Improving Soft Skill Extraction via Data Augmentation and Embedding Manipulation

Muhammad Uzair Ul Haq
University of Padova
Padua, Italy
muhammaduzair.ulhaq@phd.unipd.it

Paolo Frazzetto
University of Padova
Padua, Italy
paolo.frazzetto@phd.unipd.it

Giovanni Da San Martino
University of Padova
Padua, Italy
giovanni.dasanmartino@unipd.it

Alessandro Sperduti
University of Padova
Padua, Italy
DISI, University of Trento
Trento, Italy
alessandro.sperduti@unipd.it

ABSTRACT

Soft skills (SS) are important for Human Resource Management when recruiting suitable candidates for a job. Nowadays, enterprises aim to automatically extract such information from documents, curriculum vitae (CVs) and job descriptions, to speed up their recruitment process. State-of-the-art Large Language Models (LLMs) have been successful in Natural Language Processing (NLP) by fine-tuning them to the domain-specific task. However, annotated data for the task is very limited and costly to obtain, since it requires domain experts. Moreover, SS consists of complex long entities which are difficult to extract given few annotated examples. As a consequence, the performance of the LLMs on soft skill detection still needs improvement before being used in a real-world context. In this paper, we introduce data augmentation based entity extraction approach which shows promising performance when the entity length is long (i.e more than three tokens). Moreover, we explore the performance of pre-trained LLMs to generate synthetic data for training. The pre-trained models are used to generate contextual augmentation of the baseline dataset. We further analyse the embeddings generated by these models in aiding the extraction process of entities. We develop an Embedding Manipulation (EM) approach to further improve the performance of baseline models. We evaluated our approach on the only publicly available dataset for soft skills (SKILLSPAN), and on three Entity Extraction datasets (GUM, WNUT-2017 and CoNLL-2003) to assess the proposed approach. Empirical evidence shows that the proposed approach allows us to get 6.52% increased F_1 over the baseline model for the soft skills.

CCS CONCEPTS

• **Information systems** → *Structured text search*; **Information extraction**; **Language models**.

SAC '24, April 8–12, 2024, Avila, Spain

ACM ISBN 979-8-4007-0243-3/24/04.
https://doi.org/10.1145/3605098.3636010

KEYWORDS

Skill Extraction, Data Augmentation, Human Resource, Embeddings, NER

ACM Reference Format:
Muhammad Uzair Ul Haq, Paolo Frazzetto, Giovanni Da San Martino, and Alessandro Sperduti. 2024. Improving Soft Skill Extraction via Data Augmentation and Embedding Manipulation. In *The 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24), April 8–12, 2024, Avila, Spain.* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3605098.3636010

1 INTRODUCTION

In recent years, there has been a paradigm shift toward online job posting and recruitment portals. With the help of these platforms, candidates can effortlessly upload their data and documents, such as resumes and curriculum vitae (CV), for the chosen vacancies. On the positive side, these systems have made the job application process smoother for candidates, but on the other hand, it has made the screening process a time-consuming and labor-intensive task for recruiters—for a single job advertisement, the Human Resources (HR) department may receive a huge number of applications. Machine learning tools supporting the recruiters could potentially save a significant amount of HR resources [2, 24, 29]. Specifically, automatic information extraction from text data could greatly speed up a recruiter's job [19]. In this context, the data extraction primarily focuses on recognizing applicants' personal data, work experience, and education. Soft Skills (SS) are also one of the constructs that recruiters try to assess when screening candidates given a job profile. The job profile consists of features (e.g. education, background, hard and soft skills, past covered positions, ...) that an ideal candidate should possess to get the respective position.

However, extracting the soft skills from text data can be tedious as SS do not possess any distinctive definition or rule. To help recruiters, a tool for automatic soft skill extraction from CVs is needed. The current state-of-the-art in soft skill extraction is insufficient to cover this need. Moreover, it is difficult to improve the performance of the current state-of-the-art using conventional techniques due to the scarcity of available datasets, with the notable exception of the SKILLSPAN dataset [31].

SS could consist of a single token (communication, management, ...), multiple tokens (team player, critical thinker, ...), or even a sentence (manage and develop a competitive service product portfolio strategy ...). Recently, researchers have tackled the task of extracting SS from resumes and job postings as a classification task at sentence level [28], and at token level [31]. However, the quality of the output of these models is severely limited by the scarcity of available annotated data. Moreover, due to the complex nature of the human language, the interpretation of a word largely depends on the context. For instance, *head* in "head of management" contributes towards SS, while it does not in "head of human body".

State-of-the-art Large Language Models (LLMs) have been shown to be successful in understanding word meaning in context. These models can be fine-tuned to domain-specific tasks, yet to do so an annotated dataset is required. However, manually annotating large corpora at the token level is time-consuming and tedious. Additionally, SS detection requires the knowledge of domain experts to do quality annotations.

In this scenario, data augmentation (DA) [15] is a viable approach to generate more synthetic data, given a limited amount of golden annotations. However, the current DA approaches performs well when the length of the entities in the dataset is short. The quality of generated data via DA degrades when the length of the entities becomes longer.

In this paper, we address the problem of SS extraction especially when the length of the entities is long, i.e., more than three tokens (more details in Section 3.4). Starting from baseline DA techniques, we develop new DA approaches based on LLMs to improve the results. Moreover, we introduce the Embedding Manipulation (EM) approach to further improve the performance of LLMs.

2 RELATED WORKS

Since resumes usually follow a standard structure, keyword-matching algorithms can be leveraged to search and extract specific data in the relevant sections, such as personal details and experiences [1]. Challenges arise when detecting hard and soft skills due to their ambiguity, the need for annotated text, and domain experts.

The skill extraction task has been framed as a binary classification problem by [21], where they implemented a phrase-matching based approach to differentiate between SS phrases and something else. After comparing different neural network models such as CNNs, LSTMs, and Hierarchical Attention Model, it was found that using an input representation with tagged skills in combination with an LSTM achieved the best performance.

The SkillNER system [7] is a tool based on a support vector machine model. The training for this system was based on a collection of 5000 scientific papers, and it utilizes a classification system for SS from the O*NET database. The system is divided into two parts: clue extraction and skill extraction. Clue extraction involves identifying patterns that suggest the presence of a specific SS, and these clues are then used to identify relevant sentences in the corpus. In the skill extraction stage, these sentences are labeled, and the resulting data is used to train a support vector machine and an MLP. However, the evaluation of the system has shown that there is room for improvement.

The models proposed by [28] use BERT [5] word embedding representations in combination with POS tags (Part of Speech Tags) and DEP tags (Dependency Parsing Tags). These features were used to train various machine learning classifiers, which were then evaluated on publicly available datasets. Results showed that using these techniques improved accuracy compared to traditional methods, although the limited size of the datasets hampered further progress.

Zhang et al. [31] released a novel dataset for skill extraction on English job postings called SKILLSPAN, while also outlining the annotation guidelines created by domain experts to annotate hard and soft skills. Additionally, this research introduces two BERT models (jobBERT and jobSpanBERT) that are optimized with continuous pre-training on the job posting domain and multi-task learning techniques. Experimental results obtained with these models show that single-tasking and multitasking can improve performance significantly over non-adapted counterparts. The authors point out the need to enrich the taxonomy with unseen skills, and they addressed this issue using weak supervision in a subsequent work [32].

Finally, Imane et al. [12] provide a systematic review and classification of skills extraction techniques.

3 DATA AUGMENTATION

Data Augmentation (DA) is a well-known technique in machine learning to automatically generate more training data without an extensive annotation exercise [15]. DA has recently become popular in Natural Language Processing (NLP) due to the availability of large language models and increased interest in low-resource domains. There are various DA techniques in NLP, including random swap [27], random insertion [27], word deletion [27], back-translation, and text generation (see [8] for a survey on the topic).

