TU WIEN Informatics

Automated Skill Extraction and Classification from German Job Listings using Transfer Learning

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Malte Scheuvens
Matrikelnummer 01326499

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dr.-Ing. Fazel Ansari
Mitwirkung: Univ.Lektor Dipl.-Ing. Linus Kohl

Wien, 23. Jänner 2025

Malte Scheuvens	Fazel Ansari

TU WIEN Informatics

Automated Skill Extraction and Classification from German Job Listings using Transfer Learning

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Malte Scheuvens
Registration Number 01326499

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dr.-Ing. Fazel Ansari
Assistance: Univ.Lektor Dipl.-Ing. Linus Kohl

Vienna, January 23, 2025

Malte Scheuvens	Fazel Ansari

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Ich habe zur Kenntnis genommen, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

Diplomarbeit

nur mit Bewilligung der Prüfungskommission berechtigt bin.

Ich erkläre weiters Eides statt, dass ich meine Diplomarbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen selbstständig ausgeführt habe und alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, genannt habe. Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ bzw. „Overview of Additional Tools Used“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden.

Weiters erkläre ich, dass ich dieses Diplomarbeitsthema bisher weder im In- noch Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe und dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Wien, 23. Jänner 2025

_____

Malte Scheuvens

Danksagung

Ich möchte mich an dieser Stelle bei meinen Betreuern Linus Kohl und Fazel Ansari bedanken, ohne dessen Feedback und Geduld die Diplomarbeit in dieser Form nicht möglich gewesen wäre.

Ein weiterer Dank geht an meine Kolleginnen und Kollegen bei der Fraunhofer Austria Research GmbH, welche mir mit ihrer Erfahrung und ihrem Interesse am Thema stets geholfen und mich motiviert haben.

Des Weiteren möchte ich mich bei meinen Eltern und meiner Familie bedanken, die mich nicht nur während des Studiums, sondern auch in all den Jahren davor stets unterstützt haben. Ohne sie wäre ich heute nicht hier.

Nicht zuletzt möchte ich mich bei meiner Freundin bedanken, welche stets ein offenes Ohr hatte, wenn mal nicht alles lief wie geplant, und die mir mit ihren Ratschlägen immer wieder aufs Neue geholfen hat.

Kurzfassung

Die Fähigkeiten und Kenntnisse der Menschen spielen eine wichtige Rolle bei den sich ständig ändernden Anforderungen des Arbeitsmarktes. Die automatisierte Erfassung und Klassifizierung von Fähigkeiten können einerseits dazu beitragen, neue Trends bei den Bedarf nach neuen Fähigkeiten zu erkennen, andererseits aber auch die Vermittlung zwischen Arbeitssuchenden und Arbeitgebern zu unterstützen. Um dies zu erreichen, ist eine standardisierte Klassifizierung der Fähigkeiten erforderlich. Die vorliegende Arbeit befasst sich mit den Herausforderungen der automatisierten Extraktion von Fähigkeiten aus deutschsprachigen Stellenangeboten. Dabei werden zwei Hauptprobleme betrachtet: das Fehlen einer standardisierten Zuordnung von Fähigkeiten zu standardisierten Kompetenztaxonomien (P1) und das Fehlen öffentlich zugänglicher Benchmarking-Datensätze (P2). Um die genannten Herausforderungen zu bewältigen, untersucht die vorliegende Arbeit den Einsatz moderner Natural-Language-Processing Methoden zur Extraktion und Klassifizierung von Fähigkeiten, wobei der Schwerpunkt auf der Klassifikation nach der European Skills, Competences, Qualifications, and Occupation (ESCO) Taxonomie liegt. Die verwendete strukturierte Methodik basiert auf dem Design Science Research Process (DSRP) sowie dem Cross Industry Standard Process for Data Mining (CRISP-DM), welche den Rahmen für das gesamte Forschungsdesign bilden. Das methodische Vorgehen umfasst zudem eine systematische Literaturrecherche, um den aktuellen Stand der Technik zu evaluieren. Dabei werden Einschränkungen, wie der Fokus auf die ESCO-Taxonomie und die Abhängigkeit von bestehenden Modellen, berücksichtigt.

In der praktischen Anwendung der Ergebnisse werden auf den Anwendungsbereich und die deutsche Sprache abgestimmte Transformatormodelle, wie beispielsweise JobGBERT, eingesetzt und sprachspezifische Vorverarbeitungstechniken eingeführt, welche die Eigenheiten der deutschen Sprache adressieren, wie die Verwendung von zusammengesetzten Wörtern und die häufige Nominalisierung von Verben. Die Ergebnisse legen nahe, dass die Einbeziehung sprachspezifischer Anpassungen die Extraktions- und Klassifikationsleistung in Bezug auf Precision, Recall und F1 zwar erheblich verbessert, die Einbeziehung domänenspezifischer Modelle jedoch in bestimmten Situationen nicht unbedingt zu einer Steigerung der Gesamtleistung führt. Des Weiteren wird durch die Erstellung eines neuartigen, Benchmarking-Datensatz aus deutschen Stellenausschreibungen der Mangel an Benchmarking-Ressourcen behoben, wodurch eine reproduzierbare Forschung sowie eine vergleichbare Bewertung von Methoden zur Extraktion von Fähigkeiten ermöglicht wird.

Zusammenfassend leistet diese Arbeit einen Beitrag zur Weiterentwicklung des Forschungsgebiets durch die Erstellung eines Benchmarking-Datensatzes sowie der entwickelten Extraktions-Pipeline, welche den Vergleich verschiedener State-of-the-Art Modelle ermöglicht.

Abstract

People's skills and knowledge play an important role in the ever-changing demands of the labour market. Automated skill extraction and classification can, on the one hand, aid in the discovery of new trends in skill demand, but on the other, support the matching process between job seekers and employers. To achieve this, a standardised classification of skills is necessary, which helps in facilitating these matching processes. Previous approaches have been mainly focussing on extracting skills from English job listings. This thesis addresses the challenges of automated skill extraction from German job listings, focusing on two main problems: the absence of a standardised competency taxonomy mapping (P1) and the lack of publicly available benchmarking datasets (P2). To address these challenges, the thesis investigates the use of state-of-the-art transformer-based methods for skill extraction and classification, with particular emphasis on the European Skills, Competences, Qualifications, and Occupation (ESCO) taxonomy. The research involves creating a new benchmarking dataset of German job listings, specifically annotated using a developed set of annotation guidelines for German job listings. The structured research methodology used is based on the Design Science Research Process (DSRP) and the Cross Industry Standard Process for Data Mining (CRISP-DM), which serve as the guidelines for the overall research design. The methodology also includes a systematic literature review to assess the current state-of-the-art.

In the practical application of the findings, the thesis applies existing transformer models fine-tuned for the German language, such as JobGBERT, and introduces language-specific pre-processing techniques, designed to address the particularities of the German language, such as usage of compound words and the frequent nominalisation of verbs. The effectiveness of these methods is evaluated using standard performance metrics such as precision, recall, and the F1 score. Findings reveal that while incorporating language-specific adaptations substantially enhances extraction and classification performance, the incorporation of domain-specific models does not necessarily improve the overall performance in certain settings. Additionally, the creation of a novel annotated job listing dataset addresses the lack of benchmarking resources, allowing for reproducible research and comparable evaluation of skill extraction methods.

This thesis contributes to advancing the field of skill extraction and classification from job listings, particularly within the context of the German labour market, by providing a novel annotated German job listing dataset as well as structured skill extraction pipeline, which enables the comparison of different state-of-the-art models.

Table of Contents

1 Introduction

People's knowledge and skills play a central role in many human resources (HR) processes and the labour market. This starts with creating tailored job listings and continues along the hiring process, the selection of internal training, and efficient shift planning based on the individual skill profiles within the team (Decorte et al., 2022). At the same time, due to rapid technological advancements and increasing automation, the labour market is constantly evolving to meet new skill and job demands, and job tasks are changing at an unprecedented pace. Examples of this include the increased importance of digital skills and the demand for related job profiles, like data analysts (Shakina et al., 2021). This presents a significant challenge for both job seekers and employers in understanding the evolving labour landscape, identifying new skills, and remaining competitive in their respective fields (Steiber et al., 2021). To address these challenges, company and market data are increasingly being analysed for skill information to help identify potential trends early on (Cheng et al., 2021; Gnehm & Clematide, 2020; Grüger & Schneider, 2019). While identifying trends is only one possible use-case for skill data analysis, the gathered insights can also provide information about the training needs of employees, aid in competence-based shift planning (Ansari et al., 2023) or allow for the pre-selection of individuals for suitable job listings. For this, the skill identification accuracy and the comparability between individual skill profiles and the required needs are highly important (Konstantinidis et al., 2022).

To achieve this comparability and compatibility, structured skill taxonomies, such as the ESCO (**E**uropean **S**kills, **C**ompetences, Qualifications, and **O**ccupation) taxonomy, were created. The ESCO taxonomy is a multilingual classification system developed by the European Union to standardise and harmonise information about skills, competences, qualifications, and occupations across different countries and sectors. It was created to facilitate the exchange and comparison of labour market information, job vacancies, and resumes, enabling a better understanding and matching of skills between job seekers and employers in Europe. The ESCO taxonomy consists of two main pillars: the occupation pillar, containing 3008 individual occupations, and the skills pillar. The ESCO skills pillar consists of 13,890 individual concepts. These concepts can be subdivided into four categories: Knowledge, Skills, Transversal skills and competences, and Language skills and knowledge. Each concept has its preferred label in the 27 supported languages (European Commission, 2022a).

The complexity and granularity of these predefined taxonomies, combined with the vast amounts of available data, cause significant complexities not only for appropriate matching of demanded and supplied skills and competences but also in terms of updating and revisioning the taxonomies over time and thus motivate the use of automated skill extraction and identification methods.

1.1 Problem Definition and Research Questions

When it comes to fully automated skill extraction and identification, the difficulty now lies in a) the accurate extraction of relevant skills from unstructured text and b) the correct labelling of the skills according to a predefined taxonomy. Point a) is important because much of the available data, like job listings and internal job descriptions, are typically only available in an unstructured form with many different layouts. Point b) is necessary to transform the extracted skills into a standardised language and put individual skills into relation to one another. This allows drawing additional inferences from the skill profiles. Additionally, in the context of German job listings, the applicability to the intricacies of the German language, such as the frequent usage of compound words, the nominalisation of verbs, and much larger flexibility in terms of word order, is of high importance. For skill extraction and identification, transformer-based methods still represent the current state-of-the-art in natural language processing (NLP) (Gnehm, Bühlmann, & Clematide, 2022). Although the training of such transformers requires large amounts of raw text data, their wide applicability to written text makes them very suitable for transfer learning (Devlin et al., 2019). By utilising knowledge learned from one task or domain, transfer learning allows transformers to generalise well to new tasks with limited or no labelled training data and accelerate model training (Devlin et al., 2019). Considering the current state-of-the-art, while some papers deal with skill extraction by word frequencies (e.g. Gurcan & Cagiltay (2019) and Wu et al. (2020)), only a few deal with mapping the extracted skills to a defined taxonomy (e.g. the ESCO). However, this is central for comparability and, thus, the analysis of potential skill gaps and possible learning paths. Essential scientific publications in this context are Decorte et al. (2022), Fareri et al. (2021), Konstantinidis et al. (2022), and Zhang et al. (2022). They have in common their use of various pre-trained transformer models and the ESCO taxonomy for the classification of skills via semantic similarities, and their sole focus is on the English language (P1). A publication dealing with the applicability of these methods to other languages is Zhang, Jensen, & Plank (2022), with a focus on Danish. Works dealing with skill extraction from German texts are Gnehm et al. (2022) and Grüger & Schneider (2019), but here an integration into existing skill taxonomies is omitted, or task specific pretraining required (P1). Finally, none of the works focusing on German job listings freely published their benchmarking data sets, making a reproduction of the results impossible (P2).

Considering the discussion above, this thesis deals with two main problems:

- P1: There are currently no works, to the best of the author's knowledge, that extract skills from German job listings, without task specific pre-training, while also creating a mapping to a standardised skill taxonomy.

- P2: At the same time, no publicly available benchmarking data set exists for evaluating and comparing the performance of different skill extraction and classification approaches.

This thesis explores the applicability of current state-of-the-art transformer-based skill extraction and classification approaches on German job listings and the impact of language-specific pre-processing and pre-training on the model performance. The model performance is evaluated using a set of standard performance measures, including accuracy, precision, recall, and F1 score, the harmonic mean of precision and recall. Continuing from the problem statement, the following guiding research question will be examined: “How can language-specific adaptations improve the current state-of-the-art for automated skill extraction from German job listings?”.

This main research question can be broken down into three sub-questions:

- RQ1: To what extent do state-of-the-art skill extraction and classification methods perform effectively (considering standard performance measures) on German job description data sets?
- RQ2: What is the impact of language-specific pre-processing on the performance measures, compared to the state-of-the-art?
- RQ3: How does the selection of different pre-trained models affect the overall extraction and classification performance?

The subsequent objectives (Ox) of this thesis are thus to create a valid benchmarking data set (O1), compare the applicability of existing state-of-the-art methods on German job listings (O2), evaluate whether language-specific pre-processing or pre-training improves the extraction and classification results (O3), and finally compare the performance of different pre-trained models and pre-processing steps on the German benchmarking data set (O4).

1.2 Methodology and Limitations

The research of this thesis follows the Design Science Research Process (DSRP) proposed by Peffers et al. (2007). The DSRP is a nominal process consisting of six iterative activities for conducting research in information systems (IS). The model provides four different entry points for research, see Figure 1. In alignment with the research questions and objectives, this thesis will focus on developing an objective-centred solution. It looks at the results of an existing artefact (i.e. state-of-the-art skill extraction and classification techniques) and tries to improve them by introducing an alternative artefact (i.e. a skill extraction and classification pipeline adapted to the German language).

