# Entity Linking in the Job Market Domain

**Mike Zhang**[🧭,🤖] **Rob van der Goot**[🧭,🤖] **Barbara Plank**[🧭,▲,🚶]

[🧭]Department of Computer Science, IT University of Copenhagen, Denmark
[🤖]Pioneer Centre for Artificial Intelligence, Copenhagen, Denmark
[▲]MaiNLP, Center for Information and Language Processing, LMU Munich, Germany
[🚶]Munich Center for Machine Learning (MCML), Munich, Germany
mikejj.zhang@gmail.com

## Abstract

In Natural Language Processing, entity linking (EL) has centered around Wikipedia, but remains underexplored for the job market domain. Disambiguating skill mentions can help us to get insight into the labor market demands. In this work, we are the first to explore EL in this domain, specifically targeting the linkage of occupational skills to the ESCO taxonomy (le Vrang et al., 2014). Previous efforts linked coarse-grained (full) sentences to a corresponding ESCO skill. In this work, we link more fine-grained span-level mentions of skills. We tune two high-performing neural EL models, a bi-encoder (Wu et al., 2020) and an autoregressive model (Cao et al., 2021), on a synthetically generated mention–skill pair dataset and evaluate them on a human-annotated skill-linking benchmark. Our findings reveal that both models are capable of linking implicit mentions of skills to their correct taxonomy counterparts. Empirically, BLINK outperforms GENRE in strict evaluation, but GENRE performs better in loose evaluation (accuracy@$k$).[1]

## 1 Introduction

Labor market dynamics, influenced by technological changes, migration, and digitization, have led to the availability of job descriptions (JD) on platforms to attract qualified candidates (Brynjolfsson and McAfee, 2011, 2014; Balog et al., 2012). It is important to extract and link surface form skills to a unique taxonomy entry, allowing us to quantify the current labor market dynamics and determine the demands and needs. We attempt to tackle the problem of *entity linking* (EL) in the job market domain, specifically the linking of fine-grained span-level skill mentions to a specific taxonomy entry.

Generally, EL is the task of linking mentions of entities in unstructured text documents to their respective unique entities in a knowledge base (KB), most commonly Wikipedia (He et al., 2013). Recent models address this problem by producing entity representations from a (sub)set of KB information, e.g., entity descriptions (Logeswaran et al., 2019; Wu et al., 2020), fine-grained entity types (Raiman and Raiman, 2018; Onoe and Durrett, 2020; Ayoola et al., 2022), or generation of the input text autoregressively (Cao et al., 2021, 2022).

For skill linking specifically, we use the European Skills, Competences, Qualifications and Occupations (ESCO; le Vrang et al., 2014) taxonomy due to its comprehensiveness. Previous work classified spans to its taxonomy code via multi-class classification (Zhang et al., 2022b) without surrounding context and neither the full breadth of ESCO. Gnehm et al. (2022) approaches it as a sequence labeling task, but only uses more coarse-grained ESCO concepts, and not the full taxonomy. Last, others attempt to match the full sentence to their respective taxonomy title (Decorte et al., 2022, 2023; Clavié and Soulié, 2023).

The latter comes with a limitation: The taxonomy title does not indicate which subspan in the sentence it points to, without an exact match. We define this as an *implicit* skill, where mentions (spans) in the sentence do not have an exact string match with a skill in the ESCO taxonomy. The differences can range from single tokens to entire phrases. For example, we can link "being able to work together" to "plan teamwork".[2] If we know the exact span, this implicit skill can be added to the taxonomy as an alternative choice for the surface skill. As a result, this gives us a more nuanced view of the labor market skill demands. Therefore, we attempt to train models to the linking of both implicit and explicit skill mentions.

**Contributions.** Our findings can be summarized as follows: ① We pose the task of skill linking as an entity linking problem, showing promising

[1] The source code and data can be found at https://github.com/mainlp/el_esco

[2] See example here: https://t.ly/3VUJG.

