Improving Skill Extraction from Job Postings Using DeBERTa-v3-base on the SkillSpan Dataset

Nezar Abdilah Prakasa[1]

[1]Master's Program in Artificial Intelligence, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia

Abstract

Automatically extracting hard and soft skills from unstructured job postings is an important but challenging Named Entity Recognition (NER) task that underlies applications such as automated resume screening, labour-market analytics, and curriculum design. The SkillSpan benchmark (Zhang et al., 2022) established a domain-adapted BERT model, JobBERT, as a strong baseline, reporting a development F1-score of 56.64%. Although DeBERTa-v3 (He et al., 2023) has achieved state-of-the-art results on a wide range of NER benchmarks through disentangled attention and ELECTRA-style pre-training, it has not previously been evaluated on skill extraction. In this work, we replace the JobBERT backbone in the SkillSpan pipeline with `microsoft/deberta-v3-base`, keeping the dataset splits, preprocessing, training procedure, hyperparameters, and evaluation protocol unchanged. Our reproduced JobBERT baseline reaches 57.98% Dev F1, while DeBERTa-v3-base achieves **60.73% Dev F1** – an absolute improvement of **+4.09** percentage points over the originally reported baseline, with consistent gains on both the HOUSE (60.06%) and TECH (61.38%) domains. These preliminary results suggest that DeBERTa-v3's architectural improvements transfer effectively to the skill-extraction domain, and they motivate the more detailed error analysis presented as future work.

Keywords: skill extraction, named entity recognition, DeBERTa-v3, SkillSpan, job postings, transformer models

1 Introduction

The digital labour market produces an enormous and continuously growing volume of job postings on platforms such as LinkedIn, Indeed, and regional job boards. Embedded within these postings is detailed information about the hard skills (e.g., programming languages, software tools) and soft skills (e.g., communication, teamwork) that employers require, but this information is almost always expressed as free, unstructured text. Manually analyzing this text at scale is infeasible, which motivates automated approaches based on Named Entity Recognition (NER) – identifying and classifying skill-related spans directly from job-posting text.

The SkillSpan dataset (Zhang et al., 2022) is a widely used benchmark for this task. It provides dual-track BIO annotations distinguishing *Skill* spans from *Knowledge* spans across two domains, HOUSE (hospitality) and TECH (information technology), and establishes JobBERT – a BERT model further pre-trained on job-posting text – as the baseline encoder, reporting a development F1-score of 56.64%.

Independently, DeBERTa-v3 (He et al., 2023) has demonstrated substantial improvements over BERT- and RoBERTa-class encoders across GLUE, SQuAD, and several NER benchmarks. These gains stem from two architectural changes: (i) *disentangled attention*, which represents each token with separate content and relative-position vectors (He et al., 2021), and (ii) replacing masked-language-model pre-training with ELECTRA-style Replaced Token Detection combined with Gradient-Disentangled Embedding Sharing (GDES). Models built on DeBERTa-v3, such as GLiNER (Zaratiana et al., 2023), have shown strong zero-shot NER performance, yet DeBERTa-v3 has not been systematically evaluated on the skill-extraction task. At the same time, large language models have recently been applied to SkillSpan (Nguyen et al., 2024), achieving competitive results but at substantially higher computational cost than fine-tuning a moderately sized encoder.

This work addresses the resulting gap by asking:

- **RQ1.** What is the effect of replacing the JobBERT backbone with DeBERTa-v3-base on overall skill-extraction performance on SkillSpan?
- **RQ2.** Is any improvement consistent across both the HOUSE and TECH domains?
- **RQ3.** How do the error characteristics of DeBERTa-v3-base differ from JobBERT for hard-skill versus soft-skill spans?

We answer RQ1 and RQ2 with the preliminary experimental results reported in this paper; RQ3 is the subject of ongoing error analysis and will be reported in the full thesis. Our contributions are: (1) a controlled, single-variable comparison in which the transformer backbone is the *only* difference between the baseline and proposed pipelines; (2) the first reported evaluation of `microsoft/deberta-v3-base` on SkillSpan, showing a +4.09 percentage-point absolute improvement in Dev F1 over the originally reported JobBERT baseline; and (3) a fully documented configuration to support reproducibility.

