# NNOSE: Nearest Neighbor Occupational Skill Extraction

Anonymous ACL submission


## Abstract

The labor market is changing rapidly, prompting increased interest in the automatic extraction of occupational skills from text. With the advent of English benchmark job description datasets, there is a need for systems that handle their diversity well. We tackle the complexity in occupational skill datasets tasks—combining and leveraging multiple datasets for skill extraction, to identify rarely observed skills within a dataset, and overcoming the scarcity of skills across datasets. In particular, we investigate the retrieval-augmentation of language models, employing an external datastore for retrieving similar skills in a dataset-unifying manner. Our proposed method, **N**earest **N**eighbor **O**ccupational **S**kill **E**xtraction (NNOSE) effectively leverages multiple datasets by retrieving neighboring skills from other datasets in the datastore. This improves skill extraction *without* additional fine-tuning. Crucially, we observe a performance gain in predicting infrequent patterns, with substantial gains of up to 30% span-F1 in cross-dataset settings.


## 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). JDs consist of a collection of skills that exhibit a characteristic *long-tail pattern*, where popular skills are more common while niche expertise appears less frequently across industries (Autor et al., 2003; Autor and Dorn, 2013), such as "teamwork" vs. "system design".[1] This pattern poses challenges for skill extraction (SE) and analysis, as certain skills may be underrepresented, overlooked, or emerging in JDs. This complexity makes the extraction and analysis of skills more difficult, resulting in a *sparsity of skills* in SE datasets. We tackle this by combining three different skill datasets.

To address the challenges in SE, we explore the potential of Nearest Neighbors Language Models (NNLMs; Khandelwal et al., 2020). NNLMs calculate the probability of the next token by combining a parametric language model (LM) with a distribution derived from the k-nearest context–token pairs in the datastore. This enables the storage of large amounts of training instances without the need to retrain the LM weights, improving language modeling. However, the extent to which NNLMs enhance application-specific end-task performance beyond language modeling remains relatively unexplored. Notably, NNLMs offer several advantages, as highlighted by Khandelwal et al. (2020): First, explicit memorization of the training data aids generalization. Second, a single LM can adapt to multiple domains without domain-specific training, by incorporating domain-specific data into the datastore (e.g., multiple datasets). Third, the NNLM architecture excels at predicting rare patterns, particularly the long-tail.

Therefore, we seek to answer the question: ***How effective are nearest neighbors retrieval methods for occupational skill extraction?*** Our contributions are as follows:

- To the best of our knowledge, we are the first to investigate encoder-based $k$NN retrieval by leveraging *multiple* datasets.
- Furthermore, we present a novel domain-specific RoBERTa$_{\text{base}}$-based language model, JobBERTa, tailored to the job market domain.
- We conduct an extensive analysis to show the advantages of $k$NN retrieval, in contrast to prior work that primarily focuses on hyperparameter-specific analysis.[2]

[1] Examples are from the `CEDEFOP Skill Platform`.

[2] Code: `anonymous.4open.science/r/nnose-3B3F`.

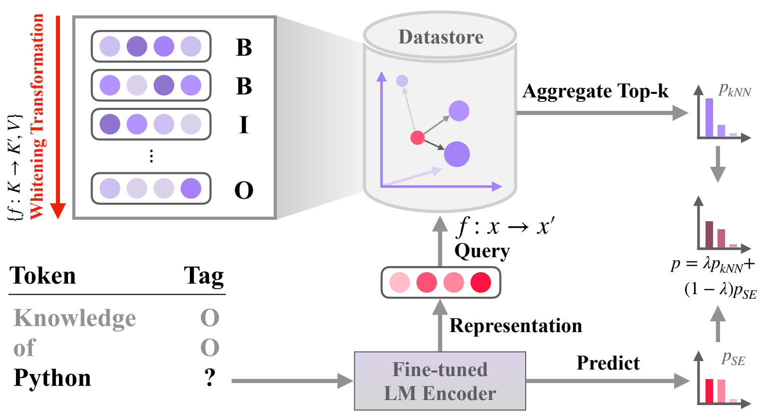


Figure 1: **Setup of NNOSE.** The datastore consists of paired contextual token representations obtained from a fine-tuned encoder and the corresponding BIO tag. We use a whitening transformation to enhance the isotropy of token representations. During inference, i.e., retrieving tokens, we use the same whitening transformation on the test token's representation to retrieve the $k$-nearest neighbors from the datastore. We interpolate the encoder and $k$NN distributions with a hyperparameter $\lambda$ as the final distribution.

## 2 Nearest Neighbor Skill Extraction

**Skill Extraction.** The task of SE is formulated as a sequence labeling problem. We define a set of job description sentences $\mathcal{X}$, where each $d \in \mathcal{X}$ represents a set of sequences with the $j^{\text{th}}$ input sequence $\mathcal{X}_d^j = \{x_1, x_2, ..., x_i\}$, with a corresponding target sequence of BIO-labels $\mathcal{Y}_d^j = \{y_1, y_2, ..., y_i\}$. The labels include "B" (beginning of a skill token), "I" (inside skill token), and "O" (any outside token). The objective is to use $\mathcal{D}$ in training a labeling algorithm that accurately predicts entity spans by assigning an output label $y_i$ to each token $x_i$.

### 2.1 NNOSE

The core idea of NNOSE is that we augment the extraction of skills during inference with a $k$NN retrieval component and a datastore consisting of context–token pairs. Figure 1 outlines our two-step approach. First, we extract skills by getting token representation $\boldsymbol{h}_i$ from $x_i$ and assign a probability distribution $p_{\text{SE}}$ for each $\boldsymbol{h}_i$ in the input sentence. Second, we use each $\boldsymbol{h}_i$ to find the most similar token representations in the datastore and get the probability distribution $p_{\text{kNN}}$, aggregated from the $k$-nearest context–token pairs. Last, we obtain the final probability distribution $p$ by interpolating between the two distributions. In addition to formalizing NNOSE, we apply the Whitening Transformation (Section 2.2) to the embeddings, an important process for $k$NN approaches as used in previous work (Su et al., 2021; Yin and Shang, 2022).

**Datastore.** The datastore $\mathcal{D}$ comprises key–value pairs $(\boldsymbol{h}_i, y_i)$, where each $\boldsymbol{h}_i$ represents the contextualized token embedding computed by a *fine-tuned* SE encoder, and $y_i \in \{$B, I, O$\}$ denotes the corresponding gold label. Typically, the datastore consists of all tokens from the training set. In contrast to the approach employed by Wang et al. (2022b) for $k$NN–NER, where they only store B and I tags in the datastore (only named entities), we also include the O-tag in the datastore. This allows us to retrieve non-named entities, which is more intuitive than assigning non-entity probability mass to the B and I tokens.

**Inference.** During inference, the NNOSE model aims to predict $y_i$ based on the contextual representation of $x_i$ (i.e., $\boldsymbol{h}_i$). This representation is used to query the datastore for $k$NN using an $L^2$ distance measure (following Khandelwal et al., 2020), denoted as $d(\cdot, \cdot)$. Once the neighbors are retrieved, the model computes a distribution over the neighbors by applying a softmax function with a temperature parameter $T$ to their negative distances (i.e., similarities). This aggregation of probability mass for each label (B, I, O) across all occurrences in the retrieved targets is represented as:

$$p_{\text{kNN}}(y_i \mid x_i) \propto \sum_{(k_i, v_i) \in \mathcal{D}} \mathbb{1}_{y=v_i} \exp\left(\frac{-d(\boldsymbol{h}_i, \boldsymbol{k})}{T}\right). \quad (1)$$

Items that do not appear in the retrieved targets have zero probability. Finally, we interpolate the nearest neighbors distribution $p_{\text{kNN}}$ with the fine-tuned model distribution $p_{\text{SE}}$ using a tuned parameter $\lambda$ to produce the final NNOSE distribution $p$:

$$\begin{aligned} p(y_i \mid x_i) = \lambda \times p_{\text{kNN}}(y_i \mid x_i) + \\ (1-\lambda) \times p_{\text{SE}}(y_i \mid x_i). \end{aligned} \quad (2)$$

### 2.2 Whitening Transformation

Several works (Li et al., 2020a; Su et al., 2021; Huang et al., 2021) note that if a set of vectors are isotropic, we can assume it is derived from the Standard Orthogonal Basis, which also indicates

| Dataset | Loc. | License | Train | Dev. | Test | $\mathcal{D}$ (tokens) |
|---|---|---|---|---|---|---|
| SKILLSPAN | * | CC-BY-4.0 | 5,866 | 3,992 | 4,680 | 86.5K |
| SAYFULLINA | UK | Unknown | 3,706 | 1,854 | 1,853 | 53.1K |
| GREEN | UK | CC-BY-4.0 | 8,670 | 963 | 336 | 209.5K |
| TOTAL | | | | | | 349.2K |

Table 1: **Dataset Statistics.** We provide statistics for all three datasets, including the location and license. Input granularity is at the token level, with performance measured in span-F1. The size of the datastore $\mathcal{D}$ is in tokens and determined by embedding tokens and their context from the training sets, resulting in approximately 350K keys. See Appendix B for examples.

that we can properly calculate the similarity between embeddings. Otherwise, if it is anisotropic, we need to transform the original sentence embedding to enforce isotrophorism, and then measure similarity. Su et al. (2021); Huang et al. (2021) applies the vector whitening approach (Koivunen and Kostinski, 1999) on BERT (Devlin et al., 2019). The Whitening Transformation (`WT`), initially employed in data preprocessing, aims to eliminate correlations among the input data features for a model. In turn, this can improve the performance of certain models that rely on uncorrelated features. Other works (Gao et al., 2019; Ethayarajh, 2019; Li et al., 2020b; Yan et al., 2021; Jiang et al., 2022b, among others) found that (frequency) biased *token* embeddings hurt final sentence representations. These works often link token embedding bias to the token embedding anisotropy and argue it is the main reason for the bias. We apply `WT` to the token embeddings like previous work for nearest neighbor retrieval (Yin and Shang, 2022). In short, `WT` transforms the mean value of the embeddings into 0 and the covariance matrix into the identity matrix, and these transformations are then applied to the original embeddings. We apply `WT` to the embeddings before putting them in the datastore and before querying the datastore. The workflow of `WT` is detailed in Appendix A.