Most techniques described above are only suitable when the entity (to extract) length is short, i.e., one or two tokens. The current approaches do not show promising results for longer entities (length greater than three tokens). Also, such techniques can only be used for text classification tasks where the annotation is only at the sentence level. These tasks are annotated via binary or multi-label schema. For instance, in a binary classification task, each sentence is annotated either *Positive* or *Negative*. Hence, preserving the golden annotation is straightforward. Such tasks can easily leverage the aforementioned DA techniques since token-label correspondence is unnecessary. However, the problem arises when it comes to fine-grained analysis tasks such as Named Entity Recognition (NER). In the NER task, each token of a document or sentence is tagged. Any manipulation of the input sequence might misalign the corresponding label. Therefore, preserving the gold labels becomes a critical task in NER problems. More details are explained in Section 3.5. In the remaining part of this section, descriptions and limitations of some of these techniques are provided.

3.1 Word Replacement

Various methods of word replacement have been proposed in the past. The approach proposed in [27] replaces words with one of their synonyms (WordNet [18]) or random word insertion, swap, or deletion. An alternative solution using word replacement based on context predicted by a bi-directional LSTM-RNN based language model has been suggested in [14]. Another approach presented

in [9] replaces a randomly chosen word in a sequence with a soft word which is a probabilistic distribution over the vocabulary of a language model. The author leveraged the use of Transformer architecture [26] for the language model. However, these techniques are suitable for short entity lengths (less than three tokens). Also, these techniques are limited in annotation-sensitive tasks such as token classification due to the token-label misalignment problem, as explained in Section 3.5.

3.2 Back-Translation

Back-translation is a popular approach in NLP where a sequence is translated into another language and then translated back to the original [22]. This approach preserves the overall semantics of the original sentences but does not guarantee to preserve token-label correspondence in token-level tasks.

3.3 Masked Language-Modelling

There have been few efforts to address the token-label misalignment problem. For instance, Ding et al. [6] proposed using DA as a conditional generation task, generating new sentences while preserving the original targets and labels. Their approach relies upon linearized labeled sequences. During linearization, the entity labels are explicitly inserted in the sequence. This approach is controllable and allows for more diversified sentence generation. Zhou et al. [33] suggest the use of Masked Entity Language Modelling (MELM) as a DA framework for low-resource NER, which addresses the token-label misalignment issue by injecting NER labels explicitly into a sentence [6]. This enables the fine-tuned MELM to predict masked entity tokens while explicitly conditioning on their labels [3]. Such techniques solve the token-label misalignment problem by injecting the label information explicitly into the model. However, the performance degrades when the lengths of entities are longer. Also, these methods require post-processing to remove noisy samples from the augmented data.

3.4 Entity Length

In this paper, we consider entities with a length of less than three tokens as short entities, whereas entities with a length greater than three tokens as long entities. In this research, we find that the length of the entities affects the data augmentation process. The current state-of-the-art approaches are viable for the token classification task when the entity length is short. Unfortunately, for longer entities the literature is limited [23]. Moreover, most datasets for token classification tasks consist of named entities, such as the name of a person, organization, or location, which usually consist of one or two tokens. However, the entity length could be much longer for niche tasks such as SS extraction.

To assess our approaches, in addition to a SS dataset, we selected one dataset with similar entity length statistics as the SS dataset, and two other datasets that represent the usual entity extraction tasks where the average length is shorter. Experimental evaluation shows that traditional DAs perform well with short entities but not well with long entities; conversely, our proposed DAs are able to perform well with long entities while being comparable to traditional DAs on short ones.

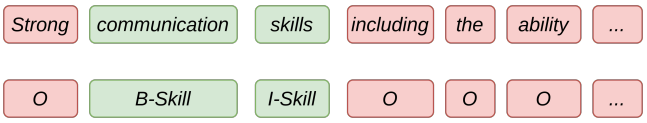


Figure 1: Annotation scheme, data (top) is annotated using BIO-tags (bottom).

In Table 1, we report the average entity length of the datasets we used in our experimental assessment. We see that, on average, a SS consists of 5 tokens, whereas on average entities in GUM datasets consist of 3 tokens. The average entity length in WNUT-2017 and CoNLL-2003 datasets is 1.5

3.5 Token-Label Misalignment

In a token-level classification task, each token in a document is assigned a corresponding label, as shown in Figure 1. One of the hard conditions for DA on such a task is to preserve the token-label correspondence in the output. The token-label misalignment is a critical problem limiting the use of DA techniques. For instance, in back-translation or text generation, the augmented output sequence is not guaranteed to be aligned with the input sequence length. By using the back translation augmentation from [17] the following generated output sequence consists of 8 tokens, whereas the input sequence is of 12 tokens:

Original Text: `Support and mentor other engineers through code reviews and pair programming sessions.`
Augmented Text: `Support mentor other through code reviews programming sessions.`

The above example limits us to only using the DA techniques where the token-label correspondence is retained. Therefore, we explore contextual augmentation as explained in Section 4.

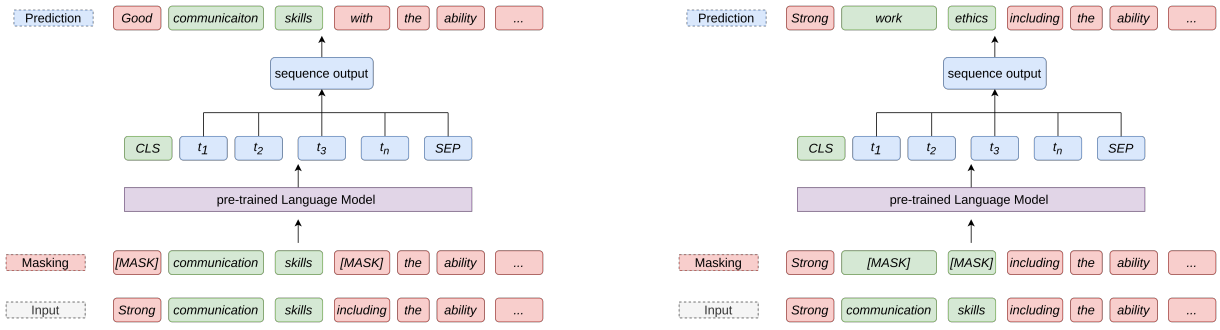
4 OUR METHODS

The proposed SS extraction workflows are shown in Figure 2. We propose contextual and keyword DA by a pre-trained LLM [26]. The LLM is used to generate synthetic data. Moreover, using Embedding Manipulation (EM) we better exploit the information from the annotated dataset. As shown in Figure 1, the input sentence is composed of two parts referred to as *context* (highlighted in light red) and *keywords* (highlighted in light green). Here, the keyword is defined as a SS in the sentence. As can be seen from Figure 1, the gold standard data is annotated using BIO (Beginning-Inside-Outside) tags.

Notice that the SS labeled by domain experts consists of unique tokens that are largely dependent on the context. To learn the representation of such soft skills given one specific context is not enough. The LLM requires more examples to discriminate between different contexts. To address this problem, we propose two types of augmentation techniques, namely *context augmentation* and *keyword augmentation*, and to further enhance the performance, we use *Embedding Manipulation*.

Table 1: Statistics of the datasets used in the proposed work are presented. The average entity length refers to the average number of tokens for each entity.

Dataset	Sentences			Tokens			Avg. Entity Length
	Train	Validation	Test	Train	Validation	Test	
SKILLSPAN	3074	1396	1522	92621	39923	42541	4.72
GUM	1435	615	805	29392	12688	17437	3.15
WNUT-2017	3394	1008	1287	62730	15734	23394	1.73
CoNLL-2003	3000	1000	1000	44429	14511	12538	1.60



(a) Contextual DA **(b) Keyword DA**

Figure 2: Pipeline of the proposed Data Augmentation flow. The tokens comprising of SS are referred to as keywords (marked in light green), whereas the rest of the tokens are referred to as context (marked in light red). A pre-trained Language Model is used to replace either the context or SS tokens. (a) refers to Contextual Augmentation, (b) refers to Keyword Augmentation.