*Findings of the Association for Computational Linguistics: EACL 2024*, pages 410–419
March 17-22, 2024 ©2024 Association for Computational Linguistics

| | Instances | Unique Titles | UNK |
|---|---|---|---|
| Train | 123,619 | 12,984 | 14,641 |
| Dev. | 480 | 149 | 233 |
| Test | 1,824 | 455 | 813 |

Table 1: **Data Statistics.** Data distribution of train, dev, and test splits. UNK indicates skills mentions that are not linked to a corresponding taxonomy title.

results of linking with two entity linking systems. ② We present a qualitative analysis showing that the model successfully links implicit skills to their respective skill entry in ESCO.

## 2 Methodology

**Definition.** In EL, we process the input document $\mathcal{D} = \{w_1, \ldots, w_r\}$, a collection of entity mentions denoted as $\mathcal{MD} = \{m_1, \ldots, m_n\}$, and a KB, ESCO in our case: $\mathcal{E} = \{e_1, \ldots, e_{13890}, \text{UNK}\}$. The objective of an EL model is to generate a list of mention-entity pairs $\{(m_i, e_i)\}_{i=1}^{n}$, where each entity $e$ corresponds to an entry in a KB. We assume that both the titles and descriptions of the entities are available, which is a common scenario in EL research (Ganea and Hofmann, 2017; Logeswaran et al., 2019; Wu et al., 2020). We also assume that each mention in the document has a corresponding valid gold entity present in the knowledge base, including UNK. This scenario is typically referred to as "in-KB evaluation". Similar to prior research efforts (Logeswaran et al., 2019; Wu et al., 2020), we also presuppose that the mentions within the document have already been tagged.

**Data.** We use ESCO titles as ground truth labels, containing 13,890 skills.[3] Table 1 presents the train, dev, and test data in our experiments. We leverage the train set introduced by Decorte et al. (2023)[4] along with the dev and test sets provided in Decorte et al. (2022).[5] The train set is synthetically generated by Decorte et al. (2023) with the gpt-3.5-turbo-0301 model (OpenAI, 2023). Specifically, this involves taking each skill from ESCO and prompting the model to generate sentences resembling JD sentences that require that particular skill. The dev and test splits, conversely, are derived from actual job advertisements sourced from the study by Zhang et al. (2022a). These JDs are annotated with spans corresponding to specific skills, and these spans have subsequently been manually linked to ESCO, as described in the work of Decorte et al. (2022). In cases where skills cannot be linked, two labels are used, namely UNDERSPECIFIED and LABEL NOT PRESENT. For the sake of uniformity, we map both of these labels to a generic UNK tag. We used several heuristics based on Levenshtein distance and sentence similarity to find the exact subspans if it exceeds certain thresholds, otherwise, it is UNK. This process is outlined in Appendix A. In addition, some data examples can be found in Appendix B. The number of UNKs in the data is also in Table 1. During inference, the UNK title is a prediction option for the models.

**Models.** We use two EL models, selected for their robust performance in EL on Wikipedia. [6]

**BLINK (Wu et al., 2020).** BLINK uses a bi-encoder architecture based on BERT (Devlin et al., 2019), for modeling pairs of mentions and entities. The model processes two inputs:

$$\texttt{[CLS]}\ \text{ctxt}_\text{l}\ \texttt{[S]}\ \text{mention}\ \texttt{[E]}\ \text{ctxt}_\text{r}\ \texttt{[SEP]}$$

Where "mention", "ctxt$_\text{l}$", and "ctxt$_\text{r}$" corresponds to the wordpiece tokens of the mention, the left context, and the right context. The mention is denoted by special tokens [S] and [E]. The entity and its description are structured as follows:

$$\texttt{[CLS]}\ \text{title}\ \texttt{[ENT]}\ \text{description}\ \texttt{[SEP]}$$

Here, "title" and "description" represent the wordpiece tokens of the skills' title and description, respectively. [ENT] is a special token to separate the two representations. We train the model to maximize the dot product of the [CLS] representation of the two inputs, for the correct skill in comparison to skills within the same batch. For each training pair $(m_i, e_i)$, the loss is computed as $\mathcal{L}(m_i, e_i) = -\,\mathrm{s}(m_i, e_i) + \log \sum_{j=1}^{B} \exp\left(\mathrm{s}(m_i, e_j)\right)$, where the objective is to minimize the distance between $m_i$ and $e_i$ while encouraging the model to assign a higher score to the correct pair and lower scores to randomly sampled incorrect pairs. Hard negatives are also used during training, these are obtained by finding the top 10 predicted skills for each training example. These extra hard negatives are added to the random in-batch negatives.