# 3 Experimental Setup

## 3.1 Data

All datasets are in English and have different label spaces. We transform all skills to the same label space and give each token a generic tag (i.e., `B`, `I`, `O`). We give a brief description of each dataset below and Table 1 summarizes them:

**SKILLSPAN (Zhang et al., 2022a).** This job posting dataset includes annotations for skills and knowledge derived from the ESCO taxonomy. To fit our approach, we flatten the two label layers into one layer (i.e., `BIO`). The baseline is the JobBERT model, which was continuously pre-trained on a dataset of 3.2 million job posting sentences. The industries represented in the data range from tech to more labor-intensive sectors.

**SAYFULLINA (Sayfullina et al., 2018)** is used for soft skill sequence labeling. Soft skills are personal qualities that contribute to success, such as teamwork, dynamism, and independence. Data originated from the UK. This is the smallest dataset among the three, with no specified industries.

**GREEN (Green et al., 2022).** A dataset for extracting skills, qualifications, job domain, experience, and occupation labels. The dataset consists of jobs from the UK, and the industries represented include IT, finance, healthcare, and sales. This is the largest dataset among the three.

## 3.2 Models

We use 3 English-based LMs: 1 general-purpose and 2 domain-specific models. Implementation details for fine-tuning and NNOSE are in Appendix C.

**JobBERT (Zhang et al., 2022a)** is a 110M parameter BERT-based model continuously pre-trained (Gururangan et al., 2020) on 3.2M English job posting sentences. It outperforms $\text{BERT}_{\text{base}}$ on several skill-specific tasks.

**RoBERTa (Liu et al., 2019).** We also use $\text{RoBERTa}_{\text{base}}$ (123M parameters). It showed to outperform JobBERT in our initial experiments and we therefore include this model as a baseline.

**JobBERTa (Ours).** Given that RoBERTa outperformed JobBERT, we create another baseline and release a model named JobBERTa. This is a $\text{RoBERTa}_{\text{base}}$ model continuously pre-trained (Gururangan et al., 2020) on the same 3.2M JD sentences as JobBERT.

| | Setting | SKILLSPAN | SAYFULLINA | GREEN | avg. span-F1 |
|---|---|---|---|---|---|
| JobBERT (Zhang et al., 2022a) | | 60.47 | 88.16 | 42.55 | 63.73 |
| + $k$NN | {D}+WT | 61.06 $\uparrow_{0.59}$ | 88.25 $\uparrow_{0.09}$ | 43.56 $\uparrow_{1.01}$ | 64.29 $\uparrow_{0.56}$ |
| + $k$NN | ∀D+WT | 60.93 $\uparrow_{0.48}$ | 88.26 $\uparrow_{0.10}$ | 44.44 $\uparrow_{1.89}$ | 64.54 $\uparrow_{0.81}$ |
| RoBERTa (Liu et al., 2019) | | 63.88 | 91.97 | 44.49 | 66.78 |
| + $k$NN | {D}+WT | 63.57 $\downarrow_{0.31}$ | 91.97 $-_{0.00}$ | 45.02 $\uparrow_{0.53}$ | 66.85 $\uparrow_{0.07}$ |
| + $k$NN | ∀D+WT | 63.98 $\uparrow_{0.10}$ | 91.97 $-_{0.00}$ | 44.86 $\uparrow_{0.37}$ | 66.94 $\uparrow_{0.16}$ |
| JobBERTa (This work) | | 63.74 | 92.06 | 49.61 | 68.47 |
| + $k$NN | {D}+WT | 64.14 $\uparrow_{0.40}$ | 91.89 $\downarrow_{0.17}$ | 50.35 $\uparrow_{0.74}$ | 68.79 $\uparrow_{0.32}$ |
| + $k$NN | ∀D+WT | **64.24** $\uparrow_{0.50}$$^{\dagger}$ | **92.15** $\uparrow_{0.09}$ | **50.78** $\uparrow_{1.17}$$^{\dagger}$ | **69.06** $\uparrow_{0.59}$ |

Table 2: **Test Set Results.** Two settings are considered for each model based on dev. set results in Appendix D: {D} refers to the in-dataset datastore, containing keys from the specific training data, while ∀D represents a datastore with keys from all available training sets. The notation +WT indicates the application of Whitening Transformation to the keys before adding them to and querying the datastore. The performance impact of using $k$NN is indicated as ↑ (increase), ↓ (decrease), or – (no change). The best-performing setup for each dataset is highlighted. For the top-performing model (JobBERTa), $^{\dagger}$ signifies statistical significance over the baseline using a token-level McNemar test (McNemar, 1947). The avg. span-F1 performance of each model across the three datasets is displayed.

## 4 Results

We evaluate the performance of fine-tuning models enhanced with NNOSE. We consider different setups: First, we compare using the Whitening Transformation (+WT) or without. Second, we explore two datastore setups: One using an in-dataset datastore ({D}), where each respective training set is stored separately, and another where all datasets are stored in the datastore (∀D). In the latter setup, we encode all three datasets with each fine-tuned model, and each model has its own WT matrix. For example, we fine-tune a model on SKILLSPAN and encode the training set tokens of SKILLSPAN, SAYFULLINA, and GREEN to populate the datastore. From the results on the development set (Table 11, Appendix D), we observe that adding WT consistently improves performance. Therefore, we only report the span-F1 scores on each test set (Table 2) *with* WT and the average over all three datasets.

**Best Model Performance.** In Table 2, we show that the best-performing baseline model is JobBERTa, achieving more than 4 points span-F1 improvement over JobBERT and 2 points higher than RoBERTa on average. This confirms the effectiveness of DAPT in improving language models (Han and Eisenstein, 2019; Alsentzer et al., 2019; Gururangan et al., 2020; Lee et al., 2020; Nguyen et al., 2020; Zhang et al., 2022a).

**Best NNOSE Setting.** We confirm the trends from dev. on test: The largest improvements come from using the setup with WT, especially in the ∀D+WT setting. All models seem to benefit from the NNOSE setup, JobBERT and JobBERTa shows the largest improvements, with the largest gains observed in the ∀D+WT datastore setup. In summary, ∀D+WT consistently demonstrates performance enhancements across all experimental setups.

## 5 Analysis

As we store training tokens from all datasets in the datastore, we expect the model to recall a greater number of skills based on the current context during inference. In turn, this would lead to improved downstream model performance. We want to address the challenges of SE datasets by predicting long-tail patterns, and if we observe improvements in detecting unseen skills in a cross-dataset setting.

To investigate in which situations our model improves, we are analyzing the following: 1 The predictive capability of NNOSE in relation to rarely occurring skills compared to regular fine-tuning (Section 5.1). Skills exhibit varying frequencies across datasets, we categorize the skill frequencies into buckets and compare the performance between vanilla fine-tuning and the inclusion of $k$NN. 2 If NNOSE actually retrieves from other datasets when they are combined (Section 5.2), and if there is a sign of leveraging multiple datasets, then; 3 How much does NNOSE enhance performance in a cross-dataset setting (Section 5.3)? Our results indicate a large performance drop when a fine-tuned SE model, trained on one dataset, is applied to

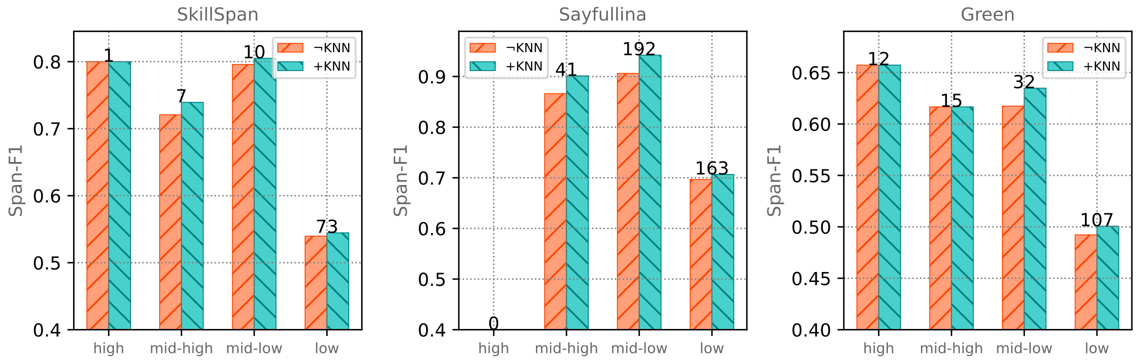


Figure 2: **Long-tail Prediction Performance.** $k$NN is based on the datastore with all the datasets. We categorize the occurrences of a skill in the test set with respect to the training set. For example, a skill in the test set occurs two times in the training set, we put this in the "low" bin. There are three frequency ranges: *high*: 10–15, *mid–high*: 7–10, *mid–low*: 4–6, *low*: 0–3. SAYFULLINA does not have any test set skills that occur more than 10 times in the training set. On top of the bars is the number of predicted skills for the test set in each bucket.

another dataset, highlighting the sparsity across datasets. We demonstrate that NNOSE helps alleviate this, both from an empirical perspective and by inspecting the prediction errors (Section 5.4).