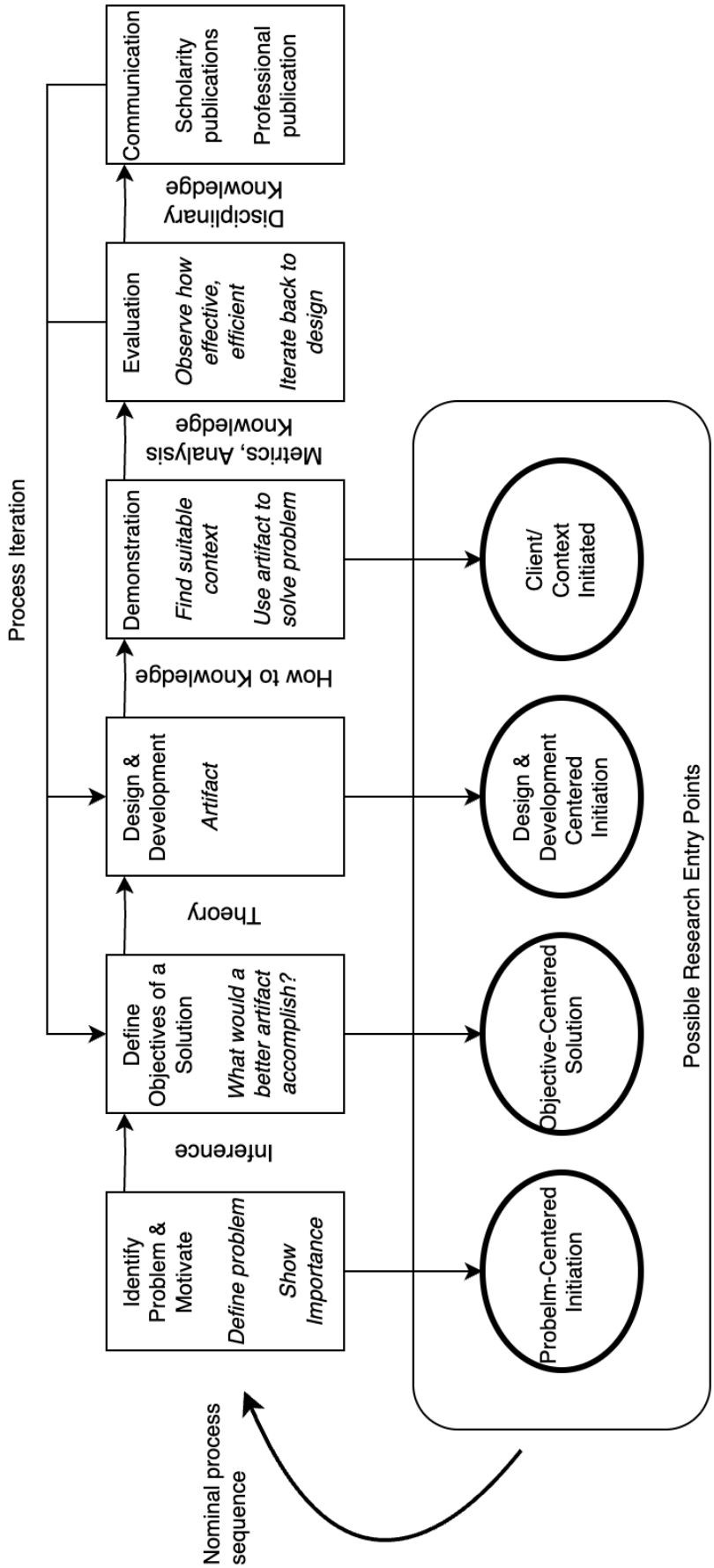

Figure 1: Design science research process (DSRP), cf. Peffer et al. (2007, p. 54)

For the design & development of the artefact, the skill extraction and classification pipeline adapted to the German language, this thesis is aligned with the Cross Industry Standard Process for Data Mining (CRISP-DM). CRISP-DM consists of six phases: business understanding, data understanding, data preparation, modelling, evaluation, and deployment, see Figure 2 (Chapman et al., 2000; Shearer, Colin, 2000). Specifically, this thesis will focus on the four middle phases: data understanding, data preparation, modelling, and evaluation. Within the first phase, data understanding, the existing data, i.e. available German job listings and the ESCO taxonomy, will be examined and discussed. The results of this step, including any language-specific findings, will then be used as the basis of the data preparation phase, where the datasets will be cleaned, transformed, and prepared for the subsequent modelling phase. Within the modelling phase, different models will be selected for the task and compared to another one using different parameter settings and pre-processing steps. The performance of the models will be compared using different performance measures, but mainly focusing on the F1 score for the span prediction and skill classification tasks. Finally, the performance results and gained insights will be discussed in terms of the research questions and the underlying problem statements of this thesis during the evaluation phase.

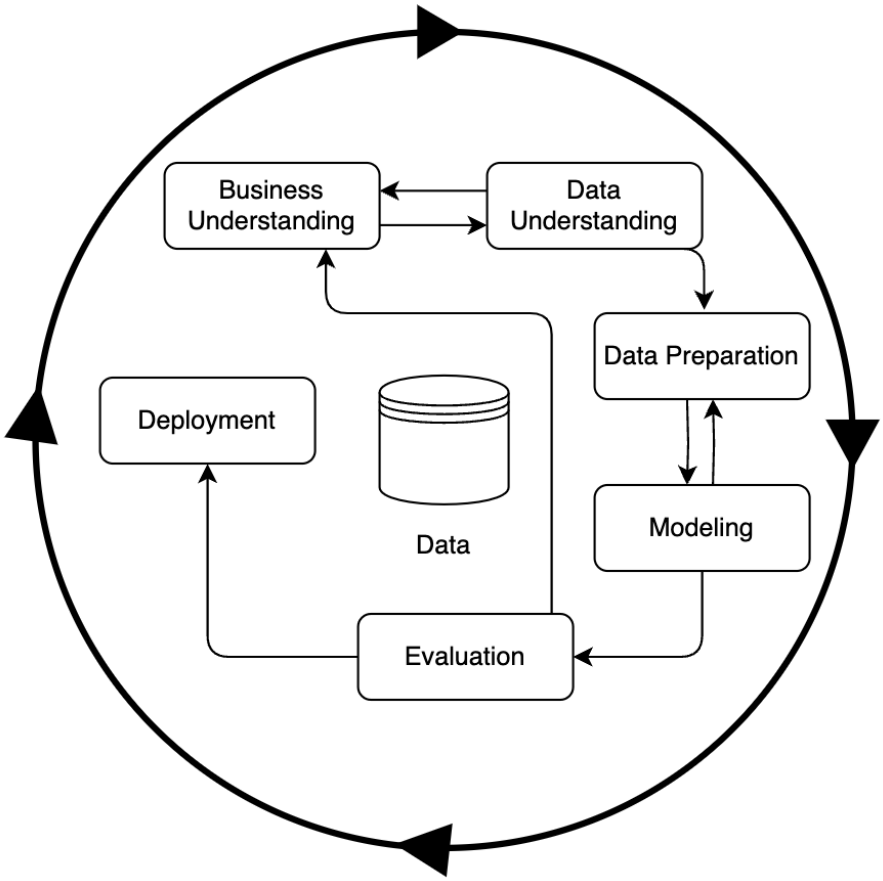

Figure 2: Process diagram showing the relationship between the different phases of CRISP-DM, cf. Chapmann et al. (2000, p. 13)

Additionally, to review the current state-of-the-art in terms of skill extraction and classification, a literature review will be carried out following the methodology described by Zonta et al. (2020). For this, a search string is constructed and used to query the IEEE, Scopus, and ScienceDirect scientific databases. Web of Science and Scopus were chosen due to their comprehensive indexing of peer-reviewed journals, also including Springer Nature journals, ensuring access to high-quality scholarly outputs. The IEEE library was selected for its specialised coverage in technology and engineering, which is key for studies on automated skill extraction and classification. The resulting state-of-the-art (SOTA) will be categorised and compared using a set of defined criteria, such as extraction objectives, predefined skill taxonomies, classification granularity, machine learning techniques, pre-trained models, and examined languages.

The main limitations of this research are two-fold: firstly, the focus is on a single skill taxonomy (ESCO) which means the approach and the results may not directly translate to other taxonomies. Secondly, due to computing constraints, no new transformer model will be trained. Instead, only existing models will be examined with only minor adaptations.

2 Theoretical Background and Related Work

In the following chapter, related work and important theoretical concepts of this thesis will be explored. These include, among other things, an overview of the ESCO Taxonomy, key concepts in the context of NLP techniques, a definition and introduction to transfer learning and finally, transfer learning in the context of NLP.

2.1 The ESCO Taxonomy

Skill taxonomies and occupational classifications provide a standardised language for skills and occupations and enable various beneficial applications, such as automatic candidate-job matching, identifying skill gaps or finding individualised career paths (le Vrang et al., 2014).

On a national level, several such occupational classifications exist, for example:

- **BERUFENET** – by the German Bundesagentur für Arbeit (German public employment service)
- **O*NET** (Occupational Information Network) – by the US Department of Labor
- **BIS** (Berufsinformationssystem) – by the Austrian Arbeitsmarktservice AMS (Austrian public employment service)

All the above also include lists of necessary skills for the included occupational profiles, as well as detailed task descriptions. To promote job mobility between different countries with different national classifications and languages, it is necessary to enable data exchange and standardised concepts between different national and international private and public employment services (le Vrang et al., 2014).

The ESCO taxonomy was specifically created to fill this gap. It is a multilingual classification of European Skills, Competences, Qualifications, and Occupations and is developed and maintained by the European Commission. It serves as a standardised taxonomy for describing and categorising skills, competences, qualifications, and occupations across different countries and sectors within the European Union, see Figure 3 (European Commission, 2017). This interoperability, combined with the fact that the ESCO taxonomy is well documented and has been published as open data, are also the main reasons why it was chosen as the reference skill knowledge base of this thesis. Other national German-speaking occupational classifications such as the German BERUFENET and the Austrian BIS were excluded due to their closed databases, which are only accessible through their respective web portals.

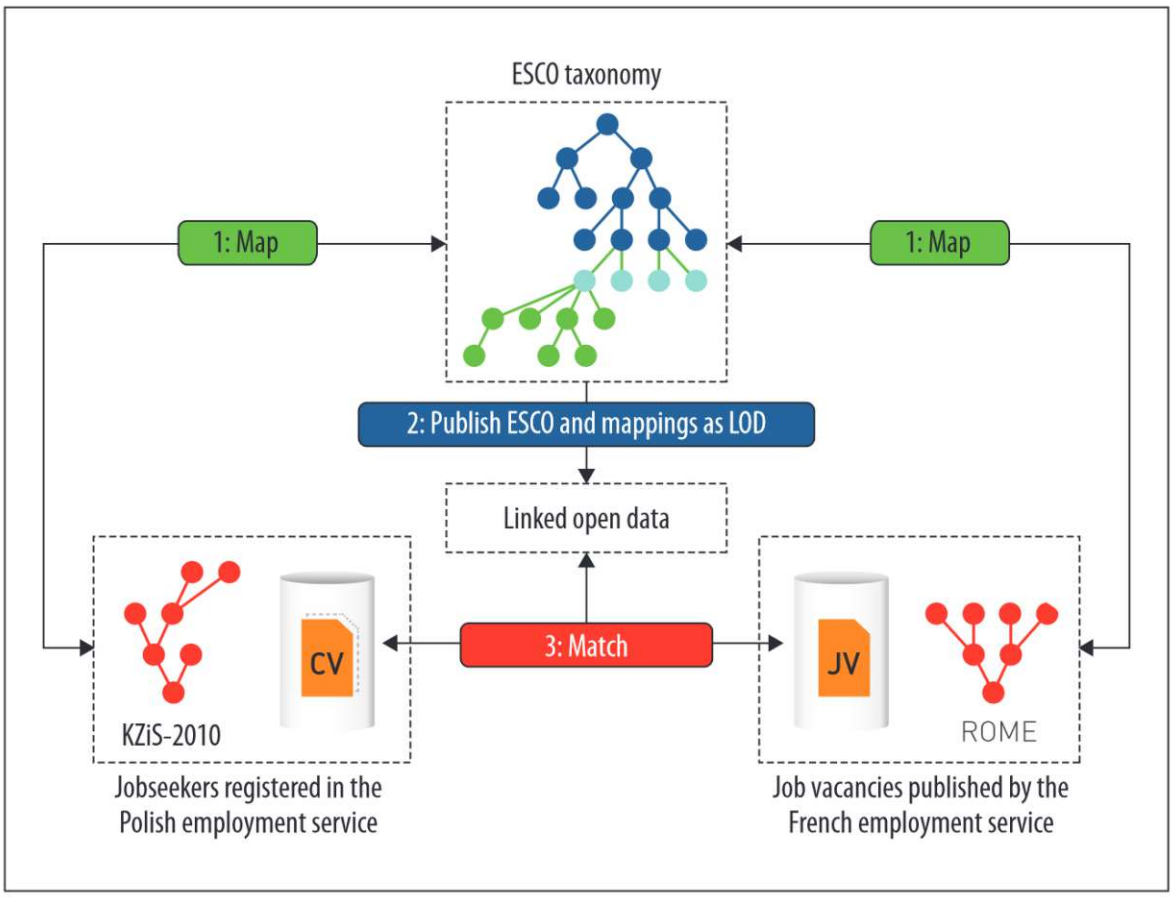

Figure 3: Schematic showing how ESCO will serve as an exchange hub for employment services using different occupational classifications and languages (le Vrang et al., 2014, p. 60).

2.1.1 Skill and Knowledge Definition

Since this thesis uses the ESCO Taxonomy as its skill knowledge base, the wording and definitions of relevant terms will also be aligned with it. ESCO uses definitions in line with the European Qualifications Framework (EQF), which are as follows:

- The term **knowledge** is defined as: "[…] the outcome of the assimilation of information through learning. Knowledge is the body of facts, principles, theories and practices that is related to a field of work or study." (European Commission, 2023b)
- A **skill** is defined as: "[…] the ability to apply knowledge and use know-how to complete tasks and solve problems." (European Commission, 2023c)
- **Competence,** on the other hand**,** is defined as: "[...] the proven ability to use knowledge, skills and personal, social and/or methodological abilities, in work or study situations and in professional and personal development." (European Commission, 2023a)

The difference between skill and competence, according to the EQF, lies within their scope. While a skill usually refers to the "use of methods or instruments in a particular setting and in relation to defined tasks", competence is broader in scope and refers to

“the ability of a person - facing new situations and unforeseen challenges - to use and apply knowledge and skills in an independent and self-directed way” (European Commission, 2023a).

In addition to the distinction between skill and competence, there also exists an ongoing discussion about the usage and the differences between competence, competency, and competencies (Moghabghab et al., 2018; Teodorescu, 2006). Even though the EQF makes a distinction between skills and competencies, the ESCO taxonomy itself does not. Within its skills pillar, only knowledge concepts and skill/competence concepts (from now on referred to as just “skill concepts”) are differentiated (European Commission, 2023f).

This is in line with other current works on the topic of skill/competence extraction, where the term “skill” is used almost exclusively (Decorte et al., 2022; Fareri et al., 2021; Konstantinidis et al., 2022; Zhang, Jensen, & Plank, 2022). Because of this, and in line with the usage within the ESCO taxonomy, this thesis will also only differentiate between skill and knowledge components for the extraction task, relying on the definitions made within the EQF.

2.1.2 Purpose and Structure

The ESCO taxonomy is designed to provide a common language and structure for matching job seekers' skills and qualifications with job vacancies, facilitating labour market transparency, and promoting mobility and employability within the EU. It includes hierarchical classifications for various skills, qualifications, and occupations, making it easier to understand and compare qualifications and job requirements across different European countries, see Figure 4.

The European Commission has developed the ESCO taxonomy with several key objectives in mind. Firstly, it aims to enhance communication between the education and training sector and the EU labour market. Secondly, ESCO intends to facilitate geographical and occupational mobility within Europe. Thirdly, it seeks to improve data transparency and accessibility for various stakeholders, including public employment services, statistical organisations, and educational institutions. Fourthly, ESCO aims to enable seamless data exchange between employers, education providers, and job seekers across different languages and countries. Lastly, the taxonomy strives to support evidence-based policy-making by enhancing data collection, comparison, and dissemination through skills intelligence and statistical tools, enabling real-time analysis of skills supply and demand using big data (European Commission, 2017).

Structurally, the ESCO taxonomy consists of three pillars: the occupations pillar, the skills pillar, and the qualifications pillar:

- The **occupations pillar** consists of 3008 individual occupation concepts (as of version 1.1). These concepts are organised hierarchically, using the International Standard Classification of Occupations (ISCO-08) for the top four levels. Each ESCO occupation is then assigned to exactly one ISCO-08 unit (European Commission, 2023d).
- The **skills pillar** consists of 13890 individual concepts (as of version v1.1). Within the skills pillar, a distinction is made between knowledge concepts and skill concepts. As already mentioned, no difference is made between skill and competences within the skill pillar (European Commission, 2023f).
- And finally, the **qualifications pillar**, which has now been integrated into Europass. It contains information on qualifications at the European level and their relationships with the skills and the occupations pillar. It is based on the European Qualifications Framework (EQF) (European Commission, 2023e).

