Paderborn University
Faculty of Economics
Department Business Informatics

Project Report AML4TA

# Skill Gap Analysis Of University Courses And Job Advertisements

by
Dennis Wüppelmann, Mohammad Niaj Uddin Ahmed, Yves Illes Moidja Nkoue, Yann Habib Tshansi Ngamo
Matriculation Numbers: 7266410, 6900003, 6866552, 6842737
dewue@mail.uni-paderborn.de, niaj@mail.uni-paderborn.de,
yves@mail.uni-paderborn.de, yannhtn@mail.uni-paderborn.de

submitted to
Prof. Dr. Oliver Müller
Matthew Caron

03.03.2023

## Eidesstattliche Erklärung

Hiermit erklären wir an Eides statt, dass wir die vorliegende Arbeit selbstständig, ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken, Tabellen, Skizzen, Zeichnungen, bildliche Darstellungen usw. sind ausnahmslos als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Paderborn, 03.03.2023

MD. Niaj uddin

Dennis Wüppelmann, Mohammad Niaj Uddin Ahmed, Yves Illes Moidja Nkoue, Yann Habib Tshansi Ngamo

## Abstract

This work aims to investigate the skill gap between university courses and the labor market, especially in the field of data science. To achieve this goal, the study uses two datasets: LinkedIn job postings and course descriptions from the University of Paderborn. To compare the skills taught at universities with the skills most in demand by employers, natural language processing is used to extract hard and soft skills from the datasets. The study presents a proof-of-concept implementation of a method that produces a machine learning-assisted analysis of the current skill gap between the job market and university teaching in the same field. In addition, the study led to the development of a transformer based model that can automatically extract hard and soft skills from unstructured text with an accuracy of 0.6602, measured as F1 score. This model can be applied to other texts in the field of data science, helping job seekers identify relevant job openings and employers identify suitable candidates by extracting skills from a resume. While developing the described method, this paper highlights areas of past and future research and improvements in skill extraction methods, as well as the skill gap in the field studied. By addressing these gaps, universities and employers can better align their curricula and job requirements with current labor market demands. The results of the study can be used by universities, employers, and students as a base for a data-driven analysis of the skill gap in data science.

**Keywords:** data science, machine learning, natural language processing, named entity recognition, job skill gap, transformer, spacy, prodigy

# Contents

# List of Figures

# List of Tables

# 1 Introduction

The digital revolution has created a lot of new jobs, including the increasingly popular job of data scientists. But what exactly does a data scientist do? Are the daily challenges in this job similar across different companies? All companies will likely be doing something with data and knowledge generation, won't they? Students who are nearing graduation and thinking about possible career paths may ask themselves if they have learned the right skills for entering this field. The University of Paderborn provides a list of typical career fields after completing a degree in business informatics, such as data science, ERP consulting, IT entrepreneurship, operations research analysis, service engineering and management, social media/social intranet management, supply chain analysis, and consulting (Paderborn University, 2020). These professions suggest a broad range of possible activities, for which students may be better or worse prepared depending on their individual focus. The gap between the skills that students learn in their degree programs and the skills required in the workplace is also known as the *skill gap*.

This paper aims to examine the job skill gap between university courses and the job market, with a particular focus on the current state of this gap. To achieve this, we will use two datasets as our examples. Firstly we analyze data from LinkedIn job advertisements as the dataset to identify the skills that are most requested by employers right now. Secondly as the current state of university teaching we will analyze the course descriptions of the Paderborn University.

The main goal of this research is create a Proof of Concept implementation of a method to create a machine learning assisted analysis of the current skill gap between the job market and university teaching in the same domain. For this we will include a state of the art natural language processing pipeline that is able to extract soft and hard skills from unstructured text. After applying this extraction process to our two datasets we will further analyze the collected skills on intersections and differences.

Overall, our approach provides a data-driven analysis of the skill gap in the specific field of 'data science' jobs. The findings of this analysis will be of interest to a range of stakeholders, including universities, employers and students. Our collected data and created model can be used to enable further research into this topic as well.

We will begin by building our dataset from scratch by scraping the LinkedIn Job Market and the required course descriptions. This step will take a great part of our work and consists of multiple steps which will result in a prepared dataset for further usage in our machine learning pipeline. To achieve this, we followed a process that involved data extraction from different sources, various steps of cleaning, language analysis and translation into a common language.

To extract the skills from the texts we will train two models each with a different architecture on a part of the dataset. The training set will be manually labeled with the use of *Prodigy*. The models will be extracting the skills using Named Entity Recognition categorized in hard skills and soft skills. The first model we will train is based on statistical methods to tokenize the input data while the second model is based on transformers. The best performing model of these two will be used as our ’skill detection’ model.

Using the trained ‘skill detection’ model we can then extract and analyze the skills. First we will provide an overview of the job market and the skills that are currently in high demand by employers. We will then move on to a detailed analysis of the skills listed in university course descriptions and the extent to which they align with the skills in demand in the job market. For this we use different calculation techniques and visualizations to describe our findings.

By comparing the skills listed in university course descriptions with the skills in demand in the job market, we hope to provide a more comprehensive and accurate analysis of the current skill gap in the data science job market.

# 2 Related Work

For our research we identified key areas of interest that heavily influenced our research method: Skill extraction from texts written in natural language, comparing different sets of job skills, improvements of course design with industry skills and skill analysis in the realm of data science in general.

Several studies have already addressed the skill gap in the realm of data science. For example, Lyon et al. (2015) conducted a study that asked two research questions about the 'talent gap' that were very similar to ours:

1. What are the skills, competencies, knowledge, experiences and education required for the distinct data science roles?
2. How do these data science role requirements map to current curriculum topics and course offerings?

The researchers analyzed job positions related to the 'data-preservation' role, which were further categorized into Data Librarian, Data Steward/Curator, and Data Archivist with ten job descriptions for each position. They also examined the course syllabi of the graduate Library and Information Sciences program at the University of Pittsburgh School of Information Sciences. The research demonstrated that employers were seeking graduates who were data-savvy and work-ready. This paper presents an interesting and transferable methodology that faculties could use to analyze their curriculum. Additionally, the authors mapped the courses to jobs to guide students. However, it is important to note that all the data used was up to date and from the US, and the analysis was done manually. This makes the method less scalable in practice.

