UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Improving Skill Extraction from Job Postings Using Synthetic Data and Advanced Language Models

Author:
Andrii MYRONENKO

Supervisor:
Hanna PYLIEVA

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2024

Declaration of Authorship

I, Andrii MYRONENKO, declare that this thesis titled, "Improving Skill Extraction from Job Postings Using Synthetic Data and Advanced Language Models" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

Improving Skill Extraction from Job Postings Using Synthetic Data and Advanced Language Models

by Andrii MYRONENKO

Abstract

Skill extraction involves the automated identification and categorization of skills from textual data, such as job postings, and is important for improving human resource management and job market analysis. In this thesis, we examine existing research on skill extraction from job postings, highlighting the primary challenges and methods used in the field. Following this review, we propose utilizing Large Language Models (LLMs) for skill extraction, specifically formulated as a sequence labeling task. We will evaluate their out-of-the-box performance and explore the potential of using synthetically generated data by these advanced LLMs to improve the performance of smaller, more efficient Domain-Specific Models.

Acknowledgements

I would like to thank my advisor, Hanna Pylieva, for her guidance throughout my research. I also want to acknowledge Oleksii Molchanovskyi for the creation of this Data Science program, and Ruslan Partsey for his management and assistance during my education at UCU.

Additionally, I am grateful to the AresAI platform for providing computation resources that were essential for my research.

Finally, I must thank my family, friends, and loved ones for their support. I couldn't have completed this work without you.

Contents

List of Figures

List of Tables

List of Abbreviations

LLM	**L**arge **L**anguage **M**odel
GPT	**T**enerative **P**re-trained **T**ransformer
BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
ISCO	**I**nternational **S**tandard **C**lassification of **O**ccupations
ESCO	**E**uropean **S**kills, **C**ompetences, Qualifications and **O**ccupations
BLEU	**B**i**L**ingual **E**valuation **U**nderstudy
NER	**N**amed **E**ntitiy **R**ecognition

Chapter 1

Introduction

1.1 Motivation and Goals

Skill Extraction (SE) is an actively researched domain in the field of labor market analysis, instrumental for various human resources (HR) functions such as talent acquisition, workforce management, and career development planning. Beyond HR, SE can also be helpful for policymakers who can leverage these insights for informed decision-making in areas like education, employment, and economic development.

From a Natural Language Processing (NLP) standpoint, the task of SE can be approached through multiple methodologies.

One method is matching resumes or job postings (JP) with a predefined skill set from established ontologies, such as the International Standard Classification of Occupations (ISCO)[1] or the European Skills, Competences, Qualifications and Occupations (ESCO)[2]. It offers a more systematic and standardized approach to job market assessment. (Bhola et al., 2020; Clavié et al., 2023; Decorte et al., 2022)

Another method involves identifying specific text segments within documents that represent skills. This process requires detailed textual analysis to locate and classify skill-related phrases, providing a granular view of the professional competencies explicitly mentioned in resumes or JPs. (Jia et al., 2018; Zhang et al., 2023; Zhang et al., 2022)

The recent advancements of transformer-based models, especially Large Language Models (LLMs), have significantly enhanced automatic analysis capabilities, setting new standards in quality. These models have revolutionized NLP tasks, enabling more sophisticated interpretation of complex language data. However, their full potential is yet to be realized due to several key factors: the most powerful models are proprietary, which makes them impossible to be deployed on controlled hardware, and the stochastic nature of these models poses challenges for getting reliable and structured output (Zhao et al., 2023).

We intend to focus our research on identifying text segments that contain skills. Essentially, we approach this problem as a sequence labeling task. We aim to explore the efficiency of modern LLMs in skill extraction especially given their "hallucination" problem. We will also look at how we can utilize them to train smaller encoder-only models, such as BERT, that are more accessible, cheaper to deploy, and widely used for sequence tagging (Pakhale, 2023).

[1] https://www.ilo.org/public/english/bureau/stat/isco/isco08

[2] https://esco.ec.europa.eu/en/classification/skill_main

1.2 Structure of the thesis

The structure of the paper is organized as follows: Chapter 2 discusses existing research in the field of skill extraction, both traditional methodologies and recent advances. Chapter 3 describes a specific area we aim to focus on that was under-researched before. This chapter will outline the scope of our novel contribution to the field, the research question, and the hypotheses. In this chapter, we will also discuss experimentation and evaluation methodologies. Chapter 4 describes the dataset that we will use for experimentation and methods of synthetic data generation. In Chapter 5, we present the results of the experimentation, and in Chapter 6, we discuss the results and outline the directions for further research.

Chapter 2

Literature review

A detailed review and classification of research in skill identification was conducted in (Khaouja et al., 2021). This paper provides an exhaustive summary of the studies in this field over the last decade, including references to numerous relevant publications.

2.1 Classical NLP Techniques for Skill Extaction

In the field of skill extraction or skill identification from JPs, four main methods are commonly recognized. These methods include Skill Count, Skill Embedding, Topic Modeling, and Deep Learning (DL)-based approaches (Khaouja et al., 2021). In this section, we will cover the first three, leaving DL methods for subsequent sections.

Skill Count, both manual and automatic, is a fundamental method for skill identification. Manual skill counting involves reading job-related texts and annotating them to identify skills, which is reliable but time-consuming. Automatic skill counting employs a technique that matches text spans directly to predefined skill databases, such as ESCO (Khaouja et al., 2021). The limitation of this method is its reliance on exact wording, which may overlook skills phrased differently in the texts.

The Skill Embedding technique involves training word embedding models on collections of job postings to produce vector representations of skills. This approach ensures that similar or co-occurring skills are closely represented in the vector space, allowing for more nuanced skill identification across different sectors. One application of this technique is an automatic skill detection system developed by Meng Zhao and Faizan Javed in 2015 (Zhao et al., 2015). In their system, they utilize Word2Vec (Mikolov et al., 2013), a popular word embedding model, to distinguish skills from other similar entities, thereby enhancing the precision. This system was evaluated through a user survey and demonstrated 80% accuracy in actual skill tagging with a 70% recall rate.

Topic Modeling algorithms analyze word distributions to identify key topics, which domain experts then interpret as skills. This method is particularly helpful in the absence of a taxonomy for skill selection. It has been used in various market studies, such as one conducted by Stephan Debortoli in 2014 (Debortoli et al., 2014), which compared the skills needed for Business Intelligence and Big Data specialists.

All previously outlined approaches, while valuable, come with limitations. Skill Count and Embedding rely on direct matches with existing skill taxonomies. Topic Modeling, on the other hand, requires interpretation by domain experts. To address these challenges, deep learning techniques have been developed, offering more robust and autonomous solutions for extracting skills from job postings.

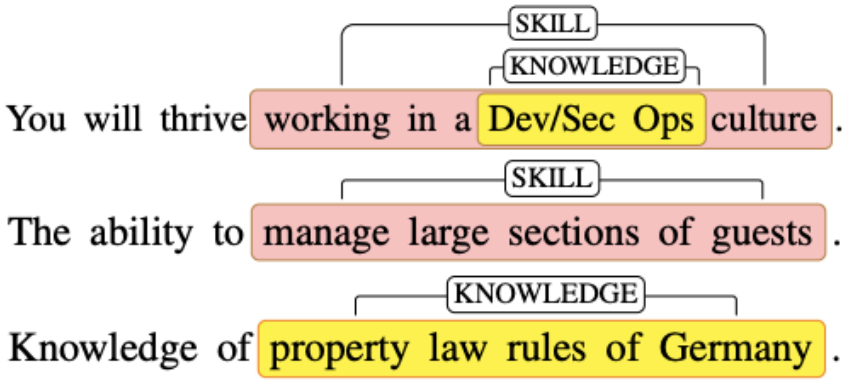

FIGURE 2.1: Skill and Knowledge Components. Adopted from Zhang et al., 2022. [2]

2.2 Skill Extraction using deep learning

The recent advancements in neural networks have significantly influenced the skill extraction domain. Two key approaches that are widely researched are Named Entity Recognition (NER) and text classification. NER, in the context of SE, involves identifying and extracting skill-related information from unstructured texts such as JPs and resumes. Text classification, on the other hand, involves assigning segments of text to predefined skills. This section will contain a review of research related to both of the above-mentioned approaches.

2.2.1 Named Entity Recognition

In 2016, Guillaume Lample introduced a novel neural network architecture using bidirectional Long Short-Term Memory (LSTM) networks specifically for Named Entity Recognition (NER) tasks (Lample et al., 2016). This approach laid the groundwork for subsequent applications in skill detection, as shown in Shanshan Jia's 2018 study (Jia et al., 2018). Jia employed LSTM initially to extract named entities, followed by Bi-gram algorithms for job entity linking. While this represented significant progress, the approach still required the use of additional algorithms alongside the neural network.