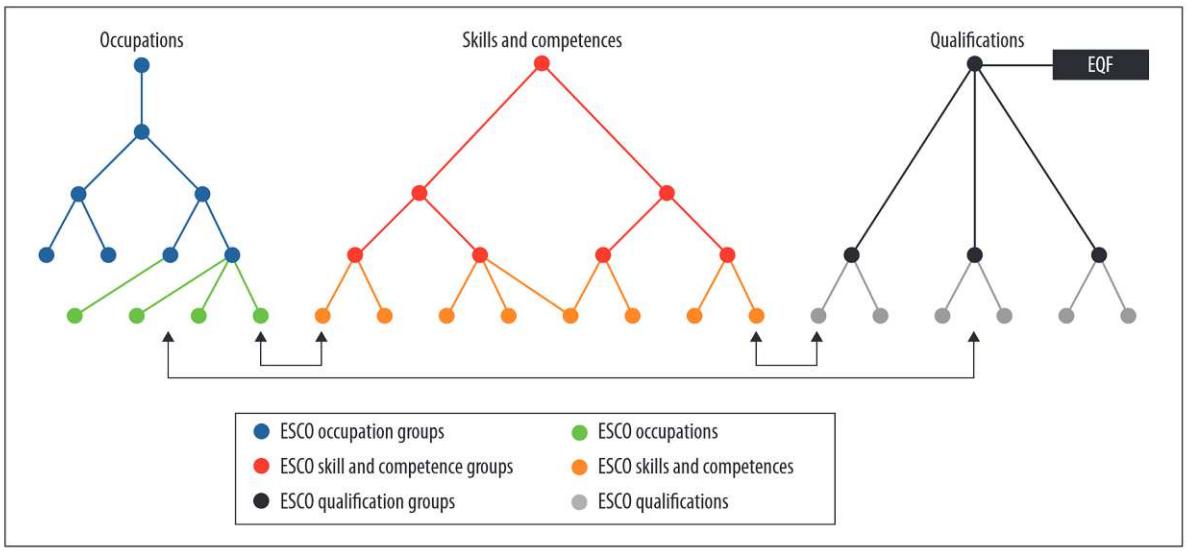

Figure 4: Overview of the three pillars of the ESCO classification and the European Qualifications Framework (EQF) (le Vrang et al., 2014, p. 58)

Within each pillar, a distinction exists between concepts and terms. A *concept* represents a universal understanding or idea and is not dependent on language—for example, the concept of a person baking bread and selling it to customers. *Terms*, on the other hand, refer to the linguistic descriptions of concepts and are language-specific. For instance, in English, the term "baker" is used for the concept mentioned earlier, while in German, it is "Bäcker/Bäckerin". In ESCO, each concept is associated with at least one term in all its 27 supported languages. Multiple terms can exist for a single concept within a language. ESCO uses three types of terms: preferred terms, which are unique and best represent the occupation or skill; non-preferred terms, which include synonyms, variations, or abbreviations of the preferred term; and hidden terms, which capture outdated, misspelt, or politically incorrect terms for indexing and searching purposes but are not visible to end users (European Commission, 2017).

2.2 Natural Language Processing (NLP)

The following chapter will give a short overview of NLP and dive deeper into the tasks of information extraction and text classification. It will then go into more detail about different representation techniques for text and finally examine transformers and their technical build-up, which are highly relevant in the context of this thesis.

2.2.1 Overview

NLP is a subfield of computer science that can be divided into natural language understanding (NLU) and natural language generation (NLG), which focus on using computational techniques to learn, understand and (re-)produce human language content (Pais et al., 2022). It has evolved from early often manual approaches in language research towards automating linguistic analysis like sentiment analysis, and technologies like machine translation, speech recognition and speech synthesis (Goyal et al., 2018). Today's advancements in NLP are attributed to four key factors: increased computing power, access to vast linguistic data, successful machine learning (ML) methods, and a deeper understanding of human language structure and its use in social contexts (Hirschberg & Manning, 2015). Common tasks encountered in NLP include, among other things: *Text classification*, where text is automatically categorised into predefined categories based on its content, like its overall sentiment, *information extraction*, where relevant information is extracted from text, like places or people's names, or, in the context of this thesis, skill requirements from job listings, and *topic modelling,* which is often used in text mining to uncover underlying topical structures within large collections of documents (Sowmya V. B et al., 2020). A generic NLP data pipeline can be seen in Figure 5.

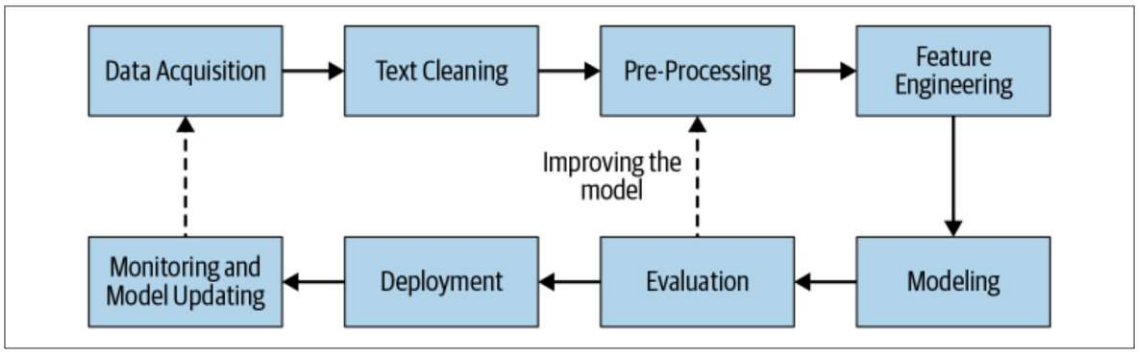

Figure 5: Generic NLP pipeline (Sowmya V. B et al., 2020, p. 38)

This pipeline consists of eight individual steps following Sowmya V. B et al. (2020):

- **Data acquisition**: Obtaining the necessary data required for the NLP application.
- **Text cleaning**: Cleaning and preparing the text data by removing noise, irrelevant information, or formatting issues.

- **Pre-processing:** Performing various text processing tasks, such as tokenisation, stemming, and lemmatisation.
- **Feature engineering:** Extracting relevant features from the pre-processed text data to be used as inputs for the NLP model.
- **Modelling:** Building and training the NLP model using the prepared data and features.
- **Evaluation:** Assessing the performance of the NLP model to ensure it meets the desired requirements and objectives.
- **Deployment:** Integrating the trained NLP model into the intended application or system.
- **Monitoring and model updating:** Continuously monitoring the performance of the deployed model and updating it as needed to maintain its accuracy and relevance.

The last two steps, namely *deployment* and *monitoring and model updating* are not within the focus of this thesis.

2.2.2 Information Extraction and Text Classification

In view of the artefact to be developed, specifically the creation of the skill extraction and classification pipeline, this thesis will focus on the NLP tasks of *information extraction in the form of skill spans and text classification* to assign the correct ESCO label to each span.

Text classification deals with categorising text data into predefined classes of one or more categories. Depending on the scope and the goal, the classification can happen on the document level, paragraph level, sentence level, and sub-sentence level (Kowsari et al., 2019). There are three types of text classifications based on the number of categories: binary, multiclass, and multilabel classification (Sowmya V. B et al., 2020):

- **Binary Classification**: In this type, the text is categorised into two classes. An example would be classifying emails as spam or not spam.
- **Multiclass Classification**: Here, the text is categorised into more than two classes. For instance, classifying the different sections of a job listing into heading, requirements, company description and so on, as was for example done by Grüger & Schneider (2019)
- **Multilabel Classification**: This type allows a document to have one or more labels/classes attached to it. Each document can belong to none, one, or multiple classes.

Information extraction (IE) involves extracting relevant information from unstructured text documents. Unlike structured data sources like databases, text lacks a predefined schema, making IE a challenging task. IE involves tasks like key phrase extraction, named entity recognition (NER), entity disambiguation and linking, and relationship extraction. Specifically, the extraction of skills from unstructured text documents can

also be treated as an NER problem, as was shown by Fareri et al. (2021). Since its inception in 1996, various methods have been employed to identify Named Entities. Initial methods were centred on handcrafted rules, effective in certain areas as noted in Goyal et al. (2018). For example, by employing patterns based on word tokens and POS tags, as seen in tools like spaCy's EntityRuler[1]. However, modern methodologies utilise machine learning to address the limitations of rule-based systems, which often lack adaptability and require significant effort and expertise for development and upkeep. Presently, neural network models, particularly those using Convolutional Neural Networks (CNNs) are being superseded by Transformer-based architectures (Devlin et al., 2019). These incorporate word embeddings and have been successfully employed in the context of skill extraction (Fareri et al., 2021; Zhang, Jensen, & Plank, 2022).

In the case of framing NER as a sequence labelling problem, the information extraction task of NER can also be reframed as a multi-label text classification problem (Fu et al., 2021), for example using the BIO format (short for **B**eginning, **I**nside, **O**utside) presented by Ramshaw & Marcus (1995), see Figure 6. This approach has also been applied to the task of skill extraction (Fareri et al., 2021; Zhang, Jensen, Sonniks, et al., 2022). Alternatively, another labelling format, BILOU (short for **B**eginning, **I**nside, **L**ast, **O**utside, **U**nit element), has been designed to overcome some of the limitations of the BIO format, such as the inability to accommodate nesting, and has been shown to significantly outperform it on the CoNLL-2003 NER shared task (Ratinov & Roth, 2009).

2.2.3 Text Representations

When it comes to information extraction and text classification, it is important to represent text in a way that is suitable for various machine learning algorithms. For this, the text needs to be converted into a mathematical representation, such as a vector of various lengths/dimensions. The representation of text as numerical vectors is known as the vector space model (VSM) or term vector model. The following chapter will cover different text representation schemes falling within the scope of VSMs. The effectiveness of each scheme depends on how well it captures the linguistic properties of the represented text (Sowmya V. B et al., 2020). To classify different text representation approaches, first, two key concepts need to be defined:

Distributional similarity: This describes the idea, that a word's meaning can be understood from the context in which it appears, i.e. the meaning is defined by the context. For example, the word "nail" in the context of "I hammered a nail in the wall to

[1] https://spacy.io/api/entityruler

hang the new painting" most likely has a different meaning than in the context of "Check out this new nail polish I got".

Distributional hypothesis: A linguistic hypothesis suggesting that words with similar contexts share similar meanings. For instance, the words "dog" and "cat" usually occur in similar contexts, which would then indicate a strong similarity in their meanings (Firth, 1957).

```
Essex   B-ORG
,       O
however O
,       O
look    O
certain O
to      O
regain  O
their   O
top     O
spot    O
after   O
Nasser  B-PER
Hussain I-PER
and     O
Peter   B-PER
Such    I-PER
gave    O
them    O
a       O
firm    O
grip    O
on      O
their   O
match   O
against O
Yorkshire       B-ORG
at      O
Headingley      B-LOC
.       O
```

Figure 6: NER labelling example in the BIO format (Sowmya V. B et al., 2020, p. 173)

With these definitions in mind, vector representations of text can be classified into two main categories: Distributional representations and distributed representations. **Distributional representations** refer to representation schemes obtained from the distribution of words in their contexts. These schemes are based on the distributional hypotheses and use high-dimensional vectors derived from co-occurrence matrices capturing word-context relationships (Ferrone & Zanzotto, 2020). **Distributed representations** are a related concept to distributional representations and are also

based on the distributional hypothesis. But instead of using sparse and high-dimensional vectors, the word representations are transformed into compact and dense vectors. The resulting vector space is known as the distributed representation (Ferrone & Zanzotto, 2020). *Distributed* because the meaning of a word is distributed across the entire vector. This compression reduces their size and improves computational efficiency for machine learning (Sowmya V. B et al., 2020).

Distributional Representations include basic statical approaches like *one-hot encoding*, *bag-of-words* (BoW) or *bag-of-n-grams* (BoN) and *term frequency-inverse document frequency* (TF-IDF).

In **one-hot encoding**, each word in the corpus vocabulary is assigned a unique integer ID between 1 and the size of the vocabulary (|V|). Words are then represented as V-dimensional binary vectors with 0s in all positions except for the index corresponding to their ID, which is set to 1. This encoding is applied to individual words and then combined to represent sentences. For example, using an example corpus with the following word IDs: dog = 1, bites = 2, man = 3, meat = 4, food = 5, eats = 6, the sentence "dog bites man" is represented as [[1 0 0 0 0 0] [0 1 0 0 0 0] [0 0 1 0 0 0]]. Each word is represented as a six-dimensional vector, where the dimensions correspond to the size of the corpus.

The main concept of **bag-of-words** (BoW) is to represent the text as a collection of words, disregarding their order and context. The underlying assumption is that in a text classification task, text belonging to a particular class in the dataset can be characterised by a unique set of words. If two texts share similar words, they likely belong to the same class (Harris, 1954). The **bag-of-n-grams (BoN)** representation, on the other hand, addresses one of the limitations of BOW, namely, that words were treated as independent units without considering phrases or word ordering (Le & Mikolov, 2014). BoN breaks the text into contiguous chunks of n words (or tokens) called n-grams. In BoN, the corpus vocabulary (V) is a collection of all unique n-grams across the entire text corpus. This allows for capturing some context, which was not possible in previous methods (Sowmya V. B et al., 2020).

TF-IDF on the other hand is a text representation technique that addresses the issue of treating all words in a document equally important. It aims to quantify the importance of a word relative to other words in the document and the entire corpus (Sprark Jones, 1972). TF-IDF is commonly used in information retrieval systems to extract relevant documents from a corpus based on a given text query.

The intuition behind TF-IDF is to identify words that are important to a specific document but not commonly found in other documents in the corpus. It achieves this by using two measures: Term Frequency (TF) and Inverse Document Frequency (IDF):

- Term Frequency measures how often a word appears in a given document. To account for different document lengths, the term frequency is normalised by dividing the number of occurrences of the term by the total number of terms in the document.
- Inverse Document Frequency measures the importance of a term across the entire corpus. It gives higher weight to rare terms and reduces the weight of common terms, such as stop words.

The TF-IDF score for a word in a document is then calculated as the product of its term frequency and its inverse document frequency. By using TF-IDF, words that are both frequent in the document and rare in the corpus are considered to be more important for representing the content of the document. This technique allows for more meaningful and relevant text representations in information retrieval and other natural language processing tasks (Salton et al., 1975).

The advantages of all the distributional representation methods described above include their interpretability and ease of implementation. The main disadvantages, on the other hand, are threefold. First, their discrete representations of words or n-grams reduce their ability to capture the context in which a word is appearing. Second, the resulting feature vectors are sparse, high-dimensional, and increase in dimensionality with the size of the vocabulary. Third and finally, the methods cannot handle out-of-vocabulary (OOV) words, which means the methods have no way of assigning a vector to a word that has not been seen in the training data of the model (Sowmya V. B et al., 2020).

To address the disadvantages of distributional representations, **distributed representations** were introduced. One of the distributed representations comes in the form of word embeddings, where each word gets assigned a dense vector representation that should capture the distributional similarities between words as well as possible. The concept of distributional similarities between words in text representation refers to the idea that words with similar meanings or contexts are likely to be related. For example, the word "USA" might be associated with other countries, like Austria and Germany, or American cities, like New York (Sowmya V. B et al., 2020). This concept was significantly advanced by Mikolov et al. (2013) with their Word2vec model. This neural network-based model could understand word analogies, like "King - Man + Woman ≈ Queen", by representing words in a low-dimensional, dense vector space. Word2vec's methodology involves learning word representations from a text corpus, with each word's meaning derived from its contextual neighbours. The model projects these meanings into a vector space where similar words cluster together, and dissimilar words are distant. This system is efficient for machine learning tasks due to its lower dimensionality and dense vector nature.

Since the training of such word-embedding models requires large amounts of data and is computationally expensive, pre-trained word embeddings, like Google's Word2vec

(Mikolov et al., 2013), Stanford's GloVe (Pennington et al., 2014), and Facebook's fasttext (Bojanowski et al., 2017) are widely used. These are embeddings trained on large text corpora, available for use without the need for individual training, saving time and computational resources.