### 5.1 Long-tail Skills Prediction

Khandelwal et al. (2020) observed that due to explicitly memorizing the training data, NNLMs effectively predict rare patterns. We analyze whether the performance of "long-tail skills" improves using NNOSE. A visualization of the long-tail distribution of skills is in Figure 8 (Appendix E).

We present the results in Figure 2. We investigate the performance of JobBERTa with and without $k$NN based on the occurrences of skills in the evaluation set relative to the train set. We count the skills in the evaluation set that occur a number of times in the training set, ranging from 0–15 occurrences and is grouped into low, mid–low, mid–high, and high–frequency bins (0–3, 4–6, 7–10, 10–15, respectively). This approach estimates the number of skills the LM recalls from the training stage.

Our findings reveal that skills with low-frequent skills are the most difficult and make up the largest bucket, and our approach is able to improve on them on all three datasets. For SKILLSPAN, we observe an improvement in the low-frequency bin, from 53.9→54.5 span-F1. Similarly, GREEN exhibits a similar trend with an improvement in the low-frequency bin (49.2→50.1). Interestingly, it also shows gains in most other frequency bins. Last, for SAYFULLINA, there is also an improvement (69.7→70.7 in the low bin). It is worth pointing out that there are many skills that fall in the low bin in SKILLSPAN and GREEN. This is exactly where NNOSE improves most for these datasets. For SAYFULLINA, we notice the largest number of predicted skills is in the mid–low bin. This is where we also see improvements for NNOSE.

### 5.2 Retrieving From All Datasets

We presented the best improvements of NNOSE in the ∀D+WT datastore in Section 4. An important question remains: Does the ∀D+WT setting retrieve from all datasets? Qualitatively, Figure 3 shows the UMAP visualization (McInnes et al., 2018) of representations stored in each ∀D+WT datastore. We mark the retrieved neighbors with orange for each downstream dev. set. In all plots, we observe that GREEN is prominent in the representation space (green), while SKILLSPAN (darkcyan) and SAYFULLINA (blue) form distinct clusters. Each plot has its own pattern: SKILLSPAN and SAYFULLINA have well-shaped clusters, while GREEN consists of one large cluster. SKILLSPAN and SAYFULLINA mostly retrieve from their own clusters. In contrast, GREEN retrieves from the entire representation space, which could explain the largest span-F1 performance gains (Table 2). This suggests that $k$NN effectively leverages multiple datasets in most cases (qualitative analysis see Appendix F).

### 5.3 Prediction of Unseen Skills

The UMAP plots in Figure 3 suggest that some datasets are closer to each other than others. To quantify this, we investigate the overlap of annotated skills between datasets and assess cross-dataset performance of NNOSE on unseen skills.

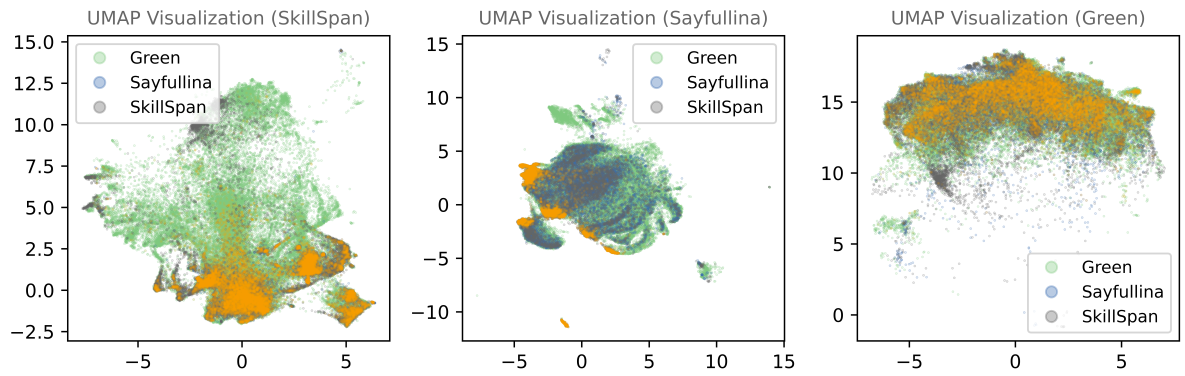


Figure 3: **UMAP Visualization of Nearest Neighbors Retrieval.** The datastore consists of the training set (+WT) of all three datasets used in this work. Each colored dot represents a non-O token from the training set. The embeddings are generated using JobBERTa. The orange shade represents the retrieved neighbors with $k = 4$ for each token that is a skill (i.e., not an O token). Note that for the middle plot, the orange shade covers the blue clusters SAYFULLINA. GREEN has the green shade and SKILLSPAN are the darkcyan colors.

| | ↓Trained on | SKILLSPAN | SAYFULLINA | GREEN |
|---|---|---|---|---|
| Vanilla | SKILLSPAN | | 18.05 | 43.17 |
| | SAYFULLINA | 9.44 | | 11.79 |
| | GREEN | 29.67 | 15.93 | |
| | ALL | 59.33 | 90.16 | 44.59 |
| +$k$NN | SKILLSPAN | | 45.86 ↑27.81 | 45.44 ↑2.27 |
| | SAYFULLINA | 26.16 ↑16.72 | | 25.38 ↑13.59 |
| | GREEN | 41.22 ↑11.55 | 46.58 ↑30.65 | |
| | ALL | 59.51 ↑0.31 | 90.33 ↑0.17 | 45.63 ↑1.04 |

Table 3: **Results of Unseen Skills based on JobBERTa (∀D+WT).** In the vanilla setting, models trained on one skill dataset are applied to another on test, showing varied performance. However, applying $k$NN improves the detection of unseen skills. Diagonal results can be found in Table 2. Refer to Table 10 for tuned hyperparameters.

**Overlap of Datasets.** We calculate the exact span overlap of skills between the training sets of the datasets using the Jaccard similarity coefficient (Jaccard, 1901): $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$, where $A$ and $B$ are sets of multi-token spans (e.g., "manage a team") from two separate training sets. The Jaccard similarity coefficients are as follows: $J$(SKILLSPAN, SAYFULLINA) = 0.35, $J$(SAYFULLINA, GREEN) = 0.10, and $J$(SKILLSPAN, GREEN) = 0.29. These Jaccard coefficients indicate overlap between unique skill spans across datasets, suggesting that NNOSE can introduce the model to new and unseen skills.

**Results.** Table 3 presents the performance of JobBERTa across datasets. For completeness, we include a baseline where JobBERTa is fine-tuned on a union of all datasets (ALL). We notice training on the union of the data never leads to the best target dataset performance. Generally, we observe that in-domain data is best, both in vanilla and NNOSE setups (diagonal in Table 3). Performance drops when a model is applied to a dataset other than the one it was trained on (off-diagonal). Using NNOSE leads to substantial improvements across the challenging off-diagonal (cross-dataset) settings, while performance remains stable within datasets. We observe the largest improvements when applied to SAYFULLINA, with *up to a 30%* increase in span-F1. This is likely due to SAYFULLINA consisting mostly of soft skills, which are less prevalent in SKILLSPAN and GREEN, making it beneficial to introduce soft skills. Conversely, when the model is trained on SAYFULLINA, the absolute improvement on SKILLSPAN is lower, indicating that skill datasets can benefit each other to different extents.