In 2022, a more sophisticated approach utilizing the BERT transformer model (Devlin et al., 2018) was employed in Mike Zhang's study (Zhang et al., 2022). Zhang's research not only focused on extracting Skills and Knowledge components, as defined in ESCO (See Fig. 2.1), from job postings but also involved the creation and release of an annotated dataset of job postings from general and IT domains called SkillSpan[1]. The study focused on the performance of BERT and SpanBERT (Joshi et al., 2019) models, trained on job domain data and fine-tuned on the annotated dataset for SE tasks. Despite its achievements, the research acknowledged certain limitations, such as challenges in detecting longer text spans.

In Zhang's latest 2023 paper, the focus shifted to developing a multilingual ESCOXLM-R model (Zhang et al., 2023), which set a new SOTA in performance on the SkillSpan dataset (Zhang et al., 2022), achieving a 3.7 point F1 score improvement over the previous SOTA (from 58.9 to 62.6). In their research, Zhang used ESCO ontology to

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

[2] License: CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

pre-train the XLM-R model, which was later further fine-tuned to a variety of tasks, including skill extraction. To our knowledge, this is the best-known result achieved using this approach.

2.2.2 Text Classification

Text classification serves as another effective method for skill extraction. In 2018, Luiza Sayfullina approached SE as a classification problem, distinguishing between sentences that contain skills and those that do not (Sayfullina et al., 2018). She evaluated various neural network models, including CNN (Kim, 2014), LSTM (Hochreiter et al., 1997), and the Hierarchical Attention Model (Yang et al., 2016), to determine their effectiveness in this context.

By 2020, Akshay Bhola advanced the field by employing the Extreme Multi-label Text Classification approach for skill detection (Bhola et al., 2020). They employed a BERT model trained to categorize job postings based on a list of 2542 predefined skills sourced from a job posting website. While Bhola's approach was effective, its applicability was limited to the specific skill list used.

Building on this, Jens-Joris Decorte introduced a novel end-to-end approach for skill extraction that did not depend on a manually labeled training corpus. Instead, this method utilized distant supervision to match skill literal in ESCO to the sentences from JPs (Decorte et al., 2022). The training involved several negative sampling strategies to address the issue of false positives. The resulting R-Precision@10 (RP@10) metric achieved 39.19 for Tech-related job postings and 38.69 for the general domain. Furthermore, for evaluation in the work they manually assigned ESCO labels to skills detected in the SkillSpan dataset (Zhang et al., 2022), or no label was assigned in cases where extracted skill contained no ESCO match. This dataset has been made publicly available for future research.[3]

2.2.3 Limitations

Finishing this section, it's important to acknowledge that the primary challenges for the discussed methods are centered around data collection and annotation. These processes are costly and require domain experts. Semi-automatic approaches, such as distant supervision, are based on literal matching. This reliance leads to problems in identifying skills that are expressed in different formulations.

2.3 Large Language models employed for Skill Extraction problem

In 2023 Nan Li introduces a tool called SkillGPT to use a Large Language Model (LLM) for SE from JPs (Li et al., 2023). SkillGPT summarizes texts and matches them to relevant ESCO skills using vector similarity. It supports multiple languages, addresses previous skill extraction challenges, and is economical for academic use. While innovative, it poses evaluation challenges due to the inherent stochastic nature of LLMs. This approach requires LLMs for summarization during inference, making its precise assessment difficult.

Complementing this, Benjamin Clavié and Guillaume Soulié, in their 2023 paper (Clavié et al., 2023) present an alternative method leveraging LLMs for SE. Their approach diverges from SkillGPT by focusing on the creation of synthetic training

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

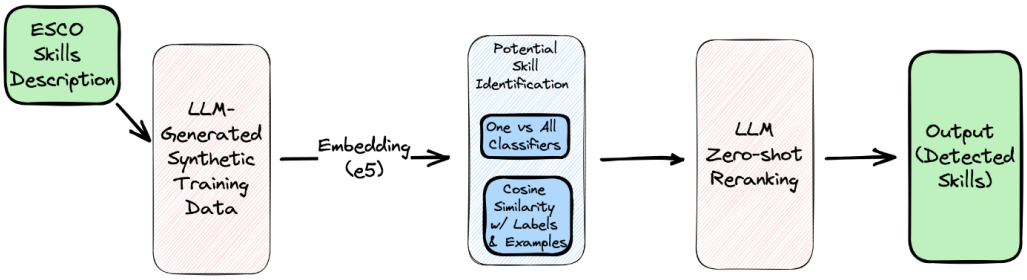

FIGURE 2.2: ESCO Skill Matching pipeline. Adopted from Clavié et al., 2023. [4]

data from the ESCO skillset. This method employs a blend of one-versus-all classifiers and similarity retrievers to propose candidate skills, which are subsequently refined and re-ranked using GPT-4. (See full pipeline schema in Fig. 2.2) This technique enhances efficiency and accuracy in skill extraction, surpassing previous models. RP@10 on the dataset released in is 61.02 for general domain JPs and 68.94 for tech, which is more than a 20-point improvement compared to the previous SOTA (Decorte et al., 2022).

Lastly in their 2023 research, Pouya Pezeshkpour generate synthetic data from a skill-occupation graph to train models for HR tasks, one of which is for skill-extraction from unlabelled texts Pezeshkpour et al., 2023. They call generated data resume-job description benchmark (RJDB). Skill extraction is approached here as a text-to-text problem where a resume is input, and the model outputs extracted skills. The paper explores the fine-tuning of three distinct T5 Raffel et al., 2019 models: one using only the Machop dataset Wang et al., 2022 (T5-M), another with a random RJDB sample (T5-R), and the third combining both datasets (T5-M+R). The focus is on the models' ability to generalize, specifically in extracting new, unlabeled skills from the Machop test set. Notably, based on the human evaluation, the T5-M+R model stands out for its ability to identify a significantly larger number of new skills with only a slight reduction in performance compared to T5-M, highlighting the benefits of combining datasets for enhanced SE.

2.4 Large Language Models for Named Entity Recognition

Considering the outstanding performance of Large Language Models across a spectrum of NLP tasks (Brown et al., 2020), it becomes important to examine their efficacy in the context of skill extraction. LLMs, essentially decoder-only models with a primary focus on text generation, could encounter specific challenges when applied to skill recognition if we formulate this task as an accurate identification of relevant text spans. A significant challenge with LLMs, including the most recent ones like GPT-4, is their susceptibility to "hallucinations" (Achiam et al., 2024). A recent GPT-NER study proposes specialized prompting and self-verification techniques to reduce hallucination frequency (Wang et al., 2023b). The encouraging outcomes of this research make it reasonable to test the standalone capabilities of LLMs in skill extraction tasks.

[4] License: CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

2.5 Summary

Classical skill extraction methods like Skill Count and Embedding laid the groundwork in this sphere, but they faced limitations like dependency on exact literal skill definition matching. Modern machine learning approaches, including NER and text classification, have overcome many of these challenges, bringing more accuracy to skill identification. The recent invention of Large Language Models (LLMs) has further revolutionized this field, offering even more sophisticated and efficient extraction capabilities. Still, there is room for improvement, and future research could lead to even better performance in SE domain.

Chapter 3

Problem Setting and Approach to Solution

3.1 Research gap and proposed approach

While reviewing the literature, related to the research topic, we became particularly interested in the transformative impact of Large Language Models on the skill extraction task, where they have led to significantly improved outcomes, particularly in matching sentences to predefined skill taxonomies (Clavié et al., 2023). However, we identified a notable research gap: there is no study on the use of LLM-generated synthetic data within the context of a sequence labeling NER approach. The encouraging results observed in the fine-tuning of the T5 model (Pezeshkpour et al., 2023) suggest a potential for enhancing encoder-only BERT and similar models' performance with the use of the generated data. Research in this area will enhance model performance by eliminating the need for costly, expert-driven manual data annotation.

Additionally, we aim to assess the capabilities of LLMs in skill extraction, specifically examining their performance in this task without any specialized fine-tuning, on the open datasets available for research (Decorte et al., 2022; Zhang et al., 2022). This exploration could provide valuable insights into the standalone effectiveness of different LLMs in the SE domain.

3.2 Research hypotheses and questions

Based on the identified research gaps, we propose the following hypotheses for our research. Validation of these hypotheses will enable us to understand both the advantages and the constraints of the advanced generative abilities of LLMs within our specialized field:

1. **Hypothesis 1** The use of synthetic training data generated by state-of-the-art Large Language Models can effectively improve the performance of smaller, encoder models in skill extraction task. This approach will lead to substantial improvements compared to relying solely on human-annotated data.

2. **Hypothesis 2** LLMs are likely to demonstrate lower efficacy in standalone skill extraction evaluations using human-annotated data compared to supervised-trained models due to the complexity and variability of skills and potential hallucination issues.