In the examples above, only one embedding exists per word/n-gram/token, regardless of the context the word is used in. Since this is not always the case, see the concept of *distributional similarity*, the words surrounding the word in question also need to be analysed to create *contextual word representations* (Smith, 2020*)*. Approaches that are designed to address these challenges range from Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), to bi-directional LSTM models such as ELMo (Peters et al., 2018). One of latest advances in this field came in the form of the transformer architecture introduced by (Vaswani et al., 2017).

2.2.4 Transformer Models

The Transformer model architecture is a neural sequence transduction model presented by Vaswani et al. (2017) that uses an encoder-decoder structure with stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, see Figure 7.

Encoder: The encoder takes an input sequence of tokens (words or subwords) and converts them into a sequence of embedding vectors. These embeddings are often referred to as the hidden state or context.

Decoder: The decoder uses the hidden state generated by the encoder to iteratively predict an output sequence of tokens. It generates tokens one at a time and uses the attention mechanism to focus on relevant parts of the input sequence during the prediction process. The output of each step is then fed back into the decoder to generate the next token until either an end-of-sequence (EOS) token is predicted, or a maximum sequence length is reached (Tunstall et al., 2022).

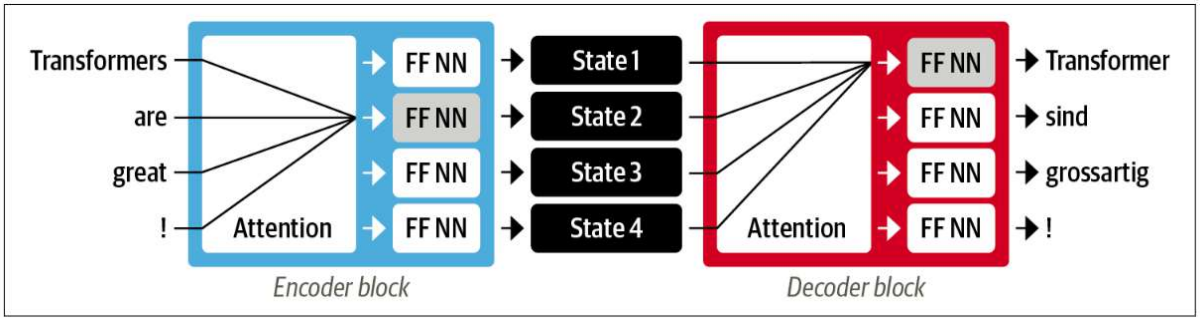

Figure 7: Encoder-decor architecture of the original Transformer (Tunstall et al., 2022, p. 6)

Instead of recurrent or convolutional layers, the Transformer model uses multi-headed self-attention, which allows for faster training times and greater parallelizability. Multi-

headed self-attention is an attention mechanism that allows the model to jointly attend to information from different representation subspaces at different positions. It works by projecting the queries, keys, and values multiple times with different, learned linear projections to different subspaces and then computing the self-attention function independently on each of these projected subspaces. The outputs of these multiple self-attention heads are concatenated and projected again, resulting in the final output of the multi-headed self-attention layer. This mechanism allows the model to capture different dependencies between different positions in the input sequence and to learn more complex relationships between them, leading to improved performance on sequence transduction tasks (Vaswani et al., 2017).

The first encoder-only model based on the Transformer architecture is the Bidirectional Encoder Representations from Transformers (BERT) model, presented by Devlin et al. (2019). BERT's distinctive feature is its unified architecture across different tasks. The pre-trained architecture is almost identical to the final downstream architecture, with minimal differences. BERT uses a multi-layer bidirectional Transformer encoder based on the original implementation described by (Vaswani et al., 2017). Unlike other language representation models at the time, BERT uses a "masked language model" (MLM) pre-training objective, inspired by the “Cloze procedure” by (Taylor, 1953), to alleviate the unidirectionality constraint of standard language models. This allows BERT to incorporate context from both directions, making it more effective for sentence-level and token-level tasks. Through this approach, BERT outperformed existing methods on a variety of NLP tasks by pre-training deep bidirectional representations from unlabelled text. These pre-trained representations can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications (Devlin et al., 2019). Many other encoder-only models have been created for specific use cases based on the original BERT model, including the following:

- SBERT: A modified version of the pre-trained BERT model, which incorporates siamese and triplet network frameworks to generate semantically significant sentence embeddings. These embeddings can then be compared efficiently using cosine similarity (Reimers & Gurevych, 2019).
- RemBERT (rebalanced multilingual BERT): a multilingual BERT model which demonstrated that decoupled embeddings enhance modelling flexibility, enabling a substantial enhancement in efficiently allocating the parameters in the input embedding of multilingual models (Chung et al., 2020).
- GBERT: A BERT-based model which has been further trained on a range of different German language corpora to improve state-of-the-art performance on German benchmarks (Chan et al., 2020).
- JobGBERT: A domain-adapted version of the GBERT model, which has been further pre-trained on German-speaking job listings (Gnehm, Bühlmann, Buchs, et al., 2022).

BERT is also regarded as one of the first Large-Language-Models (LLMs), a group of models such as OpenAI's GPT-4 model (OpenAI et al., 2024), that have revolutionised NLG. Albeit being one of the first LLMs, BERTs performance in some tasks, such as semantic clustering, is still comparative with newer models, especially when considering computational costs (Petukhova et al., 2024).

2.3 Transfer Learning

To train one of the mentioned transformer models, large amounts of training data are necessary. However, this is not feasible in every application nor is it always even possible. Transfer Learning describes the process of applying a model trained in one domain or task to a new domain or task, see Figure 8. The figure highlights a crucial concept of transfer learning: it addresses the problem of insufficient training data in the target domain by leveraging the knowledge acquired from the source domain. This enables the transfer of useful knowledge and improves the performance of the learning algorithm on the target task while addressing challenges like data scarcity, the need for model robustness, personalisation, and privacy concerns. Transfer learning is thus especially beneficial for applying complex ML models in areas with limited availability of labelled data (Yang et al., 2020). The following chapter will first describe the theoretical basics behind transfer learning and then examine how these approaches can be applied in the context of NLP and, more specifically, skill extraction and labelling.

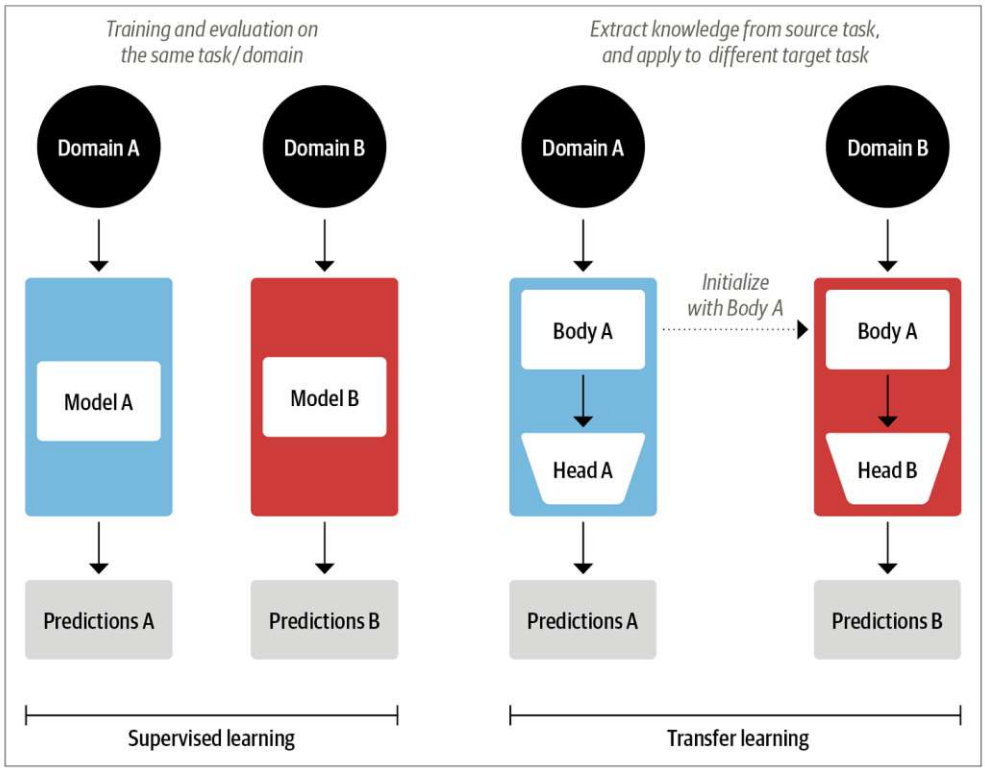

Figure 8: Comparison of traditional supervised learning (left) and transfer learning (right) (Tunstall et al., 2022, p. 7)

2.3.1 Definition

To define transfer learning, some key concepts need to be introduced, specifically, the *domain* D and *task* T. Following the notation introduced by Pan & Yang (2010) and the definitions by Yang et al. (2020), a domain D is comprised of a feature space X and a marginal probability distribution P^X, with each input instance $x \in X$. Different domains may have distinct feature spaces or marginal probability distributions. A task T is defined by a label space Y and a predictive function $f(\cdot)$, denoted as $T = \{Y, f(\cdot)\}$. The function $f(\cdot)$, is used for making predictions on unseen instances x^* and can be expressed as $P(y|x)$ in a probabilistic sense. Based on these definitions, transfer learning is defined as follows:

"Given a source domain D_s and learning task T_s, a target domain D_t and learning task T_t, transfer learning aims to help improve the learning of the target predictive function $f_t(\cdot)$ for the target domain using the knowledge in D_s and T_s, where $D_s \neq D_t$ or $T_s \neq T_t$" (Pan & Yang, 2010, p. 1347).

2.3.2 Transfer Learning Approaches

Transfer learning approaches can be classified by either the availability of labelled data in the target domain (Yang et al., 2020) or based on the relationship between source and target domain tasks (Pan & Yang, 2010). Other categorisation schemas exist, for example by the method of transfer learning, but are omitted within the scope of this chapter. Further insights into the topic of different transfer learning approaches can be found in Weiss et al. (2016) and Zhuang et al. (2021).

Following Yang et al. (2020), transfer learning approaches can be classified into three main categories Based on the availability of labelled and unlabelled data in the target domain, each requiring different approaches to leverage the available data for knowledge transfer:

- **Supervised Transfer Learning**: In this setting, only a few labelled data are available in the target domain for training. Unlabelled data from the target domain is not used during training.
- **Unsupervised Transfer Learning:** In this setting, there are no labelled data available in the target domain. The learning process relies solely on the use of unlabelled data from the target domain.
- **Semi-Supervised Transfer Learning:** In this setting, both unlabelled and labelled data are assumed to be available in the target domain. The training process benefits from having sufficient unlabelled data and a smaller set of labelled data in the target domain.

Alternatively, Transfer Learning approaches can be classified based on the relationship between source and target domain tasks, as was proposed by Pan & Yang

(2010) and is again used by Alyafeai et al. (2020), Ruder (2019), and Zhuang et al. (2021):

1. **Inductive Transfer Learning**: Here, the target task is different from the source task, regardless of whether the source and target domains are the same. This setting requires some labelled data in the target domain to develop a predictive model.
2. **Transductive Transfer Learning**: In this setting, the source and target tasks are the same, but the domains differ. No or very little labelled data are available in the target domain, but plenty in the source domain. This can be further categorized based on the feature spaces: Different feature spaces between source and target domains or the same feature spaces but different marginal probability distributions of the input data.
3. **Unsupervised Transfer Learning**: Similar to inductive transfer learning, the target task is different but related to the source task. This setting focuses on unsupervised tasks in the target domain like clustering, dimensionality reduction, and density estimation, with no labelled data in either source or target domains during training.

2.3.3 Transfer Learning in NLP

Along with the increasing prevalence of transformer and large language models in NLP comes the need for increasing availability of training data as well as computing power to train such large models. Since both cannot always be guaranteed, based on the task, the domain, and the available resources, the importance of reusing and building upon existing models becomes more and more important. For example, pretraining, as seen in the BERT model (Devlin et al., 2019), has shown success in transfer learning applications. The pre-trained model demonstrates better predictive accuracy with increasing amounts of source domain training data (Yang et al., 2020).

When following the categorisation schema based on the relationship between source and target domain tasks, as was proposed by Pan & Yang (2010) and described in the previous subchapter, current NLP transfer learning approaches can be summarised as follows, see Figure 9.

Transductive transfer learning deals with applying a pre-trained model on the same task but in a different domain, with no or only few labelled samples in the target task (Yang et al., 2020). In the case of NLP, this can be divided into using domain adaptation, if the model is adapted to a new domain, and cross-lingual learning if the pre-trained model is applied to a new language (Alyafeai et al., 2020).

Inductive transfer learning, on the other hand, deals with using a model pre-trained for one task in another different task, regardless of the source and target domains (Yang et al., 2020). Here, it can be differentiated whether the model learns the target tasks

sequentially (sequential transfer learning) or simultaneously (multi-task learning). Examples of the implementation of sequential transfer learning in NLP include:

- fine-tuning, where the initially pre-trained weights are changed using a new learning function
- adapter modules, where a new model that is smaller than the original model is trained using the output of the originally pre-trained model with unchanged model weights
- and finally, feature-based approaches, where only the embedding representations of the original model are being used for downstream tasks, but the pre-trained model itself is left unchanged

Multi-task learning is similar in nature, but instead of adapting for one task one after the other, the pre-trained model is adapted to the target tasks simultaneously, reducing the number of the resulting models down to one.

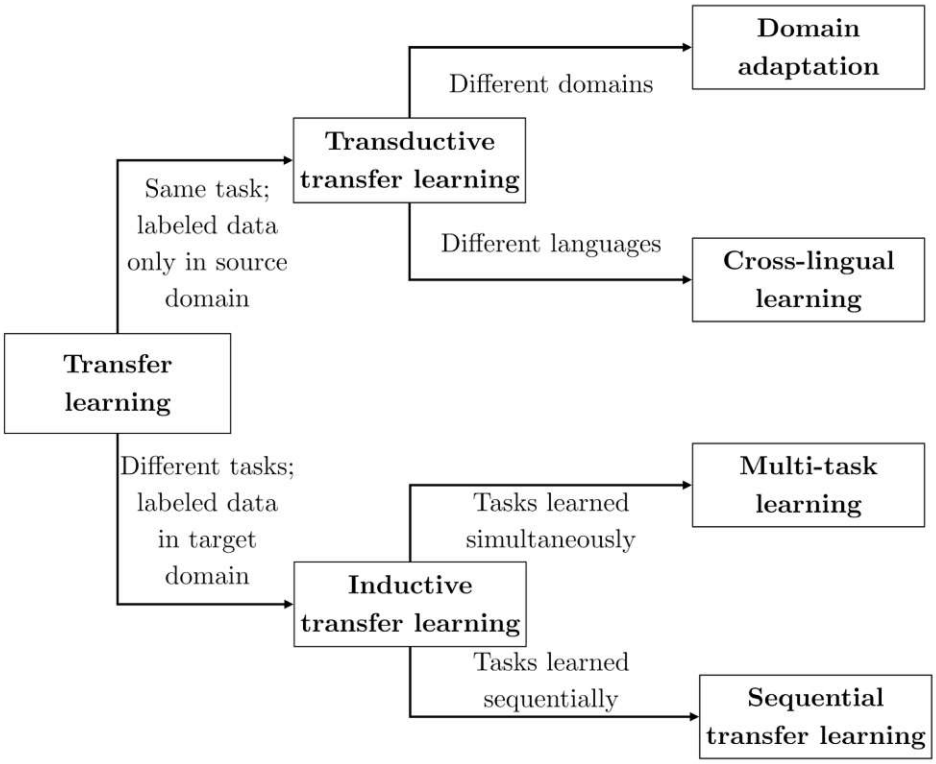

Figure 9: A taxonomy of transfer learning for NLP (Ruder, 2019, p. 46)

To standardise the process of transfer learning in NLP, as was already common practice in the realm of computer vision with models such as ResNet (He et al., 2015), a new methodology for inductive transfer learning was proposed called Universal Language Model Fine-tuning (ULMFiT) (Howard & Ruder, 2018), a framework for adapting pre-trained LSTM models. ULMFit involves three steps: pre-training on abundant text data using language modelling, model fine-tuning to the target corpus and target task data, and finally, fine-tuning of a classification layer. In the paper, it was shown that the proposed methodology can achieve state-of-the-art performance on

widely used text classification tasks. It was suggested that it could also be particularly useful for target tasks in languages where there is a lack of training data for supervised pre-training tasks, new tasks where there doesn't already exist a state-of-the-art architecture, and tasks with few labelled data and some unlabelled data for the model fine-tuning step (Howard & Ruder, 2018).