**Cross-dataset Long-tail Analysis.** Table 3 shows improvements when NNOSE is used in favor of vanilla fine-tuning. Figure 4 presents the long-tail performance analysis in the cross-dataset scenario, similar to Figure 2. We observe the largest gains with NNOSE in the low or mid–low frequency bins. However, exceptions are SKILLSPAN→GREEN and SAYFULLINA→GREEN, where most gains occur in the mid–high bin. Notably, SAYFULLINA→GREEN demonstrates higher performance with NNOSE, where all 6 skills are incorrectly predicted in the mid–high bin. An analysis of precision and recall in Table 12 (Appendix G) substantiates that the improvements are both precision and recall-based, with gains of

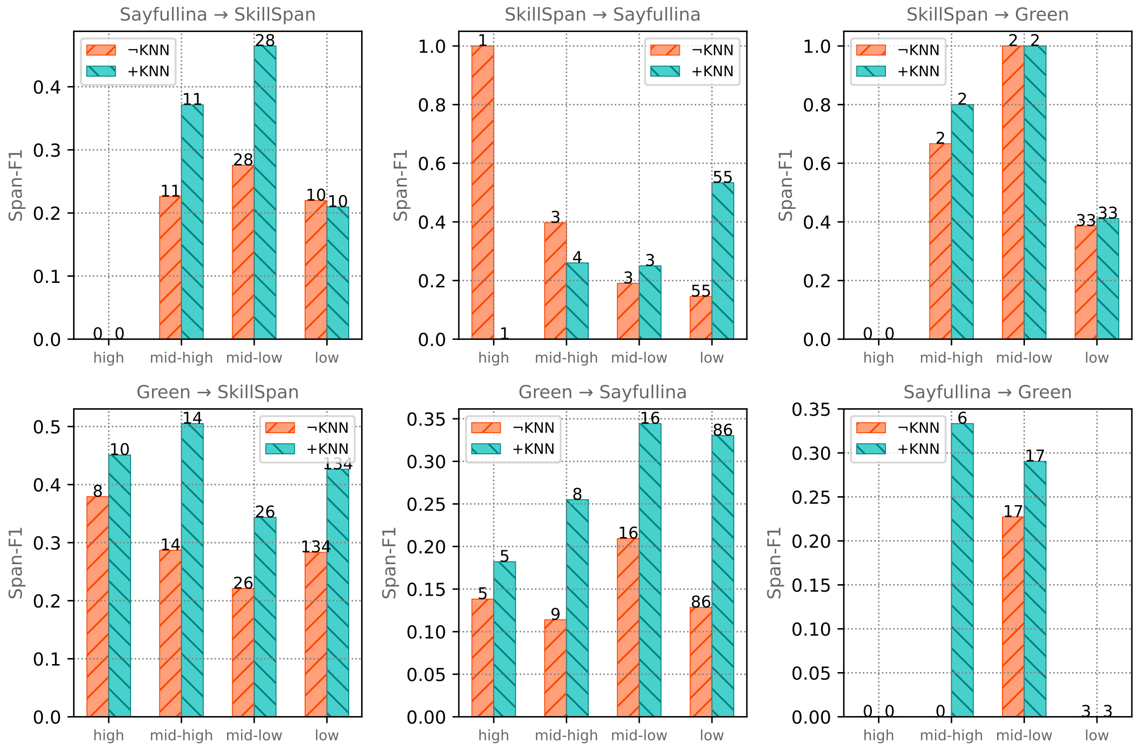


Figure 4: **Cross-dataset Long-tail Performance.** Similar to Figure 2, we plot the cross-dataset long-tail performance. NNOSE uses the datastore with all datasets. Training and evaluation data (test) are indicated in graph titles. Frequency bins are based on the training data span frequency; there are three frequency ranges: *high*: 10–15, *mid–high*: 7–10, *mid–low*: 4–6, *low*: 0–3.

up to 40 recall points and 35.4 precision points in GREEN→SAYFULLINA. There is also an improvement up to 35.5 recall points and 34.1 precision points for SKILLSPAN→SAYFULLINA. This further solidifies that memorizing tokens (i.e., storing all skills in the datastore) helps recall as mentioned in Khandelwal et al. (2020), and more importantly, highlighting the benefits of NNOSE in cross-dataset scenarios for SE.

### 5.4 Qualitative Check on Prediction Errors.

We perform a qualitative analysis on the false positives (fp) and false negatives (fn) of NNOSE predictions compared to vanilla fine-tuning for each dataset. This analysis tells us whether a prediction corresponds to an actual skill, even if it does not contribute positively to the span-F1 metric. We observe that NNOSE produces a significant number of false positives that are "similar" to genuine skills. In Table 4, for each dataset, we picked five fps and fns that represent hard, soft, and personal skills well (if applicable). We show the fps and fns for JobBERTa with NNOSE, we only show predictions that are *not* in the vainlla model predictions. In SAYFULLINA, there is only one fn. We notice from the errors, and especially the fps, that these are definitely skills, indicating the benefit of NNOSE helping to predict new skills or missed annotations.

## 6 Related Work

**Skill Extraction.** The dynamic nature of labor markets has led to an increase in tasks related to JD, including skill extraction (Kivimäki et al., 2013; Zhao et al., 2015; Sayfullina et al., 2018; Smith et al., 2019; Tamburri et al., 2020; Shi et al., 2020; Chernova, 2020; Bhola et al., 2020; Gugnani and Misra, 2020; Fareri et al., 2021; Konstantinidis et al., 2022; Zhang et al., 2022a,b,c; Green et al., 2022; Gnehm et al., 2022; Beauchemin et al., 2022; Decorte et al., 2022; Ao et al., 2023; Goyal et al., 2023; Zhang et al., 2023). These works employ methods such as sequence labeling (Sayfullina et al., 2018; Smith et al., 2019; Chernova, 2020; Zhang et al., 2022a,c), multi-label classification (Bhola et al., 2020), and graph-based methods (Shi et al., 2020; Goyal et al., 2023). Recent methodologies include domain-specific models where LMs are continuously pre-trained on unlabeled JD (Zhang et al., 2022a; Gnehm et al., 2022). However, none of these methodologies in-

| | False Positives | False Negatives |
|---|---|---|
| SKILLSPAN | cleaning<br>decisive<br>Apache Camel<br>building consumer demand for sustainable products | GCP<br>IBM MQ<br>AWS<br>budget responsible |
| SAYFULLINA | empathy<br>leadership management<br>communication<br>ability to manage and prioritise multiple assignments and tasks | leadership |
| GREEN | SQL scripting languages<br>Manage a team<br>troubleshooting activities<br>dealing with tenants | software engineering<br>development<br>DevOps<br>Cisco network administration |

Table 4: **FPs & FNs of NNOSE.** We show several examples of false positives and false negatives in each dataset. We only show the predictions of NNOSE that are *not* in the vanilla model predictions.

troduce a retrieval-augmented model like NNOSE.

**General Retrieval-augmentation.** In retrieval augmentation, LMs can utilize external modules to enhance their context-processing ability. Two approaches are commonly used: First, using a separately trained model to retrieve relevant documents from a collection. This approach is employed in open-domain question answering tasks (Petroni et al., 2021) and with specific models such as ORQA (Lee et al., 2019), REALM (Guu et al., 2020), RAG (Lewis et al., 2020), FiD (Izacard and Grave, 2021), and ATLAS (Izacard et al., 2022).

Second, previous work on explicit memorization showed promising results with a cache (Grave et al., 2017), which serves as a type of datastore. The cache contains past hidden states of the model as keys and the next word as tokens in key–value pairs. Memorization of hidden states in a datastore, involves using the $k$NN algorithm as the retriever. The first work of the $k$NN algorithm as the retrieval component was by Khandelwal et al. (2020), leading to several LM decoder-based works.

**Decoder-based Nearest Neighbor Approaches.** Decoder-based nearest neighbors approaches are primarily focused on language modeling (Khandelwal et al., 2020; He et al., 2021; Yogatama et al., 2021; Ton et al., 2022; Shi et al., 2022; Jin et al., 2022; Bhardwaj et al., 2022; Xu et al., 2023) and machine translation (Khandelwal et al., 2021; Zheng et al., 2021; Jiang et al., 2021, 2022a; Wang et al., 2022a; Martins et al., 2022a,b; Zhu et al., 2022; Du et al., 2023; Zhu et al., 2023; Min et al., 2023b,a). These approaches often prioritize efficiency and storage space reduction, as the datastores for these tasks can contain billions of tokens.

**Encoder-based Nearest Neighbor Approaches.** Encoder-based nearest neighbor approaches have been explored in tasks such as named entity recognition (Wang et al., 2022b) and emotion classification (Yin and Shang, 2022). Here, the datastores are limited to single datasets with the sentence (or token) gold label pairs. Instead, we show the potential of adding multiple datasets in the datastore.

## 7 Conclusion

We introduce NNOSE, an LM that incorporates and leverages a non-parametric datastore for nearest neighbor retrieval of skill tokens. To the best of our knowledge, we are the first to introduce the nearest neighbors retrieval component for the extraction of occupational skills. We evaluated NNOSE on three relevant skill datasets with a wide range of skills and show that NNOSE enhances the performance of all LMs used in this work *without* additionally tuning the LM parameters. Through the combination of train sets in the datastore, our analysis reveals that NNOSE effectively leverages all the datasets by retrieving tokens from each. Moreover, NNOSE not only performs well on rare skills but also enhances the performance of more frequent patterns. Lastly, we observe that our baseline models exhibit poor performance when applied in a cross-dataset setting. However, with the introduction of NNOSE, the models improve across all settings. Overall, our findings indicate that NNOSE is a promising approach for application-specific skill extraction setups and potentially helps discover skills that were missed in manual annotations.

## Limitations

We consider several limitations: One is the limited diversity of the datasets used in this work. Our study was constrained by the use of only three English datasets. By focusing solely on English data, we might have overlooked insights that exist in other languages. While these datasets were carefully selected to ensure relevance and quality, the limited scope of the data may restrict the generalizability of our findings to other SE datasets. Future research includes incorporating a wider range of datasets from diverse sources to obtain a more comprehensive understanding of the topic. Potential interesting future work should include validation on whether NNOSE works in a multilingual setting.

Another limitation is that we do skill detection and not specific labeling of the extracted spans, i.e., extracting generic `B`, `I`, `O` tags. This was to ensure that the datasets could be used in unison in the datastore. Interesting future work could extending NNOSE to include labeled skills in the datastore.