Apparently the concept of skill gap is not new as in has already been approach by professional and researchers. We did encounter a more significantly related studies to our work. The paper Shmatko and Volkova (2020) describes and experiment conducted on bridging the Skill Gap in Robotics. Their approach similar to the one we have taken consisted of collecting data from job ads websites which in their case was Indeed.com and hh.ru. skills which they later separated into 2 classes; Soft and Hard skills. Broadly their experiment mentioned technologies like Python and C which they used in their mining, their extraction and their classification. Basically rather than computing the skill gap, they wanted to establish a faithful overview of skill demanded the robotics job market, to serve as a guide in building university curriculum's. Their results showed a high demand in programming, especially C/C++ and python coming up second. As the job posts were collected both with USA basis and Russia basis, the founding were in the same way distributed amongst these 2 categories with slightly different results on the soft skill side. While the USA required organisation skills as top soft skills, Russia required English language as top soft skill.

Dong and Triche (2020) used a text mining approach and a custom text mining dictionary to analyze over 9000 entry-level data analytics jobs from 2014-2018. This more automated method identified a preliminary set of analytic competencies used in practice. The method used by the authors involved scraping the job portal indeed.com for historical data using multiple search terms, including wildcard searches for 'business analy*', 'data analy*', and 'business int*'. They specifically excluded the term 'data sc*' because they found that these positions require higher education and work experience, and are not considered entry-level. To analyze the job postings, the authors applied NLP techniques such as lemmatization, part-of-speech tagging, and chunking. This resulted in 6,067 unique nouns and noun phrases which were analyzed manually. For the main analysis, they created a dictionary of 186 keywords and phrases that contain the necessary skills for analysis. They state, that using the results of the study, universities can make informed curriculum decisions, and instructors can decide what skills to teach based on industry needs. In addition to showcasing an intriguing methodology related to our research question, the authors demonstrate how skills have evolved over time. The three largest trends identified include proficiency with Python, Tableau, and R. However, an increasing number of jobs are now emphasizing data visualization. Conversely, certain skills, such as Microsoft Access, SAP, and Cognos, have declined in popularity over the time frame studied. It is worth noting, however, that the creation of the skill dictionary involved a manual step, which requires a degree of familiarity with potential skills in order to be effective.

In a study conducted by Gardiner et al. (2018), a different approach was taken to the manual use of experts following NLP pre-processing. The study analyzed 1,216 job advertisements containing the term 'big data' in the job title, which were scraped from Indeed. The researchers made the decision not to expand their search to the body text of job advertisements, opting instead to use only the titles that specifically included the key term 'big data.' This decision was made in order to avoid significantly increasing the study's sample size. After scraping data, the researchers conducted computer-aided text analysis. They eliminated typical stop words, stemmed terms, eliminated white space, and converted all text to lower case. They collected n-grams (1-8 word phrases), and re-stamped similar n-grams like 'data science', 'data scientist', and 'data scientists' as 'data science'. This resulted in a final count of 259 re-stamped skill concepts. After pre-processing, they applied the Consensus-based Pile-Sort method for manual analysis conducted by four experts. The results of this study are presented within a conceptual framework of big data skill categories, highlighting the multi-faceted nature of big data job skills. The study found that many big data job advertisements emphasize the development of analytical information systems, and that soft skills remain highly valued, in addition to the importance of emerging hard technological skills. Furthermore, this study provides insights into how industry views the discipline of big data, and the skills linked to the data science profession.

A study similar to ours, but conducted in a different country and focused on the software domain, was carried out by Hiranrat and Harncharnchai (2018). The goal of their study was to identify the technical knowledge and soft skills currently required by the software industry in Thailand. They analyzed data collected from online job portal websites using text mining techniques. Their findings were summarized and reported to design training courses aimed at preparing students for employment. In this study, a dictionary of skills was employed, which was manually curated from previous studies and the Skills Framework for the Information Age (SFIA). The skills were categorized into three groups: explicit technical, implicit technical, and soft skills. To preprocess the data, standard natural language processing techniques were employed, similar to those described in Gardiner et al. (2018). Specifically, the sentences were automatically split, tokenized, and had stop words removed before being stemmed. Additionally, manual summarization was performed. The explicit technical terms with the same meaning were replaced by a common term manually. For example, the terms “Microsoft SQL Server”, “MS SQL”, “MSSQL Server” were replaced by “MSSQL”. The implicit technical and soft skill terms were transformed into the root words. For example, the term “communicative” and “communication” were stemmed to “commun”. Any further processing and categorizing after the skills were extracted, e.g. grouping similar jobs, was done manually. The terms found in each jobs qualification that matched to the keyword in the skill dictionary were recorded. The term occurrences and its frequency were calculated at the end to guide the analysis. The study has confirmed that the skill requirements in the IT job market vary across different countries. Their findings suggest that understanding these skill requirements can help educators to better manage curricula, design courses and training programs, and ultimately enhance the employability of graduates and the satisfaction of employers. The authors conclude their study with a call to action, suggesting that further research is necessary to bridge the gap between student skills and employer needs, and to develop training courses aimed at improving student skills. In addition, the authors emphasize the need to create automated tools for monitoring skill requirements and technology trends in the software industry to support educators in making informed decisions.

In a study that shares a similar methodology with our own, Fareri et al. (2021) developed a skill detection model using Named Entity Recognition (NER) on one dataset and applied it to another dataset to extract skills. However, their focus was solely on soft skills and they employed a combination of dictionary methods and machine learning. Their skill extraction method was based of an extraction context: soft skills were divided in a clue and a skill part e.g. ‘ability to (clue) solve difficult problems (skill)’. To identify these contexts, they manually constructed a seed list of soft skills and built two rule-based matchers to extract sentences that contained at least one clue. The extracted sentences were then annotated by experts using the Prodigy software. The authors presented two models for skill detection. The first model was based on Support Vector Machines (SVM) using *LIBSVM* and spacys *en_core_web_lg* model. The second model used a multilayer perceptron (MLP). The performance of both models was evaluated using precision, recall, and F1-score metrics at the token level. The results showed that the SVM model was more reliable, achieving an F1-score of 72.6. In their conclusion they recommend to use a newer transformer based model like BERT instead of SVM. In a case study they applied their developed model to the ESCO Database to showcase the application of such a system.

The ESCO database, which stands for European Skills, Competences, Qualifications and Occupations, is a multilingual classification of skills, competences, qualifications, and occupations used across the European Union. It is a standardized system that enables users to identify and match skills, qualifications, and competences across different countries and languages. According to ESCO, a data scientist is a professional who finds and interprets rich data sources, manages large amounts of data, merges data sources, ensures consistency of data-sets, and creates visualisations to aid in understanding data. They build mathematical models using data, present and communicate data insights and findings to specialists and scientists in their team and if required, to a non-expert audience, and recommend ways to apply the data (European Commision, 2022).