3 State-of-the-Art Analysis

This chapter presents a systematic approach to the state-of-the-art analysis for skill extraction and classification methodologies. It lays the foundation for answering RQ1: *To what extent do current state-of-the-art skill extraction and classification methods perform on German job listing data sets in terms of the performance measures?*, as well as addressing O2: *compare the applicability of existing state-of-the-art methods on German job listings*. First, an overview is given regarding the applied search strategy and the article selection process. Here, the focus on the state-of-the-art analysis is explained, and the applied methodology is detailed. The chapter then outlines the criteria for inclusion and exclusion of studies to ensure that only recent and relevant research is analysed and gives an overview of the selected studies. Second, the selected scientific publications are further categorised and critically analysed based on their research focus and prediction task, the applied methodology, and the implemented evaluation methods. This categorisation is then used to highlight several key aspects and trends in the field, including common prediction tasks, job sector specificity, language adaptability, skill types, methodologies, datasets, and evaluation metrics.

3.1 Search Strategy and Article Selection

For the state-of-the-art analysis, a structured approach following the methodology described by Zonta et al. (2020) was applied, combining predefined scientific database queries and selection criteria with a backward referencing search to identify relevant related works. Figure 10 outlines the selection process for relevant publications in a flowchart. The primary research question to be addressed through the state-of-the-art analysis is RQ1: *To what extent do current state-of-the-art skill extraction and classification methods perform on German job listing data sets in terms of the performance measures?*

To find appropriate scientific literature, both in German and in English, Web of Science, Scopus, and IEEE scientific libraries were chosen as the target databases. Web of Science and Scopus were chosen due to their comprehensive indexing of peer-reviewed journals, including Springer Nature Journals, ensuring access to high-quality scholarly outputs. The IEEE library was selected for its specialised coverage in technology and engineering, which is key for studies on automated skill extraction and classification. On the other hand, Google Scholar was excluded since it indexes a broader range of materials and, while thus offering a broader range of sources does not exclusively index peer-reviewed articles and includes a wide variety of materials, which could complicate the filtering process and dilute the specificity required for this research. The included databases thus provide a more refined and scholarly rigorous resource base, crucial for deriving reliable and precise insights for the study.

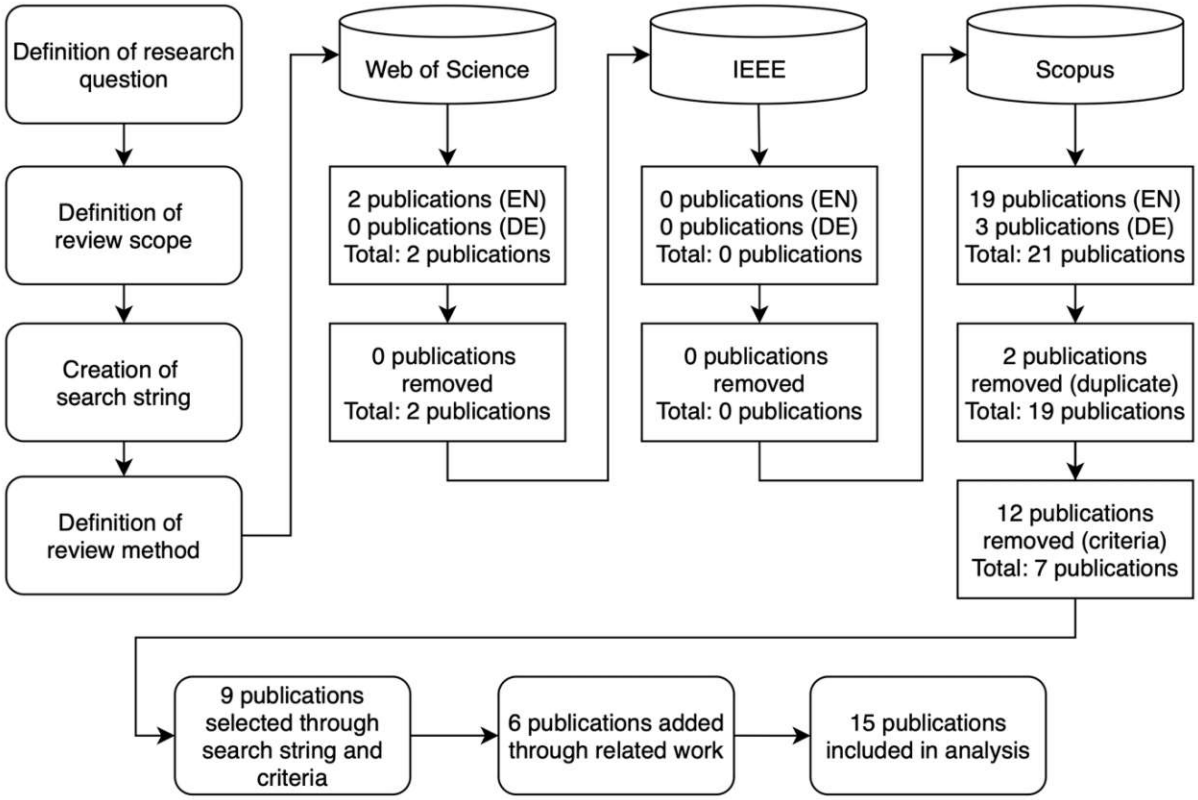

Figure 10: Overview of paper selection for state-of-the-art analysis

As the next step, the search string was defined for the databases selected, Web of Science, Scopus, and IEEE scientific libraries. The search string was based on a general template, which is shown in Figure 11. This template was then specifically adapted to comply with the unique syntax requirements of each scientific portal, as well as translated to German to also identify studies written in German, as detailed in the Appendix.

„Job" AND ("ad" OR "posting" OR "description" OR "listing") AND ("skill" OR "competency") AND ("extraction" OR "identification" OR "mining" OR "classification") AND ("weak supervision" OR "unsupervised" OR "distant supervision")*

Figure 11: English search string used for state-of-the-art analysis

For the inclusion criteria, see Table 1, it was specified that studies must be written in English or German to ensure that the findings could be accessible to the majority of the international research community and to accommodate the research team's language proficiencies. Publications were required to be from January 1st, 2018, onwards to focus on the most recent developments in the field, reflecting cutting-edge research and current methodologies. It was crucial that the full texts of the studies were accessible online to facilitate thorough analysis and replication of results. Additionally, requiring peer review and publication in recognised journals or conference materials guaranteed the credibility and academic rigour of the sources.

Conversely, the exclusion criteria were carefully chosen to maintain the focus and quality of the research corpus. Excluding studies not in English or German ensured consistency in language for analysis purposes. Publications dated before January 1st, 2018, were omitted to narrow the scope to recent developments. Sources without accessible full texts online were ruled out because these would hinder a comprehensive analysis and verification of the findings. Books and grey literature, such as conference summaries and editorials, were excluded due to their often less rigorous review processes compared to peer-reviewed articles, which could affect the reliability of the data. Finally, studies focusing on skill extraction from resumes or not employing automated methods were excluded because the research specifically targeted automated skill extraction and classification from job listings, aiming to understand and improve upon these specific techniques.

Table 1: Selection criteria for state-of-the-art analysis

Criteriu	Description
Criterium 1	Studies must be written in English or German
Criterium 2	Studies must be published from January 1st, 2018, or later
Criterium 3	Full text of studies must be accessible online
Criterium 4	Studies must be peer-reviewed and published in journals, conferences, workshops or poster sessions of conferences (exclusion of books or grey literature, as well as studies that present summaries of conferences/editorials)
Criterium 5	Studies must be related to automated skill extraction or classification from job listings (exclusion of studies that focus on skill extraction from resumes or do not employ automated skill extraction and/or classification in their analysis)

These criteria were essential to streamline the research process, ensuring that the studies collected were not only relevant and of high quality but also representative of the latest advancements in automated skill analysis in the context of job listings. This methodical approach allowed the volume of literature to be managed effectively while focusing on significant and applicable insights for the study objectives.

3.2 Discussion of Findings

Overall, the initial search across the three databases, Web of Science, IEEE, and Scopus, yielded 2 publications from Web of Science (both in English), none from IEEE, and 21 from Scopus (19 in English and 2 in German). From these initial results, no publications are removed from the Web of Science findings, while two duplicate publications are removed from the Scopus findings that were already found in the Web of Science database, leaving 19 publications found in the Scopus database. Scopus also sees a further reduction where 12 publications are removed based on the defined inclusion criteria, resulting in seven publications remaining from the Scopus scientific database. This leads to nine publications being included through the search across the databases with the predefined search string. Additionally, 6 publications were added

to the state-of-the-art analysis through the analysis of related work, leading to 15 publications. Table 2 gives an overview of the publications included in the analysis. Additionally, Figure 12 provides a general overview of the publication types included in the analysis, as well as the general yearly publication trend. Overall, one-quarter of the publications included were published in journals, whereas the other 75% were published in conferences. The yearly trend shows a strong increase in publication activities over the years, with a peak in 2022. Since the state-of-the-art analysis was conducted at the beginning of 2023, the number of included publications in that year cannot yet be seen as complete.

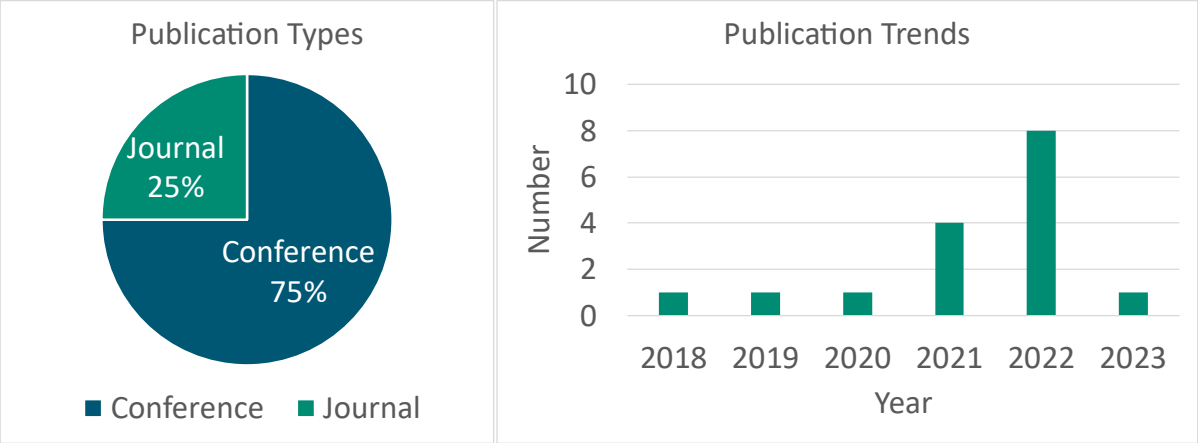

Figure 12: Overview of publication type (left) and yearly trends in publication number (right)

To analyse the current state-of-the-art in skill identification from online job listings, the final set of 15 publications included in the analysis are further categorised to illuminate current research trends and potential future directions. The categories used are based on the methodology used by Khauoja, Kassou, et al. (2021) and adapted to the research questions of this thesis.

First, the application focus of the research article and the prediction task will be examined. This includes the industrial or professional sectors in which the skill identification methods are applied, the languages in which the job listings and skill identification processes are conducted, and the specific types of skills that are being identified, such as hard/technical, soft, or transversal skills. Subsequently, the methodological approaches will be categorised, such as the skill databases and taxonomies used as references for skill identification and the computational models and algorithms employed and tested. Finally, the evaluation methods used in the papers will be examined, including performance metrics used to evaluate the effectiveness of the methodology, the datasets utilised for training and evaluation, as well as the availability of these datasets for public use and replication of the results. A comprehensive overview of the categorisations can be seen in Table 3 and Table 4 on Page 39f.

Table 2: Overview of publications included in the state-of-the-art analysis

Author	Year	Type	Publisher	Journal/Conference	Name
Ao et al.	2023	Journal	Elsevier	Information Processing and Management	Skill requirements in job advertisements: A comparison of skill-categorization methods based on wage regressions
Zhang, Jensen, van der Goot et al.	2022	Conference	ACM	Recommender Systems (RecSys)	Skill Extraction from Job Postings using Weak Supervision
Zhang, Jensen, Sonniks, et al.	2022	Conference	NAACL	Human Language Technologies	SkillSpan: Hard and Soft Skill Extraction from English Job Postings
Zhang, Jensen, & Plank	2022	Conference	ELRA	Language Resources and Evaluation Conference (LREC)	KOMPETENCER: Fine-grained Skill Classification in Danish Job Postings via Distant Supervision and Transfer Learning
Vermeer et al.	2022	Conference	ACM	Computational Jobs Marketplace (compjobs)	Using RobBERT and eXtreme Multi-Label Classification to Extract Implicit and Explicit Skills from Dutch Job Descriptions
Konstantinidis et al.	2022	Conference	ACM	EETN Conference on Artificial Intelligence (SETN)	Knowledge-driven Unsupervised Skills Extraction for Graph-based Talent Matching
Gnehm, Bühlmann, & Clematide	2022	Conference	ELRA	Language Resources and Evaluation (LREC)	Evaluation of Transfer Learning and Domain Adaptation for Analyzing German-Speaking Job Advertisements
Gnehm, Bühlmann, Buchs, et al.	2022	Conference	ACL	Natural Language Processing and Computational Social Science (NLP+CSS)	Fine-Grained Extraction and Classification of Skill Requirements in German-Speaking Job Ads
Decorte et al.	2022	Conference	ACM	Recommender Systems (RecSys)	Design of Negative Sampling Strategies for Distantly Supervised Skill Extraction
Khaouja et al.	2021	Conference	IEEE	Information Reuse and Integration for Data Science (IRI)	Unsupervised Skill Identification from Job Ads
Fareri et al.	2021	Journal	Elsevier	Expert Systems with Applications	SkillNER: Mining and mapping soft skills from any text
Decorte et al.	2021	Conference	ECML PKDD	Fair, Effective and Sustainable Talent management using data science (FEAST)	JobBERT: Understanding Job Titles through Skills
Gnehm & Clematide	2020	Conference	ACL	Natural Language Processing and Computational Social Science (NLP+CSS)	Text Zoning and Classification for Job Advertisements in German, French and English
Grüger & Schneider	2019	Conference	SCITEPRESS	Web Information Systems and Technologies (WEBIST)	Automated Analysis of Job Requirements for Computer Scientists in Online Job Advertisements
Lovaglio et al.	2018	Journal	Wiley	Statistical Analysis and Data Mining: The ASA Data Science Journal	Skills in demand for ICT and statistical occupations: evidence from web based job vacancies

3.2.1 Focus and Prediction Task

A. Prediction Task

NLP tasks such as skill span extraction, skill classification, and multi-label classification approaches are central to skill identification methods. Categorising research by task helps in understanding each method's specific capabilities and focus. Skill span extraction involves identifying and extracting relevant skill phrases from text, while skill classification assigns these phrases to predefined categories. Multi-label classification approaches instead assign multiple entailed or relevant skill to a single text unit. This unit can be a phrase, a sentence, a paragraph or even a full document. This categorisation ensures that a complete overview of skill identification processes is given.