[3] Per version 1.1.1, accessed on 01 August 2023.

[4] https://t.ly/edqkp

[5] https://t.ly/LcqQ7

[6] For the hyperparameter setups, we refer to Appendix C.

| | Train Source | Acc@1 | Acc@4 | Acc@8 | Acc@16 | Acc@32 |
|---|---|---|---|---|---|---|
| Random | | 0.22±0.00 | 0.88±0.00 | 1.76±0.00 | 3.52±0.00 | 7.04±0.00 |
| TF-IDF | | 2.25±0.00 | | | | |
| BLINK (`bert-base`) | ESCO | 12.74±0.49 | 22.81±0.79 | 27.70±0.82 | 32.44±1.33 | 36.46±1.07 |
| BLINK (`bert-large`) | ESCO | 12.77±0.94 | 22.58±1.47 | 27.24±1.23 | 31.75±0.89 | 36.10±1.28 |
| BLINK (`bert-large`) | Wiki (0-shot) | 23.30±0.00 | **32.89±0.00** | **38.16±0.00** | 42.60±0.00 | 45.56±0.00 |
| BLINK (`bert-large`) | Wiki + ESCO | **23.55±0.14** | 32.63±0.16 | 37.38±0.09 | **43.25±0.13** | **48.98±0.21** |
| GENRE (`bart-base`) | ESCO | 1.47±0.05 | 4.84±1.74 | 10.46±6.81 | 11.30±4.18 | 15.51±4.62 |
| GENRE (`bart-large`) | ESCO | 2.33±0.44 | 5.74±1.43 | 8.18±2.21 | 11.13±2.42 | 15.26±2.66 |
| GENRE (`bart-large`) | Wiki (0-shot) | 6.91±0.00 | 12.34±0.00 | 15.52±0.00 | 21.60±0.00 | 33.17±0.00 |
| GENRE (`bart-large`) | Wiki + ESCO | **11.48±0.41** | **21.26±0.43** | **27.40±0.78** | **37.21±0.69** | **49.78±1.05** |

Table 2: **Skill Linking Results.** We show the results of the various models used. There are two `base` and four `large` models. Training sources are either ESCO or a combination of Wikipedia and ESCO. The results are the average and standard deviation over five seeds. For the 0-shot setup, we apply the fine-tuned models from the work of Wu et al. (2020) and Cao et al. (2021) to the ESCO test set once. We have a random and a TF-IDF-based baseline.

**GENRE (Cao et al., 2021).** GENRE formulates EL as a retrieval problem using a sequence-to-sequence model based on BART (Lewis et al., 2020). This model generates textual entity identifiers (i.e., skill titles) and ranks each entity $e \in \mathcal{E}$ using an autoregressive approach: $\mathrm{s}(e \mid x) = p_\theta(y \mid x) = \prod_{i=1}^{N} p_\theta\left(y_i \mid y_{<i}, x\right)$, where $y$ represents the set of $N$ tokens in the identifier of entity $e$ (i.e., entity tile), and $\theta$ denotes the model parameters. During decoding, the model uses a constrained beam search to ensure the generation of valid identifiers (i.e., only producing valid titles that exist within the KB, including `UNK`).

**Setup.** We train a total of six models: for BLINK, these are BERT$_{\text{base}}$ and BERT$_{\text{large}}$ (uncased; Devlin et al., 2019) trained on ESCO, and another large version trained on Wikipedia and ESCO sequentially. GENRE has the same setup, but then with BART (Lewis et al., 2020). Additionally, we apply the released models from both BLINK and GENRE (large, trained on Wikipedia) in a zero-shot manner and evaluate their performance. The reason we use Wikipedia-based models is that we hypothesize this is due to many skills in ESCO also having corresponding Wikipedia pages (e.g., Python[7] or teamwork[8]), thus could potentially help linking. Next, to address unknown entities (`UNK`), we include them as possible label outputs.