## Ethics Statement

The subject of job-related language models is a highly contentious topic, often sparking intense debates surrounding the issue of bias. We acknowledge that LMs such as JobBERTa and NNOSE possess the potential for inadvertent consequences, such as unconscious bias and dual-use when employed in the candidate selection process for specific job positions. There are research efforts to develop fairer recommender systems in the field of human resources, focusing on mitigating biases (e.g., Mujtaba and Mahapatra, 2019; Raghavan et al., 2020; Deshpande et al., 2020; Köchling and Wehner, 2020; Sánchez-Monedero et al., 2020; Wilson et al., 2021; van Els et al., 2022; Arafan et al., 2022). Nevertheless, one potential approach to alleviating such biases involves the retrieval of sparse skills for recall (e.g., this work). It is important to note, however, that we have not conducted an analysis to ascertain whether this particular method exacerbates any pre-existing forms of bias.

## References


Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Ziqiao Ao, Gergely Horváth, Chunyuan Sheng, Yifan Song, and Yutong Sun. 2023. Skill requirements in job advertisements: A comparison of skill-categorization methods based on wage regressions. *Information Processing & Management*, 60(2):103185.

Adam Mehdi Arafan, David Graus, Fernando P Santos, and Emma Beauxis-Aussalet. 2022. End-to-end bias mitigation in candidate recommender systems with fairness gates. In *Proceedings of RecSys in HR'22: The 2nd Workshop on Recommender Systems for Human Resources, in conjunction with the 16th ACM Conference on Recommender Systems*.

David H Autor and David Dorn. 2013. The growth of low-skill service jobs and the polarization of the us labor market. *American economic review*, 103(5):1553–1597.

David H Autor, Frank Levy, and Richard J Murnane. 2003. The skill content of recent technological change: An empirical exploration. *The Quarterly journal of economics*, 118(4):1279–1333.

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.

David Beauchemin, Julien Laumonier, Yvan Le Ster, and Marouane Yassine. 2022. "FIJO": a French Insurance Soft Skill Detection Dataset. In *Proceedings of the Canadian Conference on Artificial Intelligence*. Canadian Artificial Intelligence Association (CAIAC).

Rishabh Bhardwaj, George Polovets, and Monica Sunkara. 2022. Adaptation approaches for nearest neighbor language models. *ArXiv preprint*, abs/2211.07828.

Akshay Bhola, Kishaloy Halder, Animesh Prasad, and Min-Yen Kan. 2020. Retrieving skills from job descriptions: A language model based extreme multi-label classification framework. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5832–5842, Barcelona, Spain (Online). International Committee on 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.

Mariia Chernova. 2020. Occupational skills extraction with FinBERT. *Master's Thesis*.

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

Ketki V Deshpande, Shimei Pan, and James R Foulds. 2020. Mitigating demographic bias in ai-based resume filtering. In *Adjunct publication of the 28th ACM conference on user modeling, adaptation and personalization*, pages 268–275.

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.

Yichao Du, Zhirui Zhang, Bingzhe Wu, Lemao Liu, Tong Xu, and Enhong Chen. 2023. Federated Nearest Neighbor Machine Translation. In *The Eleventh International Conference on Learning Representations*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

Silvia Fareri, Nicola Melluso, Filippo Chiarello, and Gualtiero Fantoni. 2021. Skillner: Mining and mapping soft skills from any text. *Expert Systems with Applications*, 184:115544.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ann-Sophie Gnehm, Eva Bühlmann, and Simon Clematide. 2022. Evaluation of transfer learning and domain adaptation for analyzing german-speaking job advertisements. In *Proceedings of the Language Resources and Evaluation Conference*, pages 3892–3901, Marseille, France. European Language Resources Association.

Gene H Golub and Christian Reinsch. 1971. Singular value decomposition and least squares solutions. *Linear algebra*, 2:134–151.

Nidhi Goyal, Jushaan Kalra, Charu Sharma, Raghava Mutharaju, Niharika Sachdeva, and Ponnurangam Kumaraguru. 2023. JobXMLC: EXtreme multi-label classification of job skills with graph neural networks. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2181–2191, Dubrovnik, Croatia. Association for Computational Linguistics.

Edouard Grave, Armand Joulin, and Nicolas Usunier. 2017. Improving neural language models with a continuous cache. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Thomas Green, Diana Maynard, and Chenghua Lin. 2022. Development of a benchmark corpus to support entity recognition in job descriptions. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1201–1208, Marseille, France. European Language Resources Association.

Akshay Gugnani and Hemant Misra. 2020. Implicit skills extraction using document embedding and its use in job recommendation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13286–13293. AAAI Press.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. 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 4238–4248, Hong Kong, China. Association for Computational Linguistics.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient nearest neighbor language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5703–5714, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. WhiteningBERT: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021. Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *ArXiv preprint*, abs/2208.03299.

Paul Jaccard. 1901. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bull Soc Vaudoise Sci Nat*, 37:241–272.

Hui Jiang, Ziyao Lu, Fandong Meng, Chulun Zhou, Jie Zhou, Degen Huang, and Jinsong Su. 2022a. Towards robust k-nearest-neighbor machine translation. *ArXiv preprint*, abs/2210.08808.

Qingnan Jiang, Mingxuan Wang, Jun Cao, Shanbo Cheng, Shujian Huang, and Lei Li. 2021. Learning kernel-smoothed machine translation with retrieved examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7280–7290, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022b. Promptbert: Improving bert sentence embeddings with prompts. *ArXiv preprint*, abs/2201.04337.

Xuyang Jin, Tao Ge, and Furu Wei. 2022. Plug and play knowledge distillation for kNN-LM with external logits. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 463–469, Online only. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ilkka Kivimäki, Alexander Panchenko, Adrien Dessy, Dries Verdegem, Pascal Francq, Hugues Bersini, and Marco Saerens. 2013. A graph-based approach to skill extraction from text. In *Proceedings of TextGraphs-8 Graph-based Methods for Natural Language Processing*, pages 79–87, Seattle, Washington, USA. Association for Computational Linguistics.

Alina Köchling and Marius Claus Wehner. 2020. Discriminated by an algorithm: a systematic review of discrimination and fairness by algorithmic decision-making in the context of hr recruitment and hr development. *Business Research*, 13(3):795–848.

AC Koivunen and AB Kostinski. 1999. The feasibility of data whitening to improve performance of weather radar. *Journal of Applied Meteorology and Climatology*, 38(6):741–749.

Ioannis Konstantinidis, Manolis Maragoudakis, Ioannis Magnisalis, Christos Berberidis, and Vassilios Peristeras. 2022. Knowledge-driven unsupervised skills extraction for graph-based talent matching. In *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*, pages 1–7.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020a. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020b. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, and Levy. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Pedro Martins, Zita Marinho, and Andre Martins. 2022a. Efficient machine translation domain adaptation. In *Proceedings of the 1st Workshop on Semiparametric Methods in NLP: Decoupling Logic from Knowledge*, pages 23–29, Dublin, Ireland and Online. Association for Computational Linguistics.

Pedro Henrique Martins, Zita Marinho, and André FT Martins. 2022b. Chunk-based nearest neighbor machine translation. *ArXiv preprint*, abs/2205.12230.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. 2018. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Sewon Min, Suchin Gururangan, Eric Wallace, Hannaneh Hajishirzi, Noah A Smith, and Luke Zettlemoyer. 2023a. Silo language models: Isolating legal risk in a nonparametric datastore. *arXiv preprint arXiv:2308.04430*.

Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wen-tau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2023b. Nonparametric masked language modeling. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2097–2118, Toronto, Canada. Association for Computational Linguistics.

Dena F Mujtaba and Nihar R Mahapatra. 2019. Ethical considerations in ai-based recruitment. In *2019 IEEE International Symposium on Technology and Society (ISTAS)*, pages 1–7. IEEE.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. 2020. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 469–481.

Javier Sánchez-Monedero, Lina Dencik, and Lilian Edwards. 2020. What does it mean to 'solve' the problem of discrimination in hiring? social, technical and legal perspectives from the uk on automated hiring systems. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 458–468.

Luiza Sayfullina, Eric Malmi, and Juho Kannala. 2018. Learning representations for soft skill matching. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 141–152.

Baoxu Shi, Jaewon Yang, Feng Guo, and Qi He. 2020. Salience and market-aware skill extraction for job targeting. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 2871–2879. ACM.

Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. 2022. Nearest neighbor zero-shot inference. *ArXiv preprint*, abs/2205.13792.

Ellery Smith, Martin Braschler, Andreas Weiler, and Thomas Haberthuer. 2019. Syntax-based skill extractor for job advertisements. In *2019 6th Swiss Conference on Data Science (SDS)*, pages 80–81. IEEE.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *ArXiv preprint*, abs/2103.15316.

Damian A Tamburri, Willem-Jan Van Den Heuvel, and Martin Garriga. 2020. 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)*, pages 391–394. IEEE.

Jean-Francois Ton, Walter Talbott, Shuangfei Zhai, and Joshua M. Susskind. 2022. Regularized training of nearest neighbor language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 25–30, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Sarah-Jane van Els, David Graus, and Emma Beauxis-Aussalet. 2022. Improving fairness assessments with synthetic data: a practical use case with a recommender system for human resources. In *Proceedings of The First International Workshop on Computational Jobs Marketplace: A WSDM 2022 Workshop*.