The remainder of this paper is organized as follows. Section 2 reviews related work and situates the research gap. Section 3 provides background on NER, the SkillSpan annotation scheme, self-attention, and DeBERTa's disentangled attention mechanism. Section 4 describes the dataset,

model configurations, and training procedure. Section 5 presents preliminary results, and Section 6 concludes with limitations and future work.

2 Related Work

2.1 Skill Extraction from Job Postings

Early approaches to skill extraction relied on rule-based pattern matching and curated dictionaries to identify and normalize skill mentions (Zhao et al., 2015). Subsequent work introduced shallow neural representations: Sayfullina et al. (2018) used CNN and RNN encoders to learn representations for soft-skill matching, while Gugnani and Misra (2020) combined neural embeddings with taxonomy matching to detect implicit skills not explicitly named in the text. Cao and Zhang (2021) applied a BERT encoder with a BiLSTM-CRF head to analyze the social requirements expressed in Data Analyst job postings. Khaouja et al. (2021) and Xu et al. (2023) provide broader surveys of this growing literature, highlighting the shift from rule-based and feature-engineered systems toward pretrained transformer encoders.

2.2 Sequence Labeling and Transformer NER

Conditional Random Fields (Lafferty et al., 2001) and BiLSTM-CRF architectures (Lample et al., 2016) established the sequence-labeling foundations on which modern NER systems are built. The introduction of the Transformer architecture (Vaswani et al., 2017) and large pretrained encoders – BERT (Devlin et al., 2019), SpanBERT (Joshi et al., 2020), Longformer (Beltagy et al., 2020), and ELECTRA (Clark et al., 2020) – substantially advanced NER performance across domains. Gururangan et al. (2020) further showed that continued, domain-adaptive pre-training on in-domain corpora improves downstream task performance, a principle underlying JobBERT's domain adaptation for job-posting text.

2.3 DeBERTa and DeBERTa-v3

He et al. (2021) introduced *disentangled attention*, in which each token is represented by separate content and relative-position vectors, and an Enhanced Mask Decoder; this improved performance over BERT and RoBERTa on GLUE and SQuAD while keeping a comparable parameter budget. He et al. (2023) subsequently proposed DeBERTaV3, which replaces masked-language-model pre-training with ELECTRA-style Replaced Token Detection and introduces Gradient-Disentangled Embedding Sharing (GDES) to resolve the "tug-of-war" between the generator and discriminator embeddings. As a result, the 86M-parameter `deberta-v3-base` model outperforms much larger `deberta-large` models on several benchmarks. Zaratiana et al. (2023) built GLiNER, a generalist NER model, on a DeBERTa-v3 backbone and reported zero-shot performance exceeding ChatGPT on several NER benchmarks – but, like He et al. (2023), did not evaluate skill extraction specifically.

2.4 SkillSpan and Domain Adaptation

Zhang et al. (2022) introduced SkillSpan, a dataset of job postings from the HOUSE and TECH domains with dual-track BIO annotations for Skill and Knowledge spans, and proposed JobBERT, a BERT model further pre-trained on job-posting text, as the baseline encoder (Dev F1 = 56.64%). Zhang et al. (2023) extended this direction with ESCOXLM-R, a multilingual encoder pre-trained using the ESCO occupational taxonomy. Most recently, Nguyen et al. (2024) evaluated large language models (GPT-3.5/GPT-4) on the TECH subset of SkillSpan, finding them competitive with fine-tuned encoders but considerably more expensive at inference time.

2.5 Research Gap

Taken together, the literature shows that (i) DeBERTa-v3's architectural improvements have not been evaluated on the skill-extraction task, and (ii) the SkillSpan baseline has not been re-examined with a modern, disentangled-attention encoder. While LLM-based approaches are competitive, their computational cost limits practical deployment. A moderately sized encoder with an improved architecture, such as `deberta-v3-base`, may offer a better performance–efficiency trade-off. This paper directly addresses this gap by substituting DeBERTa-v3-base for JobBERT within the existing SkillSpan pipeline, changing no other component.