For evaluation, we assess the accuracy of generated mention-entity pairs in comparison to the ground truth. Here, we use the evaluation metric Accuracy@$k$, following prior research (Logeswaran et al., 2019; Wu et al., 2020; Zaporojets et al., 2022). We calculate the correctness between mentions and entities in the KB as the sum of correct hits or true positives (TP) if the ground truth for instance $i$ is in the top-$k$ predictions, formally:

$$\text{Accuracy@}k = \frac{1}{n}\sum_{i=1}^{n} \text{TP in top-}k \text{ for instance } i. \tag{1}$$

## 3 Results

Table 2 presents the results. Each model is trained for five seeds, and we report the average and standard deviation. We make use of a random and TF-IDF-based baseline.

Firstly, we observe that the strict linking performance (i.e., Acc@1) is rather modest for both BLINK and GENRE. But most models outperform the baselines. Notably, the top-performing models in this context are the BERT$_{\text{large}}$ and BART$_{\text{large}}$ models, which were further fine-tuned from Wikipedia EL with ESCO. As expected, scores improve considerably as we increase the value of $k$. Secondly, for both BLINK and GENRE, model size seems not to have a substantial impact when trained only on ESCO. Specifically for BLINK, the performance remains consistent for Acc@1 and exhibits only a slight decline as we relax the number of candidates for performance evaluation. For GENRE, the observed trend remains largely unchanged, even with a larger $k$.

[7] `https://en.wikipedia.org/wiki/Python_(programming_language)`

[8] `https://en.wikipedia.org/wiki/Teamwork`

| Mention | BLINK | GENRE |
|---|---|---|
| ① Work in a way that is **patient-centred** and inclusive. | person centred care (K0913) | work in an organised manner (T) |
| ② You can **ride a bike**. | sell bicycles (S1.6.1) | drive two-wheeled vehicles (S8.2.2) |
| ③ It is expected that you are a super user of the **MS office tools**. | use Microsoft Office (S5.6.1) | tools for software configuration management (0613) |
| ④ **Picking and packing**. | carry out specialised packing for customers (S6.1.3) | perform loading and unloading operations (S6.2.1) |
| ⑤ You are expected to be able to further **develop your team** - both personally and professionally. GOLD: **manage a team** (S4.8.1) | manage personal professional development (S1.14.1) | shape organisational teams based on competencies (S4.6.0) |
| ⑥ Our games are developed using Unity so we expect all our programmers to have solid knowledge of mobile game development in Unity3D and **C#**. | C# (K0613) | C# (K0613) |

Table 3: We show six qualitative examples. The mention is indicated with purple and we show the predictions ($\mathbf{k=1}$) of BLINK and GENRE. Green predictions mean correct, and red indicates wrong linking with respect to the ground truth. We also show the ESCO ID, indicating the differences in concepts. The results show successful linking of implicit mentions of skills. In example (5), we show how the linked results are still valid while being different concepts. However, evaluation does not count it as a correct hit.

Remarkably, the zero-shot setup performance of both BLINK and GENRE, when trained on Wikipedia, surpasses that of models trained solely on ESCO. For Wikipedia-based evaluation, GENRE usually outperforms BLINK. We notice the opposite in this case. For BLINK, this improvement is approximately 11 accuracy points for $k = 1$. Meanwhile, for GENRE, we observe an increase of roughly 9 accuracy points when trained on both Wikipedia and ESCO. This trend persists for a larger $k$, reaching up to a 12.5 accuracy point improvement for BLINK and a 34 accuracy point improvement for GENRE in the case of Acc@32. Furthermore, we show that further fine-tuning the Wikipedia-trained models on ESCO contributes to an improved EL performance at $k = \{1, 16, 32\}$ for both models. We confirm our hypothesis that Wikipedia has concepts that are also in ESCO, this gives the model strong prior knowledge of skills.

For UNK-specific results, we refer to Appendix D. Additionally, we show a direct comparison to previous work in Appendix E.

## 4 Discussion

**Qualitative Analysis.** We manually inspected a subset of the predictions. We present qualitative examples in Table 3. We found the following trends upon inspection:

- The EL models exhibit success in linking implicit and explicit mentions to their respective taxonomy titles (e.g., ①, ②, ④, ⑥).
- In cases of hard skills (③, ⑥), BLINK correctly matches “MS office tools” to “using Microsoft Office”, which is not an exact match. Both models predict the explicit mention “C#” correctly to the C# taxonomy title.
- We found that the models predict paraphrased versions of skills that could also be considered correct (④, ⑤), even being entirely different concepts (i.e., different ESCO IDs).