Dexin Wang, Kai Fan, Boxing Chen, and Deyi Xiong. 2022a. Efficient cluster-based $k$-nearest-neighbor machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2175–2187, Dublin, Ireland. Association for Computational Linguistics.

Shuhe Wang, Xiaoya Li, Yuxian Meng, Tianwei Zhang, Rongbin Ouyang, Jiwei Li, and Guoyin Wang. 2022b. KNN-NER: Named entity recognition with nearest neighbor search. *ArXiv preprint*, abs/2203.17103.

Christo Wilson, Avijit Ghosh, Shan Jiang, Alan Mislove, Lewis Baker, Janelle Szary, Kelly Trindel, and Frida Polli. 2021. Building and auditing fair algorithms: A case study in candidate screening. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 666–677.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Frank F Xu, Uri Alon, and Graham Neubig. 2023. Why do nearest neighbor language models work? *ArXiv preprint*, abs/2301.02828.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.

Wenbiao Yin and Lin Shang. 2022. Efficient nearest neighbor emotion classification with BERT-whitening. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4738–4745, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Dani Yogatama, Cyprien de Masson d'Autume, and Lingpeng Kong. 2021. Adaptive semiparametric language models. *Transactions of the Association for Computational Linguistics*, 9:362–373.

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 Language Resources and Evaluation Conference*, pages 436–447, Marseille, France. European Language Resources Association.

Mike Zhang, Kristian Nørgaard Jensen, Rob van der Goot, and Barbara Plank. 2022c. Skill extraction from job postings using weak supervision. In *Proceedings of RecSys in HR'22: The 2nd Workshop on Recommender Systems for Human Resources, in conjunction with the 16th ACM Conference on Recommender Systems*.

Mike Zhang, Rob van der Goot, and Barbara Plank. 2023. ESCOXLM-R: Multilingual taxonomy-driven pre-training for the job market domain. *ArXiv preprint*, abs/2305.12092.

Meng Zhao, Faizan Javed, Ferosh Jacob, and Matt McNair. 2015. SKILL: A system for skill identification and normalization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 4012–4018. AAAI Press.

Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. 2021. Adaptive nearest neighbor machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 368–374, Online. Association for Computational Linguistics.

Wenhao Zhu, Shujian Huang, Yunzhe Lv, Xin Zheng, and Jiajun Chen. 2022. What knowledge is needed? towards explainable memory for knn-mt domain adaptation. *ArXiv preprint*, abs/2211.04052.

Wenhao Zhu, Qianfeng Zhao, Yunzhe Lv, Shujian Huang, Siheng Zhao, Sizhe Liu, and Jiajun Chen. 2023. knn-box: A unified framework for nearest neighbor generation. *ArXiv preprint*, abs/2302.13574.

## A Whitening Transformation Algorithm

**Algorithm 1:** Whitening Transformation Workflow

```
1 input: Embeddings {x_i}_{i=1}^N;
2 Compute μ = (1/N) Σ_{i=1}^N x_i and Σ of {x_i}_{i=1}^N
3 Compute U, Λ, U^⊤ = SVD(Σ)
4 Compute W = U √(Λ^{-1})
5 for i = 1, 2, ..., n do
6   |  x̃_i = (x_i − μ) W
7 end
8 return {x̃_i}_{i=1}^N;
```

We apply the whitening transformation to the query embedding and the embeddings in the datastore. We can write a set of token embeddings as a set of row vectors: $\{x_i\}_{i=1}^N$. Additionally, a linear transformation $\widetilde{x}_i = (x_i - \mu)\, W$ is applied, where $\mu = \frac{1}{N}\sum_{i=1}^N x_i$. To obtain the matrix $W$, the following steps are conducted: First, we obtain the original covariance matrix

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^\top (x_i - \mu) . \qquad (3)$$

Afterwards, we obtain the transformed covariance matrix $\widetilde{\Sigma} = W^\top \Sigma W$, where we specify $\widetilde{\Sigma} = I$. Therefore, $\Sigma = \left(W^\top\right)^{-1} W^{-1} = \left(W^{-1}\right)^\top W^{-1}$. Here, $\Sigma$ is a positive definite symmetric matrix that satisfies the following singular value decomposition (SVD; Golub and Reinsch, 1971) as indicated by Su et al. (2021): $\Sigma = U \Lambda U^\top$. $U$ is an orthogonal matrix, $\Lambda$ is a diagonal matrix, and the diagonal elements are all positive. Therefore, let $W^{-1} = \sqrt{\Lambda} U^\top$, we obtain the solution: $W = U\sqrt{\Lambda^{-1}}$. Putting it all together, as input, we have the set of embeddings $\{x_i\}_{i=1}^N$. We compute $\mu$ and $\Sigma$ of $\{x_i\}_{i=1}^N$. Then, we perform SVD on $\Sigma$ to obtain matrices $U$, $\Lambda$, and $U^\top$. Using these matrices, we calculate the transformation matrix $W$. Finally, we apply the transformation to each embedding in the set by subtracting $\mu$ and multiplying by $W$. We are left with $\widetilde{x}_i = (x_i - \mu)\, W$. Note that we do `WT` *before* we store the embedding in the datastore, and apply `WT` to the token embedding before we query the datastore.

We show the Whitening Transformation procedure in Algorithm 1. Note that Li et al. (2020a); Su et al. (2021) introduced a dimensionality reduction factor $k$ on $W$ ($W[:, : k]$). he diagonal elements in the matrix $\Lambda$ obtained from the SVD algorithm are in descending order. One can decide to keep the first $k$ columns of $W$ in line 6. This is similar to PCA (Abdi and Williams, 2010). However, empirically, we found that reducing dimensionality had a negative effect on downstream performance, thus we omit that in this implementation.

## B Data Examples

| | |
|---|---|
| SKILLSPAN | Figure 5 |
| SAYFULLINA | Figure 6 |
| GREEN | Figure 7 |

Table 5: Data example references for each dataset.

In Table 5, we refer to several listings of examples of the datasets. Notably in SKILLSPAN, the original samples contain two columns of labels. These refer to skills and knowledge. To accommodate for the approach of NNOSE, we merge the labels together and thus removing the possible nesting of skills. Zhang et al. (2022a) mentions that there is not a lot of nesting of skills. Following Zhang et al. (2022a), we prioritize the skills column when merging the labels. When there is nesting, we keep the labels of skills and remove the knowledge labels.

## C Implementation Details

**General Implementation.** We obtain all LMs from the Transformers library (Wolf et al., 2020) and implement JobBERTa using the same library. All learning rates for fine-tuning are $5 \times 10^{-5}$ using the AdamW optimizer (Loshchilov and Hutter, 2019). We use a batch size of 16 and a maximum sequence length of 128 with dynamic padding. The models are trained for 20 epochs with early stopping using a patience of 5. We implement the retrieval component using the FAISS library (Johnson et al., 2019), which is a standard for nearest neighbors retrieval-augmented methods.[3]

**JobBERTa.** We apply domain-adaptive pre-training (Gururangan et al., 2020), which involves continued self-supervised pre-training of a large LM on domain-specific text. This approach enhances the modeling of text for downstream tasks within the domain. We continue pre-training on a `roberta-base` checkpoint with 3.2M job posting

[3] `https://faiss.ai/`

```
Experience  O
in          O
working     B
on          I
a           I
cloud-based I
application I
running     O
on          O
Docker      B
.           O

A           O
degree      B
in          I
Computer    I
Science     I
or          O
related     O
fields      O
.           O
```

Figure 5: **Data Example for SkillSpan.** In SKILLSPAN, note the long skills.

```
ability     O
to          O
work        B
under       I
stress      I
condition   O

due         O
to          O
the         O
dynamic     B
nature      O
of          O
the         O
group       O
environment O
,           O
the         O
ideal       O
candidate   O
will        O
```

Figure 6: **Data Example for Sayfullina.** In SAYFULLINA, the skills are usually soft-like skills.

```
A             O
sound         O
understanding O
of            O
the           O
Care          B
Standards     I
together      O
with          O
a             O
Nursing       B
qualification I
and           O
current       O
NMC           B
registration  I
are           O
essential     O
for           O
this          O
role          O
```

Figure 7: **Data Example for Green.** There are many qualification skills (e.g., certificates).

sentences from Zhang et al. (2022a). We use a batch size of 8 and run MLM for a single epoch following Gururangan et al. (2020). The rest of the hyperparameters are set to the defaults in the Transformer library.[4]

**NNOSE Setup.** Following previous work, the keys used in NNOSE are the 768-dimensional representation logits obtained from the final layer of the LM (input to the softmax). We perform a single forward pass over the training set of each dataset to save the keys and values, i.e., the hidden representation and the corresponding gold BIO tag. The FAISS index is created using all the keys to learn 4096 cluster centroids. During inference, we retrieve $k$ neighbors. The index looks up 32 cluster centroids while searching for the nearest neighbors. For all experiments, we compute the squared Euclidean ($L^2$) distances with full precision keys. The difference in inference speed is almost negligible, with the $k$NN module taking a few extra seconds compared to regular inference. For the exact hyperparameter values, we indicate them in the next paragraph.