The jungle of AI specialist jobs led to a study by Kortum et al. (2022) who analyzed skills used in Natural Language Processing specialists jobs compared to skills in Computer Vision specialists jobs. The researchers crawled two groups of job advertisements from indeed.com extracted skills using NER and compared the skill groups of both. They found that there is no general requirement profile of an artificial intelligence specialist. In their paper they stated that NER methods rely on large amounts of labeled training data, and since there was a lack of suitable data to build a dedicated training base, they decided to use a cloud service to support them in the process of skill identification. In concrete the Microsoft Azure Analytics Service was used to retrieve terms from the corpus referring to skill entities. Followed by a manual analysis of the skills. Of the initial 3,712 terms identified as skills, 523 unique terms remained after validation. The skills were manually assigned to one of the following categories: AI, programming, programming language, general skills, domain, soft skills, and qualification. Based on the extracted sequences, bag-of-words frequency and term frequency–inverse document frequency matrices (TF-IDF) were used to provide more differentiated statements about the relevance of a term for a single document. The similarity was analyzed with cosine similarity, distribution plots, heatmaps and displayed as venn diagram too.

Della Volpe and Esposito (2020), have shown a similar approach to our work, where the research paper explores Italian universities' courses whether they verify the qualification for the data scientist jobs. The authors employed Natural Language Processing software, specifically the NooJ environment, to analyze data from LinkedIn job advertisements. The paper presented a cluster-based classification approach, grouping data science skills into nine clusters: Analytical Skills, Educational Requirements, Effective Communication, Machine Learning, Knowledge Management, Mathematics and Statistics, Programming and Software Development, Soft Skills, and Visualization Skills. These clusters were annotated with words and expressions related to each skill group. For example, 'Soft Skills' contained terms like team working, problem-solving, leadership, and motivation. Other skills such as, artificial intelligence, neural networks, deep learning are clustered as machine learning, and Python, Java, No-SQL are clustered as programming and software development.

Further more, in paper Ahadi et al. (1970), whose purpose was to identify the skill gap between the content taught by an institution and the job market that they are targeting, it was established that these analysis could be applied to other scenarios like when trying to build. In this manner provide a better allocation of resources in teaching. They used the curricula of the University of Sydney making sure to extract all areas that were in their opinion good sources of skill content of the courses. These areas included content, learning objectives,graduate attribute and others. Then, with the help of an API for skill tagging which uses Natural Language Processing (provided by EMSI-Burning Glass Technology), they were able to extract skills form the curricula extracted data and compared it to the 144 000 job advertisement they also collected. This data sets permitted them to compute the skill gap by weighing the relative importance of each skill using the Revealed Comparative Advantage (RCA), which is an index in labour economics for calculative relative advantage or importance. With the RCA method, they were able to weight relative importance of specific skills and sort them out. They then selected 10 courses from the university that potentially prepared students for position like data scientist, data engineer. From this point they could compute the pairwise similarity matrix of the courses and the occupations. With the help of a heatmap plotted using the bag of skills return from each course, they were for example able to observed that Information technology related courses had the lowest skill gap with the above cited positions.

The above papers all used a broad set of job titles related to the hard-to-define job of a data scientist. Ho et al. (2019) tried to define the current job of a data scientist and their skills with the approach of a data scientist. The approach was scraping job postings from an online job site, pre-process the data, explore the data, then transform the data into high-dimensional vectors to cluster, classify, and analyze. The application of NLP, universal sentence encoder, and machine learning techniques led to quantifiable visualizations, results, and features. The created clusters were reviewed to find closest neighbors to the data scientist job postings – closest neighbors being statistician and data analyst. The final step was to find the features of importance for each of the three roles. As described they applied a fully automated process including NLP and ML techniques and found based on the information gathered in job postings, when an employer is looking for a Data Scientist, they look for the following: *"A Data Scientist codes, communicates, and collaborates – transforming data into insights using statistical, analytical, and machine learning techniques."*

After reviewing studies already done in the field we combined the approaches and build a method to answer our own research questions.

# 3 Method

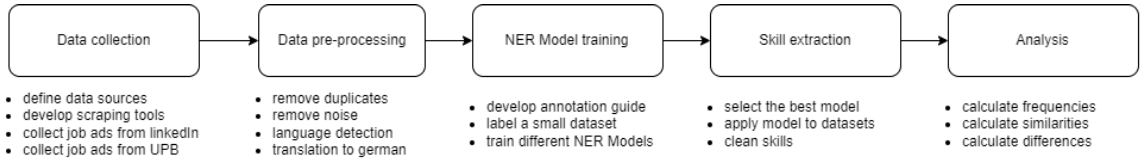

Figure 3.1: Our research method

The final result should provide a proof of concept which contains an automated pipeline to detect hard and soft skills in the first step as well as a second step that is able to compare two given groups of skills.

Compared to the approaches in the related work section our approach includes:

- Gathering job skills data from LinkedIn Jobs in Germany
- Collecting course descriptions from Paderborn University
- Utilizing NER models instead of other NLP techniques such as POS tagging and lookups with dictionaries
- No manual labor is required during the skill extraction or categorization steps, with the exception of the annotation process needed to create an initial training set

## 3.1 Data collection and pre-processing

Before we can start any analysis we need a dataset. Because our goal is to compare the current state of the job market with the current state of university teaching in the field of data science, two datasets are required. We decided to create these datasets by ourselves to ensure that the job postings and course descriptions are up to date and reflect the current state.

### 3.1.1 Job Advertisements

For the job market we collected job ads from the LinkedIn Jobsearch. The search portal offers the possibility, among other filters, to search the ads by job title and location. When searching for the title 'Data Scientist' throughout Germany LinkedIn states to find 48.000 jobs. The search was focused on the data scientist job because we are interested in the comparism with a university therefore the entry job definition of Dong and Triche (2020) is not sufficient for us and the compared Paderborn University states the 'Data Scientist' as a possible future job of their business informatics graduates (Paderborn University, 2020). The study by Ho et al. (2019) further encouraged us to use the 'Data Scientist' title as it is the current description on the market.

To collect the relevant data from the portal the *Selenium framework* was leveraged to automatically scrape the adverts with their *WebDriver*. The decision to use Selenium was driven by the fact that LinkedIn is a dynamic website not static and therefore not easily readable with other frameworks like *Beautiful Soup*.