4.1 Context Augmentation

As shown in Figure 2a, an input sentence of the training set is divided into keywords (light green) and context (light red), according to the gold labels provided by the annotators. In contextual augmentation, we leverage the pre-trained contextual embedding of the language model. Given an input sentence S containing context tokens C_t and Keyword tokens as K_t, we mask the C_t with M, and the model task is to predict the M given K_t. The new substitutes for tokens C_t can be sampled from a given probability distribution over the vocabulary of the language model. We choose the top 5 tokens to generate 5 different augmentations of the same sentence S. We generate similar sentences with the pre-trained model, constraining it to replace only tokens of the input sentence. The original token-label correspondence is maintained since we perform augmentation by substituting the tokens instead of random insertion or deletion. The generated augmented sentence is shown at the top of Figure 2a, where the pre-trained model predicts masked tokens M. In this way, we generate more contexts given a single K_t. Additionally, the sequence length for all the augmentations is the same as the input sequence, so the token-label correspondence is preserved.

4.2 Keyword Augmentation

In keyword augmentation, using the same approach mentioned in Section 4.1, given an input sentence S containing context tokens C_t and Keyword tokens as K_t, we mask the K_t with M, and the model task is to predict the M given C_t. The new substitutes for tokens K_t are sampled similarly as mentioned in Section 4.1. This setting allows us to generate more sentences with different keywords given the one specific context C_t. The keyword augmentation is shown in Figure 2b.

4.3 Embedding Manipulation

In this paper, we propose a simple yet effective approach to explore the embeddings of a pre-trained language model. Given an annotated dataset, the gold label for each token T is known beforehand. For instance, the Skillspan dataset D is annotated with soft skills using BIO tagging. With this information, we can extract all the soft skills (where a single soft skill is referred to as SS_i, $i = 1, \ldots, n$) in the training set. Each SS_i in the training set consists of a different number of tokens $|SS_i|$. All the extracted SSs are padded to the maximum length m of soft skills in the training set. The padded SS_i is then passed through a LLM [26] to generate the sequence of m embedding vectors $E^i_1, \ldots, E^i_m$ corresponding to each token T_j, $j = 1, \ldots, m$, in SS_i. All the generated embedding vectors are then averaged out to produce a single embedding vector E^i representing SS_i. Likewise, for each SS_i in the training set E^i is calculated. Finally, all the calculated embedding vectors E^i are averaged out to produce a single global embedding vector E_{avg}, which supposedly contains the average representation in vector space of all SSs in the training set.

During fine-tuning of LLM, given a sentence S with k input tokens $\{T_j \mid j = 1, \ldots, k\}$, we obtain the embedding representation

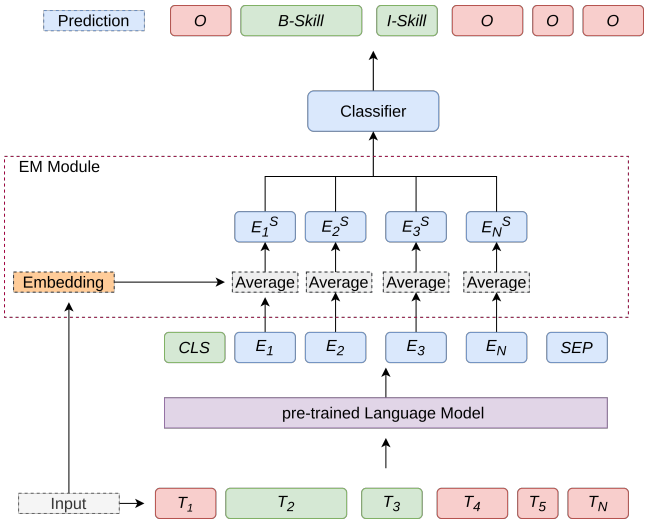


Figure 3: Proposed methodology for fine-tuning LLM via Embedding Manipulation (EM). T_i are input tokens to the pre-trained LM. E_i refers to the embedding representation of each token generated by the pre-trained LM. Embedding (highlighted in orange) is the average representation of entities, calculated by extracting entities in the training set. E_i^S is the average embedding representation of each token.

of each token $T_j \overset{LLM}{\longmapsto} E_j^S$. We use the pre-calculated E_{avg} and average with each E_j^S, $j = 1, \ldots, k$, obtaining the resulting vector corresponding to each token j as $\widehat{E_j^S} = (E_j^S + E_{avg})/2$ (see Figure 3). Then, the resulting embedding vectors $\widehat{E_j^S}$, $j = 1, \ldots, k$, are passed through the classifier for final prediction. Since E_{avg} contains the average representation of entities, performing a simple averaging operation on each E_j^S with E_{avg} raises the score of E_j^S which are closer to E_{avg}. This allows the LLM to be driven toward entities of interest in the dataset.

Likewise, during testing we use the EM module (refer to Figure 3). As we do not have the information of golden annotation, we use the same E_{avg} as calculated from the training set. For each sentence S with k input tokens $\{T_j \mid j = 1, \ldots, k\}$, we pass all the embeddings of tokens $E_1^i, \ldots, E_m^i$ through EM module which allows us to obtain $\widehat{E_j^S}$, $j = 1, \ldots, k$ (explained in above paragraph), which is then passed through classifier for final predictions. The proposed setting allows the increase in the score of E_j^S close to E_{avg} in the embedding space.

5 DATASETS AND EXPERIMENTAL SETUP

This section describes our empirical assessment of the proposed DA techniques. Due to the scarcity of SS datasets, in addition to SKILLSPAN, we tested our approach on three datasets, i.e., GUM, WNUT-2017, and CoNLL-2003. For CoNLL-2003, we used a subset of comparable size to the other two datasets to "simulate" a training regime involving a relatively small number of labeled examples. For all the datasets used in this work, we compare our DA approach versus the baselines obtained by using BERT and RoBERTa. For SKILLSPAN, we also report the performance of jobBERT, jobSpanBERT [31], and GPT-4 [20].

5.1 Datasets

In the following, we provide details of the SKILLSPAN dataset and three additional entity extraction datasets used in our experimental assessment. Table 1 presents the statistics of the datasets we used in our experimental assessment. As explained in Section 4.1, when considering our two DA techniques, the training set for the datasets reported in Table 1 is augmented with N more samples via an LLM, increasing its size by N+1 times.

SKILLSPAN dataset, described in [31], is collected from job postings and labeled by domain experts. The authors of the dataset divided the data into three categories BIG, HOUSE, and TECH. The BIG dataset has not been released, whereas the HOUSE and TECH datasets are publicly available. The HOUSE dataset contains the various categories of job ads from 2012 to 2020, whereas the TECH dataset is restricted to only technical job postings. In this paper, the HOUSE and TECH datasets are merged to increase baseline data.

GUM dataset [30] is an open-source dataset of distinct annotated texts from twelve different text types. GUM comprises diverse text sources, such as interviews, news stories, academic writing, biographies, Wikipedia articles, political speeches, ... The dataset contains nine different entities.

WNUT-2017 dataset [4] focuses on identifying unusual, previously unseen entities in the context of emerging discussions. It evaluates the ability to detect and classify novel, emerging, singleton-named entities in noisy text.

CoNLL-2003 dataset [25] is a token classification dataset with 4 classes of entities extracted from a news corpus. To better simulate the low resource settings, we choose a subset of CoNLL-2003 to match best the token statistics of the other two datasets.

5.2 Experimental Setup

All the experiments are conducted in the same `Python` environment. We used the [10] transformers repository for the model implementation. Each experiment is executed with early stopping with patience of 3 and a cut-off of 0.3. The batch size is 32; the initial learning rate is set to 2×10^{-5} with AdamW optimizer [13], and weight decay of 0.2. All experiments are conducted on NVIDIA RTX *A*6000 GPU.

NER-MODEL For fine-tuning, we use the approach described in the original work by Vaswani et al. [26]. We utilize the base [5] and RoBERTa [16] model for our experiments with the linear classifier [10] on top.

MLM-MODEL For augmentation, we use the baseline RoBERTa [16] model and replace the linear classification head with the Language Modeling Head (LMH) [10] on top.