To prove or reject the above-mentioned hypotheses, we formulated a set of narrower research questions we will strive to answer in our thesis:

1. **Research Question 1** Is it feasible for LLMs to generate a versatile synthetic dataset of sentences that incorporate skills in context, given a specific skill and occupation (potentially sourced from the ESCO ontology)?
2. **Research Question 2** What are the most effective prompting techniques to ensure consistent, high-quality results in skill extraction using Large Language Models of different sizes, and how can these methods be optimized to mitigate the issue of hallucinations?
3. **Research Question 3:** How do the outcomes of fine-tuning encoder models with only human-labeled data compare to those achieved by combining both synthetic and human-annotated data?
4. **Research Question 4** At what point does the volume of synthetic data become sufficient for fine-tuning, and how many skills should be included to maximize performance improvements?

3.3 Research plan

To begin with the research, we have to define our metrics and select an appropriate dataset and models for evaluation.

3.3.1 Selected dataset

The literature review indicates that the SkillSpan dataset (Zhang et al., 2022) is the most suitable for our task. The dataset is divided into three parts, with two publicly available. The first contains labeled job postings related to IT sourced from Stack Overflow, while the second contains vacancies from various professional fields. This dataset has been annotated by a team of three, including a professional linguist. The annotators adhered to publicly released guidelines based on the definitions of Skill[1] and Knowledge[2] from the ESCO ontology. This dataset allows us to evaluate models in the detection of both skill and knowledge entities separately.

3.3.2 Metrics

As for the metrics, we will use span-level Recall, Precision, and F1 score, where the prediction span is counted as a true positive (TP) only if it matches exactly the annotated span. The span is counted as a false negative (FN) if it was not predicted, and the prediction is counted as a false positive (FP) if an incorrect span is predicted.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Where

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

[1] https://esco.ec.europa.eu/en/about-esco/escopedia/escopedia/skill
[2] https://esco.ec.europa.eu/en/about-esco/escopedia/escopedia/knowledge

3.3.3 Evaluation of Standalone LLM Performance

After obtaining the dataset and selecting the appropriate metrics, the next step is to evaluate the performance of LLMs for skill extraction. To achieve this, we will prompt the models to extract skills and knowledge entities on the test set using Skill and Knowledge definitions from ESCO.

Our preferred language models are GPT-4 and GPT-3.5; the former because it is the most powerful, and the latter for its balance between quality and cost-efficiency. The primary challenge is getting consistent and reproducible results because of the stochastic nature of Large Language Models and the "hallucination" problem. Here, the results of the GPT-NER (Wang et al., 2023b) paper might be useful.

3.3.4 Synthetic data generation

The main part of our research is to create synthetic data utilizing LLMs and ESCO ontology. This involves injecting skills, knowledge components, and occupations from the ontology into prompts, and prompting the LLM to generate sentences resembling the ones that can be encountered in real job postings. The reason behind using this approach is that although LLMs may not be suitable for accurately annotating skill and knowledge components, they can efficiently generate sample job-posting context around the predefined skill.

An important research challenge is ensuring the synthetic data is contextually rich and accurately mirrors the complexity of actual job postings. Furthermore, the data must exhibit sufficient versatility to effectively train models to generalize. To assess this, we plan to use human evaluations to determine if the generated data is a viable job posting part and employ metrics like Self-BLEU (Zhu et al., 2018) for evaluating the generated text's diversity. Specifically, we will use the fast-bleu (Alihosseini et al., 2019) implementation [3].

3.3.5 Experimentation

After generating the synthetic data, we plan to use it to augment the SkillSpan training dataset and train a BERT-base model using it to extract skills. We will compare the performance of the model trained only on real data with the one trained using an augmented dataset.

BERT-base (bert-base-cased [4]) was selected because this model is relatively small and allows us to iterate quickly by doing experiments with different amounts of synthetic data.

To determine the optimal quantity of synthetic data for model training, we will use the following approach:

1. Generate 100 samples using a specific prompting method.
2. If this method shows promise based on Self-BLEU scores and human evaluation, we then produce synthetic data equal to 20% of our training set size. We train the BERT model by combining human and synthetic data.
3. If the BERT model demonstrates improvement after training with the synthetic data, generate and use more synthetic data. Evaluate the impact of newly added samples on model performance.

[3] https://github.com/Danial-Alh/fast-bleu
[4] https://huggingface.co/google-bert/bert-base-cased

The method proposed is inspired by a paper (Kaddour et al., 2023). In this research, the authors used a mix of subsets of real and synthetic data to fine-tune models for text classification. The difference in our approach is that we add synthetic data to the entire training set. Considering the vast number of skills and the fact that existing human-labeled data includes only a portion of all possible skills, we believe this approach will work.

3.4 Summary

To summarize, our research plan can be outlined as follows:

1. **Dataset Generation:** Review SkillSpan and ESCO datasets that will be used for the research. Test different prompting techniques for synthetic data generation to enrich training data.

2. **Reproduce Existing Research:** This step involves replicating the previous research using the original dataset. Since only two out of three parts of the dataset are publicly available, we need to train and evaluate BERT using them to set up the baseline.

3. **Construct LLM Evaluation Pipeline:** Develop a pipeline for evaluating the standalone performance of Large Language Models for skill extraction.

4. **Experimentation:** Use the pipeline from the previous step to test GPT-3.5 and GPT-4 performance on the SkillSpan dataset. Train BERT model using augmented training data, evaluate the performance, and compare the results with the baseline.

As a result, we aim to develop a model that significantly improves skill extraction capabilities compared to the baseline performance of BERT fine-tuned on the SkillSpan dataset.

Chapter 4

Dataset Examination and Augmentation

In this chapter, we will analyze the SkillSpan and ESCO datasets and explore various strategies for enriching the former with the latter. We will identify a list of promising augmentation approaches and generate datasets using these methods so we can later utilize them to train a model with improved performance in the skill extraction task.

4.1 Dataset analysis

4.1.1 SkillSpan

As discussed in the previous chapter, the SkillSpan dataset selected for our experiments consists of two publicly available parts. The first part primarily includes tech sector postings, referred to as TECH in the original research. The second part, sourced from a governmental agency and referred to as HOUSE, contains vacancies from various domains. Sentences in the dataset are formatted in BIO format, where skills and knowledge entities are annotated separately.

The original paper (Zhang et al., 2022) examines the dataset's structure in terms of the total number of skills and knowledge components and provides relevant statistics. However, it offers limited information on the structure of individual sentences. We aim to address this gap in Table 4.1, which provides both the most important statistics from the original paper as well as an analysis of the dataset at the sentence level.

TABLE 4.1: Descriptive Statistics of SkillSpan Dataset by Category

Parameter	House	Tech	Total
N of Training Sentences	1,674	3,156	4,830
N of Development Sentences	1,022	2,187	3,209
N of Test Sentences	1,216	2,352	3,568
N of Total Sentences	3,912	7,695	11,607
Average Sentence length	12.86	19.98	15.27
% Sentences with Skills/Knowledge	28.24%	33.66%	30.07%
N of Sentences with Skills	1,200	987	2,187
N of Sentences with Knowledge	1,326	575	1,901
Average Skill length (words)	4.16	4.58	4.37
Average Knowledge length (words)	2.12	1.74	1.84
N of Unique Skills	1,965	1,878	3,757
N of Unique Knowledge	2,083	1,091	3,055

The table indicates that approximately one-third of the sentences in the annotated job postings contain references to skills or knowledge. To maintain this ratio in the synthetic data, we will generate approximately two sentences without entities for every sentence containing them. According to the data, we need to create 966 samples to constitute 20% of the training data.

Additionally, it would be useful to compare the lengths of skills and knowledge entries in the annotated dataset with those in the ESCO ontology. This will help us understand if they can be used interchangeably.

4.1.2 ESCO

For our research, we will utilize version 1.1.2 [1] of the ESCO dataset, which is available in 29 European languages. Our analysis will focus on the English version. This dataset contains a list of occupations divided into subcategories. It also includes a list of skills divided into two types: skills/competences (treated as a single, undifferentiated concept) and knowledge. The definitions of these two concepts were used to develop annotation guidelines for the SkillSpan dataset (Zhang et al., 2022). Each occupation in the ESCO dataset is associated with essential and non-essential skills and knowledge. Table 4.2 presents basic statistics of the ESCO dataset.

TABLE 4.2: Descriptive Statistics of the ESCO Ontology Dataset

Parameter	**Value**
N of Skill/competence components	10,831
N of Knowledge components	3,059
N of Occupations	3,006
N of Average Skill length (words)	3.87
N of Average Knowledge length (words)	2.65

Note: For skill and knowledge length we use preferredLabel column in the dataset.