1) Skill Span Prediction

Many publications included in the analysis focus on skill span prediction to extract skill phrases from job listings. This is mostly done by framing the problem as a sequence labelling task, following a labelling schema such as BIO (Beginning, Inside, Outside), as is done by Zhang, Jensen, Sonniks, et al. (2022) using transformer models to identify the start and end points of skill mentions within job listings. Zhang, Jensen, van der Goot, et al. (2022) focus on skill span prediction from job listings using weak supervision, leveraging the ESCO taxonomy to identify similar skills in job listings through latent representations, showing superior performance over token-level and syntactic pattern baselines. SkillNER (Fareri et al., 2021) is a data-driven method for extracting soft skills from text using NER, trained on a scientific corpus, and validated by psychologists. It enables the detection of communities of job profiles and soft skills, proving useful for firms and institutions. On the other hand, an automated system for identifying skills is presented in German-language job listings by Grüger & Schneider (2019). The skill span prediction utilises POS templates and a combination of the fasttext model and Levenshtein distance for assigning skills to classes.

2) Skill Span Classification

In contrast to skill span prediction, skill span classification deals with assigning the correct skill label to an already predefined skill span. This approach is applied by Zhang, Jesen, & Plank (2022) for English and Danish skill spans, utilising multilingual language models and the ESCO taxonomy for fine-grained skill classification. On the other hand, a combined approach of skill span prediction and classification is explored by Gnehm, Bühlmann, Buchs, et al. (2022). Here, the study adapts pre-trained transformer-based models and incorporates context from job listings and the ESCO taxonomy to improve unsupervised multi-label classification. The skill span extraction was done on a coarse level, only differentiating between Education, Experiences and

Language Skills, while the fine-grained classification was then done as a separate test case, utilising a set of 40 predefined job listing skills terms.

3) Multi-Label Classification

Instead of focussing on extracting skills on the span level, multi-label classification assigns a list of potentially relevant skills to a text segment. These skills then become available for other downstream tasks, such as analysing similarities between job listings or creating requirement profiles for applicants. Differences exist in the length of the text passage that is being examined. For example, eXtreme Multi-Label Classification (XML) is used to predict not only explicit but also implicit skills on the job listing level, achieving high recall and MRR values on Dutch job descriptions (Vermeer et al., 2022). Khaouja, Mezzour, et al. (2021) present a new methodology for unsupervised skill identification from job listings, focusing on identifying skills expressed in sentences and as technical words using Wikipedia, focusing on explicitly mentioned skills. The methodology involves splitting job descriptions into sentences, embedding them along with the skills in the skill base, and using sentence similarity to identify required skills. While the classification is carried out on the sentence level, the evaluation of the performance is then carried out on the job listing level. Decorte et al. (2022) on the other hand, explores different negative sampling strategies to improve the performance of multi-label skill classification on the sentence level.

4) Other

Other prediction tasks covered in the analysed papers include the prediction of wage variation across different job profiles (Ao et al., 2023), Document Classification, Text Zoning and ICT term recognition (Gnehm, Bühlmann, & Clematide, 2022; Gnehm & Clematide, 2020; Lovaglio et al., 2018), resume-to-job matching (Konstantinidis et al., 2022) and job title prediction (Decorte et al., 2021).

B. Examined Job Sector

Different job sectors such as IT, healthcare, manufacturing, and education have unique skill requirements and terminology. By categorising research based on the job sector, the applied skill identification methods can be seen in the context of sector-specific needs, which in turn also helps in comparing how different publications approach skill identification with the specific challenges of the examined job-sector, such as the dynamic nature of technology skills in IT versus the more stable skill sets in traditional applications.

1) Skill Identification from Specific Sectors

The studies included in the state-of-the-art analysis that focus on a specific sector exclusively target the computer science and statistics sector to understand the soft and hard skill requirements due to the high demand for IT-related job positions and the

aforementioned dynamic nature of technology skills (Grüger & Schneider, 2019; Lovaglio et al., 2018). These two studies being among the oldest studies included, also shows that while initially, skill identification focused on single sectors or specific occupations to manage the volume of job listings, the rise of complex data collection technologies has amplified the possibility for large-scale and real-time skill identification across the job market.

2) Skill Identification from Multiple Sectors

More recent studies included in the analysis aim to identify skills from multiple sectors for comprehensive skill mapping, automatically handling large job listing collections. This leads to almost all publications included in the review not focussing on any specific job sector in their analysis. Instead, most purposefully construct a diverse data set, that includes listings from many different job sectors for skill extraction (Decorte et al., 2022; Gnehm, Bühlmann, Buchs, et al., 2022; Zhang, Jensen, & Plank, 2022) but also for other tasks such as wage variation prediction (Ao et al., 2023) among others. For instance, Ao et al. (2023) analysed job listings from sectors encompassing a wide range of industries such as healthcare, finance, and technology. This comprehensive approach ensures the findings are relevant across different fields.

C. Languages of Data Sets

The language in which job listings and resumes are written affects the performance of different NLP techniques. Categorising research by the language of the dataset allows us to evaluate the adaptability of methods to different languages and scripts. This is crucial because some NLP models are primarily trained on English and might not perform as well on other languages without significant adaptation. It also helps in identifying the need for multilingual models or specific preprocessing steps for languages with different grammatical structures.

English (EN) is the predominantly used language in the examined studies (Ao et al., 2023; Decorte et al., 2022; Zhang, Jensen, van der Goot, et al., 2022). English is also a common choice due to the availability of datasets and most commonly used pre-trained models being trained on mostly English data.

Other languages include Dutch (NL), where RobBERT, a BERT variant pre-trained on Dutch text, was used to specifically apply a model for effective skill extraction in languages other than English (Vermeer et al., 2022). Other data sets used in the studies include German (DE) (Gnehm, Bühlmann, Buchs, et al., 2022; Gnehm, Bühlmann, & Clematide, 2022; Grüger & Schneider, 2019) and Italian (IT) (Lovaglio et al., 2018).

Studies specifically highlighting and analysing the transferability of their approaches between different languages are (Gnehm & Clematide, 2020; Konstantinidis et al.,

2022; Zhang, Jensen, & Plank, 2022). Gnehm & Clematide (2020) experiment with multilingual modelling and machine translation-based approaches to handle sparse data problems in German, French (FR) and English job listings. Zhang, Jensen, & Plank (2022) analyse the applicability of differently fine-tuned language models on Danish job listings. The study highlights the benefits of domain adaptive pretraining, which improves performance for both English and Danish fine-tuning. However, it also notes the trade-off between the short-term gains from pretraining on unlabelled data and the long-term gains from annotating additional data.

D. Skill Types

Skills can be broadly categorised into “hard” technical and sector-specific skills, and “soft” transversal skills. Categorising by skill type helps in understanding the focus of the research and its applicability. Understanding the type of skills targeted by the research allows for a better assessment of the method’s scope and relevance.

Most of the analysed publications collectively emphasise both hard and soft skills, when it comes to the task of skill extraction, be it through span prediction (Gnehm, Bühlmann, Buchs, et al., 2022; Zhang, Jensen, Sonniks, et al., 2022; Zhang, Jensen, van der Goot, et al., 2022), span classification (Zhang, Jensen, & Plank, 2022) or multilabel classification (Decorte et al., 2022; Khaouja, Mezzour, et al., 2021; Vermeer et al., 2022), most reflecting an integrated approach to skill extraction from job listings. Fareri et al. (2021) on the other hand specifically target the extraction of soft skills using their proposed method SkillNER, indicating a primary emphasis on interpersonal and cognitive skills as opposed to technical skills. Grüger & Schneider (2019) instead focus primarily on hard skills relevant to computer science and related domains to collect job requirements for computational fields.

3.2.2 Applied Methodology

E. Skill Base

The foundational skills taxonomy or ontology (such as ESCO or O*NET) used in the research standardises the skill identification process and ensures comparability across different methods. A consistent skill base facilitates interoperability between different systems and databases, allowing for more comprehensive analysis and integration of findings. It also helps in evaluating the completeness and robustness of the skill taxonomy used and whether it adequately covers the required skill sets for the job sector being examined.

Most of the studies that rely on a standardised taxonomy use the ESCO skills and occupations taxonomy. For example, on study utilises the ESCO taxonomy as a knowledge base to match skills from profile experience descriptions using a Siamese BERT-Network for sentence embeddings (Konstantinidis et al., 2022). Another

employs ESCO for weak supervision in skill extraction from job listings, leveraging the taxonomy to label spans that relate to ESCO skills in embedding space (Zhang, Jensen, van der Goot, et al., 2022). Instead of relying solely on predefined skill taxonomies Khaouja, Mezzour, et al. (2021) extend the ESCO taxonomy by cross-referencing the analysed segments with Wikipedia to identify software names, frameworks, certifications, and licenses. This helps in recognising technical skills that may not be present in the predefined skill base.

Other standardised taxonomies include JDCO (Jobdigger Classification of Occupations) (Vermeer et al., 2022), O*NET (Fareri et al., 2021), or the DISCO (European Dictionary of Skills and Competences) framework (Ao et al., 2023). JDCO is a proprietary job classification developed by the Dutch company Jobdigger and is used to help identify both explicit and implicit skills that are relevant to the job listing, even if they are not directly mentioned in the text, since each JDCO code is associated with a list of the most relevant skills for that particular occupation class (Vermeer et al., 2022). DISCO is used to classify job listings into 25 domains of DISCO using word embeddings and similarity measurements, then calculate the skill intensity for each job's domain (Ao et al., 2023). Finally, O*NET is utilized to perform a cross-validation process, which involves identifying the most frequently mentioned soft skills in the literature that are also present in an occupational framework, ensuring that the skills are concise and easily traceable in the text (Fareri et al., 2021).

F. Models and Category

Different computational models, including transformer-based models (e.g. BERT-based models), other machine learning approaches such as support vector machines (SVMs), or unsupervised topic modelling methods such as Latent Dirichlet Allocation (LDA), exhibit varying strengths and weaknesses. Categorising research based on these models helps understand technological approaches, compare performance, identify effective models for specific tasks or sectors, and comprehend computational and data requirements.

1) Topic Modelling

Word count and topic modelling fall in the category of unsupervised learning. The approach presented by Ao et al. (2023) uses these models to analyse the frequency and distribution of words and phrases within job listings to identify prevalent skills and themes. For this, models like LDA (Latent Dirichlet Allocation), PLSA (Probabilistic Latent Semantic Analysis) and BERTopic are compared for uncovering underlying topics in large text corpora of job listings, providing insights into the general landscape of skill demands without requiring extensive labelled data. For the task of explaining wage variation between jobs, LDA outperformed other methods.

2) Transformer Architecture

Models based on the transformer architecture are utilised in the vast majority of publications, 10 out of 15 publications in total. The employed transformer models range from base BERT implementation, over multilingual models such as RemBERT (Zhang, Jensen, & Plank, 2022), domain-adapted models such as JobBERT (Decorte et al., 2021), another transformer model fine-tuned on job descriptions, and JobSpanBERT (Zhang, Jensen, Sonniks, et al., 2022), to language-specific and domain-adapted models such as the German JobGBERT (Gnehm, Bühlmann, & Clematide, 2022) and the Danish DaJobBERT (Zhang, Jensen, & Plank, 2022). JobBERT was compared with LASER, fastText, and Sentence-BERT models for job title normalisation, with JobBERT showing showing considerable improvements in test scores, outperforming Sentence-BERT and the other models (Decorte et al., 2021). Similarly, the DaJobBERT was able to improve upon the test score of other Danish language models, that were not domain-adapted. Notably, the much larger and multilingual RemBERT, which was fine-tuned on both English and Danish data, showed substantial improvement even over the domain-adapted DaJobBERT in both zero-shot and few-shot settings (Zhang, Jensen, & Plank, 2022). Conversely, Vermeer et al. (2022) showed that that a monolingual BERT model like RobBERT outperforms multilingual models, showcasing the effectiveness of RobBERT-XMLC approach in skill extraction through extreme-multi-label classification from Dutch job descriptions.

3) Embedding Models and Other ML Methods

Besides Transformer models, pre-trained embedding models, such as sent2vec or fasttext are also commonly used. The sent2vec model outperformed the transformer-based SBERT model on the sentence-level multi-label classification task (Khaouja, Mezzour, et al., 2021). Different embedding models, such as fasttext and word2vec, were also combined with a set of distance measures for semantic matching. A combination of the fasttext model and Levenshtein distance showed the best performance when assigning skills to semantically similar skill classes (Grüger & Schneider, 2019). For other machine learning methods, no pre-trained embedding models were used. Instead, a bag of words approach was combined with a range of classification algorithms, such as a support vector classifier, logistic regression and a random forest classifier (Lovaglio et al., 2018).

3.2.3 Evaluation Methods

G. Data Set and Availability

The characteristics of the dataset, such as its size, diversity, and domain, significantly impact the results of skill identification methods. Analysing the research by the dataset used helps in understanding the empirical basis of the research and its generalizability. Large, diverse datasets provide robust training and evaluation opportunities, while

smaller or domain-specific datasets might limit the applicability of the findings. Additionally, the public availability of the used datasets is crucial for replication and ensuring that findings can be validated and built upon by other researchers. Open datasets and models foster transparency and collaboration, enabling continuous improvement and innovation.

1) Job Listings

The majority of the analysed papers use some form of job listings as their data set, which originate from various sources with differing degrees of availability. The KOMPETENCER dataset includes a set of 60 Danish job listings along with 391 English job listings with annotated spans and is available online[2] (Zhang, Jensen, & Plank, 2022). The SkillSpan dataset contains English job listings collected from June 2020 to September 2021 from three sources: BIG (a large job platform), HOUSE (an in-house dataset), and TECH (the StackOverflow job platform) (Zhang, Jensen, Sonniks, et al., 2022). The dataset consists of 14.5K sentences and over 12.5K annotated spans and is also available online[3]. A variation of SkillSpan and the Sayfullina dataset (Sayfullina et al., 2018) were modified[4] and used for weak supervision in conjunction with the ESCO taxonomy (Zhang, Jensen, van der Goot, et al., 2022). SkillSpan was used as the basis for the data set by Decorte et al. (2022), where it the existing skill spans were manually annotated with their corresponding ESCO skills. The final data set contains 1459 annotated spans with ESCO labels and is available online[5]. Another publicly available data set[6] contains a set of 30,926 English vacancy titles labelled with corresponding ESCO occupations (Decorte et al., 2021). Available on request is a data set of over one million job listings from the UK job market along with the corresponding wage data (Ao et al., 2023). Finally, other studies using job listings include (Gnehm, Bühlmann, Buchs, et al., 2022; Grüger & Schneider, 2019; Vermeer et al., 2022). Here, no dataset availability is mentioned (Grüger & Schneider, 2019; Vermeer et al., 2022), or the data set is only available on request and unlabelled (Gnehm, Bühlmann, Buchs, et al., 2022) or only partially labelled (Gnehm, Bühlmann, & Clematide, 2022).

2) Other

Other datasets include a resume dataset used by Konstantinidis et al. (2022), which in its unlabelled form is available on Kaggle[7] and includes 2,400 resumes extracted from PDF files sourced from an online career portal. Finally a soft skill dataset based on over 5,000 scientific papers and abstracts and annotated by domain experts and

[2] https://github.com/jjzha/kompetencer
[3] https://github.com/kris927b/SkillSpan
[4] https://github.com/jjzha/skill-extraction-weak-supervision
[5] https://github.com/jensjorisdecorte/Skill-Extraction-benchmark
[6] https://github.com/jensjorisdecorte/JobBERT-evaluation-dataset
[7] https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset

psychologists was created to train and evaluate SkillNER (Fareri et al., 2021). While the SkillNER tool itself is available online, no mention is made of availability of the data set itself.

H. Evaluation Metrics

Evaluation metrics are essential for assessing and comparing the performance of different skill identification models across studies when used on the same data set.