While building the LinkedIn scraper we faced three main problems and limitations:

Firstly it is stated that there are 48000 jobs in the portal but only the first 1000 results are shown for each search query. The job results are paginated for 25 jobs per page. So if you scroll to the bottom of the list result list about 40 times you will reach the end and no new jobs are loaded. To solve this limitation we did one search for each of the 80 largest German cities with a population over 100.000 (Statistisches Bundesamt, 2022). The default 40km distance of the search portal was kept, which resulted in 62844 fetched jobs. After removing duplicates (identified by the same URL) 27777 adverts were left. The fetched data contained the content of the ad as plain text and the following metadata: job title, company, date of listing, listed location and the link to the ad.

Secondly the job ads were written in multiple languages. To create a uniform dataset we first collected the different languages using *spaCy_langdetect* and then translated the original content into the most common language German, using *googletrans*. In the process of the translation we also removed ads that contained no content or only less than a sentence of content.

The third and arguably the most critical problem was the content of the ads itself. The dataset is polluted with noisy job ads. On the one hand, when searching for job titles in our collected data only 654 titles mimic the exact 'Data Scientist' of our search query. There are different writing styles like all uppercase or all lowercase too. On the other hand there are far more jobs containing ’Software’ than ’Data’ which indicates we have fetched job ads not mainly focused in the ’Data Science’ sector. For the first point we simply transform all the titles lowercase to get a more uniform dataset. But for the second problem we have multiple possible reasons: companies are creative with job titles and LinkedIn tries to recommend other jobs that are related to the search query. Here we encountered a similar issue of job titles being ambiguously labeled, as noted in a previous study by Ho et al. (2019). The study states that different job sites utilize varying algorithms, leading to job postings that appear to be for a data scientist position but are, in fact, for a completely different role. For the general research in the current state of the job market that might not be a problem but because we want to train a NER model with the data, a quality training set is needed. Therefore we decided to create a smaller training set with job titles matching the initially search term ’data scientist’. Similar to the process of Dong and Triche (2020) we used a wildcard search pattern, ’*data scien*’ to filter and collect 1117 job ads used for further processing in our training pipeline.

After the last step of cleaning/selection is finished we end up with the numbers of job ads shown in Table 3.1 which reflects an example of the currently requested job positions in the German market when searching for ’Data Scientist’.

| filter | count |
|---|---|
| full dataset | 27.516 |
| ’data scien’ | 1.117 |
| ’data’ | 2.640 |
| ’software’ | 9.887 |

Table 3.1: LinkedIn Dataset Overview

Using *spaCy* and the *xx_sent_ud_sm* model we took an initial look on our training dataset of 1117 jobs. To see if there is already a tendency of hard skills, we used *part-of-speech* tagging and collected 16010 unique nouns. The first 25 most used nouns already contained words like Daten (3rd), Python (5th), Mathematik (17th) and Statistik (25th) which hints that the selected job ads have similar content. Also, you can see in Figure 3.2 that the length of job ads is 3610 characters on average. Therefore, our dataset seems manageable for manual annotation, and we divided it into chunks of 250 for labeling by the four members of our group.

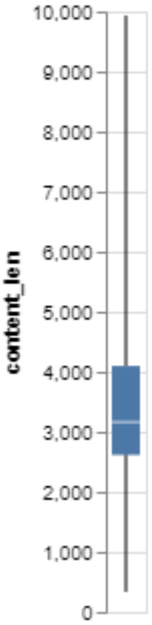

Figure 3.2: trainingset job ad content length boxplot

### 3.1.2 University Courses

As our purpose was to built a simple model that worked before scaling it, we considered the position of data scientist. This indeed made it clear that the principal university majors of interest would be the computer science which is deeply rooted in statistics and mathematics and the business informatics related majors. In order to constitute our dataset therefore, we accessed 2 sources namely the computer science major module catalogue and selected 10 courses from business informatics major. We made our selection taking care of selecting those courses that were closer to the data science profession. The strategy was to scrape both sources, extract the sections we needed (Content, learning outcomes...etc) and finally combine these 2 as a unique dataset, following that they were most likely to be similarly structured.

#### Computer Science Module Catalogue

The Computer Science module catalogue of the University of Paderborn is made available for every student on the university website. We were therefore expecting a web page with the information listed as HTML tags. But rather, we found a well formatted PDF document with all the courses and their details listed. We spotted a total of 40 courses including the master thesis.

We proceeded in scraping the PDF document. To perform this we leverage the python library *PyPDF2*, which enabled us not only to read but also to transform PDF files in python. Upon exploring the file, we notice that there was a lot of information that would not be significant for our experiment. We then went on to extract all the information present between the tags 'Inhalte' and 'Prüfungsleistung'. This way we captured the course content and the learning outcomes.

Although the data was entirely in German, we encountered a few outliers that were in English language. The proportion was 99 % for German language and 1 % for English language. Though this could pose a problem, some exploratory analysis enable us to notice that this little proportion of English was depicted by concept and technology denominations mostly. By concepts we mean machine learning, regression, and so on. And as they were also present in our training for the NER model, they turned on to be useful for our model performance.The following step was then to combine this data set to 10 selected courses from Business Informatics major ( after proper cleaning), as this will constitute our data from the university side.

### Business Informatics Module Catalogue

The university website module catalogue presents an overview of modules that can be selected according to the degree program. So, we selected business informatics and try to collect data from the web page containing the information listed as HTML tags. To extract them, we used *Selenium*, set up the chrome *WebDriver* and got the HTML page source with the Beautiful Soup library. We noticed directly that the information (content and learning outcomes) needed were in the table tag and decided to scrape all elements within tables by collecting the related table header and table data. Finally, we export the data in *JSON* format considering the encoding *utf8*.

We repeat this process 10 times for each of the 10 most related modules to a data science or machine learning job application. In this way, we were able to collect data from the university website module catalogue.

After the collection, it was necessary to clean and translate the data. In fact, we faced the problem of multiple languages (English and German) in the data. So, we merged the 10 collected data sets, processed them using the small model *de_core_news_sm* from *spaCy*, and detect the proportion of each language with the *spaCy_langdetect*. Although we cleaned our raw merged data by replacing unnecessary characters with empty spaces and created a *spaCy* document, we first got English as the average language in the document with a score of 99 %. Then, we decided to split our document into sentences and calculate the proportion of each language and obtain this time much more representative result: English 45 % and German 55 %.

Furthermore, we translate the document into German and for this task, we used the *googletrans* library. It is important to note that, the google API allows translation of up to 5000 characters. To solve this problem, we split the text into chunks of 4500 characters, translate each chunk and concatenate the translated chunks. Finally, we obtained our data translated into German and store it in a *JSON* file always considering the encoding *utf8*.