4.1.3 Comparison

The tables show that mean knowledge and skill lengths differ between annotated and taxonomy datasets. Figure 4.1 reveals that annotators tend to select longer skill components than those in the ESCO dataset, with some outliers reaching up to 23 words. However, the median length remains at four words. In contrast, knowledge components in the SkillSpan dataset are usually shorter. This difference is largely due to the dataset's significant number of IT job postings, which often feature short terms like "Java" and "JavaScript."

Previous research (Decorte et al., 2022) shows that 64.5% of the components in the SkillSpan dataset match an ESCO entity. The significant match rate implies that ESCO can be a valuable resource for enriching the SkillSpan dataset. However, the difference in component lengths between the ESCO and SkillSpan datasets may suggest that the skills and knowledge concepts can be formulated differently in the real world compared to ESCO definitions. Therefore, when generating sentences, we should prompt the model to adapt the skills to sound naturally within the sentence.

[1] https://esco.ec.europa.eu/en/about-esco/escopedia/escopedia/esco-v112

FIGURE 4.1: Box Plots of Skill and Knowledge Components Word Count Distribution in ESCO and SkillSpan Datasets

4.2 Data Generation

This section describes the techniques we employed to generate synthetic data to enrich the training portion of the SkillSpan dataset. We used the **gpt-4-0613** model for data generation, accessed through the OpenAI API.

The central concept in all our strategies was to prompt the LLM to generate sentences with skills enclosed by @@ symbols and knowledge components surrounded by ## symbols. We then used these highlights to convert our generated sentences into the BIO format of the SkillSpan dataset for further training.

Before using the ESCO dataset for the data generation, we conducted a preprocessing step. We observed that the ontology often includes 'ICT' in the names of IT occupations and skills, such as 'ICT system administrator.' We removed these terms since they never appear in job postings from the training set.

It is also important to note that only IT professions with *iscoGroup* 25 were used in all approaches to enrich the TECH dataset. We did not apply this restriction to the HOUSE one, as it contains domains from various fields.

4.2.1 Zero-shot Generation

Our first data generation idea was to create synthetic data without using existing sentences from the original dataset. The generation algorithm looked like this:

1. Select a random occupation from the ESCO dataset.
2. Take two[2] random qualities associated with this occupation.
3. Constuct the prompt using the selected occupation and the skills (See the template in Appendix A.1).

The main reason we tried this approach is that we believed GPT-4 could generate diverse and original sentences and annotate skills that are given as references.

[2]Two is a median quantity of skills or knowledge components in the SkillSpan dataset sentences with at least one annotated quality.

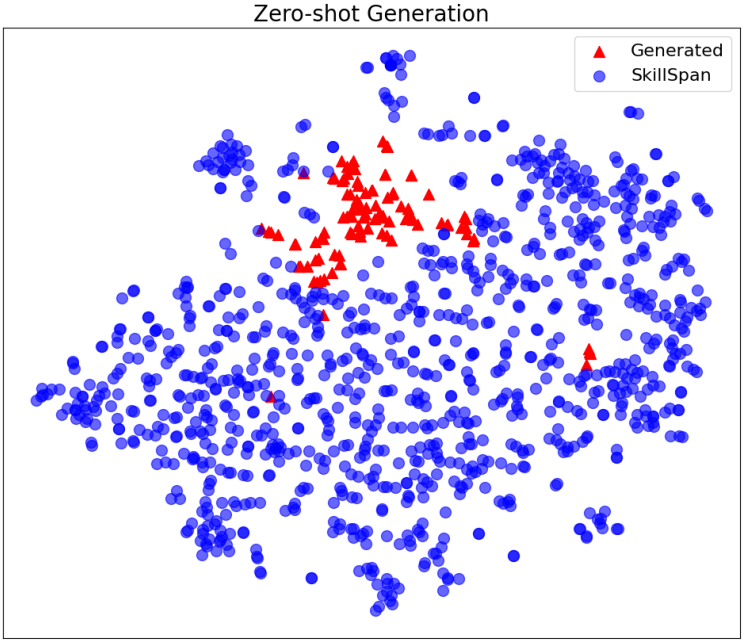

FIGURE 4.2: t-SNE Visualization of Synthetic Zero-shot vs. Real Training Data Embeddings

The model could do it with a temperature set to one. However, we encountered the problem that the model often inserted additional skills or knowledge into the generated sentences, even though it was explicitly prompted to avoid it. That made the generated sentences unsuitable for our task. We solved this issue by setting the temperature parameter to zero.

To evaluate the effectiveness of our algorithm, we generated a hundred sentences for IT occupations. See Table A.1 for the examples of the generated sentences.

While reviewing the results, we saw that many sentences were quite similar. For instance, ten samples began with "We are seeking" and twelve with "Seeking an." Such similarity could potentially worsen the quality of the training data. Self-BLEU-2 of 0.76 suggests that the diversity of the generated data is poor.

To analyze the data in more detail, we utilized sentence embeddings model *all-MiniLM-L6-v2*[3]. We generated the embeddings for both the generated sentences and the original training sentences from TECH dataset, which contain at least one annotated component.

Figure 4.2 displays a t-SNE visualization of these embeddings. It clearly shows that the generated sentences predominantly cluster in a specific plot region, which doesn't contain the original sentences. This clustering suggests that the generated sentences are too similar to each other and also different from the original sentences in the dataset; thus, training them may negatively affect BERT's performance on the test part of the dataset.

Based on the analysis, we have decided to stop the experimentation with Zero-shot data generation and try utilizing existing training samples for data generation instead.

[3] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

4.2.2 Template-based Generation

We developed a new strategy for data generation to enhance diversity by utilizing sentences from job postings as templates for incorporating ESCO skills. The algorithm for this strategy can be described as follows:

1. Select a random occupation from the ESCO dataset.
2. Select a random sentence from the training part of the SkillSpan dataset containing at least one annotated component
3. Select the same amount of skills or knowledge components related to the occupation as in the original sentence.
4. Prompt the model to insert it into the sentence doing only minimal grammatical adjustments using the prompt in Appendix A.2,

We initially generated one hundred synthetic sentences utilizing TECH dataset as in the previous experiment. We assessed the diversity of these sentences and noted a significant improvement; the Self-BLEU-2 score, this time, was 0.46. Moreover, the visualization of the embeddings for the generated sentences, alongside real training samples (see Figure 4.3), shows that the synthetic data no longer forms a separate from the original data cluster.

Considering the improved quality, we chose this approach for further experimentation. For this, we generated a number of synthetic sentences equivalent to 20% of the training samples in both HOUSE and TECH datasets. Later in this text, we will refer to this dataset as the TEMPLATE-BASED one.

Analyzing the sentences, we found that most of the generated examples make sense and could be a part of the job postings for the occupations not present in the training data (Examples in Table A.2). However, sometimes, as in the fourth and fifth rows of the table, the reference text and randomly selected skills are so different that the model struggles to create a coherent sentence. We tried to address this in the next data generation strategy.

4.2.3 Template-based Generation with Rephrasing

This method aims to resolve issues noted in the previous experiment, such as incoherent sentences, to improve the quality of the context for annotated skills and knowledge components. The algorithm we designed for this method looks like this:

1. Using previously generated samples, construct a validation prompt to automatically assess which sentences are coherent and can be used in training without additional rephrasing and which can't. The prompt is in the Appendix A.3.1.
2. Rephrase the sentences detected as problematic in the previous step by prompting GPT to rephrase them through the prompt (Appendix A.3.2).
3. Evaluate the rephrased sentences using the same prompt as in step one to determine that our approach works.

The primary challenge in implementing this method was crafting a prompt that effectively rephrases sentences without breaking the annotations. The results displayed in Table A.3 show the improvement in the coherence of the selected sentences.

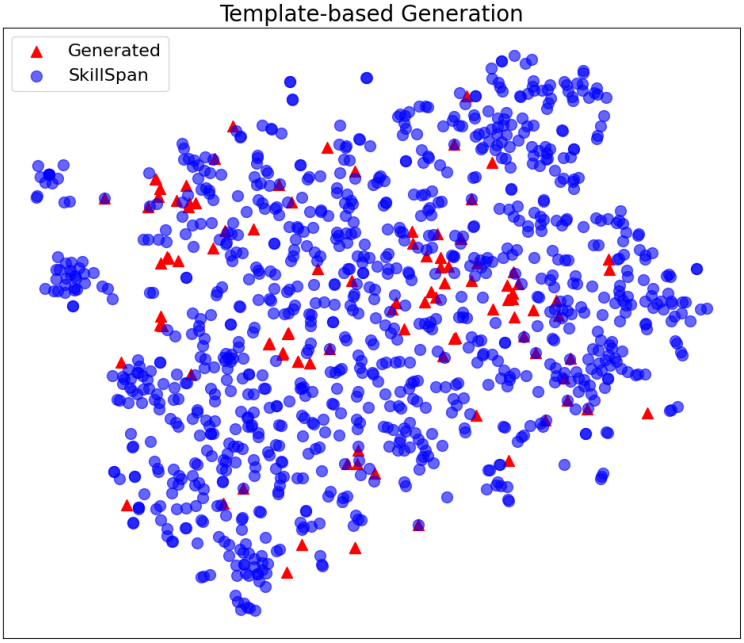

FIGURE 4.3: t-SNE Visualization of Synthetic Template-based vs. Real Training Data Embeddings

One issue that persists can be observed in row five. Some sentences remain odd after rephrasing. This issue happens because randomly selected skills do not always fit the training sentences. For example, here, we tried to insert the skills associated with the "audio and video equipment specialised seller" occupation into the sentence from the "Associate Professor" job posting.