**Hyperparameters NNOSE.** The best-performing hyperparameters and search space can be found in Table 6, Table 7, Table 8, and Table 9. We report the $k$-nearest neighbors, $\lambda$ value, and softmax temperature $T$ for each dataset and model.

In Table 10, we show the hyperparameters for the cross-dataset analysis. In the vanilla setting, we apply the models trained on a particular skill dataset to another skill dataset, similar to transfer learning. We observe a significant discrepancy in performances cross-dataset, indicating a wide range of skills. However, when $k$NN is applied, it improves the detection of unseen skills. The datastore contains tokens from all datasets.

## D Development Set Results

We show the dev. set results in Table 11. Overall, the patterns of improvements hold across datasets and models. We base the test set result on the best-performing setups in the development set, i.e.,

[4] https://github.com/huggingface/transformers/blob/main/examples/pytorch/language-modeling/run_mlm.py

| Dataset → | | SkillSpan | Sayfullina | Green |
|---|---|---|---|---|
| JobBERT | $k$ | 4 | 4 | 16 |
| | $\lambda$ | 0.3 | 0.3 | 0.15 |
| | $T$ | 0.1 | 2.0 | 10.0 |
| RoBERTa | $k$ | 32 | 4 | 64 |
| | $\lambda$ | 0.3 | 0.3 | 0.25 |
| | $T$ | 10.0 | 0.1 | 10.0 |
| JobBERTa | $k$ | 16 | 4 | 8 |
| | $\lambda$ | 0.2 | 0.1 | 0.1 |
| | $T$ | 5.0 | 10.0 | 10.0 |
| Search Space | $k$ | {4, 8, 16, 32, 64, 128} | | |
| | $\lambda$ | {0.1, 0.15, 0.2, 0.25, ..., 0.9} | | |
| | $T$ | {0.1, 0.5, 1.0, 2.0, 3.0, 5.0, 10.0} | | |

Table 6: **Tuned Hyperparameters on Dev.** These are for $\{\mathcal{D}\}$.

| Dataset → | | SkillSpan | Sayfullina | Green |
|---|---|---|---|---|
| JobBERT | $k$ | 4 | 4 | 64 |
| | $\lambda$ | 0.35 | 0.35 | 0.4 |
| | $T$ | 2.0 | 0.1 | 5.0 |
| RoBERTa | $k$ | 32 | 4 | 16 |
| | $\lambda$ | 0.35 | 0.45 | 0.25 |
| | $T$ | 0.1 | 0.1 | 1.0 |
| JobBERTa | $k$ | 64 | 128 | 128 |
| | $\lambda$ | 0.25 | 0.35 | 0.45 |
| | $T$ | 10.0 | 0.5 | 10.0 |
| Search Space | $k$ | {4, 8, 16, 32, 64, 128} | | |
| | $\lambda$ | {0.1, 0.15, 0.2, 0.25, ..., 0.9} | | |
| | $T$ | {0.1, 0.5, 1.0, 2.0, 3.0, 5.0, 10.0} | | |

Table 7: **Tuned Hyperparameters on Dev.** These are for $\{\mathcal{D}\} + WT$.

| Dataset → | | SkillSpan | Sayfullina | Green |
|---|---|---|---|---|
| JobBERT | $k$ | 4 | 16 | 32 |
| | $\lambda$ | 0.3 | 0.25 | 0.15 |
| | $T$ | 10.0 | 5.0 | 10.0 |
| RoBERTa | $k$ | 16 | 8 | 8 |
| | $\lambda$ | 0.15 | 0.1 | 0.1 |
| | $T$ | 10.0 | 10.0 | 10.0 |
| JobBERTa | $k$ | 8 | 4 | 8 |
| | $\lambda$ | 0.2 | 0.15 | 0.1 |
| | $T$ | 0.5 | 0.1 | 10.0 |
| Search Space | $k$ | {4, 8, 16, 32, 64, 128} | | |
| | $\lambda$ | {0.1, 0.15, 0.2, 0.25, ..., 0.9} | | |
| | $T$ | {0.1, 0.5, 1.0, 2.0, 3.0, 5.0, 10.0} | | |

Table 8: **Tuned Hyperparameters on Dev.** These are for `∀D`.

| Dataset → | | SkillSpan | Sayfullina | Green |
|---|---|---|---|---|
| JobBERT | $k$ | 32 | 4 | 128 |
| | $\lambda$ | 0.3 | 0.3 | 0.4 |
| | $T$ | 1.0 | 0.5 | 2.0 |
| RoBERTa | $k$ | 128 | 128 | 64 |
| | $\lambda$ | 0.35 | 0.1 | 0.25 |
| | $T$ | 0.1 | 0.5 | 0.1 |
| JobBERTa | $k$ | 32 | 8 | 128 |
| | $\lambda$ | 0.15 | 0.3 | 0.2 |
| | $T$ | 0.1 | 0.1 | 2.0 |
| Search Space | $k$ | {4, 8, 16, 32, 64, 128} | | |
| | $\lambda$ | {0.1, 0.15, 0.2, 0.25, ..., 0.9} | | |
| | $T$ | {0.1, 0.5, 1.0, 2.0, 3.0, 5.0, 10.0)} | | |

Table 9: **Tuned Hyperparameters on Dev.** These are for `∀D+WT`.

| ↓Trained on | Hyperparams. | SkillSpan | Sayfullina | Green |
|---|---|---|---|---|
| SkillSpan | $k$ | | 16 | 32 |
| | $\lambda$ | | 0.9 | 0.7 |
| | $T$ | | 0.1 | 0.5 |
| Sayfullina | $k$ | 64 | | 32 |
| | $\lambda$ | 0.9 | | 0.8 |
| | $T$ | 0.1 | | 0.1 |
| Green | $k$ | 32 | 32 | |
| | $\lambda$ | 0.85 | 0.9 | |
| | $T$ | 0.5 | 0.1 | |
| All | $k$ | 4 | 128 | 32 |
| | $\lambda$ | 0.25 | 0.6 | 0.65 |
| | $T$ | 1.0 | 1.0 | 0.5 |
| Search Space | $k$ | {4, 8, 16, 32, 64, 128} | | |
| | $\lambda$ | {0.1, 0.15, 0.2, 0.25, ..., 0.9} | | |
| | $T$ | {0.1, 0.5, 1.0, 2.0, 3.0, 5.0, 10.0} | | |

Table 10: **Results of Unseen Skills (Development Set) based on JobBERTa.**

`{D}+WT` and `∀D+WT`.

## E Frequency Distribution of Skills

We show the skill frequency distribution of the datasets in Figure 8, as mentioned in Section 5.1. Here, we show evidence of the long-tail pattern in skills for each dataset. There is a cut-off at count 15 for Green, indicating that there are skills in the development set that occur more than 15 times.

## F Qualitative Results NNOSE

We show several qualitative results of NNOSE. In Table 13, we show a qualitative sample of using JobBERTa on SkillSpan. The current token is "IT" with gold label `O`. The language model puts 0.4 softmax probability on the tag `I`. By retrieving the nearest neighbors, the final probability mass gets shifted towards `O` with probability 0.43, which is the correct tag.

In Table 14, we show a qualitative sample of using JobBERTa on SkillSpan with multi-token annotations and how this behaves. The current skill is "coding skills" with gold labels `B` and `I` respectively. Both the model and $k$NN puts high confidence in the correct label. Note that the nearest neighbors of "coding" are quite varied, which shows the ben-

| Dataset (Dev.) → | Setting | SKILLSPAN | SAYFULLINA | GREEN | avg. Span-F1 |
|---|---|---|---|---|---|
| JobBERT (Zhang et al., 2022a) | | 61.08 | 89.26 | 37.27 | 62.54 |
| + $k$NN | `{D}` | 61.56 $\uparrow_{0.48}$ | 89.69 $\uparrow_{0.43}$ | 37.48 $\uparrow_{0.21}$ | 62.91 $\uparrow_{0.37}$ |
| + $k$NN | `{D}+WT` | 61.77 $\uparrow_{0.69}$ | 89.78 $\uparrow_{0.52}$ | 38.07 $\uparrow_{0.80}$ | 63.21 $\uparrow_{0.67}$ |
| + $k$NN | `∀D` | 61.58 $\uparrow_{0.50}$ | 89.50 $\uparrow_{0.24}$ | 37.27 $-_{0.00}$ | 62.78 $\uparrow_{0.24}$ |
| + $k$NN | `∀D+WT` | 61.50 $\uparrow_{0.42}$ | 89.37 $\uparrow_{0.11}$ | 38.19 $\uparrow_{0.92}$ | 63.02 $\uparrow_{0.48}$ |
| RoBERTa (Liu et al., 2019) | | 65.02 | 92.91 | 40.33 | 66.09 |
| + $k$NN | `{D}` | 65.36 $\uparrow_{0.34}$ | 92.76 $\downarrow_{0.15}$ | 40.53 $\uparrow_{0.20}$ | 66.22 $\uparrow_{0.13}$ |
| + $k$NN | `{D}+WT` | 65.34 $\uparrow_{0.32}$ | 93.07 $\uparrow_{0.16}$ | 41.22 $\uparrow_{0.89}$ | 66.54 $\uparrow_{0.45}$ |
| + $k$NN | `∀D` | 64.98 $\downarrow_{0.04}$ | 92.78 $\downarrow_{0.13}$ | 40.60 $\uparrow_{0.27}$ | 66.12 $\uparrow_{0.03}$ |
| + $k$NN | `∀D+WT` | 65.38 $\uparrow_{0.36}$ | 92.92 $\uparrow_{0.01}$ | 41.11 $\uparrow_{0.77}$ | 66.47 $\uparrow_{0.38}$ |
| JobBERTa (This work) | | 65.15 | 92.09 | 40.59 | 65.94 |
| + $k$NN | `{D}` | 65.25 $\uparrow_{0.10}$ | 91.99 $\downarrow_{0.10}$ | 41.31 $\uparrow_{0.72}$ | 66.18 $\uparrow_{0.24}$ |
| + $k$NN | `{D}+WT` | 65.21 $\uparrow_{0.06}$ | 92.10 $\uparrow_{0.01}$ | 41.41 $\uparrow_{0.82}$ | 66.24 $\uparrow_{0.30}$ |
| + $k$NN | `∀D` | 65.15 $-_{0.00}$ | 92.04 $\downarrow_{0.05}$ | 40.83 $\uparrow_{0.24}$ | 66.01 $\uparrow_{0.07}$ |
| + $k$NN | `∀D+WT` | 65.22 $\uparrow_{0.07}$ | 92.13 $\uparrow_{0.04}$ | 41.45 $\uparrow_{0.86}$ | 66.26 $\uparrow_{0.32}$ |