GPT-4 We used prompting to extract SS using the GPT-4 [20] model. Using the whole test set to extract SS using GPT-4 is expensive. Therefore, we sampled 50 examples from the test set by randomly shuffling data with three different seeds (42, 3143, and 4314). The experiments are performed by varying the temperature value from 0.3 to 0.7 with a 0.2 increment. The best temperature value was found to be 0.7. To better estimate the quality of the model three different prompts were used, as presented below:

Table 2: Number of generated augmented sentences N and F_1 score on the validation set for each dataset and each considered model/augmentation technique. The baseline value is for $N = 0$, i.e. no augmentation. Best performance for each model/data augmentation is in bold.

SKILLSPAN

Model	N					
	0	1	2	3	4	5
BERT	45.85	51.82	**53.03**	52.09	52.49	52.85
jobBERT	48.01	55.47	**56.13**	55.14	53.41	54.49
jobSpanBERT	47.80	56.13	**56.67**	55.53	56.07	54.07
RoBERTa	49.20	56.56	**57.69**	57.34	57.20	57.03

WNUT-2017

Model	N					
	0	1	2	3	4	5
BERT	46.87	**53.78**	51.25	49.45	48.98	49.14
RoBERTa	63.85	**65.37**	65.18	63.81	59.66	56.42

CONLL-2003

Model	N					
	0	1	2	3	4	5
BERT	**91.93**	91.68	91.70	91.95	90.60	90.54
RoBERTa	94.51	**94.53**	93.75	9290	92.31	92.25

GUM

Model	N					
	0	1	2	3	4	5
BERT	48.85	54.48	**57.24**	55.95	55.33	55.02
RoBERTa	55.27	57.61	59.85	**61.94**	59.35	59.40

(1) Extract list of tokens corresponding to soft skills in the given input sentence.
(2) Extract soft skills from the input sentence. Return tokens corresponding to soft skills in a list.
(3) Extract a list of skills from the input sentence.

The prompts were engineered to cast the output of the model as a token classification task. The best prompt is chosen which returns the best F_1 score on the test samples. The proposed prompting approach reduces the post-processing work of the output generated by GPT-4 model and also allows us to make better comparisons with the other approaches presented in Table 3. The results using the best prompt are presented in Table 3, whereas some examples can be visualized in Table 5.

Hyperparameter Tuning and Model Selection To determine the optimal setting for augmentation rate N, we conduct a grid search on hyperparameters in the range $[1, 2, 3, 4, 5]$[1]. We used the pre-trained baseline models and generated the augmented data on the dataset following our methods described in Section 4. The augmented data was used to fine-tune the model, and its performance on the validation set was recorded. The best model was chosen based on the best F_1 score on the validation set. The number of augmentations for each dataset is presented in Table 2. It can be noted that for datasets with longer entities—SKILLSPAN and GUM—the performance degrades after 3 rounds of augmentation, whereas for datasets with short entities, i.e., WNUT-17 and CoNLL-2003, a similar trend is observed after 2 rounds of augmentation.

[1]Please, recall that we choose the top 5 tokens to generate five different augmentations of the same sentence.

6 RESULTS

As already stated, due to the scarcity of datasets for SS extraction, we decided first to assess our DA techniques proposed in Section 4 on three entity extraction tasks. For these three datasets, we considered two LLMs, BERT [5] and RoBERTa [16]. In contrast, for improving the performance of SS in job postings and CVs (SKILLSPAN dataset) we also considered jobBERT [31] and jobSpanBert [31]. The performance of the GPT-4 model [20] is also evaluated for SS extraction using the methodology mentioned in Section 5.2.

6.1 Soft Skills

To show the effectiveness of the proposed DA approaches, we perform an extensive comparison with traditional DA approaches such as (word deletion, synonym replacement, word swap, and spelling augmentation [27]) and recently introduced DA approaches for token classification tasks (MELM [33] and DAGA [6]). From Table 3, we observe that the proposed DA approaches (keyword and context augmentation) used in the experiments achieve a higher F_1 score than the baseline counterpart. The improvement is significant for all the models used in experimental assessment. It can be observed from empirical evidence that the performance of context augmentation surpasses keyword augmentation. We hypothesize that this increase in performance is because when the entity length is long, LLM requires more contextually diverse examples to learn better representations of entities. From Table 3, we notice the previous state-of-the-art models jobBERT and jobSpanBERT for SS extraction can perform better than the BERT model. However, the RoBERTa model achieves the highest performance in the SS extraction task overall.

Moreover, we notice that EM plays a significant role in further enhancing the performance of LLMs. To test whether the EM leads to a higher F_1 score without assuming normality and homoscedasticity, we resorted to the Wilcoxon signed-ranks test [11]. It gives $W = 224$ and p-value$= 0.017$, revealing statistical evidence from the experiments that using EM improves the performance over baseline models. The RoBERTa model with EM fine-tuned on contextual augmented dataset achieves the highest F_1 score of 54.46, which leads to a 6.52% improvement over the baseline counterpart. We also evaluate our approach against the off-the-shelf GPT-4 model using the prompting approach (described in Section 5.2). The GPT-4 model achieves 48.01% in absolute F_1, which is 6.41% less than our proposed approach.

Table 3: The F_1 along with standard deviation, precision, and recall are reported on the test set of SKILLSPAN dataset. The values are averaged over three different random initializations. The results are reported with and without EM. The bold highlight shows the highest F_1 score.