In Table 4.3, you can see statistics for the number of unconnected sentences before and after rephrasing. We completely eliminated unconnected sentences in the synthetic TECH dataset and halved them in the HOUSE dataset. The issue persists in the HOUSE dataset because it includes skills from all available ESCO occupations, whereas the TECH subset only includes IT-related ones.

TABLE 4.3: Automatic Evaluation of the Coherence of Generated Sentences.

Parameter	**Before Rephrasing**	**After Rephrasing**
N of Incoherent Sentences (TECH)	16	0
% Incoherent Sentences (TECH)	7.84%	0%
N of Incoherent Sentences (HOUSE)	23	11
% Incoherent Sentences (HOUSE)	22.11%	10.58%

Overall, we consider this improvement substantial and will use the data generated with this approach in BERT training experiments. Later in this thesis, we will refer to the data created using this approach as the TEMPLATE-BASED WITH REPHRASING dataset.

4.2.4 Generation of Training Sentences without Skills

In order to keep the dataset balanced and avoid introducing bias by generating only sentences containing Skill or Knowledge components, we needed a strategy for generating ones without them.

We decided to use the same GPT-4 model for the generation and to prompt it to rephrase real sentences from SkillSpan (See Appendix A.4). In the same manner, as in the previous sections, we generated a quantity equal to 20% of the original dataset's size and added these sentences to both TEMPLATE-BASED and TEMPLATE-BASED WITH REPHRASING datasets. The examples are in the Table A.4.

4.2.5 Summary

In this section, we explored the structure of the SkillSpan and ESCO datasets and discussed how ESCO can be used to enrich SkillSpan. We also outlined the strategies we employed to create two synthetic datasets: TEMPLATE-BASED and TEMPLATE-BASED WITH REPHRASING. In the next chapter, we will conduct experiments using these synthetic datasets in combination with the original data to improve performance in the skill extraction task.

Chapter 5

Experiments and Results

This chapter describes the experiments with different models on Skill Extraction and their results. At first, we trained BERT using the public SkillSpan dataset to establish a baseline for further experimentation. Then, we assessed the performance of GPT-3.5 and GPT-4 models using the test part of the same dataset. Lastly, we experimented with BERT training using data augmented with the generated sentences and evaluated the results of this approach.

5.1 Experimental Setup

For the training of **BERT-base**, we utilized the code provided by Zhang et al., 2022, which is accessible in the GitHub repository[1]. For information about the hyperparameters, please refer to the original paper. Hardware used for the experimentation is Intel® Xeon® Gold 6442Y CPU and 1/4 of NVIDIA® H100 Tensor Core GPU. Each experiment was conducted five times with the seeds 3044792, 4236855, 6676809, 8679308, and 9979325.

For the performance evaluation of LLMs, we utilized the code released along with the GPT-NER paper by Wang et al., 2023b. We evaluated both **gpt-3.5-turbo-instruct** and **gpt-4-0613** on skill extraction using this approach. Since the original code does not support GPT-4, we adapted it to accommodate this model. The forked version, along with the adapted code and datasets used for evaluation, is available in our repository[2].

Lastly, all the code for the generation of synthetic datasets, along with the actual generated data used in the experiments, as well as the code for the analysis of the datasets, can be found in the repository [3].

5.2 Baseline Evaluation

Before the experimentation we needed to establish a baseline to evaluate the impact of synthetic data on model performance, as well as compare LLM performance to a supervised model. The results from Zhang et al., 2022 weren't suitable because their study assessed performance using the entire SkillSpan dataset, while we had access only to the two publicly released parts.

As concluded in the original research, single-task learning outperformed multi-task learning in Skill Extraction. Therefore, we trained separate models for skill and knowledge extraction, each five times, using the combined House and Tech datasets. Table 5.1 presents the baseline performance on each part of the dataset.

[1] https://github.com/kris927b/SkillSpan
[2] https://github.com/andrii-myronenko/GPT-NER
[3] https://github.com/andrii-myronenko/skill-extraction-diploma

TABLE 5.1: Baseline Performance of BERT on SkillSpan Test Dataset

Entity Type	House	Tech
Skill	49.47±0.78	48.52±1.45
Knowledge	54.81±1.03	67.53±0.70

*Note: The values represent span-level **F1 scores** and their standard deviations across five evaluations for each entity type.*

Overall, the results were lower than those reported for the BERT model in the original research due to the smaller sizes of the dataset used in our study.

5.3 LLMs for Skill Extraction

In this section, we aimed to evaluate modern LLMs like GPT-3.5 and GPT-4 in Skill Extraction. One of the main challenges in evaluation is to obtain structured results from the language models' text generation. To address this, we utilized NER-style prompting (Wang et al., 2023b) due to its demonstrated efficacy in named entity recognition tasks.

For the experiment, we used 4-shot in-context learning. We selected four training examples based on their similarity to the currently evaluated sentence using k-Nearest Neighbors (kNN) and incorporated them into the prompt. The similarity was determined by comparing embeddings generated using the BERT-base model with the SimSCE framework (Gao et al., 2021).

In the training examples used in the prompts, we included only sentences that contained named entities, meaning we selected the nearest neighbors exclusively from such sentences. This approach was chosen based on our experimentation with prompting, which showed the best results when there were more examples of sentences containing detected skills. See the example of the prompt in Appendix B.1.1.

Also, following the original research, we tested a self-verification prompt to improve the models' performance. You can find the example of this prompt in Appendix B.1.3.

TABLE 5.2: Performance of Large Language Models on SkillSpan Test Dataset

Entity Type	Model	House	Tech
Skill	**GPT 3.5**	24.84	18.28
	GPT 3.5$_{+ \text{ verification}}$	**31.33**	**29.34**
	GPT 4	22.56	18.18
	GPT 4$_{+ \text{ verification}}$	24.41	21.31
Knowledge	**GPT 3.5**	25.64	41.73
	GPT 3.5$_{+ \text{ verification}}$	28.51	49.10
	GPT 4	29.03	39.13
	GPT 4$_{+ \text{ verification}}$	**39.77**	**52.39**

Note: GPT-4 model is evaluated on the 5% sample of the dataset because of budget limitations
*Note: The values represent span-level **F1 scores**. Values in bold indicate the best LLM performance in the experiments.*

As shown in the Table 5.2. The LLMs, even the most capable ones, currently fail to surpass a supervised baseline. This may be attributed to several factors, which we discuss in Section 5.5.1.

5.4 Synthetic Data Experiments

5.4.1 Comparison of Data Generation Methods

The next portion of our experiments included the addition of synthetic data into the SkillSpan dataset. With the combined datasets, we trained BERT-base models to identify skill and knowledge components. During the synthetic data generation experiments, we generated two distinct datasets: TEMPLATE-BASED and TEMPLATE-BASED WITH REPHRASING. Each dataset constituted 20% of the SkillSpan Train dataset.

We observed promising results with a 20% increase in the size of the training dataset. Consequently, we expanded the dataset by an additional 20% and experimented with this portion as well. The results of the experiments are presented in Table 5.3.

TABLE 5.3: Performance of BERT Trained with Synthetic Data on SkillSpan Test Dataset

Entity Type	Dataset	House	Tech
Skill	**S**	49.47±0.78	48.52±1.45
	S + $T_{20\%}$	**50.68±0.82**	49.47±0.78
	S + $TR_{20\%}$	49.97±0.82	**49.99±1.59**
	S + $T_{40\%}$	49.68±1.76	48.85±1.92
	S + $TR_{40\%}$	50.27±1.43	49.70±2.04
Knowledge	**S**	54.81±1.03	67.53±0.70
	S + $T_{20\%}$	56.90±1.18	67.10±1.08
	S + $TR_{20\%}$	54.97±1.83	67.75±1.42
	S + $T_{40\%}$	**58.39±1.38**	68.12±0.52
	S + $TR_{40\%}$	56.21±1.42	**68.18±1.67**

Dataset Abbreviations: **S** - SKILLSPAN (original training dataset), **T** - TEMPLATE-BASED, **TR** - TEMPLATE-BASED WITH REPHRASING

*Note: The values represent span-level **F1 scores** and their standard deviations across five evaluations for each entity type. Values in bold indicate the best performance in the experiments.*

5.4.2 Effect of Increasing Synthetic Dataset Size on Performance

After noting that the TEMPLATE-BASED WITH REPHRASING method performs worse than the TEMPLATE-BASED method, we decided to conduct further experiments using TEMPLATE-BASED method by increasing the size of the synthetic set up to 100% of the training dataset. The results of this experiment are presented in the Table 5.4.