3 Background

3.1 Named Entity Recognition and the SkillSpan Annotation Scheme

NER is the task of identifying spans of text and assigning each a category label. A common encoding is the BIO scheme, in which each token is labeled `B-`*tag* (beginning of a span), `I-`*tag* (inside a span), or `O` (outside any span). SkillSpan uses a *dual-track* annotation: every token receives both a *Skill-tag* and a *Knowledge-tag*, allowing the two label sequences to overlap. Table 1 shows an example in the CoNLL format used by the dataset, where each line contains a token followed by its skill-tag and knowledge-tag.

Table 1: Example of the dual-track CoNLL annotation format used in SkillSpan.

Token	Skill-tag	Knowledge-tag
Python	O	B-Knowledge
programming	B-Skill	I-Knowledge
skills	I-Skill	O
required	O	O

In this example, "programming skills" is annotated as a *Skill* span, while "Python programming" is annotated as a *Knowledge* span – the token "programming" therefore carries two labels simultaneously, one from each track.

3.2 Self-Attention

The Transformer's self-attention mechanism (Vaswani et al., 2017) computes, for each token, a weighted combination

of *value* vectors V, where the weights are derived from the compatibility between *query* vectors Q and *key* vectors K:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \quad (1)$$

Multi-head attention runs h independent copies of this computation in parallel, each with its own learned projections, and concatenates the results:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2)$$

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W^O \quad (3)$$

This allows the model to capture different types of token relationships (e.g., syntactic and semantic) simultaneously. Figure 1 illustrates the scaled dot-product attention computation.

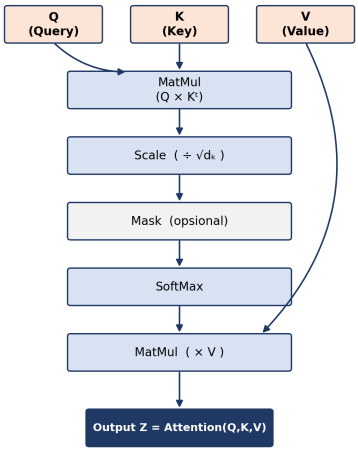

Figure 1: Scaled dot-product self-attention mechanism.

3.3 Disentangled Attention in DeBERTa

Standard Transformer encoders add position information directly to content embeddings before computing attention. DeBERTa instead represents each token i with two separate vectors: a content vector H_i and a relative-position vector $P_{i|j}$. The attention score between tokens i and j is computed as the sum of three terms – content-to-content, content-to-position, and position-to-content:

$$\begin{aligned} A_{i,j} &= H_i W_{q,c}(H_j W_{k,c})^\top \\ &\quad + H_i W_{q,c}(P_{j|i} W_{k,r})^\top \\ &\quad + P_{i|j} W_{q,r}(H_j W_{k,c})^\top \end{aligned} \quad (4)$$

where $W_{q,c}, W_{k,c}, W_{q,r}, W_{k,r}$ are learned projection matrices for content and relative-position queries and keys. Figure 2 illustrates how these three terms are combined.

DeBERTa-v3 retains this disentangled-attention architecture but changes the pre-training objective from masked language modeling to ELECTRA-style *Replaced Token Detection* (RTD), in which a discriminator learns to detect tokens replaced by a smaller generator network. Sharing embeddings between the generator and discriminator naively causes conflicting gradient signals; DeBERTa-v3