**Evaluation Limitation.** We qualitatively demonstrate the linking of skills that are implicit and/or valid. Empirically, we observe that the strict linking of skills leads to an underestimation of model performance. We believe this limitation is rooted in evaluation. In train, dev, and test, there is only *one* correct gold label. We reciprocate the findings by Li et al. (2020), where they found that a large number of predictions are “technically correct” but limitations in Wikipedia-based evaluation falsely penalized their model (i.e., a more or less precise version of the same entity). Especially ⑤ in Table 3 shows this challenge for ESCO, we can consider multiple links to be correct for a mention given a particular context. This highlights the need for appropriate EL evaluation sets, not only for ESCO, but for EL in general.

## 5 Conclusion

We present entity linking in the job market domain, using two existing high-performing neural models. We demonstrate that the bi-encoder architecture of BLINK is more suited to the job market domain compared to the autoregressive GENRE model. While strict linking results favor BLINK over GENRE, if we relax the number of candidates, we observe that GENRE performs slightly better. From a qualitative perspective, the performance of strict linking results is modest due to limitations in the evaluation set, which considers only one skill correct per mention. However, upon examining the predictions, we identify valid links, suggesting the possibility of multiple correct links for a particular mention, highlighting the need for more comprehensive evaluation. We hope this work sparks interest in entity linking within the job market domain.

## Limitations

In the context of EL for ESCO, our approach has several limitations. Firstly, it only supports English, and might not generalize to other languages. However, several works are working on multilingual entity linking (e.g., Botha et al., 2020; De Cao et al., 2022) and ESCO itself consists of 28 European languages. This work could be extended by supporting it for more languages.

Secondly, our EL model is trained on synthetic training data, which may not fully capture the intricacies and variations present in real-world documents. The use of synthetic data could limit its performance on actual, real JD texts. Nevertheless, we have human-annotated evaluation data.

Moreover, in our evaluation process, we use only one gold-standard ESCO title as the correct answer. This approach may not adequately represent a real-world scenario, where multiple ESCO titles could be correct as shown in Table 3.

In Table 2, we show that providing in-domain data for continuous pre-training shows larger improvements for GENRE than for BLINK. We did not conduct a detailed analysis on the underlying reasons for these positive variations.

## Acknowledgements

We thank the MaiNLP and NLPnorth group for feedback on an earlier version of this paper, and the reviewers for their insightful comments. This research is supported by the Independent Research Fund Denmark (DFF) grant 9131-00019B and in parts by ERC Consolidator Grant DIALECT 101043235.

## References

Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022. ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 209–220, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Krisztian Balog, Yi Fang, Maarten De Rijke, Pavel Serdyukov, and Luo Si. 2012. Expertise retrieval. *Foundations and Trends in Information Retrieval*, 6(2–3):127–256.

Jan A. Botha, Zifei Shan, and Daniel Gillick. 2020. Entity Linking in 100 Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, Online. Association for Computational Linguistics.

Erik Brynjolfsson and Andrew McAfee. 2011. *Race against the machine: How the digital revolution is accelerating innovation, driving productivity, and irreversibly transforming employment and the economy*. Brynjolfsson and McAfee.

Erik Brynjolfsson and Andrew McAfee. 2014. *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. WW Norton & Company.

de Nicola Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Nicola de Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2022. Multilingual autoregressive entity linking. *Transactions of the Association for Computational Linguistics*, 10:274–290.

Benjamin Clavié and Guillaume Soulié. 2023. Large language models as batteries-included zero-shot esco skills matchers. *arXiv preprint arXiv:2307.03539*.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2022. Multilingual autoregressive entity linking. *Transactions of the Association for Computational Linguistics*, 10:274–290.

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

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

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

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark. Association for Computational Linguistics.

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

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–34, Sofia, Bulgaria. Association for Computational Linguistics.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. 2020. Efficient one-pass end-to-end entity linking for questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6433–6441, Online. Association for Computational Linguistics.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, Florence, Italy. Association for Computational Linguistics.