TABLE 5.4: Effect of Increasing Synthetic Dataset Size on BERT Performance for Skill Extraction

Entity Type	Dataset	House	Tech
Skill	**S**	49.47±0.78	48.52±1.45
	S + T$_{20\%}$	50.68±0.82	49.47±0.78
	S + T$_{40\%}$	49.68±1.76	48.85±1.92
	S + T$_{60\%}$	**51.15±1.13**	**50.21±1.98**
	S + T$_{80\%}$	50.47±2.69	49.47±2.42
	S + T$_{100\%}$	49.43±1.74	50.19±2.07
Knowledge	**S**	54.81±1.03	67.53±0.70
	S + T$_{20\%}$	56.90±1.18	67.10±1.08
	S + T$_{40\%}$	58.39±1.38	**68.12±0.52**
	S + T$_{60\%}$	**58.99±1.71**	67.75±0.74
	S + T$_{80\%}$	56.93±0.48	66.07±1.67
	S + T$_{100\%}$	57.90±1.97	67.31±1.17

Dataset Abbreviations: **S** - SKILLSPAN (original training dataset), **T** - TEMPLATE-BASED
*Note: The values represent span-level **F1 scores** and their standard deviations across five evaluations for each entity type. Values in bold indicate the best performance in the experiments.*

5.5 Results Discussion

5.5.1 LLMs for Skill Extraction

When we assessed the results of the experiments with LLMs, our findings aligned with the initial expectations. In the Skill Extraction task, both GPT-3.5 and GPT-4 perform worse than the supervised baseline.

We also observed that the self-verification prompt improves the performance of both models in every experiment. This suggests that more complex prompting techniques, such as the Chain-of-Thought (Wei et al., 2022), could further boost the efficacy of LLMs in SE. However, a significant challenge would still be getting consistent output from the models.

By analyzing the output, we found several error patterns that LLMs show within the test data:

1. **Irrelevant detections**. Sometimes, models detect terms completely unrelated to the definition of skills or knowledge in the prompts.

   Example of incorrectly detected skill: *You can @@choose to work either full time or part time##.*

   Example of incorrectly detected knowledge: *Manage and be part of all processes of a typical @@sales cycle## from A-Z .*

2. **Malformed responses**. Another portion of errors include the sentences that don't return the correct annotation, even though we specifically asked for this in a prompt.

   Example of malformed response: *You will be reporting to the @@Head of Sales .*

3. **Merged entities**. This error occurs when distinct skills are incorrectly detected as a single entity, failing to distinguish between them.

   Example of the error: *@@Danish, Swedish, and Norwegian## will be an advance because of our many customers in <LOCATION> .*

The possible reason for these errors is that GPT models are optimized for general-purpose text generation, and the NER task requires a specific format, which is much different from the gigabytes of text data the LLM was trained on. Even though Wang et al., 2023b reports that GPT models perform comparably to supervised models on NER tasks, our observations show that they significantly underperform in skill extraction, especially in extracting skill components. While these models efficiently extract shorter, one-to-two-word knowledge components, they struggle with detecting more complex entities like skills.

5.5.2 Experimentation with Synthetic Data

The results we obtained from adding synthetic data generated during the experiments to the training set were mixed. Table 5.5 presents the best performance on the HOUSE and TECH datasets for skill and knowledge extraction. This table includes only the models that demonstrated the highest performance among the experiments, along with the improvement compared to the baseline model, which was trained only on the real data and the statistical significance of these improvements, as determined by t-tests (Student, 1908). As shown, we achieved statistically significant improvements in knowledge detection on the HOUSE dataset. Additionally, we observed a slight yet significant improvement in skill detection on this dataset.

TABLE 5.5: Best models' performance for Skill Extraction

Evaluation Dataset	Training Dataset	Mean F1	Improvement	p-value
Skill HOUSE	**S + T**$_{60\%}$	51.15	1.68	**< 0.05**
Skill TECH	**S + T**$_{60\%}$	50.21	1.69	> 0.05
Knowledge HOUSE	**S + T**$_{60\%}$	58.99	4.18	**< 0.005**
Knowledge TECH	**S + TR**$_{40\%}$	68.18	0.65	> 0.05

Dataset Abbreviations: **S** - SKILLSPAN (original training dataset), **T** - TEMPLATE-BASED, **TR** - TEMPLATE-BASED WITH REPHRASING

Our initial plan to generate more training samples without using existing datasets didn't yield good results. The model we used for data generation (GPT-4) was surprisingly unreliable in generating the synthetic training dataset. And with the low temperature settings, the results weren't versatile enough to use in model training.

The second TEMPLATE-BASED approach produced better results. Here, our strategy was to employ GPT-4 to insert Skills and Knowledge components from ESCO into the sentences from the original datasets. It wasn't a simple insertion, but we also prompted the model to maintain grammar consistency to accommodate newly inserted entities.

Since this approach proved to be the best among those tested, we also experimented with increasing the dataset size up to 100% of the initial data size. The results showed that the model's performance peaks at 60%, with a decline observed when more synthetic data is added. We believe this decline happens because the model begins to overfit on the synthetic data.

Beyond the 60% threshold, the non-ideal quality of the added data starts to become more noticeable. This data may introduce noise and patterns that fail to accurately reflect the complexity of real-world data.

The third experiment (TEMPLATE-BASED WITH REPHRASING) involved prompting the generative model to rewrite the sentences significantly. We hoped rephrasing

would reduce the noise and boost the quality of our synthetic dataset. However, as seen in Table 5.3, this rephrasing didn't bring good results and worsened the model's performance in knowledge extraction.

Let's analyze what went wrong:

1. **Skill Entities Detection Showed Modest Improvements**: In the experiment with training BERT on the TEMPLATE-BASED dataset, skill detection showed statistically significant yet minor improvements. A likely reason is the greater difficulty integrating skills into sentences compared to shorter and simpler knowledge components without significant rephrasing. The unsatisfactory results of the TEMPLATE-BASED WITH REPHRASING experiment will be described in the next paragraph.

2. **Rephrasing Worsened the Quality**: Although manual analysis of the first 20% of rephrases indicated that the sentences improved in quality, in generating the next portion of data, our rewrite prompt removed the annotations in several sentences, simply adding a portion of unannotated skills and knowledge as noise.

In conclusion, although current GPT models can create mostly coherent synthetic data, the reliability of this data is still an issue. Without an automated process, the quality of fine-tuning smaller models for different downstream tasks could be negatively affected.

Chapter 6

Summary and Further Research Directions

6.1 Summary

In this thesis, we explored the application of LLMs to improve skill extraction from job postings using synthetic data. The primary goal was to evaluate the performance of LLMs like GPT-3.5 and GPT-4 in this task and to improve smaller models such as BERT by augmenting training datasets with synthetic data generated by LLM.

What was done:

- **Literature Review and Dataset Selection**: We reviewed existing approaches for Skill Extraction from Job Postings. Additionally, we identified and selected an annotated dataset to evaluate the performance in this task.
- **Initial Experiments with LLMs**: In the initial phase of our investigation, we experimented with both open-source and proprietary LLMs to extract skills in the given text samples. We explored various methods to incorporate instructions for detecting job-related skills and knowledge within the given text into the prompt. We also experimented with different input sizes to determine the optimal text length for skill extraction.
- **Baseline Evaluation**: We reproduced the experiments described in Zhang et al., 2022 and established a BERT baseline for further experimentation in skill extraction.
- **NER-style LLM Skill Extraction**: We utilized techniques outlined in Wang et al., 2023b for Named Entity Recognition (NER) style prompting to methodically extract skills using LLMs. This included employing few-shot in-context learning approaches and self-validation prompts. The results were analyzed, evaluated, and compared to the baseline.
- **Synthetic Data Generation Experiments**: We experimented with different methods to generate synthetic data, both without references and by incorporating ESCO skills into existing sentences. We evaluated the quality of generated sentences both manually and by the construction of embeddings to compare generated sentences to real ones in the dataset. To improve coherence, we added validation and rephrasing steps, resulting in datasets used for subsequent BERT training experiments.
- **Evaluation of Models Trained on the Augmented Datasets**: We generated synthetic data amounting to 20% and 40% of the real data size, used it to train BERT along with the real data, and compared the results to the baseline.