## 3.2 Annotation

The skill labeling process was conducted by our research team consisting of four students and did not involve any experts from human resources, psychology, or university teaching (like it was done in the related work). In order to ensure a consistent and standardized approach to the labeling process, we developed a set of guidelines for our annotators, which were based on the related work previously described. The primary objective of these guidelines was to achieve a homogeneous training set.

We categorized skills into two main categories: hard and soft skills, which were labeled as *HARD* and *SOFT*, respectively. This decision was made based on the recognition, similar to the findings of Gardiner et al. (2018), that both types of skills are crucial in job requirements. The terms 'explicit technical' and 'implicit technical' used by Hiranrat and Harncharnchai (2018) were merged into the category of hard skills, while the term 'soft skills' was retained. We applied the same definitions as used by Hiranrat and Harncharnchai (2018) to these terms. As mentioned earlier Della Volpe and Esposito (2020) organized similar skills into clusters. While we draw ideas on what to label as a skill from these clusters we decided not to further divide skills into clusters like, Machine Learning or Programming and Software development and rather combine them into the hard skills.

### 3.2.1 Guidelines

- Hard skills refer to the specific software product or technology as well as to the more abstract concept, competencies, and knowledge needed for a job (Hiranrat and Harncharnchai, 2018):

For instance, 'SQL' is an example of a technology-based hard skill, while 'Verwaltung und Wartung von Datensätzen' represents an abstract hard skill.

- Soft skills refer to types of behaviors required for successful performance involving personality traits, social interaction abilities, communication, and personal habits (Hiranrat and Harncharnchai, 2018):
  For instance 'Kreatives und innovatives Denken', 'Planen und Organisieren' or 'Zeigen von Lernbereitschaft' (European Commision, 2022)
- When labeling the skills, only use the core of the skill, and not the accompanying 'clue'. For instance, for the skill 'Arbeiten mit Python' only annotate 'Python' as the skill.
- Keep the annotation as short as possible, but include longer described skills that are more than one word, like: 'maschinelles Lernen' or 'Wartung von Datensätzen'
- At the bottom of the Prodigy page, a link to the original job posting is provided. In case of any doubts regarding incorrect translations or other issues, you can refer to the original job post.
- In case of uncertainty regarding a particular skill, one can refer to the ESCO database and its search engine:
  - ESCO data scientist page: `http://data.europa.eu/esco/occupation/258e46f9-0075-4a2e-adae-1ff0477e0f30`
  - ESCO skill search: `https://esco.ec.europa.eu/en/classification/skill_main`
- To speed up the labeling process, make use of the various shortcuts available in Prodigy, such as pressing '1' or '2' to select the appropriate label, 'a' to finish the current job position, and 'ctrl+s' to save the annotations.

### 3.2.2 Process

To label the job ads, we utilized the annotation tool, *Prodigy*. We prepared the ads by transforming them into a format compatible with the tool, ensuring that each document used for annotation corresponded to a single job ad. We initialized a blank *spaCy* pipeline to tokenize the German texts, but all other tasks were performed manually. A total of 1,117 job ads for data scientists were pre-selected and divided into four chunks, which were assigned to the research team for annotation. Each team member was tasked with annotating approximately 75 ads from their assigned chunk. After completing the annotation task, the labeled data from each team member was combined to create a single dataset. During the annotation process, we identified redundant texts, resulting in a final annotated dataset of 291 job ads. The annotated ads were further split into a training and validation set with an 80/20 split. We used these two datasets for further training of both models

## 3.3 Training

To train our NER models, we utilized the annotated dataset created in the previous step. We developed two different models using spaCy's training pipeline, which we selected for its suitability in building a proof of concept. Rather than relying on a cloud provider as in Kortum et al. (2022), we opted to test and demonstrate the complete but pragmatic process of building our own model, starting with data collection and proceeding to state of the art model training and prediction, before conducting the skill analysis.

In our process we tested two models with the goal to pick the best performing for the final usage. The first model is a statistical entity recognition system that was build with a spaCy pipeline of *tok2vec* and *ner*. More precisely we used their pre-built architectures: *Tok2Vec.v2* with *MultiHashEmbed.v2* and *MaxoutWindowEncoder.v2* as well as the *TransitionBasedParser.v2* with *Tok2VecListener.v1*.

We trained a transformer-based model using spaCy, which served as our second model. spaCy made it convenient by providing for us a *base-config* file, with the pipeline set to *transfomer* and *ner*. SpaCy also provides a system with a *gpu-allocator* set to *pytorch*. The transformer model *de_dep_news_trf* from spaCy was then installed in *python*, the training and validation set fit via the config file, and the model trained. We had multiple reasons for choosing this model. Firstly, we wanted a comparison model for the basic spaCy configuration. Secondly, as recommended by Fareri et al. (2021), we opted for newer transformer-based models like BERT. Moreover, we assumed that a transfer learning technique like transformers could practically improve on the accuracy of our model performance.

## 3.4 Skill extraction

The two models successfully trained, we proceeded to use either of them to annotate our unseen data ( at this point, our unseen data was the university data and the unlabelled jod data), to all not annotated data science jobs and the university courses. In this case we chose the best performance model. In order to do this, we loaded the *model-best* config file we got as output after training the model, and loaded it in the *nlp* function using **spaCy.load**, then passed our university data set in the function. The transformer model performed significantly good and enabled us to extract the skills from the university data set into our two main classes, SOFT and HARD. The output was saved a json files, which we would then use to performed a similarity analysis against the LinkedIn data set.

## 3.5 Similarity computation methods

Following our researches, there exist multiple machine learning techniques to evaluate similarity between data sets. the choice of the methods mainly depends on the nature of the data sets. Amongst others, we have the most famous being *Cosine similarity* and *Euclidean distance.* In our case, we were faced with a problem of *semantic similarity* computation. According to Slimani (2013), semantic similarity measures compute the similarity between concepts/terms included in knowledge sources in order to perform estimations. It measures the semantic distance between 2 concepts according to ontology. Studies also showed that there are different categories of semantic similarity measures. In our project, considering the nature of the data sets we had, the ideal category would then be *Knowledge-based similarity* or *Feature-based similarity.* It is a category of semantic similarity measure that focuses on concepts and computing the similarity in the meaning of these concepts. We therefore had to figure out a way to perform this test for similarity while including the semantic parameter. Our researches led us to the spacy which offered a similarity computation based on cosine similarity; **spacy.similarity**.

The next chapter will present further analysis and the results.