Model	Technique	Without EM			With EM		
		Precision	Recall	F_1	Precision	Recall	F_1
GPT-4	Prompt	$39.61_{\pm 2.00}$	$60.95_{\pm 3.00}$	$48.01_{\pm 2.35}$	-	-	-
BERT	Baseline	$39.12_{\pm 0.70}$	$51.16_{\pm 3.19}$	$44.29_{\pm 1.05}$	$41.92_{\pm 0.87}$	$46.29_{\pm 3.89}$	$43.74_{\pm 2.64}$
	Word Deletion	$43.92_{\pm 2.70}$	$54.48_{\pm 2.88}$	$48.23_{\pm 2.47}$	$42.04_{\pm 4.84}$	$59.37_{\pm 3.02}$	$49.00_{\pm 2.26}$
	Synonym Replacement	$47.97_{\pm 0.72}$	$45.61_{\pm 3.43}$	$46.71_{\pm 1.53}$	$45.62_{\pm 2.83}$	$47.28_{\pm 3.19}$	$45.06_{\pm 1.98}$
	Word Swap	$40.93_{\pm 4.48}$	$57.71_{\pm 4.03}$	$47.71_{\pm 0.94}$	$45.46_{\pm 0.89}$	$55.82_{\pm 4.08}$	$50.06_{\pm 1.85}$
	Spelling Augmentation	$43.97_{\pm 3.91}$	$53.39_{\pm 2.18}$	$47.69_{\pm 2.31}$	$48.05_{\pm 1.15}$	$50.76_{\pm 4.54}$	$49.26_{\pm 1.54}$
	MELM et al. [33]	$44.75_{\pm 2.31}$	$57.59_{\pm 1.49}$	$50.31_{\pm 0.92}$	$43.23_{\pm 0.44}$	$56.82_{\pm 0.15}$	$49.10_{\pm 0.23}$
	DAGA et al. [6]	$40.93_{\pm 0.37}$	$60.48_{\pm 0.16}$	$48.82_{\pm 0.31}$	$41.01_{\pm 2.28}$	$59.58_{\pm 0.88}$	$48.55_{\pm 1.30}$
	Keyword Augmentation	$42.15_{\pm 2.80}$	$57.55_{\pm 3.39}$	$48.54_{\pm 1.06}$	$41.47_{\pm 1.18}$	$59.12_{\pm 0.35}$	$48.73_{\pm 0.75}$
	Context Augmentation	$47.24_{\pm 2.27}$	$57.18_{\pm 1.31}$	**$51.71_{\pm 1.26}$**	$46.70_{\pm 2.14}$	$55.59_{\pm 0.18}$	$50.74_{\pm 1.20}$
jobBERT	Baseline	$42.18_{\pm 0.61}$	$53.26_{\pm 0.66}$	$47.07_{\pm 0.14}$	$44.46_{\pm 2.27}$	$52.31_{\pm 2.52}$	$48.00_{\pm 0.48}$
	Word Deletion	$45.34_{\pm 3.66}$	$60.90_{\pm 1.94}$	$51.87_{\pm 1.80}$	$42.87_{\pm 0.47}$	$62.65_{\pm 1.46}$	$50.90_{\pm 0.18}$
	Synonym Replacement	$44.97_{\pm 2.63}$	$58.40_{\pm 4.07}$	$50.68_{\pm 0.35}$	$44.36_{\pm 1.16}$	$62.76_{\pm 2.70}$	$52.04_{\pm 0.64}$
	Word Swap	$46.44_{\pm 1.04}$	$60.30_{\pm 0.98}$	$51.92_{\pm 0.31}$	$42.74_{\pm 1.81}$	$62.33_{\pm 1.02}$	$50.68_{\pm 0.93}$
	Spelling Augmentation	$45.61_{\pm 0.76}$	$54.25_{\pm 6.58}$	$49.76_{\pm 2.84}$	$44.36_{\pm 1.02}$	$62.76_{\pm 1.34}$	$51.97_{\pm 0.60}$
	MELM et al. [33]	$48.16_{\pm 3.34}$	$56.96_{\pm 4.26}$	$52.06_{\pm 0.18}$	$47.81_{\pm 1.77}$	$57.34_{\pm 1.87}$	$52.12_{\pm 0.28}$
	DAGA et al. [6]	$46.63_{\pm 2.62}$	$58.38_{\pm 2.05}$	$51.78_{\pm 0.82}$	$47.51_{\pm 0.61}$	$59.33_{\pm 0.83}$	$52.76_{\pm 0.65}$
	Keyword Augmentation	$44.76_{\pm 0.68}$	$58.31_{\pm 1.89}$	$50.63_{\pm 0.74}$	$45.02_{\pm 0.81}$	$60.57_{\pm 0.63}$	$51.64_{\pm 0.31}$
	Context Augmentation	$49.05_{\pm 1.74}$	$58.03_{\pm 0.63}$	$53.15_{\pm 0.78}$	$48.46_{\pm 1.37}$	$59.35_{\pm 0.50}$	**$53.34_{\pm 0.65}$**
jobSpanBERT	Baseline	$43.26_{\pm 1.88}$	$50.32_{\pm 4.64}$	$46.18_{\pm 3.26}$	$45.22_{\pm 2.78}$	$55.03_{\pm 4.25}$	$49.49_{\pm 0.03}$
	Word Deletion	$45.99_{\pm 4.60}$	$58.29_{\pm 4.3}$	$51.17_{\pm 1.34}$	$46.68_{\pm 1.70}$	$59.12_{\pm 1.40}$	$52.14_{\pm 0.82}$
	Synonym Replacement	$45.66_{\pm 3.23}$	$59.37_{\pm 1.64}$	$51.54_{\pm 1.39}$	$46.64_{\pm 1.34}$	$59.23_{\pm 3.63}$	$52.13_{\pm 1.23}$
	Word Swap	$48.44_{\pm 0.50}$	$57.20_{\pm 1.63}$	$52.44_{\pm 0.40}$	$47.12_{\pm 2.26}$	$57.94_{\pm 2.63}$	$51.89_{\pm 0.27}$
	Spelling Augmentation	$45.50_{\pm 3.10}$	$58.52_{\pm 4.29}$	$51.03_{\pm 0.60}$	$46.10_{\pm 2.32}$	$61.61_{\pm 1.28}$	$52.70_{\pm 1.20}$
	MELM et al. [33]	$49.45_{\pm 1.29}$	$56.79_{\pm 1.75}$	$52.84_{\pm 0.49}$	$50.25_{\pm 0.65}$	$58.10_{\pm 1.37}$	$53.89_{\pm 0.96}$
	DAGA et al. [6]	$48.03_{\pm 1.26}$	$56.58_{\pm 0.9}$	$51.94_{\pm 0.34}$	$48.98_{\pm 1.81}$	$57.64_{\pm 2.29}$	$52.93_{\pm 1.44}$
	Keyword Augmentation	$48.16_{\pm 1.02}$	$59.35_{\pm 0.42}$	$53.16_{\pm 0.49}$	$49.87_{\pm 1.54}$	$58.94_{\pm 0.2}$	$53.68_{\pm 0.92}$
	Context Augmentation	$46.20_{\pm 1.91}$	$60.80_{\pm 1.57}$	$52.48_{\pm 1.01}$	$49.87_{\pm 2.36}$	$58.94_{\pm 2.57}$	**$53.95_{\pm 0.33}$**
RoBERTa	Baseline	$43.64_{\pm 3.53}$	$53.38_{\pm 3.47}$	$47.94_{\pm 2.64}$	$46.75_{\pm 1.24}$	$56.52_{\pm 1.34}$	$51.16_{\pm 0.83}$
	Word Deletion	$48.10_{\pm 2.24}$	$53.00_{\pm 4.19}$	$50.09_{\pm 2.56}$	$48.57_{\pm 2.87}$	$57.66_{\pm 3.94}$	$52.55_{\pm 0.71}$
	Synonym Replacement	$50.31_{\pm 4.69}$	$55.95_{\pm 4.84}$	$52.70_{\pm 0.75}$	$52.43_{\pm 2.46}$	$54.04_{\pm 4.2}$	$53.02_{\pm 2.11}$
	Word Swap	$48.71_{\pm 0.40}$	$56.61_{\pm 5.71}$	$52.26_{\pm 2.64}$	$45.94_{\pm 2.65}$	$61.98_{\pm 1.65}$	$52.74_{\pm 2.05}$
	Spelling Augmentation	$51.98_{\pm 1.47}$	$49.50_{\pm 2.85}$	$50.67_{\pm 1.51}$	$53.21_{\pm 2.71}$	$52.10_{\pm 2.56}$	$52.56_{\pm 0.17}$
	MELM et al. [33]	$49.92_{\pm 5.56}$	$57.67_{\pm 1.69}$	$53.45_{\pm 0.79}$	$51.95_{\pm 1.72}$	$54.86_{\pm 2.19}$	$53.33_{\pm 0.75}$
	DAGA et al. [6]	$47.04_{\pm 3.69}$	$59.83_{\pm 2.59}$	$52.54_{\pm 1.43}$	$50.37_{\pm 0.91}$	$58.69_{\pm 0.48}$	$54.20_{\pm 0.33}$
	Keyword Augmentation	$46.73_{\pm 0.91}$	$59.47_{\pm 1.32}$	$52.33_{\pm 0.59}$	$45.16_{\pm 3.96}$	$60.42_{\pm 2.49}$	$51.55_{\pm 1.93}$
	Context Augmentation	$49.42_{\pm 0.23}$	$59.09_{\pm 1.44}$	$53.82_{\pm 0.47}$	$49.15_{\pm 0.49}$	$61.06_{\pm 0.40}$	**$54.46_{\pm 0.39}$**

Table 4: The F_1 along with standard deviation, precision, and recall are reported on the test set of the NER dataset. The values are averaged over three different random initializations. The results are reported with and without EM. The bold highlight shows the highest F_1 score.