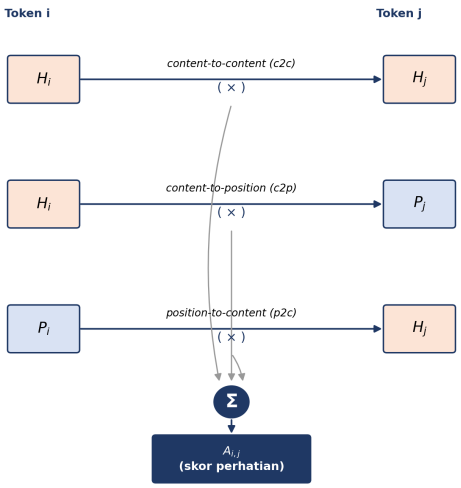

Figure 2: Disentangled attention: content-to-content (c2c), content-to-position (c2p), and position-to-content (p2c) terms are summed to produce the attention score $A_{i,j}$.

resolves this with *Gradient-Disentangled Embedding Sharing* (GDES), which shares the embedding matrix while stopping the discriminator's gradient from flowing back into the generator's embedding. This combination allows `deberta-v3-base` (86M backbone parameters, 128K-token SentencePiece vocabulary) to match or exceed the performance of much larger models on GLUE, SQuAD, and CoNLL-style NER benchmarks (He et al., 2023).

4 Methodology

4.1 Dataset

We use the SkillSpan dataset as released in its official repository[1], which contains job postings from two domains, HOUSE (hospitality) and TECH (information technology), in the CoNLL format described in Section 3.1. We use the official train/development/test splits without modification, and combine both domains for overall training and evaluation while also reporting per-domain results, consistent with Zhang et al. (2022).

4.2 Model Architectures

We compare two encoders within an otherwise identical token-classification pipeline:

- **JobBERT** (`jjzha/jobbert-base-cased`) – the baseline encoder from Zhang et al. (2022), a BERT-base model further pre-trained on job-posting text using a WordPiece tokenizer (30K vocabulary).
- **DeBERTa-v3-base** (`microsoft/deberta-v3-base`, proposed) – 12 transformer layers, hidden size 768, ∼86M backbone parameters, using a SentencePiece tokenizer with a 128K vocabulary.

In both cases, the encoder is followed by a linear token-classification head with a softmax over BIO labels, as illustrated in Figure 3. Subword alignment is handled automatically: labels are assigned to the first subword of each word, and ignored for subsequent subwords.

[1] https://github.com/kris927b/SkillSpan

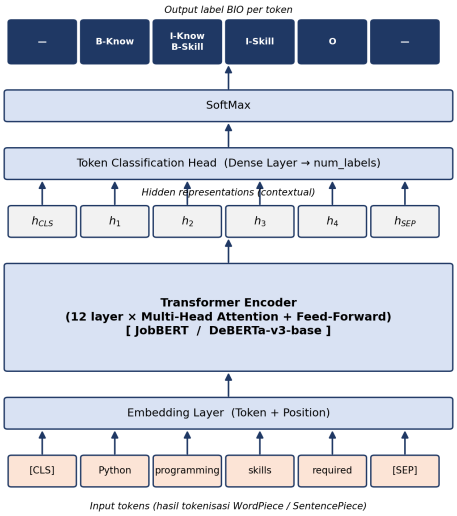

Figure 3: Token-classification architecture shared by both encoders: an embedding layer, a 12-layer transformer encoder (JobBERT or DeBERTa-v3-base), and a linear classification head producing BIO labels per token.

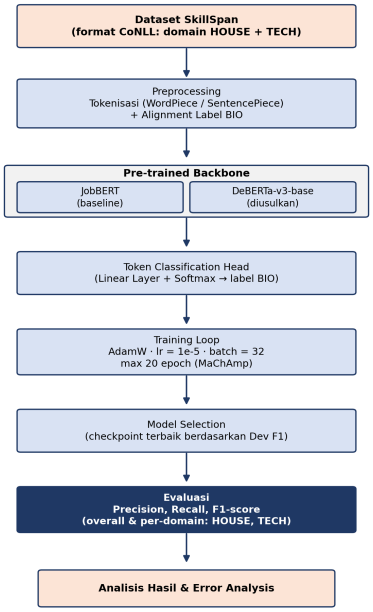

Figure 4: End-to-end experimental pipeline. The backbone-selection step (JobBERT vs. DeBERTa-v3-base) is the only point of divergence between the baseline and proposed configurations.