Table 11: **Development Set Results.** There are four settings for each model. `{D}`: in-dataset datastore (i.e., the datastore only contains the keys from the specific training data it is applied on). `∀D`: The datastore contains the keys from all available training datasets. $+W$: Whitening Transformation is applied to the keys before adding them to the datastore or querying the datastore. We indicate the performance increase (↑), decrease (↓), or no change (–) when using $k$NN compared to not using $k$NN. Additionally, we show the average span-F1 performance of each model across the three datasets. In the development set, it seems that an in-dataset datastore works best.

| | Vanilla | | +$k$NN | |
|---|---|---|---|---|
| Setup↓ | Precision | Recall | Precision | Recall |
| SAYFULLINA→SKILLSPAN | 10.20 | 10.50 | 37.67$\uparrow_{27.47}$ | 29.62$\uparrow_{19.12}$ |
| GREEN→SKILLSPAN | 28.40 | 33.56 | 46.00$\uparrow_{11.60}$ | 46.29$\uparrow_{12.73}$ |
| SKILLSPAN→SAYFULLINA | 15.19 | 23.42 | 49.25$\uparrow_{34.06}$ | 58.95$\uparrow_{35.53}$ |
| GREEN→SAYFULLINA | 12.80 | 21.58 | 48.21$\uparrow_{35.41}$ | 61.87$\uparrow_{40.29}$ |
| SKILLSPAN→GREEN | 52.01 | 37.42 | 55.37$\uparrow_{3.36}$ | 38.74$\uparrow_{1.32}$ |
| SAYFULLINA→GREEN | 17.79 | 7.64 | 39.83$\uparrow_{22.04}$ | 18.31$\uparrow_{10.67}$ |

Table 12: **Precision & Recall Numbers Cross-dataset on Test.** We show the precision and recall numbers in the cross-dataset setup. We use the `∀D+WT` setup here, with JobBERTa as the backbone model.

efit of NNOSE. Note that all the retrieved "skills" tokens are from different contexts.

In Table 15, we show a qualitative sample of using JobBERTa on SKILLSPAN. The current token is "optimistic" with gold label `B`. This is a so-called "soft skill". The language model puts high confidence in the tag `B`, which is the correct tag. The retrieved neighbors are frequently relevant, but sometimes less. This indicates that the retrieved neighbors (all soft skills) occur in similar contexts.

In Table 16, we show a qualitative sample of using JobBERTa on SKILLSPAN. The current token is "optimistic" with gold label `B`. This is a so-called "soft skill". The language model puts high confidence in the tag `B`, which is the correct tag. The retrieved neighbors are frequently relevant, but sometimes less. This indicates that the retrieved neighbors (all soft skills) occur in similar contexts.

## G Further Cross-dataset Analysis

**Precision and Recall Scores Cross-dataset.** In Table 12, we checked the precision and recall numbers for the cross-dataset setup with `∀D+WT` and JobBERTa as the backbone model. When using NNOSE, we generally notice an increase in precision, with the largest when applied to SAYFULLINA. The largest gains are with respect to recall, we notice a significant gain in all setups, where the recall and precision increase is mixed. This indicates that NNOSE is a useful method for

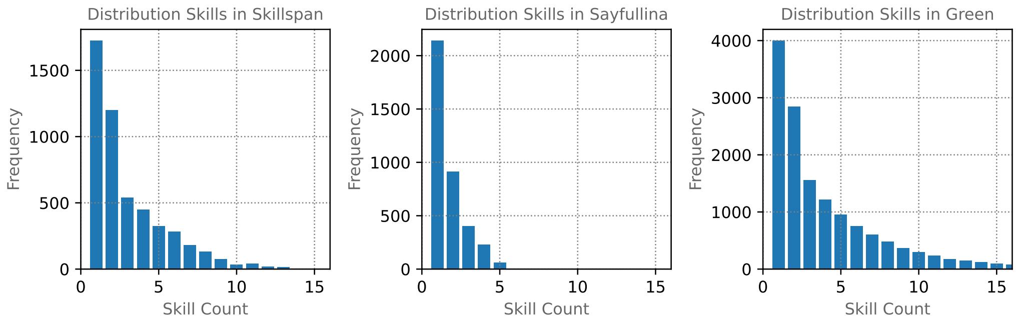


Figure 8: **Frequency Distribution of Skill Occurrences in the Train Set.** We display the frequency distribution of skill occurrences in each train set. *How to read*: For instance, in the case of Sayfullina, there are over 2,000 skills that occur only **once** in the training set. We demonstrate that all skill datasets exhibit an inherent long-tail pattern.

both precision-focused and recall-focused applications, as we are storing skills in the datastore to be retrieved.

| JobBERTa → SKILLSPAN | |
|---|---|
| Current token | IT |
| Gold label | O |
| LM prediction probs | [0.277, 0.404, 0.319] |
| Nearest neighbors ($k = 8$) | ['IT', 'Software', 'Software', 'Cloud', 'Cloud', 'Database', 'Ag', 'software'] |
| Aggregated $k$NN scores | [0.000, 0.132, 0.868] |
| Final predicted probs | [0.221, 0.350, 0.429] |

Table 13: **Cherry Picked Qualitative Sample NNOSE of Higher Precision.** We show a qualitative sample of using JobBERTa on SKILLSPAN. In this case, we see more weight being put on a specific tag, resulting in higher precision.

| JobBERTa → SKILLSPAN | |
|---|---|
| Current token | coding |
| Gold label | B |
| LM prediction probs | [0.988, 0.000, 0.012] |
| Nearest neighbors ($k = 8$) | ['programming', 'coding', 'programming', 'debugging', 'scripting', 'writing', 'coding', 'programming'] |
| Aggregated $k$NN scores | [1.000, 0.000, 0.000] |
| Final predicted probs | [0.991, 0.000, 0.009] |
| | |
| Current token | skills |
| Gold label | I |
| LM prediction probs | [0.000, 0.990, 0.010] |
| Nearest neighbors ($k = 8$) | ['skills', 'skills', 'skills', 'skills', 'skills', 'skills', 'skills', 'skills'] |
| Aggregated $k$NN scores | [0.000, 1.000, 0.000] |
| Final predicted probs | [0.000, 0.992, 0.008] |

Table 14: **Cherry Picked Qualitative Sample NNOSE of Multiple Tokens.** We show a qualitative sample of using JobBERTa on SKILLSPAN with multi-token annotations and how this behaves.

| JobBERTa → GREEN | |
|---|---|
| Current token | tools |
| Gold label | I |
| LM prediction probs | [0.250, 0.374, 0.379] |
| Nearest neighbors ($k = 8$) | ['tools', 'tools', 'transport', 'transport', 'transport', 'transport', 'car', 'transport'] |
| Aggregated $k$NN scores | [0.124, 0.626, 0.250] |
| Final predicted probs | [0.234, 0.399, 0.366] |

Table 15: **Cherry Picked Qualitative Sample NNOSE of Randomness.** We show a qualitative sample of using JobBERTa on SKILLSPAN.The language model puts high confidence on the tag I, which is the correct tag. Here the retrieved neighbors do not seem too relevant, which in this case is mostly random chance that it got it correctly.

| JobBERTa → SKILLSPAN | |
|---|---|
| Current token | optimistic |
| Gold label | B |
| LM prediction probs | [0.998, 0.000, 0.002] |
| Nearest neighbors ($k = 8$) | ['proactive', 'responsible', 'holistic', 'operational', 'positive', 'open', 'professional', 'agile'] |
| Aggregated $k$NN scores | [1.000, 0.000, 0.000] |
| Final predicted probs | [0.999, 0.000, 0.001] |

Table 16: **Cherry Picked Qualitative Sample NNOSE of Variety.** We show a qualitative sample of using JobBERTa on SKILLSPAN. The language model puts high confidence in the tag B, which is the correct tag. The retrieved neighbors are frequently relevant.