Yasumasa Onoe and Greg Durrett. 2020. Fine-grained entity typing for domain independent entity linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8576–8583.

OpenAI. 2023. Chatgpt (march version).

Jonathan Raiman and Olivier Raiman. 2018. Deeptype: multilingual entity linking by neural type system evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

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

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Klim Zaporojets, Lucie-Aimée Kaffee, Johannes Deleu, Thomas Demeester, Chris Develder, and Isabelle Augenstein. 2022. Tempel: Linking dynamically evolving and newly emerging entities. *Advances in Neural Information Processing Systems*, 35:1850–1866.

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

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

Fangwei Zhu, Jifan Yu, Hailong Jin, Lei Hou, Juanzi Li, and Zhifang Sui. 2023. Learn to not link: Exploring NIL prediction in entity linking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10846–10860, Toronto, Canada. Association for Computational Linguistics.

**Algorithm 1:** Find the most similar n-gram to a target subspan

```
Data: $sentence$: The input sentence
$target\_subspan$: The target subspan
$threshold$: The Levenshtein distance similarity threshold
Result: $most\_similar\_ngram$: The most similar n-gram
$all\_ngrams \leftarrow$ GenerateAllNgrams($sentence$)
$filtered\_ngrams \leftarrow$ FilterNgrams($all\_ngrams$, $target\_subspan$, $threshold$)
$most\_similar\_ngram \leftarrow$ None
$max\_similarity \leftarrow 0$
for $ngram$ in $filtered\_ngrams$ do
    $subspan\_embedding \leftarrow$ EncodeWithSBERT($target\_subspan$)
    $ngram\_embedding \leftarrow$ EncodeWithSBERT($ngram$)
    $similarity \leftarrow$ CosineSimilarity($subspan\_embedding$, $ngram\_embedding$)
    if $similarity > max\_similarity$ and $similarity > 0.5$ then
        $max\_similarity \leftarrow similarity$
        $most\_similar\_ngram \leftarrow ngram$
    else
        $most\_similar\_ngram$ = UNK
return $most\_similar\_ngram$
```

## A Data Preprocessing

We outline the preprocessing steps for the training set. In Decorte et al. (2023), there are sentence–ESCO skill title pairs. The data is synthetically generated by GPT-3.5. Where for each ESCO skill title a set of 10 sentences is generated. A crucial limitation for entity linkers is that the generated sentence does not have the ESCO skill title as an exact match in the sentence, but at most slightly paraphrased. To find the most similar subspan in the sentence to the target skill, we have to apply some heuristics. In Algorithm 1, we denote our algorithm to find the most similar subspan. Our method is a brute force approach, where we create all possible n-grams until the maximum length of the sentence, and compare the target subspan against each n-gram. Based on Levenshtein distance, we filter the results, where we only take the top 80% n-grams. Then, we encode both target subspan and n-gram with SentenceBERT (Reimers and Gurevych, 2019), the similarity is based on cosine similarity. If the similarity does not exceed 0.5, the candidate subspan is `UNK` and the ESCO title will also be `UNK`, otherwise, we take the most similar n-gram. Empirically, we found that these thresholds worked best. Note that this method is not error-prone, but allows us to generate implicit and negative examples to train entity linkers. We show two qualitative examples in Figure 1 and discuss the quality in Appendix B.

## B Data Examples

We show a couple of data examples from the training (Figure 1) and development set (Figure 2). In the training examples, we show an example with a mention that is the same as the original ESCO title ("young horse training"). In addition, we have an example where there is an "implicit" mention (i.e., the mention does not exactly match with the label title). This shows that our algorithm works to an extent. For the development example, this is another implicit mention. However, these samples are human annotated. There are also quite some `UNKs` given the training data. We show that this is helping the model predict `UNK`.

