. Cole, O. Mac Aodha, T. Lorieul, P. Perona, D. Morris, N. Jojic, Multi-label learning from single positive labels, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 933–942.
[18] D. Zhou, P. Chen, Q. Wang, G. Chen, P.-A. Heng, Acknowledging the unknown for multi-label learning with single positive labels, arXiv preprint arXiv:2203.16219 (2022).
[19] M. C. Du Plessis, G. Niu, M. Sugiyama, Analysis of learning from positive and unlabeled data, Advances in neural information processing systems 27 (2014).
[20] L. Sterckx, T. Demeester, J. Deleu, C. Develder, Knowledge base population using semantic label propagation, Knowledge-Based Systems 108 (2016) 79–91.
[21] J.-J. Decorte, J. Van Hautte, T. Demeester, C. Develder, Jobbert: Understanding job titles through skills, arXiv preprint arXiv:2109.09605 (2021).
[22] J. Robinson, C.-Y. Chuang, S. Sra, S. Jegelka, Contrastive learning with hard negative samples, arXiv preprint arXiv:2010.04592 (2020).
[23] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.
[25] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, arXiv preprint arXiv:1908.10084 (2019).

A. Annotation guidelines

Each item that needs to be annotated is a **span**, thus a part of a longer job posting sentence. Both the span and the complete sentence are shown to provide the right context for annotation. When a span is ambiguous, the full sentence must be read to understand the meaning of the span.

The task is to annotate the correct and most specific skill that is mentioned or implied by the span. The place of the candidate labels within the shortlist has no importance during annotation. In the case that no correct skill is found in the shortlist, you may search for the correct skill using the ESCO interface [12]. If you still cannot find a correct label, select *LABEL NOT PRESENT*. If you find that the span can generally not be interpreted as a skill, select *UNDERSPECIFIED*.

A.1. Examples

- Given the span "partner continuously with your many stakeholders" and the candidate labels *Communicate With Stakeholders*, *Negotiate With Stakeholders* and "Liaise With Shareholders", only the first two labels are considered correct. "Communicate With Stakeholders" is most specific with regards to the span, so this label should be selected.
- Spans such as "apply your depth of knowledge" or "apply your expertise" are classified as *UNDERSPECIFIED*.

B. Training details

The separate classifiers are implemented as a simple logistic regression model, using the popular scikit-learn toolkit [24]. All parameters are set to their default values, except for the inverse regularization strength parameter C, which is set to 0.1 for stronger regularization. The RoBERTa model and the mean pooling operation are implemented using the Sentence-BERT library [25].