# 4 Results

## 4.1 Model training

Overall both models yielded significantly good performances. The transformer based model yielded a better performance on the validation set with an F-Score of 0.6602 then the basic model (0.5932). However it took about 2 hours longer to train on the GPU than the basic model on the CPU.

| **Model** | **Precision** | **Recall** | **F-Score** |
|---|---|---|---|
| Spacy basic NER | 0.5620 | 0.6280 | 0.5932 |
| label HARD | 0.5921 | 0.6718 | 0.6294 |
| label SOFT | 0.3964 | 0.4090 | 0.4026 |
| Spacy transformer NER | 0.7054 | 0.6204 | 0.6602 |
| label HARD | 0.7412 | 0.6536 | 0.6946 |
| label SOFT | 0.5235 | 0.4545 | 0.4866 |

Table 4.1: NER-trained Model Results

Table 4.1 shows a detailed overview of the overall performance of both model with respect to each annotation class. We noticed that both models had hard times in predicting soft skills as compared to hard skills. There could be various reasons for this, firstly the job advertisements were highly flooded with technical skills and technologies which were classified as hard skills. This wasn't the case for soft skill, causing a class imbalance. Secondly, hard skills are easier to enumerate as they are specific and easily identifiable. Soft skills on the other hand, which are mostly qualitative appreciations of people are less intuitive in identification. Moreover soft skills are not easy to cite as the make more sense as sentences, than single words, thus increase difficulty in identification.

Figure 4.1 to Figure 4.4 are visual representation of the population for each class. The imbalance we mentioned about earlier can easily be notice via these word clouds, specifically with the university data set, and accounts for the poor performance of the model on this minority class. Following the results from training our models, we moved on to use the best performing model(transformer) to annotated our unseen data.

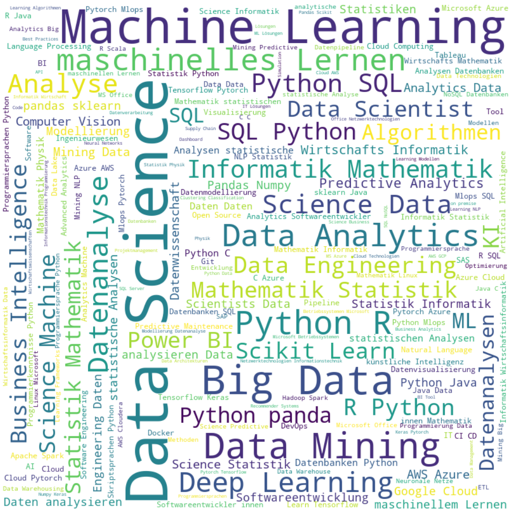

Figure 4.1: Word Cloud of LinkedIn Hard Skills

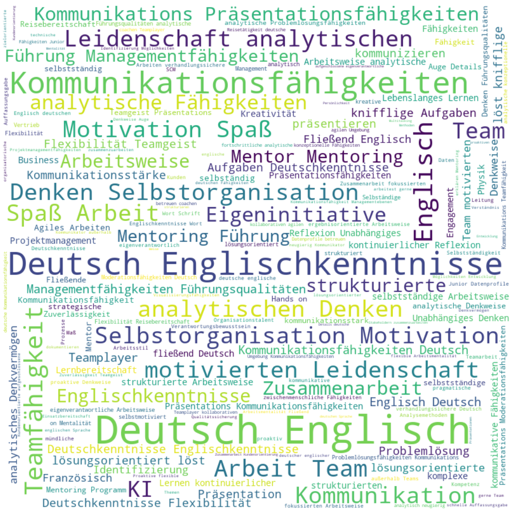

Figure 4.2: Word Cloud of LinkedIn Soft Skills

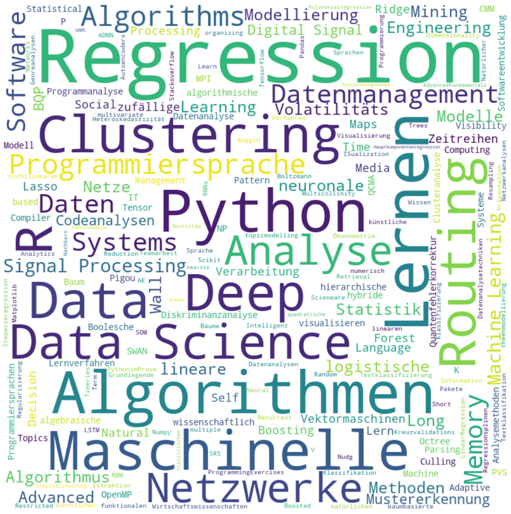

Figure 4.3: Word Cloud of University Hard Skills

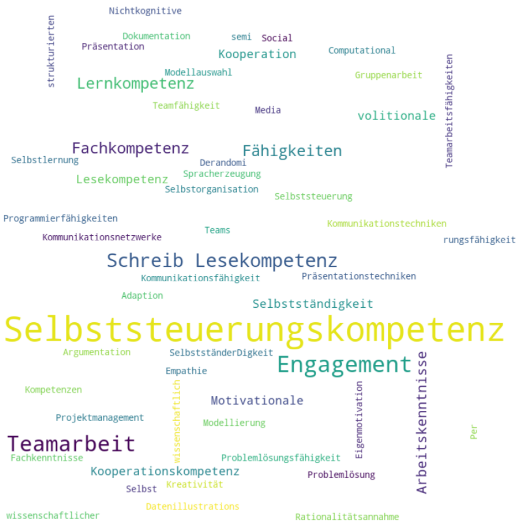

Figure 4.4: Word Cloud of University Soft Skills

## 4.2 Skill extraction

The skill extraction from the unseen data served as a good test for our NER model. We wanted to annotated this data in a smart way, but at the same time observe how it performs, or how it was able to recognise skills from text and classify them into soft and hard skills. The model (transformer) was able to extract 25760 skills from the job data set and 566 from the university data set.

| **Data set** | **Soft** | **Hard** | **Total** |
|---|---|---|---|
| University | 107 | 459 | 566 |
| LinkedIn | 4425 | 21335 | 25760 |

Table 4.2: Extracted Skills Counts

| **Soft** | **Counts** | Hard | **Counts** |
|---|---|---|---|
| deutsch | 224 | python | 1093 |
| englisch | 199 | data science | 677 |
| englischkenntnisse | 143 | sql | 492 |
| kommunikationsfähigkeiten | 135 | mathematik | 469 |
| deutschkenntniss | 79 | statistik | 443 |
| teamfähigkeit | 70 | r | 416 |
| motivation | 65 | informatik | 397 |
| flexibilität | 65 | machine Learning | 337 |
| spaß | 62 | datenbanken | 221 |
| selbstorganisation | 56 | datamining | 221 |