Key findings include:

- **Standalone LLMs Are Insufficient**: LLMs demonstrated lower performance in skill extraction compared to supervised models like BERT. Issues included irrelevant detections, malformed responses, and merged entities.
- **Synthetic Data Improved Knowledge Extraction**: The addition of sentences generated using the TEMPLATE-BASED methodology significantly improved knowledge extraction. The minimal context adjustments required by this method were sufficient to effectively integrate knowledge entities into the training sentences, leading to performance improvements.
- **Data Quality Concerns**: The rephrasing strategy did not improve the results. We found that GPT occasionally removed annotations during the rephrasing process, which compromised the dataset's quality and negatively impacted the training using the sentences generated with this method.

6.2 Further Research Directions

- **Better Post-Evaluation of Generated Data**: Future research should consider additional steps to ensure the quality of synthetic data. Simple prompt verifications are insufficient due to the stochastic nature of large language models. Automated checks to verify the presence and accuracy of annotations are essential to identify and correct errors, thereby improving the overall effectiveness of the data augmentation process.
- **Comparison With Traditional Augmentation Techniques**: The best result in knowledge detection was achieved by prompting GPT to minimally adjust the context. To determine whether this improvement was due to the better context or simply the introduction of new entities, further research should compare this approach to existing methods like Entity Swapping.
- **Experimentation With Open-Source Large Language Models**: As the quality of open-source LLMs continues to improve (Zhao et al., 2023), additional research could explore their usage in the skill extraction domain. Potentially, these models could be utilized in a similar way as in our thesis for synthetic data generation. Additionally, fine-tuning open-source models for standalone skill extraction using open datasets would be a valuable area of research.
- **Advanced Prompting Techniques for LLM Skill Extraction**: Additional research should explore more advanced prompting to enhance the performance of standalone LLMs in skill extraction. This can include investigating the use of autonomous agents (Wang et al., 2023a) for this task. If models can reliably prompt themselves to verify extraction results, the quality of detected skills could significantly improve. However, the cost-effectiveness of this approach should also be considered.

6.3 Final Remarks

Our research found that incorporating synthetic data into training datasets can improve the performance of supervised models like BERT for skill extraction from job postings. While LLMs like GPT-3.5 and GPT-4 showed potential without additional

fine-tuning, they did not outperform the baseline. The study highlights the importance of careful data generation and validation. Future research should refine these techniques and explore advanced prompting methods to enhance LLM performance on this task, leading to more accurate and efficient workforce analysis tools and better talent management.

Appendix A

Data Generation Prompts and Results

This chapter includes prompts and tables of example outputs for different synthetic data generation strategies. The prompts contain placeholders enclosed in curly brackets. The corresponding tables show the specific values for these placeholders and the example results.

A.1 Zero-shot Generation

Prompt template

A knowledge component is understood as factual or theoretical information, including specifics, principles, and certifications relevant to a field.
Skill component represents an ability or expertise, often demonstrated through verbs or adjective-noun combinations, focused on job-related tasks.
Given the abovementioned definitions write a sample sentence from a job posting for a "**{Occupation Name}**" role. It should contain the following components:
{Skill or Knowledge Components}
Annotate these components in your response, using '##' at the beginning and end for skills, and '@@' at the beginning and end for knowledge components. The response should not contain any additional skills and knowledge components, apart from the ones mentioned above. The response should not contain addional job details. The response should be consise, yet creative.

A.2 Template-based Generation

Prompt template

Take this sentence from a Job Posting.
"**{Example Sentence}**"
A part surrounded by @@ is a job-related skill and a part surrounded by ## is job-related knowledge. Write a sentence preserving annotations, but instead of the skills and the knowledge, you should put the following entities: {Skill or Knowledge Components}. Only adjust grammar minimally to maintain the readability and consistency of the sentence with the new entities.

TABLE A.1: Examples of GPT-4 Output for Zero-shot Synthetic Data Generation Prompts

No.	Occupation Name	Skill or Knowledge Components	Result
1	software architect	**skill component**: use an application-specific interface, **skill component**: define technical requirements.	We are seeking a software architect who can ##use an application-specific interface## with precision and efficiency, and who has the ability to ##define technical requirements## with clarity and foresight.
2	software developer	**skill component**: use technical drawing software, **knowledge component**: integrated development environment software.	We are seeking a software developer who can ##use technical drawing software## to create detailed designs and has a deep understanding of @@integrated development environment software@@ to streamline the coding process.
3	system administrator	**knowledge component**: system user requirements, **knowledge component**: operating systems.	As an system administrator, you will be expected to have a deep understanding of @@system user requirements@@ and a comprehensive knowledge of various @@operating systems@@, enabling you to effectively manage and optimize our IT infrastructure.

TABLE A.2: Examples of GPT-4 Output for Template-based Synthetic Data Generation Prompts

No.	Example Sentence	Skill or Knowledge Components	Result
1	* Engage with tools and operations teams to @@address failure patterns and incidents@@	**Skills:** implement anti-virus software.	* Engage with tools and operations teams to @@implement anti-virus software@@
2	You are @@responsibile@@ to @@create update implement and communicate internal policies and procedures@@ to ensure the company's and its customers' high security standards are met.	**Skills:** monitor system performance, replicate customer software issues	You are @@responsibile@@ to @@monitor system performance@@ and @@replicate customer software issues@@ to ensure the company's and its customers' high security standards are met.
3	* Knowledge of ##industry-wide technology trends and best practices## e.g .	**Knowledge:** software anomalies.	Knowledge of ##software anomalies##
4	We are looking to expand our team with a Node.js Developer who is excited to @@solve complex problems@@ and open to @@evolve and grow new skills@@ like ##serverless## and ##machine learning##.	**Skills:** build business relationships, train employees. **Knowledge:** PHP, cloud technologies.	We are looking to expand our team with a Node.js Developer who is excited to @@build business relationships@@ and open to @@train employees@@ like ##PHP## and ##cloud technologies##.
5	You are a team player with a @@result-oriented mindset@@ and an @@empathetic@@ person.	**Skills:** ensure equipment availability, set up the controller of a machine.	You are a team player with a @@ensure equipment availability@@ and an @@set up the controller of a machine@@ person.

A.3 Template-based Generation with Rephrasing

A.3.1 Validation Prompt

Prompt template

Job-related knowledge is surrounded by ## symbols. Job-related skill is surrounded by @@. You have a sentence that is supposed to be from a job posting.
Sentence: "**{Original sentence}**"
You should examine if the text is coherent, makes sense, and could potentially be present in a real job posting. Consider how well skills or knowledge fit the outer context. If the text contains just skill or knowledge components without additional context, consider it coherent, Answer just "Yes" or "No".

A.3.2 Rephrase Prompt

Prompt template

Job-related knowledge is surrounded by ## symbols. Job-related skill is surrounded by @@. You have a sentence that is supposed to be from a job posting.
Sentence: "**{Original sentence}**"
It lacks coherence. Rephrase it to make it sound less synthetic. Keep existing @@ and ## annotations. Dont́ create new skills or knowledge components.

A.4 Rephrasing of Sentences without Skills

Prompt template

Take this sentence from a Job Posting.
Sentence: "**{Original sentence}**"
Rewrite it by finding synonyms but not changing the meaning. Don't change the style of the text. The sentence should look like it is from the vacancy.

TABLE A.3: Examples of GPT-4 Rephrases of Incoherent Generated Sentences

No.	Original sentence	Result
1	We are looking to expand our team with a Node.js Developer who is excited to @@build business relationships@@ and open to @@train employees@@ like ##PHP## and ##cloud technologies##.	We are seeking a Node.js Developer to join our expanding team. The ideal candidate will be enthusiastic about @@building business relationships@@ and open to @@training employees@@. A solid understanding of ##PHP## and ##cloud technologies## is essential for this role.
2	You are a team player with a @@ensure equipment availability@@ and an @@set up the controller of a machine@@ person.	You are a team player with the skill to @@ensure equipment availability@@ and the ability to @@set up the controller of a machine@@.
3	You @@design database in the cloud@@ company ##systems development life-cycle##, ##cyber security##, and ##system programming##.	We are looking for someone who can @@design database in the cloud@@. You should have a solid understanding of the companyś ##systems development life-cycle##, ##cyber security##, and ##system programming##.
4	As a person you are passionate about @@interpreting financial statements@@ and the business through ##financial jurisdiction##.	We are looking for someone who is passionate about @@interpreting financial statements@@ and has a strong understanding of ##financial jurisdiction## to help guide our business.
5	A full-time position as Associate Professor in ##manufacturerś instructions for audiovisual equipment## with focus on ##e-commerce systems## involved in the ##product comprehension## is open at the Department of <ORGANIZATION> <ORGANIZATION> <ORGANIZATION>.	The Department of <ORGANIZATION> <ORGANIZATION> <ORGANIZATION> is seeking a full-time Associate Professor with a focus on ##e-commerce systems##. The ideal candidate should have a deep understanding of ##manufacturer's instructions for audiovisual equipment## and be involved in the ##product comprehension##.