For skill span prediction, the most used metrics include precision, recall and F1 score (Fareri et al., 2021; Gnehm, Bühlmann, Buchs, et al., 2022), with less emphasis on accuracy due to the large presence of true negatives in the NER tasks. Instead, accuracy is mostly used in studies focusing on single-label classification (Gnehm & Clematide, 2020; Grüger & Schneider, 2019; Lovaglio et al., 2018). The F1 score is also applied in a strict and a loose setting (Zhang, Jensen, van der Goot, et al., 2022). The strict F1 score requires exact matches for the skill spans, whereas the loose F1 score allows for partial matches. Other variants include the weighted macro F1 score, which adjusts the F1 score to account for class imbalance by giving different weights to different classes based on their frequency (Zhang, Jensen, & Plank, 2022).

When it comes to regression tasks, the adjusted R^2 is used to measure the model's ability to explain the variability in wage prediction across job listings (Ao et al., 2023).

Lastly, metrics such as MRR (Mean Reciprocal Rank), Recall@k, and nDCG (normalised Discounted Cumulative Gain), are used to evaluate different ranking models and are being used in studies focussing on multi-label-classification (Decorte et al., 2022; Vermeer et al., 2022).

3.3 Summary of Findings

The state-of-the-art analysis of current skill extraction and classification methods highlights several key aspects and trends in the field, including common prediction tasks, job sector specificity, language adaptability, skill types, methodologies, datasets, and evaluation metrics, see and .

The most common tasks included in the analysis include skill span prediction, skill span classification, and multi-label classification. Skill span prediction predominantly employs sequence labelling techniques, such as the BIO schema, and uses transformer models like BERT to identify skill phrases within job listings. Notable methods include weak and distant supervision. Skill span classification assigns predefined labels to identified skill spans, often leveraging multilingual language models and domain-specific taxonomies like ESCO. Multi-label classification, on the other hand, assigns multiple relevant skills to text units ranging from phrases to full documents, facilitating broader applications like job matching and profiling.

Table 3: Categorization overview of state-of-the-art publications – Part 1

Author	Focus and Prediction Task				Applied Methodology			Evaluation Methods		
	Prediction Task	Sectors	Lang-uages	Skill Types	Skill Base	Models	Category	Metrics	Data Set	Availability
Ao et al.	Other (Prediction of Wage Variation)	All sectors	EN	Hard + Soft Skills	DISCO	LDA, PLSA, BERTopic	Word Count, Topic Modelling	R2adj	Job Listings	On Request
Zhang, Jensen, van der Goot et al.	Skill Span Prediction	All sectors	EN	Hard + Soft Skills	ESCO	RoBERTa, JobBERT	Transformer Architecture	F1 (Strict and loose)	Job Listings	Yes
Zhang, Jensen, Sonniks, et al.	Skill Span Prediction	All sectors	EN	Hard + Soft Skills	ESCO	(Span-, Job-, JobSpan-) BERT, Longformer	Transformer Architecture	Span-F1	Job Listings	Yes
Zhang, Jensen, & Plank	Skill Span Classification	All sectors	DA	Hard + Soft Skills	ESCO	(Da-)BERT, (Da-) JobBERT,	Transformer Architecture	weighted macro F1	Job Listings	Yes
Vermeer et al.	Multi-Label Classification (Job Listing Level)	All sectors	NL	Hard + Soft Skills	JDCO	(Rob-)BERT-XMLC	Transformer Architecture	MRR, Recall-@k, nDCG	Job Listings	No
Konstantinidis et al.	Other (Resume Matching)	All sectors	EN	Hard + Soft Skills	ESCO	all-mpnet-base-v2	Embedding-Model	MAP, ADCG	Re-sumes	Partially (unlabelled)
Gnehm, Bühlmann, & Clematide	Other (Doc. Classif., Text Zoning, ICT Term Recog.)	All sectors	DE	NA	NA	jobGBERT, jobBERT-de	Transformer Architecture	F1	Job Listings	Partially (partially labelled)

Table 4: Categorization overview of state-of-the-art publications – Part 2

Author	Focus and Prediction Task				Applied Methodology			Evaluation Methods		
	Prediction Task	Sectors	Lang-uages	Skill Types	Skill Base	Models	Category	Metrics	Data Set	Avail-ability
Gnehm, Bühlmann, Buchs, et al.	Skill Span Prediction and Classification	All sectors	DE	Hard + Soft Skills	ESCO	jobBERT-de	Transformer Architecture	Precision, Recall, F1	Job Listings	Partially (unla-belled)
Decorte et al.	Multi-Label Classification (Sentence Level)	All sectors	EN	Hard + Soft Skills	ESCO	RoBERTa	Transformer Architecture,	MRR, R-Precision @k	Job Listings	Yes
Khaouja et al.	Multi-Label Classification (Job Listing Level)	All sectors	EN	Hard + Soft Skills	ESCO, Wikipedia	Sent2Vec, SBERT	Embedding-Model, Transformer Arch.	Precision, Recall	Job Listings	No
Fareri et al.	Skill Span Prediction	All sectors	EN	Soft Skills	O*NET	SVM, MLP	ML-Models	Precision, Recall, F1	Scientific Abstracts	No
Decorte et al.	Other (Job Title Prediction)	All sectors	EN	NA	ESCO	JobBERT, MLP	Transformer Architecture	MRR, Recall@k	Job Listings	Yes
Gnehm & Clematide	Other (Text zoning, Text classification)	All sectors	DE, FR, EN	NA	NA	BERT, FLAIRSJM M+FT	Transformer Architecture	Accuracy	Job Listings	Partially
Grüger & Schneider	Skill Span Prediction	Computer Science	DE	Hard Skills	NA	SVC, fasttext	Embedding-Model	Accuracy	Job Listings	No
Lovaglio et al.	Other (Document Classification)	ICT	IT	Hard + Soft Skills	NA	SVM, kNN, Log.Reg., GB, RF	Embedding-Model	Accuracy, Brier Score	Job Listings	No

Regarding job sector specificity, initial studies concentrated on particular sectors like IT and statistics due to the high demand and dynamic skill requirements in these fields. Recent studies, however, adopt a broader approach, analysing skills across multiple sectors to ensure comprehensive skill mapping and relevance across diverse industries.

Language adaptability is another critical aspect, with English being the primary language due to the availability of datasets and pre-trained models. Nonetheless, other languages like Dutch, German, and Italian are also explored using tailored models and multilingual approaches. The need for multilingual models and domain-specific pretraining is emphasised to improve performance in non-English contexts.

Skill types in the research generally integrate both hard (technical) and soft (interpersonal) skills, reflecting their dual importance in job listings. Specific studies, such as SkillNER, focus primarily on soft skills, while others target hard skills in technical domains.

Many methodologies employed involve the use of standardised skill bases, such as ESCO and O*NET, to ensure consistency and comparability. Some studies extend these taxonomies with additional sources like Wikipedia. Transformer-based models dominate the landscape, with variations like JobBERT and domain-adapted models showing significant performance improvements. Embedding models and other machine-learning methods are also utilised for specific tasks.

The datasets used in this research are sourced from a variety of platforms, including specialised job boards, online exchanges, and public repositories like Kaggle. Their availability ranges from publicly accessible sources to those available only upon request or not explicitly mentioned in the research documentation. The datasets used in the analysed studies predominantly consist of job listings from various sources, with some exceptions like resumes and annotated scientific papers. Of the predominantly English datasets, some are publicly available for replication and further research, like SkillSpan and KOMPETENCER. Of the German datasets, only a few are publicly available, although none of these are available with span-level skill annotations.

Common evaluation methods include precision, recall, F1 score, and accuracy, tailored to specific tasks such as skill span prediction and classification. For multi-label classification and ranking tasks, metrics like Mean Reciprocal Rank (MRR), Recall@k, and normalised Discounted Cumulative Gain (nDCG) are vital for evaluating the effectiveness of the classification and recommendation systems.

Within the state-of-the-art analysis, no study was found that extracts skills from German job listings without task-specific pre-training, while also creating a mapping to a standardised skill taxonomy (P1) and at the same time, no publicly available benchmarking data set exists for evaluating and comparing the performance of

different skill extraction and classification approaches on German job listings (P2). In the following chapter, this research gap will be addressed by creating a valid benchmarking data set (O1), testing the applicability of the relevant state-of-the-art methods described above on German job listings (O2), evaluating whether language-specific pre-processing or pre-training improves the extraction and classification results (O3), and finally comparing the performance of different pre-trained models and pre-processing steps on the German benchmarking data set (O4). See Figure 13 for a graphical representation.

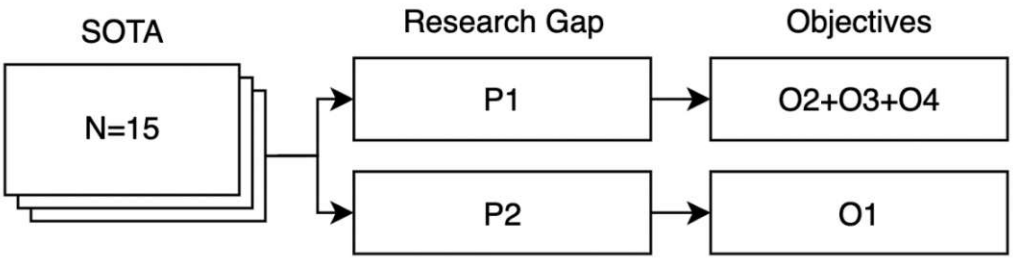

Figure 13: Overview of research gap and resulting research objectives to close the gap

4 Data Acquisition and Analysis

In this chapter, a systematic approach to data acquisition and analysis is detailed. The findings within this chapter lay the foundations for answering i) RQ2: *What is the impact of language-specific pre-processing on the performance measures, compared to the current state-of-the-art?* as well as addressing O3: *evaluate whether language-specific pre-processing or pre-training improves the extraction and classification results* and ii) realize O1: *create a valid benchmarking data set.* First, the ESCO taxonomy is thoroughly examined to uncover its underlying patterns and semantic structure. This analysis forms the foundation for understanding the taxonomy's role in the subsequent tasks. Second, the methodology for constructing the annotated dataset is presented, beginning with the data acquisition process. This section also covers the iterative development of the annotation guidelines, including the measures employed to evaluate their reliability. Finally, the chapter concludes with an analysis of the resulting annotated dataset, discussing the insights and implications derived from this analysis.

4.1 ESCO Taxonomy

The ESCO taxonomy, specifically its skills pillar, is used as the reference point within this thesis for what is to be considered a skill in a job listing and what is not. Because of this, it is necessary to get a good understanding of the underlying data and distributions. For the analysis and all further processing steps, a local CSV export of the ESCO version v1.1.1 (European Commission, 2022b) is used, which was released on September 26th, 2022, and is accessible through the ESCO download portal[8]. All further mentions of the ESCO taxonomy will also reference this version.

4.1.1 Structure of Skills Pillar

The ESCO skills pillar contains a total of 13.890 individual skill concepts. Each concept has the following, in part optional, properties, see Table 5.

The structure of the ESCO skills pillar is hierarchical and multi-layered, designed to comprehensively cover the range of skill concepts relevant across various occupations and sectors. At the top of this hierarchy are four higher-level categories that broadly classify the types of concepts. Each concept can be assigned to at least one of the following four categories:

- **Knowledge**: Theoretical or practical understanding of a subject necessary for performing tasks in a specific field or occupation.

[8] https://esco.ec.europa.eu/en/use-esco/download

- **Skills:** The ability to apply knowledge and use know-how to complete tasks and solve problems. This includes specific job-related skills and broader core competences.
- **Transversal skills and competences:** Broad skills relevant across various jobs and sectors, such as problem-solving, communication, and teamwork, which are important for personal development and employment.
- **Language skills and knowledge:** Proficiency in understanding, speaking, reading, and writing in various languages, which are important for communication in a globalised context.

Table 5: Overview of data properties of skill concepts within ESCO

Type	Property Name	Data Type
Required Property	conceptType	categorical
	conceptUri	uniform resource identifier (URI)
	skillType	categorical
	preferredLabel	string
	description	string
	reuseLevel	categorical
	status	categorical
	modifiedDate	dateTime
	InScheme	comma separated list of strings (foreign keys)
Optional Property	altLabels	comma separated list of strings
	hiddenLabels	comma separated list of strings
	scopeNote	string
	definition	string

These four higher-level categories are further divided into 27 individual skill groups, see Appendix ESCO Skill *Group Codes* for a full list of all sub-categories. Within these skill groups exist further subdivisions into subgroups, that break down the skill groups into more specific domains, allowing for a detailed categorisation of individual skills, see Figure 4 on page 10 for a schematic overview.

It is important to note that since the structure is designed to be flexible and interconnected, the high-level categories are not necessarily mutually exclusive on the lowest level. This means that an individual skill concept might belong to more than one high-level category or skill group. For example, the skill concept "analyse big data" is categorised under the skill group “working with computers” as well as the transversal skill group “thinking skills and competences”, reflecting the necessary combination of generalizable thinking skills, combined with the needed IT skills. This non-mutually exclusive categorisation allows for a more nuanced and comprehensive mapping of skills, acknowledging that certain skills can have broad applications and facilitating the identification, analysis, and combination of skills across various professional and educational contexts.

4.1.2 Skill Distribution and Composition

To get a better understanding of the underlying distributions and properties, this next chapter examines the skill pillar of the ESCO taxonomy more closely. This is an important step within the CRISP-DM process and serves as the basis for the following modelling steps.

Distribution of skill type and reusability level

First, the number of skill and knowledge concepts will be analysed, including their reusability level, which is stored in the reuselevel property. The skill reusability level indicates how well a knowledge or skill concept can be applied across occupations and sectors. Within ESCO, four levels of skill reusability exist:

- **Transversal**: These are core or soft skills relevant across many occupations and sectors, fundamental for personal development and the acquisition of more specialised skills. Examples include teamwork, communication, and basic office software usage.
- **Cross-sectoral**: These skills are applicable across multiple economic sectors. An example is "animal welfare," which is relevant in agriculture, veterinary activities, and recreation sectors.
- **Sector-specific**: These skills are relevant within a specific sector but can apply to various occupations within that sector. For instance, "monitor livestock" is relevant across different occupations within animal husbandry.
- **Occupation-specific**: These are highly specialised skills usually relevant to a single occupation or its specialisms, such as "milking operations" for a farm milk controller.

Analysing the number of skill and knowledge concepts and their reuseLevel property, a general trend within the ESCO taxonomy can be discovered, see Figure 14.

The analysis of the number of skill and knowledge concepts and their reuseLevel property within the ESCO taxonomy reveals several important trends, see Figure 14. Overall, the ESCO taxonomy contains a significantly higher number of skill concepts (10,831) compared to knowledge concepts (3,059). This discrepancy may be attributed to the fact that skills are often more task-specific and frequently updated to reflect the evolving demands of various industries. In contrast, knowledge concepts tend to be more stable and less granular, representing fundamental understanding that is less prone to frequent changes.

When considering the reusability of these concepts, a pattern emerges where sector-specific concepts dominate both skill and knowledge categories. Sector-specific concepts are likely more prevalent due to the detailed and specialised nature of tasks and knowledge required in distinct industries. Cross-sector concepts follow in frequency, with occupation-specific skills being almost as frequent as cross-sector concepts. The more pronounced gap between occupation-specific and cross-sector

knowledge concepts may indicate that knowledge, unlike skills, are more generalisable and not as tightly bound to specific occupations. The much lower frequency of transversal concepts, which are applicable across all sectors and occupations, underscores the taxonomy's focus on specific, industry-related skills and knowledge. However, this does not diminish the importance of transversal skills; rather, it highlights their universal applicability, making them valuable despite their lower numbers.