Table 4.3: LinkedIn Top 10 Skills

| **Soft** | **Counts** | Hard | **Counts** |
|---|---|---|---|
| Selbststeuerungskompetenz | 27 | Python | 8 |
| Engagement | 9 | Data Science | 6 |
| Teamarbeit | 8 | R | 6 |
| Schreib- und Lesekompetenz | 6 | Programmiersprache | 5 |
| Lernkompetenz | 4 | Regression | 4 |
| Fachkompetenz | 4 | Machine Learning | 4 |
| Arbeitskenntnisse | 3 | Clustering | 4 |
| Lesekompetenz | 2 | logistiche Regression | 3 |
| Selbstständigkeit | 2 | Algorithms | 3 |
| Motivationale und volitionale Fähigkeiten | 2 | Statistik | 3 |

Table 4.4: University Top 10 Skills

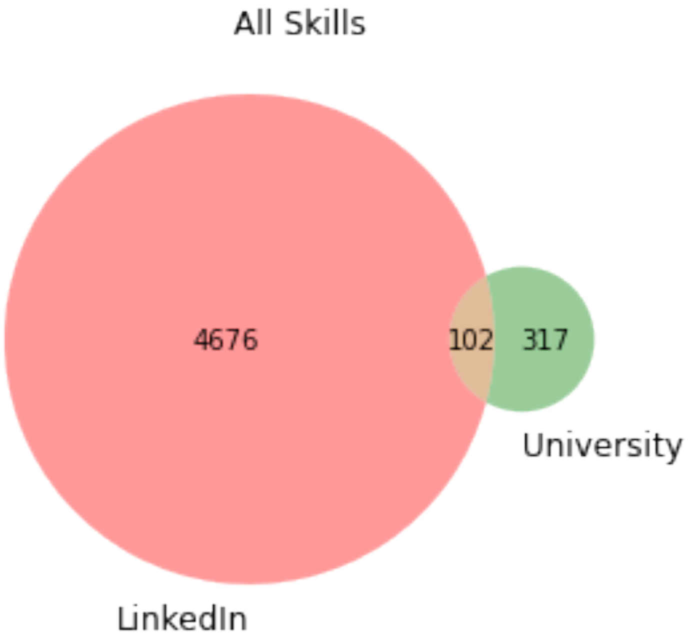

Figure 4.5: Venn Diagram of unique skills count

## 4.3 Similarities comparison

To check the similarities between the two data sets, we used the *token.similarity* method from *spacy*, which computes an estimation of *semantic similarity* between documents and returns a scalar similarity score from 0 to 1, where 1 represents the perfect similarity.

Since we had two data sets, we first separated each one into hard and soft skills data frames as shown in the tables above, then set up them as *spacy documents*, and finally investigate the similarities between each data frame with the same category hard or soft skills. We obtained a similarity score of 94.13 % on hard skills and 84.16 % on soft skills. This means that the two data sets have a high degree of semantic similarity, and they are likely to contain a large number of overlapping hard and soft skills.

However, although these significantly high results, we tried to investigate more with our analysis because the LinkedIn data set showed some technologies like AWS, PyTorch, and Apache Spark for example that were not in the university data. Therefore, we decided to use a *Venn diagram* from the *matplotlib* library and try to visualize the skills in another way, mainly by showing those specific only to LinkedIn, to the university, and the two data sets. The result was pretty good as it showed technologies in the LinkedIn circle as expected, no technologies in the university circle, and similar hard skills in the intersection.

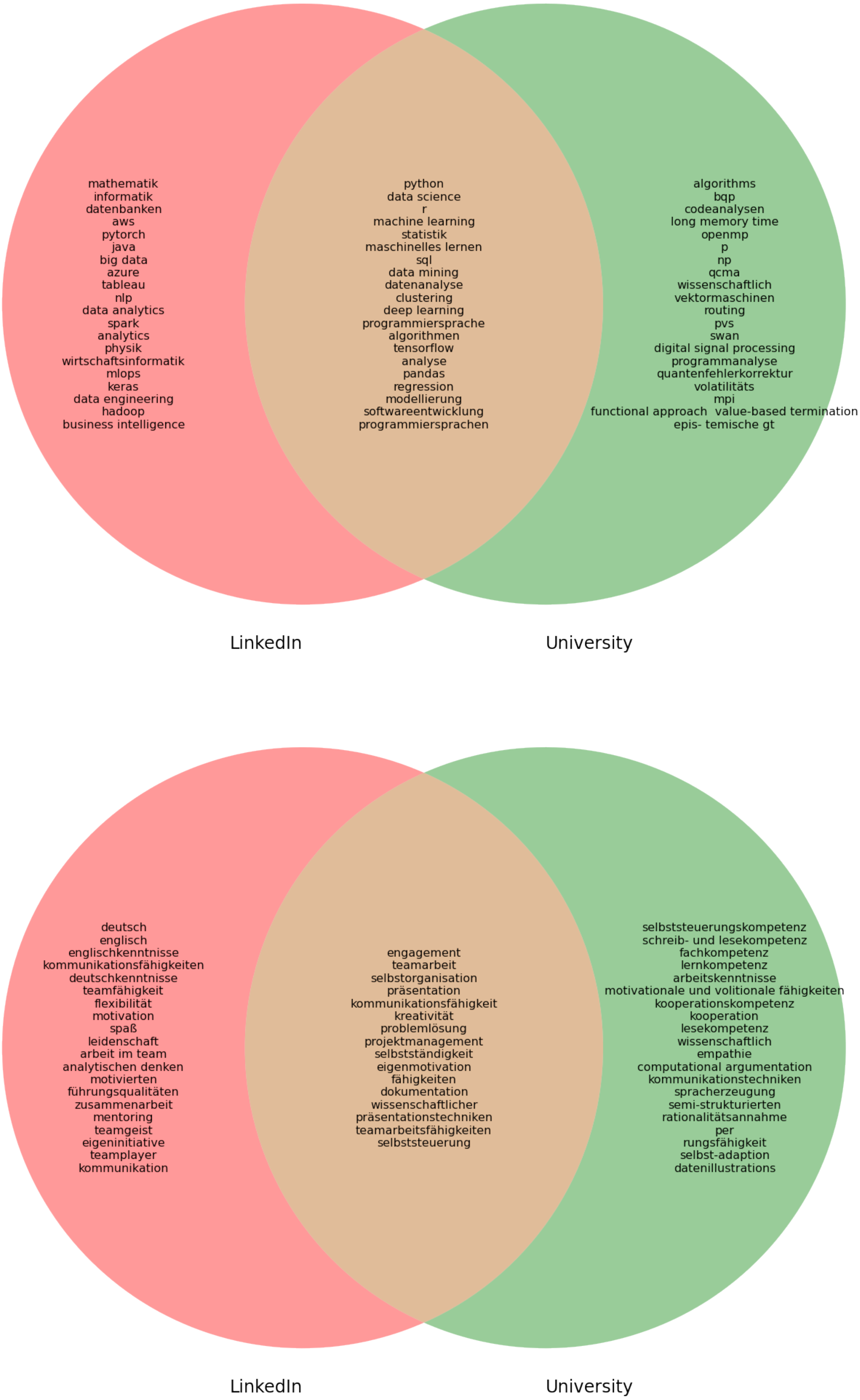

Figure 4.6: Venn Diagram of Hard Skills (upper) and Soft Skills (lower)