4.3 Training Procedure

Both models are fine-tuned using the MaChAmp multi-task framework with the AdamW optimizer, a learning rate of 1×10^{-5}, a batch size of 32, and a maximum sequence length of 256 tokens, for up to 20 epochs. The checkpoint with the highest development F1-score is selected for evaluation. Crucially, the transformer backbone is the *only* component that differs between the two runs – all other hyperparameters, data splits, and the training framework are held constant, so that any difference in performance can be attributed to the backbone itself. Figure 4 shows the end-to-end experimental pipeline, which is identical for both configurations except for the backbone-selection step. All experiments were run on a single NVIDIA T4 GPU via Google Colab.

4.4 Evaluation Metrics

Following Zhang et al. (2022), we report micro-averaged precision, recall, and F1-score over predicted Skill and Knowledge spans:

$$P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN}, \quad F_1 = \frac{2PR}{P+R} \tag{5}$$

where TP, FP, and FN denote true positives, false positives, and false negatives at the span level. We report F1 overall (combining both domains) and separately for the HOUSE and TECH domains.

5 Preliminary Results

Table 2 summarizes the development-set F1-scores. Our reproduction of the JobBERT baseline reaches 57.98% Dev F1, slightly above the 56.64% originally reported by Zhang et al. (2022), which is within the range of variation expected from differences in software versions and random seeds. Replacing only the backbone with DeBERTa-v3-base raises Dev F1 to **60.73%** – an absolute improvement of **+4.09** percentage points over the originally reported baseline, and **+2.75** percentage points over our own reproduction.

Table 2: Development F1-score comparison.

Model	Dev F1 (%)
JobBERT (original paper)	56.64
JobBERT (our reproduction)	57.98
DeBERTa-v3-base (proposed)	**60.73**

Per-domain results (RQ2). The improvement from DeBERTa-v3-base is consistent across both domains: F1 reaches **60.06%** on HOUSE and **61.38%** on TECH, both above the overall JobBERT baseline. This consistency suggests that the gains are not driven by a single domain, although the slightly higher TECH score may reflect closer alignment between DeBERTa-v3's pre-training corpus and technology-related vocabulary – a hypothesis we plan to examine further.

Convergence. The best checkpoints were obtained at epoch 19 (DeBERTa-v3-base) and epoch 18 (JobBERT, reproduction), indicating that both models converge within a comparable number of epochs under identical training hyperparameters. The observed improvement therefore reflects representational quality rather than additional training time.

Discussion. These preliminary results provide an affirmative, single-variable answer to RQ1 and RQ2: substituting DeBERTa-v3-base for JobBERT, with no other change to the pipeline, yields a substantial and domain-consistent improvement in skill-extraction F1 on SkillSpan. This is consistent with DeBERTa-v3's reported gains on other NER benchmarks (He et al., 2023; Zaratiana et al., 2023) and supports the hypothesis that its disentangled attention and RTD/GDES pre-training transfer effectively to

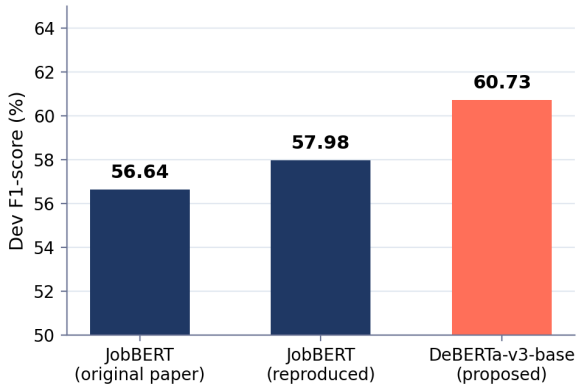

Figure 5: Development F1-score comparison between the original JobBERT baseline, our reproduction, and the proposed DeBERTa-v3-base model.

the job-posting domain. RQ3 – a finer-grained analysis of error types for hard-skill versus soft-skill spans – is the subject of ongoing work, described in Section 6.