Model	Technique	WNUT-2017			CoNLL2003			GUM		
		Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
BERT	Baseline	$49.84_{\pm 4.38}$	$29.54_{\pm 0.89}$	$37.06_{\pm 2.04}$	$86.99_{\pm 0.86}$	$90.18_{\pm 0.67}$	$88.56_{\pm 0.55}$	$44.55_{\pm 1.14}$	$53.37_{\pm 1.09}$	$48.56_{\pm 1.11}$
	Word Deletion	$56.88_{\pm 1.38}$	$31.09_{\pm 0.16}$	$40.20_{\pm 0.46}$	$88.17_{\pm 0.74}$	$90.62_{\pm 0.36}$	$89.38_{\pm 0.55}$	$48.89_{\pm 0.99}$	$57.70_{\pm 1.22}$	$52.93_{\pm 1.08}$
	Synonym Replacement	$57.04_{\pm 4.59}$	$29.44_{\pm 0.88}$	$38.77_{\pm 0.75}$	$87.81_{\pm 0.43}$	$90.02_{\pm 0.40}$	$88.90_{\pm 0.08}$	$49.03_{\pm 4.06}$	$57.17_{\pm 3.24}$	$52.78_{\pm 3.73}$
	Word Swap	$54.29_{\pm 0.80}$	$31.52_{\pm 1.79}$	$39.85_{\pm 1.21}$	$87.49_{\pm 2.05}$	$89.91_{\pm 0.78}$	$88.68_{\pm 1.42}$	$49.62_{\pm 2.62}$	$58.72_{\pm 1.48}$	$53.78_{\pm 2.17}$
	Spelling Augmentation	$58.84_{\pm 0.93}$	$30.10_{\pm 0.93}$	$39.82_{\pm 0.65}$	$88.29_{\pm 1.45}$	$90.18_{\pm 0.31}$	$89.22_{\pm 0.64}$	$47.87_{\pm 3.39}$	$56.01_{\pm 3.65}$	$51.62_{\pm 3.51}$
	MELM et al. [33]	$56.96_{\pm 4.02}$	$33.84_{\pm 0.26}$	$\mathbf{42.42_{\pm 1.10}}$	$87.06_{\pm 0.50}$	$89.40_{\pm 0.39}$	$88.21_{\pm 0.31}$	$51.13_{\pm 0.85}$	$58.35_{\pm 0.37}$	$54.51_{\pm 0.64}$
	DAGA et al. [6]	$56.86_{\pm 4.30}$	$30.58_{\pm 1.88}$	$39.61_{\pm 1.13}$	$86.89_{\pm 0.65}$	$89.26_{\pm 0.65}$	$88.06_{\pm 0.65}$	$49.05_{\pm 2.31}$	$56.71_{\pm 2.43}$	$52.60_{\pm 2.37}$
	Keyword Augmentation	$63.06_{\pm 2.05}$	$30.98_{\pm 1.33}$	$41.52_{\pm 0.98}$	$85.42_{\pm 0.33}$	$89.98_{\pm 0.55}$	$87.64_{\pm 0.20}$	$51.40_{\pm 1.40}$	$59.44_{\pm 0.42}$	$55.12_{\pm 0.96}$
	Context Augmentation	$59.18_{\pm 3.26}$	$29.41_{\pm 2.18}$	$39.19_{\pm 1.34}$	$88.91_{\pm 0.84}$	$89.45_{\pm 0.61}$	$89.17_{\pm 0.32}$	$52.99_{\pm 0.50}$	$60.43_{\pm 0.43}$	$56.47_{\pm 0.44}$
	Baseline + EM	$48.22_{\pm 3.33}$	$29.33_{\pm 2.50}$	$36.45_{\pm 2.66}$	$87.60_{\pm 1.49}$	$90.46_{\pm 0.67}$	$89.01_{\pm 1.08}$	$47.22_{\pm 0.50}$	$54.37_{\pm 0.50}$	$50.55_{\pm 0.50}$
	Word Deletion + EM	$55.78_{\pm 3.13}$	$32.61_{\pm 2.46}$	$41.10_{\pm 2.11}$	$88.24_{\pm 0.67}$	$91.14_{\pm 0.66}$	$\mathbf{89.67_{\pm 0.66}}$	$52.12_{\pm 1.21}$	$59.93_{\pm 1.41}$	$55.75_{\pm 1.28}$
	Synonym Replacement + EM	$56.28_{\pm 2.89}$	$28.87_{\pm 3.13}$	$38.15_{\pm 3.40}$	$88.26_{\pm 0.43}$	$90.42_{\pm 0.20}$	$89.33_{\pm 0.23}$	$52.76_{\pm 0.39}$	$60.25_{\pm 0.45}$	$56.26_{\pm 0.41}$
	Word Swap + EM	$55.70_{\pm 2.59}$	$29.91_{\pm 1.93}$	$38.88_{\pm 1.72}$	$87.67_{\pm 0.47}$	$90.48_{\pm 0.12}$	$89.06_{\pm 0.28}$	$51.22_{\pm 0.65}$	$58.69_{\pm 0.69}$	$54.70_{\pm 0.12}$
	Spelling Augmentation + EM	$58.84_{\pm 1.92}$	$31.14_{\pm 3.04}$	$40.63_{\pm 2.23}$	$88.18_{\pm 0.12}$	$90.33_{\pm 0.08}$	$89.24_{\pm 0.06}$	$53.07_{\pm 1.85}$	$59.72_{\pm 1.09}$	$56.20_{\pm 1.52}$
	MELM et al. [33] + EM	$57.38_{\pm 4.83}$	$33.28_{\pm 1.25}$	$42.04_{\pm 0.96}$	$86.46_{\pm 0.51}$	$90.51_{\pm 0.09}$	$88.43_{\pm 0.29}$	$53.32_{\pm 0.50}$	$60.46_{\pm 0.77}$	$56.67_{\pm 0.55}$
	DAGA et al. [6] + EM	$58.09_{\pm 2.34}$	$31.44_{\pm 2.8}$	$40.71_{\pm 2.02}$	$86.67_{\pm 0.51}$	$88.85_{\pm 0.51}$	$87.74_{\pm 0.51}$	$49.94_{\pm 2.92}$	$57.4_{\pm 3.18}$	$53.41_{\pm 3.05}$
	Keyword Augmentation + EM	$49.65_{\pm 4.36}$	$34.21_{\pm 1.16}$	$40.49_{\pm 2.17}$	$84.26_{\pm 0.51}$	$89.37_{\pm 0.50}$	$86.74_{\pm 0.47}$	$52.38_{\pm 0.64}$	$59.88_{\pm 0.66}$	$55.88_{\pm 0.65}$
	Context Augmentation + EM	$60.84_{\pm 2.00}$	$28.21_{\pm 1.79}$	$38.52_{\pm 1.86}$	$88.07_{\pm 0.53}$	$89.22_{\pm 0.09}$	$88.64_{\pm 0.26}$	$53.98_{\pm 0.16}$	$60.35_{\pm 0.25}$	$\mathbf{57.01_{\pm 0.20}}$
RoBERTa	Baseline	$56.57_{\pm 4.35}$	$46.38_{\pm 5.08}$	$50.74_{\pm 2.54}$	$90.22_{\pm 0.63}$	$92.47_{\pm 0.26}$	$91.33_{\pm 0.45}$	$50.10_{\pm 2.06}$	$58.68_{\pm 1.80}$	$54.05_{\pm 1.97}$
	Word Deletion	$63.98_{\pm 4.72}$	$47.21_{\pm 0.19}$	$54.29_{\pm 1.87}$	$90.28_{\pm 0.35}$	$93.00_{\pm 0.11}$	$91.62_{\pm 0.14}$	$56.00_{\pm 0.64}$	$62.49_{\pm 0.41}$	$59.06_{\pm 0.53}$
	Synonym Replacement	$62.03_{\pm 1.15}$	$44.78_{\pm 0.71}$	$52.01_{\pm 0.28}$	$89.84_{\pm 0.39}$	$91.88_{\pm 0.74}$	$90.85_{\pm 0.55}$	$53.83_{\pm 2.25}$	$60.18_{\pm 2.47}$	$56.83_{\pm 2.35}$
	Word Swap	$68.55_{\pm 1.16}$	$45.77_{\pm 0.99}$	$54.88_{\pm 0.42}$	$89.92_{\pm 0.45}$	$92.84_{\pm 0.24}$	$91.35_{\pm 0.21}$	$55.90_{\pm 0.55}$	$62.82_{\pm 0.80}$	$59.16_{\pm 0.11}$
	Spelling Augmentation	$62.56_{\pm 2.47}$	$45.91_{\pm 0.27}$	$53.89_{\pm 1.07}$	$90.31_{\pm 0.35}$	$92.27_{\pm 0.25}$	$91.28_{\pm 0.30}$	$53.55_{\pm 1.81}$	$60.21_{\pm 1.99}$	$56.68_{\pm 1.87}$
	MELM et al. [33]	$64.84_{\pm 1.22}$	$46.22_{\pm 0.57}$	$53.96_{\pm 0.72}$	$89.17_{\pm 0.30}$	$91.85_{\pm 0.21}$	$90.49_{\pm 0.10}$	$56.12_{\pm 0.32}$	$61.97_{\pm 0.97}$	$58.89_{\pm 0.55}$
	DAGA et al. [6]	$66.57_{\pm 1.92}$	$46.33_{\pm 3.5}$	$54.55_{\pm 2.07}$	$89.27_{\pm 0.1}$	$91.9_{\pm 0.1}$	$90.57_{\pm 0.1}$	$54.86_{\pm 1.23}$	$60.97_{\pm 1.14}$	$57.75_{\pm 1.16}$
	Keyword Augmentation	$66.80_{\pm 1.69}$	$44.79_{\pm 1.73}$	$53.59_{\pm 0.77}$	$87.18_{\pm 0.64}$	$91.60_{\pm 0.45}$	$89.51_{\pm 0.52}$	$55.24_{\pm 1.09}$	$61.56_{\pm 0.99}$	$58.22_{\pm 0.92}$
	Context Augmentation	$65.97_{\pm 2.18}$	$45.91_{\pm 2.62}$	$54.08_{\pm 1.28}$	$90.10_{\pm 0.14}$	$92.12_{\pm 0.45}$	$91.10_{\pm 0.19}$	$58.12_{\pm 0.41}$	$64.09_{\pm 0.43}$	$60.96_{\pm 0.42}$
	Baseline + EM	$57.72_{\pm 0.65}$	$46.13_{\pm 0.59}$	$51.28_{\pm 0.31}$	$90.36_{\pm 0.48}$	$92.97_{\pm 0.21}$	$91.65_{\pm 0.34}$	$54.81_{\pm 0.65}$	$60.08_{\pm 0.50}$	$57.33_{\pm 0.54}$
	Word Deletion + EM	$66.51_{\pm 1.91}$	$45.28_{\pm 1.53}$	$53.85_{\pm 0.57}$	$90.23_{\pm 0.20}$	$93.33_{\pm 0.31}$	$\mathbf{91.75_{\pm 0.25}}$	$57.93_{\pm 0.82}$	$63.35_{\pm 0.91}$	$60.52_{\pm 0.79}$
	Synonym Replacement + EM	$66.21_{\pm 0.95}$	$41.87_{\pm 1.31}$	$51.28_{\pm 0.75}$	$89.96_{\pm 0.03}$	$92.32_{\pm 0.35}$	$91.12_{\pm 0.16}$	$58.16_{\pm 0.21}$	$63.90_{\pm 0.96}$	$60.89_{\pm 0.36}$
	Word Swap + EM	$65.85_{\pm 1.86}$	$47.70_{\pm 0.99}$	$\mathbf{55.31_{\pm 0.27}}$	$89.96_{\pm 0.81}$	$93.00_{\pm 0.31}$	$91.46_{\pm 0.57}$	$56.64_{\pm 0.16}$	$62.35_{\pm 0.69}$	$59.36_{\pm 0.28}$
	Spelling Augmentation + EM	$63.41_{\pm 2.80}$	$46.35_{\pm 2.20}$	$53.49_{\pm 0.50}$	$90.38_{\pm 0.21}$	$92.92_{\pm 0.51}$	$91.64_{\pm 0.36}$	$58.59_{\pm 0.42}$	$64.23_{\pm 0.75}$	$61.28_{\pm 0.52}$
	MELM et al. [33] + EM	$65.96_{\pm 0.28}$	$44.54_{\pm 0.90}$	$53.17_{\pm 0.74}$	$88.91_{\pm 0.18}$	$91.53_{\pm 0.51}$	$90.20_{\pm 0.17}$	$56.76_{\pm 0.29}$	$62.44_{\pm 0.78}$	$59.47_{\pm 0.40}$
	DAGA et al. [6] + EM	$62.03_{\pm 4.26}$	$47.54_{\pm 3.88}$	$53.64_{\pm 1.58}$	$89.68_{\pm 0.48}$	$91.69_{\pm 0.48}$	$90.67_{\pm 0.48}$	$58.42_{\pm 0.65}$	$63.64_{\pm 0.85}$	$60.92_{\pm 0.73}$
	Keyword Augmentation + EM	$61.07_{\pm 4.34}$	$46.33_{\pm 0.50}$	$52.64_{\pm 1.55}$	$86.79_{\pm 0.46}$	$91.26_{\pm 0.23}$	$88.97_{\pm 0.31}$	$54.67_{\pm 0.65}$	$60.85_{\pm 0.93}$	$57.59_{\pm 0.73}$
	Context Augmentation + EM	$66.38_{\pm 1.46}$	$44.62_{\pm 1.12}$	$53.37_{\pm 1.27}$	$90.61_{\pm 0.39}$	$92.12_{\pm 0.06}$	$91.36_{\pm 0.17}$	$58.80_{\pm 0.87}$	$64.15_{\pm 0.29}$	$\mathbf{61.35_{\pm 0.55}}$