Then, we repeat the process to compare the similarities between soft skills and obtained the Figure 4.6. It shows that on one hand the labour market requests more specific soft skills like languages (English, German) and teamwork focused skills like mentoring. And on the other hand that the university tries to teach competences regarding reading, writing, learning and kooperation.

Overall 102 of the 419 unique skills collected from the University of Paderborn are also present in the set of skills from LinkedIn (see Figure 4.5 for a visual representation). This results in a 24.34% representation of university taught skills in the job market domain of data science and concludes our results.

# 5 Conclusion

Overall, our process resulted in the development of a transformer based model that can automatically extract hard and soft skills from job postings with an accuracy of 0.6602 measured as F1-score. An interactive demo of the NER can be found at `https://aml4ta-skills.streamlit.app/`.

A future development or training of the model could improve the capability of skill recognition and could be applied in other domains than ‘data scientist jobs’. In addition to further analysing the job skill gap of other domains the model can be used to help job seekers identify relevant job opportunities or to help employers identify suitable candidates by extracting skills from a CV.

In addition to the created extraction pipeline the built web scraper and the created dataset enable further analysis for example:

1. collect data over a longer time period and analyze changes in the skills
2. analyze the already collected full dataset of over 27.000 jobs instead only the job titles matching ’data scien*’

While training our model, we created a set of guidelines to lead annotators for Named Entity Recognition in the realm of skill extraction. These guidelines could be utilized for future research in this domain.

Apart from these achievements there is some critic or room for improvements. Our skill extraction method was based on full texts and the NER Model but nothing else. This could have been enhanced by first extract interesting parts of the ad like sentences or nouns from the full ad followed by extracting with NER from the pre selected sentences like other papers Dong and Triche (2020) did. However it would have needed more manual labour which was against the idea our proof of concept. We also could have improved the dataset by spending more time on removing duplicates. Companies uploaded the same job ad in different cities which we noticed while labeling and slowed us down.

Another possible improvements could be the expansion of the dataset. We only trained with a really small set of training data which all consisted of job ads from the data science realm. To create a more general purpose skill detection pipeline the data should be extended by multiple domains and used again for training because the job ads are all really similar and don't reflect a broad set of skills. In addition to that the ratio of university course description to job ads was quite off, because of the low number of available course descriptions of the particular selected university. This could be expanded to include more (German) universities.

Nevertheless, we are confident that the insights we have gained from our current dataset are still valuable and can help inform decision-making processes in this area. To sum up, we could accomplish our goal to get an overview about the job skill gap of university teaching and the job market of the data science domain. Our findings include that the course descriptions of the University of Paderborn do have a different focus than the skills mentioned in job advertisements. Although they overlap on the core skills, technologies and concepts. The extracted overviews of skills now can be handed out to Data Science related faculties and they can decide if they want to introduce more explicitly mentioned skills into their course descriptions to provide a better guidance for students.

# Bibliography

Ahadi, A., Kitto, K., Rizoiu, M.-A., and Musial, K. (1970). Skills taught vs skills sought: Using skills analytics to identify the gaps between curriculum and job markets.

Della Volpe, M. and Esposito, F. (2020). How universities fill the talent gap: The data scientist in the italian case. *African Journal of Business Management*, 14(2):53–64.

Dong, T. and Triche, J. (2020). A Longitudinal Analysis of Job Skills for Entry-Level Data Analysts. *Journal of Information Systems Education*, 31(4):312–326.

European Commision (2022). ESCO (European Skills, Competences, Qualifications and Occupations). `https://esco.ec.europa.eu/en`. Online; accessed 27-February-2022.

Fareri, S., Melluso, N., Chiarello, F., and Fantoni, G. (2021). Skillner: Mining and mapping soft skills from any text. *Expert Systems with Applications*, 184:115544.

Gardiner, A., Aasheim, C., Rutner, P., and Williams, S. (2018). Skill requirements in big data: A content analysis of job advertisements. *Journal of Computer Information Systems*, 58(4):374–384.

Hiranrat, C. and Harncharnchai, A. (2018). Using text mining to discover skills demanded in software development jobs in thailand. In *Proceedings of the 2nd International Conference on Education and Multimedia Technology*, ICEMT '18, page 112–116, New York, NY, USA. Association for Computing Machinery.

Ho, A., Nguyen, A., Pafford, J. L., and Slater, R. (2019). A data science approach to defining a data scientist. *SMU Data Science Review*, 2(3).

Kortum, H., Rebstadt, J., and Thomas, O. (2022). Dissection of ai job advertisements: A text mining-based analysis of employee skills in the disciplines computer vision and natural language processing. In *Proceedings of the 55th Hawaii International Conference on System Sciences.*

Lyon, L., Mattern, E., Acker, A., and Langmead, A. (2015). Applying translational principles to data science curriculum development. In *iPres 2015.*

Paderborn University (2020). JOBPROFIL FÜR WIRTSCHAFTSINFORMATIKER*INNEN. `https://www.uni-paderborn.de/en/studyoffer/course_of_study/wirtschaftsinformatik-master`. Online; accessed 27-February-2022.

Shmatko, N. and Volkova, G. (2020). Bridging the skill gap in robotics: Global and national environment. *Sage Open*, 10(3):2158244020958736.

Slimani, T. (2013). Description and evaluation of semantic similarity measures approaches. *International Journal of Computer Applications*, 80(10):25–33.

Statistisches Bundesamt (2022). Einwohnerzahl der größten Städte in Deutschland am 31. Dezember 2021. `https://de.statista.com/statistik/daten/studie/1353/umfrage/einwohnerzahlen-der-grossstaedte-deutschlands/`. Online; accessed 19-February-2022.