6 Conclusion and Future Work

This paper presented a controlled, single-variable comparison in which the JobBERT backbone of the SkillSpan pipeline was replaced with `microsoft/deberta-v3-base`, with all other components – data splits, preprocessing, training hyperparameters, and evaluation protocol – held constant. DeBERTa-v3-base achieved 60.73% Dev F1, a +4.09 percentage-point absolute improvement over the originally reported JobBERT baseline (56.64%) and +2.75 points over our own reproduction (57.98%), with consistent gains on both the HOUSE (60.06%) and TECH (61.38%) domains. These results indicate that DeBERTa-v3's architectural improvements – disentangled attention together with ELECTRA-style RTD and GDES pre-training – transfer effectively to the skill-extraction task, offering a favorable performance–efficiency trade-off compared to recently explored LLM-based approaches (Nguyen et al., 2024).

This work has several limitations. The evaluation is currently restricted to the English-language HOUSE and TECH domains of SkillSpan, and compares only one alternative backbone (DeBERTa-v3-base) against JobBERT. The results reported here are preliminary outputs of an ongoing thesis project.

Future work will focus on (1) a detailed error analysis comparing hard-skill and soft-skill span predictions between JobBERT and DeBERTa-v3-base (RQ3), including qualitative examination of false positives and false negatives; (2) evaluating additional backbones, such as ESCOXLM-R (Zhang et al., 2023) and DeBERTa-v3-based generalist NER models such as GLiNER (Zaratiana et al., 2023); and (3) assessing cross-domain generalization beyond HOUSE and TECH. The complete analysis, together with the final implementation and configuration details, will be presented in the full thesis.

Reproducibility

All experiments use publicly available datasets and pretrained models (`jjzha/jobbert-base-cased` and `microsoft/deberta-v3-base`) and the publicly available SkillSpan repository and MaChAmp training framework. Hyperparameters are reported in full in Section 4.3 to facilitate reproduction.

References

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Cao, Y. and Zhang, L. (2021). Understanding social requirements for data analyst jobs: A deep learning approach. *Journal of Information Science*, 47(5):621–635.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations (ICLR)*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186. Association for Computational Linguistics.

Gugnani, A. and Misra, H. (2020). Implicit skill extraction from job postings using neural representations and taxonomy matching. In *Proceedings of RecSys-HR 2020*.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL 2020*, pages 8342–8360.

He, P., Gao, J., and Chen, W. (2023). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations (ICLR)*.

He, P., Liu, X., Gao, J., and Chen, W. (2021). Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations (ICLR)*.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

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

Lafferty, J. D., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 282–289.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT 2016*, pages 260–270.

Nguyen, M. et al. (2024). Large language models for skill extraction from job postings.

Sayfullina, L., Malmi, E., Liao, Y., and Jung, A. (2018). Learning representations for soft skill matching. In *Proceedings of the Analysis of and Mining Unstructured Big Data Using Semi-Structured Representations Workshop*. Springer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Xu, H. et al. (2023). A comprehensive survey on skill extraction from text. *ACM Computing Surveys*, 56(3):1–35.

Zaratiana, U., Tomeh, N., Holat, P., and Charnois, T. (2023). Gliner: Generalist model for named entity recognition using bidirectional transformer.

Zhang, M., Jensen, K. N., Sonniks, S., and Plank, B. (2022). Skillspan: Hard and soft skill extraction from english job postings. In *Proceedings of NAACL-HLT 2022*, pages 4962–4984. Association for Computational Linguistics.

Zhang, M., van der Goot, R., and Plank, B. (2023). Escoxlm-r: Multilingual taxonomy-driven pre-training for the job market domain. In *Proceedings of ACL 2023*, pages 11871–11890.

Zhao, M., Javed, F., Jacob, F., and McNair, M. (2015). Skill: A system for skill identification and normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.