6.2 Entity Extraction Datasets

We report the results on three entity extraction datasets to better evaluate the proposed approach. As explained in Section 3.4, the GUM dataset consists of long entities with an average entity length of 3.15, whereas WNUT-17 and CoNLL-2003 datasets consist of short entities where the average entity length is 1.73 and 1.60 respectively.

In the GUM dataset, the proposed contextual augmentation technique with embedding manipulation produces the overall best F_1 score compared to baselines for both models, i.e., BERT and RoBERTa. The RoBERTa model with EM fine-tuned on contextual augmented dataset achieves the highest F_1 score of 61.35 with a performance improvement of 7.30% over the baseline counterpart.

Table 5: Comparison of soft skills predictions on examples randomly sampled from the test set. We choose the best-performing model from Table 3, i.e. RoBERTa. The Baseline Model column shows predictions of the RoBERTa model fine-tuned with the baseline dataset, GPT-4 column shows the predictions of GPT-4 on test examples. The Data-Augmented Model shows the results of the RoBERTa model fine-tuned with augmented-dataset and EM. Highlighted texts stand for gold labels in the first column, and the corresponding predictions by the models.

№	Gold Labels	Baseline Model	GPT-4	Data-Augmented Model
1.	Share your knowledge and keep up the high level of quality in our team through reviews pairing and mentoring	Share your knowledge and keep up the high level of quality in our team through reviews pairing and mentoring	Share your knowledge and keep up the high level of quality in our team through reviews pairing and mentoring	Share your knowledge and keep up the high level of quality in our team through reviews pairing and mentoring
2.	Continuously improve and maintain components and systems to ensure its functionality scalability uptime and security	Continuously improve and maintain components and systems to ensure its functionality scalability uptime and security	Continuously improve and maintain components and systems to ensure its functionality scalability uptime and security	Continuously improve and maintain components and systems to ensure its functionality scalability uptime and security
3.	That means defining and implementing services that make up a vital sub-system in your area	That means defining and implementing services that make up a vital sub-system in your area	That means defining and implementing services that make up a vital sub-system in your area	That means defining and implementing services that make up a vital sub-system in your area
4.	NET services and APIs according to product specifications; Ensure to maintain strong Backend code quality with good practices	NET services and APIs according to product specifications; Ensure to maintain strong Backend code quality with good practices	NET services and APIs according to product specifications; Ensure to maintain strong Backend code quality with good practices	NET services and APIs according to product specifications; Ensure to maintain strong Backend code quality with good practices
5.	supervision and other relevant management functions.	supervision and other relevant management functions.	supervision and other relevant management functions.	supervision and other relevant management functions.