## C Implementation Details

For training both BLINK[9] and GENRE,[10] we use their respective repositories. All models are trained for 10 epochs, for a batch size of 32 for training and 8 for evaluation. For both BLINK and GENRE we use 5% warmup. For the base models we use learning rate $2 \times 10^{-5}$ and for the large models we

[9] https://github.com/facebookresearch/BLINK

[10] https://github.com/facebookresearch/genre

Table 4: **UNK Linking Results.** We show the results of BLINK and GENRE predicting UNK. We use the best-performing models, based on Table 2.

| | **Train Source** | **Acc@1** | **Acc@4** | **Acc@8** | **Acc@16** | **Acc@32** |
|---|---|---|---|---|---|---|
| BLINK (bert-large) UNK | Wiki + ESCO | 1.38±0.12 | 3.32±0.22 | 4.67±0.33 | 7.68±0.42 | 10.70±0.58 |
| GENRE (bart-large) UNK | Wiki + ESCO | 1.65±0.20 | 4.99±0.50 | 9.23±0.58 | 16.01±0.48 | 24.70±2.52 |

```
{
    "context_left": "we're looking for someone who is passionate
    about",
    "context_right": "and eager to share their knowledge with
    others.",
    "mention": "young horse training",
    label_title": "young horses training",
    "label": "Principles & techniques of educating young horses
    important simple body control exercises.",
    "label_id": 2198
}
{
    "context_left": "Hands-on experience with",
    "context_right": "is a must-have qualification for this
    job.",
    "mention": "various hand-operated printing devices",
    "label_title": "types of hand-operated printing devices",
    "label": "Process of creating various types hand-operated
    printing devices, such as stamps, seals, embossing labels or
    inked pads and their applications.",
    "label_id": 10972
}
```

Figure 1: **Two Training Examples.** The training examples are in the format for BLINK, there is the left context, right context, and the mention. The label title is the ESCO skill, and the label is the description of the label title. The label ID is the ID that refers to the label title.

use $2 \times 10^{-6}$. The maximum context and candidate length is 128 for both models. Each model is trained on an NVIDIA A100 GPU with 40GBs of VRAM and an AMD Epyc 7662 CPU. The seed numbers the models are initialized with are 276800, 381552, 497646, 624189, 884832. We run all models with the maximum number of epochs (10) and select the best-performing one based on validation set performance for accuracy@1.

## D UNK Evaluation

In Table 4, we show the performance of both BLINK and GENRE on the UNK label. We use the best-performing models based on Table 2. Generally, we observe that GENRE is better in predicting UNKs than BLINK. However, the exact linking results (i.e., Acc@1) are low. This can potentially be alleviated by actively training for predicting UNKs (Zhu et al., 2023).

## E Comparison To Previous Work

We argue that an entity linking approach to match skill spans to ESCO taxonomy codes is the correct direction as it could provide more transparency in the linked span in the sentence. Consequentially, this is a more challenging setup. In Table 5, we provide a direct comparison to previous work from Decorte et al. (2023) and Clavié and Soulié (2023), where they link sentences with skills directly. For context, we are not using re-rankers as

```
{
    "context_left": "You must have an",
    "context_right": "with a high-quality mindset.",
    "mention": "analytical proactive and structured workstyle",
    "label_title": "work in an organised manner",
    "label": "Stay focused on the project at hand, at any time.
    Organise, manage time, plan, schedule and meet deadlines.",
    "label_id": 3884
}
```

Figure 2: **One Evaluation Example.** The evaluation example is in the format for BLINK, there is the left context, right context, and the mention. The label title is the ESCO skill, and the label is the description of the label title. The label ID is the ID that refers to the label title.

in the previously mentioned works.

| **Approach** | **Setup** | **MRR** |
|---|---|---|
| Decorte et al. (2023) | `SentenceBERT, sentence-level, re-ranking` | 47.8±0.0 |
| Clavié and Soulié (2023) | `GPT4, sentence-level, re-ranking` | 51.6±0.0 |
| **This work** | `BLINK, mention-level, no re-ranker` | 28.8±0.1 |
| **This work** | `GENRE, mention-level, no re-ranker` | 17.5±0.2 |

Table 5: We show a comparison to previous work, in a more challenging setup. We measure the performance in mean reciprocal rank (MRR). Note that previous work separates the splits in the ESCO matching dataset by Decorte et al. (2023), we average them here. We highlight the differences in setup, which indicates the unfair comparison. We show the results of the best-performing models (i.e., BLINK/GENRE `large` with Wikipedia and ESCO as training data).