TABLE A.4: Examples of GPT-4 Rephrases of Sentences without Skills

No.	Original sentence	Result
1	The group supports the corresponding project groups and offshore production assets.	The team aids the respective project teams and offshore production resources.
2	We reserve the right not to consider material received after the deadline and not to consider applications that do not live up to the abovementioned requirements.	We retain the authority to disregard submissions received post deadline and to dismiss applications that fail to meet the aforementioned criteria.
3	For guidance on how to complete the application form see here.	For instructions on how to fill out the application form, refer here.
4	Deadline January 10th we will invite for interviews on a continuing basis so if you are interested please do not hesitate to apply.	Cut-off date is January 10th, we will be conducting interviews on an ongoing basis, so if youŕe keen, please feel free to submit your application.
4	You will be given responsibility for one or more areas of competence that suit your interests and/or your experience from previous jobs.	You will be assigned accountability for one or more areas of expertise that align with your interests and/or your background from prior roles.

Appendix B

Skill Extraction Prompts and Results

B.1 NER Prompts for Skill and Knowledge Extraction

B.1.1 Skill Extraction Prompt and Output

Skill Extraction Prompt

You are a top-notch linguist and data labeler. Within the SkillSpan dataset, the task is to analyze a sentence from a job posting and label skill entities that are indicated by a verb or an adjective-noun combination, reflecting an executable ability or a specific way of performing a task. Make sure not to include irrelevant, company-specific information. Below are some examples, and you should make the same prediction as the examples.
The given sentence: Have experience with people management and building teams
The labeled sentence: Have experience with people management and @@building teams##
{Three more labeled examples}
The given sentence: You have a natural interest in managing people and your CV shows at least two years of management experience .
The labeled sentence:

Output

You have a natural interest in @@managing people## and your CV shows at least two years of @@management experience## .

B.1.2 Knowledge Extraction Prompt and Output

Knowledge Extraction Prompt

You are a top-notch linguist and data labeler. Within the SkillSpan dataset, the task is to analyze a sentence from a job posting and label knowledge entities that refer to non-executable information that an individual possesses, often indicated by specific fields, industries, or certifications and is distinct from direct action or skills. Make sure not to include irrelevant, company-specific information. Below are some examples, and you should make the same prediction as the examples.
The given sentence: Research develop and implement new methodologies and techniques that enable superior business performance
The labeled sentence: @@Research develop and implement new methodologies and techniques## that @@enable superior business performance##
{Three more labeled examples}
The given sentence: Furthermore excellence in at least one of the following categories is highly desirable: data visualisation tools cloud platforms machine learning techniques and algorithms .
The labeled sentence:

Output

Furthermore excellence in at least one of the following categories is highly desirable: @@data visualisation tools## @@cloud platforms## @@machine learning techniques## and @@algorithms## .

B.1.3 Verification Prompt

Verification Prompt

You are an excellent linguist. The task is to verify whether the word is a skill entity extracted from the given sentence. Skill entities are indicated by a verb or an adjective-noun combination, reflecting an executable ability or a specific way of performing a task.
The given sentence: We take candidates into the recruitment process continuously and close the position down once we have found the right candidate.
Is the phrase "take candidates into the recruitment process" in the given sentence a skill entity? Please answer with yes or no.

Bibliography

Achiam, J. et al. (2024). "GPT-4 Technical Report". In: *arXiv:2303.08774 [cs.CL]*.

Alihosseini et al. (2019). "Jointly Measuring Diversity and Quality in Text Generation Models". In: *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 90–98. DOI: 10.18653/v1/W19-2311. URL: https://www.aclweb.org/anthology/W19-2311.

Bhola, A. et al. (2020). "Retrieving Skills from Job Descriptions: A Language Model Based Extreme Multi-label Classification Framework". In: *Proc. of COLING 2020*, pp. 5832–5842.

Brown, T. et al. (2020). "Language models are few-shot learners". In: *arXiv:2005.14165 [cs.CL]*.

Clavié, B. et al. (2023). "Large Language Models as Batteries-Included Zero-Shot ESCO Skills Matchers". In: *arXiv:2307.03539 [cs.CL]*.

Debortoli, S. et al. (2014). "Comparing Business Intelligence and Big Data Skills: A Text Mining Study Using Job Advertisements". In: *Business & Information Systems Engineering* 6.5, pp. 289–300.

Decorte, J.-J. et al. (2022). "Design of Negative Sampling Strategies for Distantly Supervised Skill Extraction". In: *Proc. of the 2nd Workshop on Recommender Systems for Human Resources (RecSys-in-HR 2022)*. Ed. by M. Kaya et al. Vol. 3218. CEUR.

Devlin, J. et al. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv:1810.04805 [cs.CL]*.

Gao et al. (2021). "SimCSE: Simple Contrastive Learning of Sentence Embeddings". In: *arXiv:2104.08821 [cs.CL]*.

Hochreiter, S. et al. (1997). "Long short-term memory". In: *Neural Computation* 9.8, pp. 1735–1780.

Jia, S. et al. (2018). "Representation of Job-Skill in Artificial Intelligence with Knowledge Graph Analysis". In: *2018 IEEE Symposium on Product Compliance Engineering - Asia (ISPCE-CN)*. Shenzhen, China, pp. 1–6.

Joshi, M. et al. (2019). "SpanBERT: Improving Pre-training by Representing and Predicting Spans". In: *arXiv:1907.10529 [cs.CL]*.

Kaddour et al. (2023). "Text Data Augmentation in Low-Resource Settings via Fine-Tuning of Large Language Models". Version 1. In: *arXiv:2310.01119 [cs.CL]*.

Khaouja, I. et al. (2021). "A Survey on Skill Identification From Online Job Ads". In: *IEEE Access* 9, pp. 118134–118153.

Kim, Y. (2014). "Convolutional Neural Networks for Sentence Classification". In: *arXiv:1408.5882 [cs.CL]*.

Lample, G. et al. (2016). "Neural Architectures for Named Entity Recognition". In: *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California, pp. 260–270.

Li, N. et al. (2023). "SkillGPT: A RESTful API Service for Skill Extraction and Standardization Using a Large Language Model". In: *arXiv:2304.11060 [cs.CL]*.

Mikolov, T. et al. (2013). "Efficient Estimation of Word Representations in Vector Space". In: *arXiv:1301.3781 [cs.CL]*.

Pakhale, K. (2023). "Comprehensive Overview of Named Entity Recognition: Models, Domain-Specific Applications and Challenges". In: *arXiv:2309.14084 [cs.CL]*.

Pezeshkpour, P. et al. (2023). "Distilling Large Language Models Using Skill-Occupation Graph Context for HR-Related Tasks". In: *arXiv:2311.06383 [cs.CL]*.

Raffel, C. et al. (2019). "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *arXiv:1910.10683 [cs.CL]*.

Sayfullina, L. et al. (2018). "Learning Representations for Soft Skill Matching". In: *Analysis of Images, Social Networks and Texts (AIST 2018), Lecture Notes in Computer Science*. Ed. by W. van der Aalst et al. Vol. 11179. Springer, Cham.

Student (1908). "The probable error of a mean". In: *Biometrika*, pp. 1–25.

Wang, J. et al. (2022). "Machop: An End-to-End Generalized Entity Matching Framework". In: *Proc. of the Fifth International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*, pp. 1–10.

Wang, Lei et al. (2023a). "A Survey on Large Language Model-based Autonomous Agents". In: *arXiv:2308.11432 [cs.CL]*.

Wang, S. et al. (2023b). "GPT-NER: Named Entity Recognition via Large Language Models". In: *arXiv:2304.10428 [cs.CL]*.

Wei et al. (2022). "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: *arXiv:2201.11903 [cs.CL]*.

Yang, Z. et al. (2016). "Hierarchical Attention Networks for Document Classification". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. San Diego, California, pp. 1480–1489.

Zhang, M. et al. (2022). "SkillSpan: Hard and Soft Skill Extraction from English Job Postings". In: *Proc. of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States, pp. 4962–4984.

— (2023). "ESCOXLM-R: Multilingual Taxonomy-Driven Pre-training for the Job Market Domain". In: *The 61st Annual Meeting of the Association for Computational Linguistics*. Vol. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 11871–11890.

Zhao, M. et al. (2015). "SKILL: A System for Skill Identification and Normalization". In: *Proc. of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, pp. 4012–4017.

Zhao, W. X. et al. (2023). *A Survey of Large Language Models*. arXiv:2303.18223 [cs.CL].

Zhu, Y. et al. (2018). "Texygen: A Benchmarking Platform for Text Generation Models". In: *arXiv:1802.01886 [cs.CL]*.