The performance of the proposed DA techniques on the entity extraction WNUT-2017 dataset is reported in Table 4. We can observe that contextual augmentation can achieve better performance over the baseline. However, the performance of contextual DA is marginally less than the baseline augmentation approaches. The RoBERTa model finetuned on augmentation generated by swapping random words and using EM achieves the highest F_1 score of 55.31, which is a 1.2% improvement over the contextual DA.

On the other entity extraction dataset (CoNNL-2003), the augmentation is not as effective as in the previous case of (GUM and SKILLSPAN). We notice a similar trend as in the WNUT-17 dataset. From Table 4, we observe that all augmentation techniques show marginal improvement over the baseline model. The RoBERTa model fine-tuned on a dataset augmented with word deletion and using EM gains 0.42% performance over the baseline.

We remark that the performance of data augmentation is marginally effective (WNUT-17) and, in some cases, degrades (CoNLL-2003). However, the performance gain is significant for SKILLSPAN and GUM datasets.

It can be observed that, for both datasets, the model/augmentation technique that reaches the best F_1 score in the validation set returns the best results on the test set as well. This fact supports the conclusion that the assessment of the NER datasets can be considered positive overall.

7 INFERENCE

Table 5 shows the inference on the test set. We compare the predictions of the RoBERTa model fine-tuned on the baseline and augmented dataset against the gold labels. Both models correctly predict Example 1. In Examples 2 and 3, the baseline model predicts some extra tokens not annotated in the gold labels, whereas the augmented model predicts the tokens correctly. Likewise, in Example 4, both models fail to predict the tokens "NET services and APIs ..."; however, the second part of the sentence is predicted correctly by the augmented model, showing superior performance over the baseline. Example 5 reveals interesting outcomes: the input phrase "supervision and other relevant management functions" would be considered a soft skill in general. However, due to human mistakes, it is not labeled as such in the gold dataset. The augmented model is still able to denote it as a SS. The GPT-4 model can extract a few tokens corresponding to SS in Examples 1, 2, and 5, whereas it fails to identify any SS in Examples 3 and 5.

8 CONCLUSIONS AND FUTURE WORK

We have presented a simple yet effective approach for improving the performance of Large Language Models when the entity length is long and gold annotations are limited. We demonstrate our approach on four different token classification datasets. By using DA techniques, we generated synthetic data that improved the performance of the baseline models. We also exploit the gold-annotated datasets to extract additional information via EM. We demonstrate that the proposed approach can effectively improve the SS extraction from job descriptions and CVs. This approach can help to speed

up the recruitment process by automatically extracting information about soft skills from candidate documents without requiring expensive and time-consuming manual annotations. However, it is worth noting that more research is needed to improve these models' performance further and test their effectiveness in a real-world scenario. Additionally, in future work, we plan to investigate how to use these models fairly and ethically to ensure that they do not perpetuate existing biases in the recruitment process.

9 ACKNOWLEDGEMENT

We gratefully acknowledge the support of the AMAJOR SRL SB. for sponsoring the research and providing the feedback. We also thank University of Padova for providing the necessary computational resources to carry out the research.

REFERENCES

[1] Wahiba Karra Ben Abdessalem and Soumaya Amdouni. 2011. E-recruiting support system based on text mining methods. *International Journal of Knowledge and Learning* 7, 3-4 (2011), 220–232.

[2] Christian Bizer, Ralf Heese, Malgorzata Mochól, Radoslaw Oldakowski, Robert Tolksdorf, and Rainer Eckstein. 2005. The Impact of Semantic Web Technologies on Job Recruitment Processes. In *Wirtschaftsinformatik*.

[3] Xiang Dai and Heike Adel. 2020. An Analysis of Simple Data Augmentation for Named Entity Recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 3861–3867. https://doi.org/10.18653/v1/2020.coling-main.343

[4] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*. Association for Computational Linguistics, Copenhagen, Denmark, 140–147. https://doi.org/10.18653/v1/W17-4418

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv* abs/1810.04805 (2019).

[6] Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. 2020. DAGA: Data Augmentation with a Generation Approach for Low-resource Tagging Tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6045–6057. https://doi.org/10.18653/v1/2020.emnlp-main.488

[7] S. Fareri, Nicola Melluso, Filippo Chiarello, and Gualtiero Fantoni. 2021. SkillNER: Mining and Mapping Soft Skills from any Text. *Expert Syst. Appl.* 184 (2021), 115544.

[8] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. 2021. A Survey of Data Augmentation Approaches for NLP. *ArXiv* abs/2105.03075 (2021).

[9] Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. Soft Contextual Data Augmentation for Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5539–5544. https://doi.org/10.18653/v1/P19-1555

[10] Hugging Face. 2023. Transformers APIs. https://huggingface.co/docs/transformers/index. Accessed: 2023-01-21.

[11] Janez Demsar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* 7 (2006), 1–30. https://api.semanticscholar.org/CorpusID:7553535

[12] Imane Khaouja, Ismail Kassou, and Mounir Ghogho. 2021. A Survey on Skill Identification From Online Job Ads. *IEEE Access* 9 (2021), 118134–118153.

[13] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).

[14] Sosuke Kobayashi. 2018. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 452–457. https://doi.org/10.18653/v1/N18-2072

[15] Pei Liu, Xuemin Wang, Chao Xiang, and Weiye Meng. 2020. A Survey of Text Data Augmentation. In *2020 International Conference on Computer Communication and Network Security (CCNS)*. 191–195. https://doi.org/10.1109/CCNS50731.2020.00049

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv* abs/1907.11692 (2019).

[17] Edward Ma. 2019. NLP Augmentation. https://github.com/makcedward/nlpaug.

[18] George A. Miller. 1994. WordNet: A Lexical Database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*. https://aclanthology.org/H94-1111

[19] Lamiaa Mostafa and Sara Beshir. 2021. Job Candidate Rank Approach Using Machine Learning Techniques. In *AMLTA*.

[20] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[21] Luiza Sayfullina, Eric Malmi, and Juho Kannala. 2018. Learning Representations for Soft Skill Matching. In *International Joint Conference on the Analysis of Images, Social Networks and Texts*.

[22] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 86–96. https://doi.org/10.18653/v1/P16-1009

[23] Oleksii Shatalov and Nataliya Ryabova. 2021. Named Entity Recognition Problem for Long Entities in English Texts. In *2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*, Vol. 1. 76–79. https://doi.org/10.1109/CSIT52700.2021.9648768

[24] Stefan Strohmeier. 2022. *Handbook of Research on Artificial Intelligence in Human Resource Management*. Edward Elgar Publishing.

[25] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. 142–147. https://www.aclweb.org/anthology/W03-0419

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[27] Jason Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. 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)*. Association for Computational Linguistics, Hong Kong, China, 6382–6388. https://doi.org/10.18653/v1/D19-1670

[28] Ivo Wings, Rohan Nanda, and Kolawole John Adebayo. 2021. A Context-Aware Approach for Extracting Hard and Soft Skills. *Procedia Computer Science* (2021).

[29] Kun Yu, Gang Guan, and M. Zhou. 2005. Resume Information Extraction with Cascaded Hybrid Model. (2005).

[30] Amir Zeldes. 2017. The GUM Corpus: Creating Multilayer Resources in the Classroom. *Language Resources and Evaluation* 51, 3 (2017), 581–612. https://doi.org/10.1007/s10579-016-9343-x

[31] Mike Zhang, Kristian Nørgaard Jensen, Sif Dam Sonniks, and Barbara Plank. 2022. SkillSpan: Hard and Soft Skill Extraction from English Job Postings. In *North American Chapter of the Association for Computational Linguistics*.

[32] Mike Zhang, Kristian Nørgaard Jensen, Rob van der Goot, and Barbara Plank. 2022. Skill Extraction from Job Postings using Weak Supervision. *ArXiv* abs/2209.08071 (2022).

[33] Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. MELM: Data Augmentation with Masked Entity Language Modeling for Low-Resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 2251–2262. https://doi.org/10.18653/v1/2022.acl-long.160