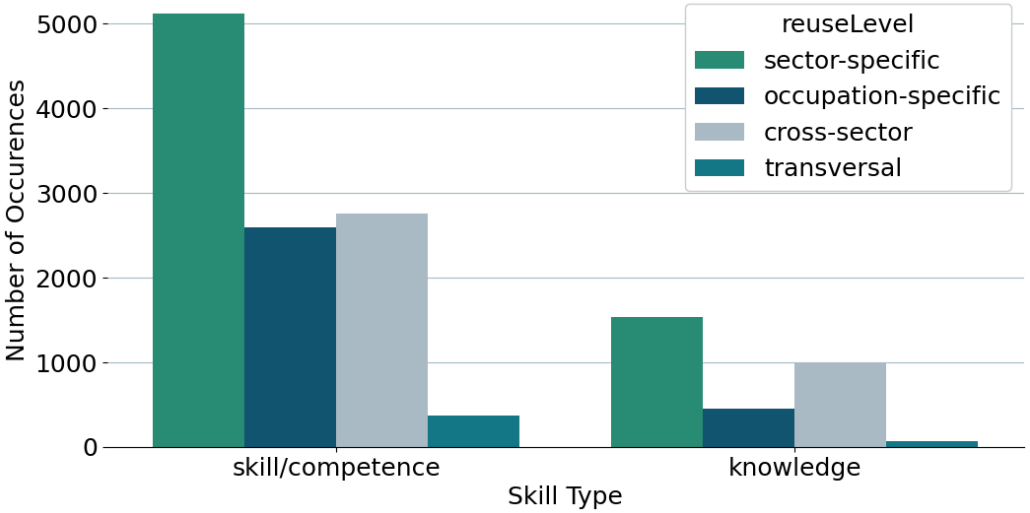

Figure 14: Analysis of skill reusability level within ESCO

One possible implication for assigning labels through semantic similarity to an ESCO concept is, that matching algorithms are statistically more likely to identify a match with a skill type than with a knowledge type, given the sheer number of skill concepts available. The higher density of sector- and occupation-specific concepts may also lead to more precise matches in those categories, although this precision may not necessarily translate into broader applicability or relevance outside of specific contexts.

Overall, this analysis shows a trend within the ESCO taxonomy to a higher density of detailed sector- and occupation-specific concepts, although this higher density does not implicate higher relevance of these more specific concepts, since more universal entities such as transversal or cross-sector concepts, lend themselves to much higher reusability within ESCO occupation. The analysis of the Occupation Pillar and the usage of the skill and knowledge concepts are not within the scope of this thesis.

Co-occurrence across skill groups

Since on the lowest level the skill concepts are not mutually within the high-level skill groups, it is important to analyse the structure of the co-occurrences and underlying trends for the later classification steps.

Taking a look at the bar plot in Figure 15 the frequency in which a skill concept simultaneously belongs to one or more skill groups, it can be seen that the majority of concepts are uniquely assigned. The maximum number of skill groups of non-uniquely

assigned concepts is four, with a steep drop-off after two skill groups, which is especially pronounced for knowledge concepts. This suggests a strong preference within ESCO for clearly defined, distinct skill concepts and likely reflects the specialised nature of many skills. The concepts that are assigned to two or more skill groups indicate that some skills possibly have broader applicability, such as transferable skills.

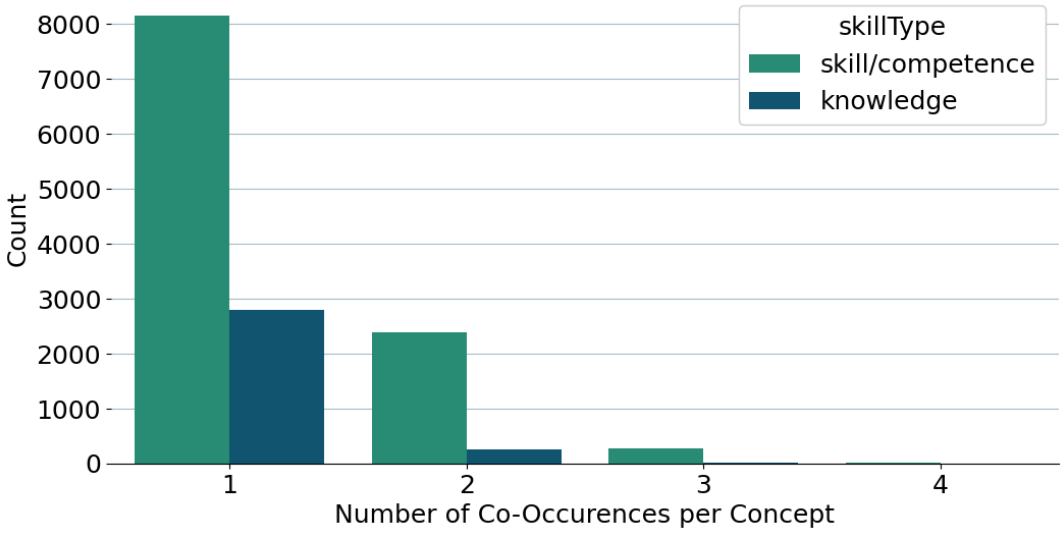

Figure 15: Bar plot showing the frequency in which a skill concept simultaneously belongs to one or more skill groups

The co-occurrence matrix in Figure 16 provides deeper insights into how different skill groups overlap. It does so by visualising only the number of overlapping skill concepts within the rows and columns of the matrix. Overall, the matrix shows that the highest overlap exists between the transversal skill group T4 (social and communication skills and competences) and skill group S1 (communication, collaboration and creativity), which makes sense given the similar domain. T4 also has a string overlap with S4 (management skills), although to a lesser extent (507 to 161). Generally, a high number of overlaps exists between the transversal skill groups and the regular skill groups. This overlap is logical, as transversal skills are inherently designed to be applicable across various fields, making them likely to co-occur with more specialised skill groups. Between regular skill groups, a high overlap between S1, S4 and S2 (information skills) exists. This overlap can be explained by the high prevalence of communication skills included in these skill groups. Finally, there is comparatively little overlap between individual knowledge groups and especially between knowledge and skill groups, although this overlap exists. This indicates a more specialised and distinct categorisation of knowledge concepts, leading to fewer co-occurrences.

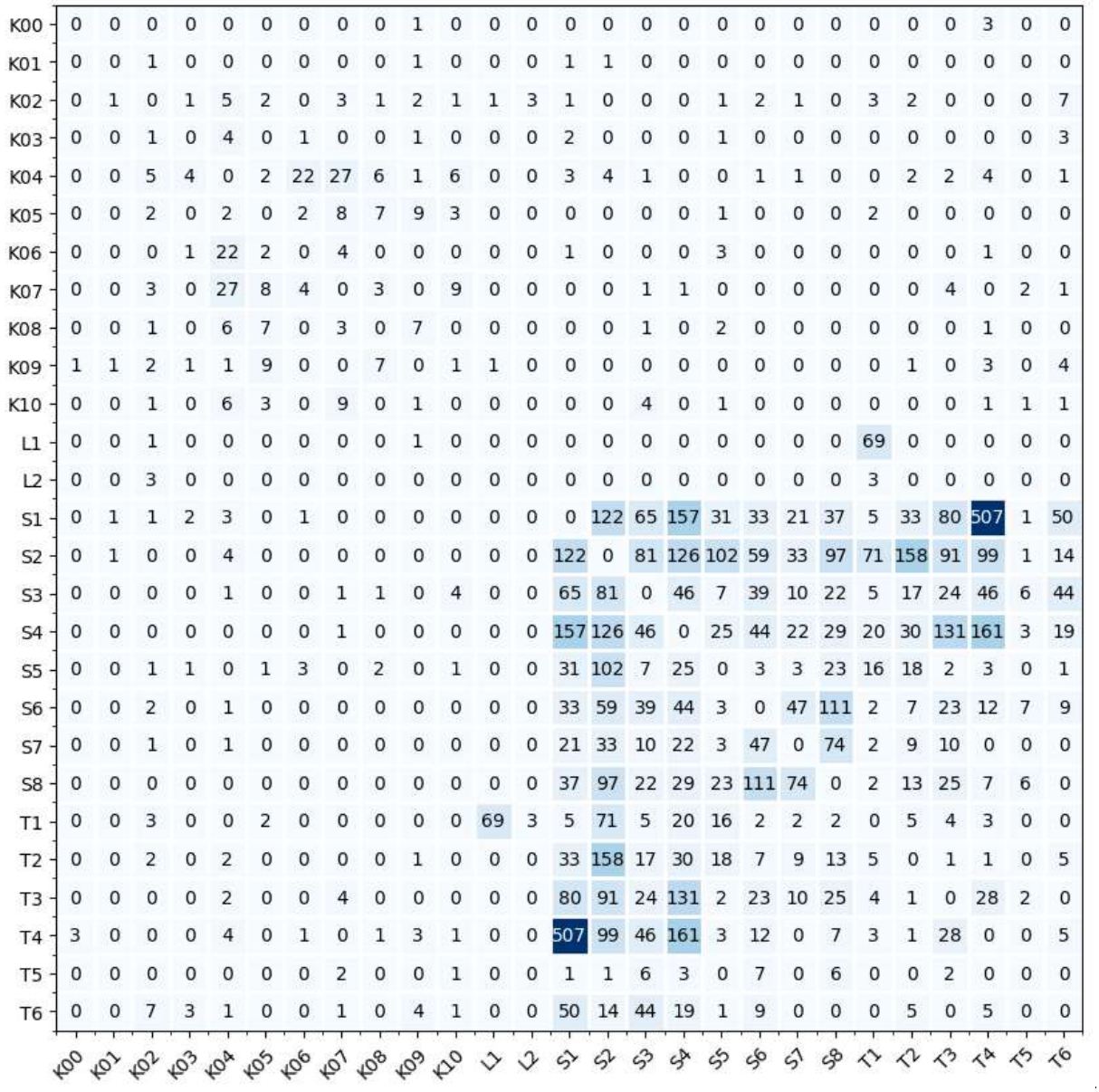

Figure 16: Co-occurrence matrix of ESCO skill concepts within skill groups

These patterns have important implications for the classification and retrieval of skills within ESCO. The predominance of uniquely assigned skills suggests that classification systems can rely on the distinctiveness of skills for precise matching in sector-specific contexts. However, the existence of overlapping concepts indicates the need for flexible classification mechanisms that can account for the multidimensional nature of certain skills, particularly those that span multiple skill groups. To operate on the skill group level, systems need to be put in place to handle cases where a skill might belong to multiple categories to ensure accurate classification and retrieval. Conversely, the near non-existent overlap between knowledge and skill concepts lends itself to a categorisation that only distinguishes between skill and knowledge concepts.

Semantic structure of skill concepts

Next, the semantic structure of the preferred labels of the skill and knowledge concepts is analysed. Understanding these structures provides valuable insights for designing effective skill extraction pipelines and offers important insights for a possible language-specific preprocessing step. Analysing the distribution of lengths of the skill and knowledge concept labels in Figure 17, some distinct patterns can be identified. Knowledge concepts are predominantly concise, with the majority consisting of a single word. The brevity of knowledge concepts, often represented by single words (e.g., "Physics", "Biology"), reflects their nature as specific domains or fields of expertise. In contrast, skill concepts tend to be more elaborate, commonly spanning two to four words. The longer phrases observed in skill concepts (e.g., "data analysis", and "project management") suggest that these terms often describe actions, processes, or abilities requiring more detailed descriptions. Notably, some outliers extend up to 17 words in length, such as the skill concept “Apply basic rules of care and maintenance for leather goods and footwear machinery” with a length of 12.

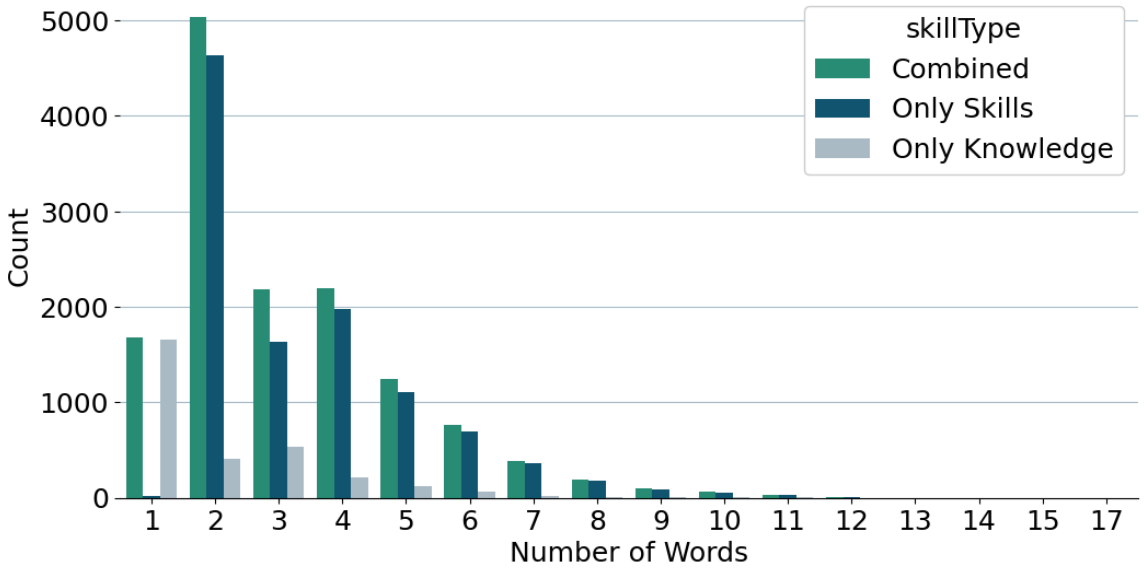

Figure 17: Distribution of word counts in preferred labels of ESCO skill concepts across skill types

When not only considering the word count per label, but only the respective part of speech tags (POS-tags), some additional patterns in the data become apparent. Overall, the most common POS-sequences are the combination of noun-verb, followed by noun-only label and the combination of noun-adposition-noun-verb, see Figure 18.

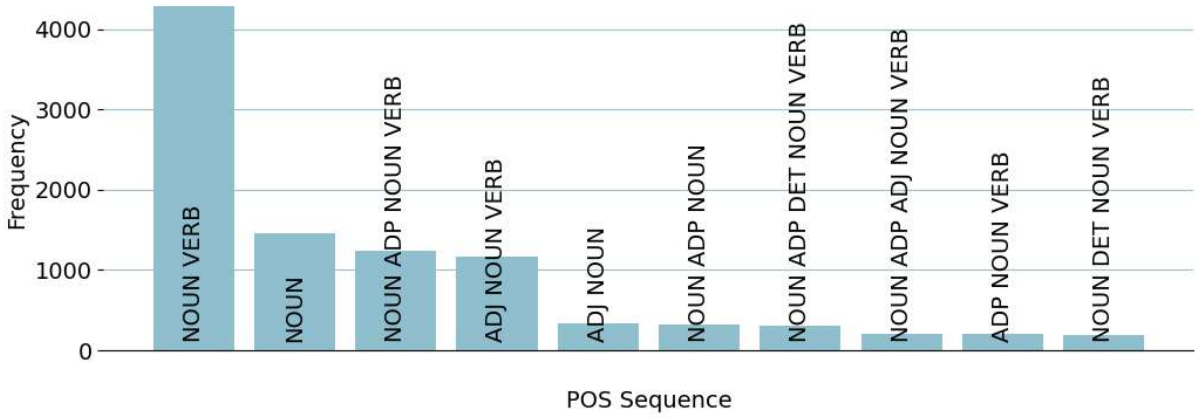

Figure 18: Part-of-speech sequences of skill and knowledge concepts combined

Differentiating between skill and knowledge concepts, the most common POS sequence for skill concepts is the combination of noun-verb, again indicating a focus on actions or processes, see Figure 19. Conversely, knowledge concepts are dominated by noun-only sequences, aligning with their role as specific fields or subjects, followed by the combination of adjective-noun and noun-adposition-noun.

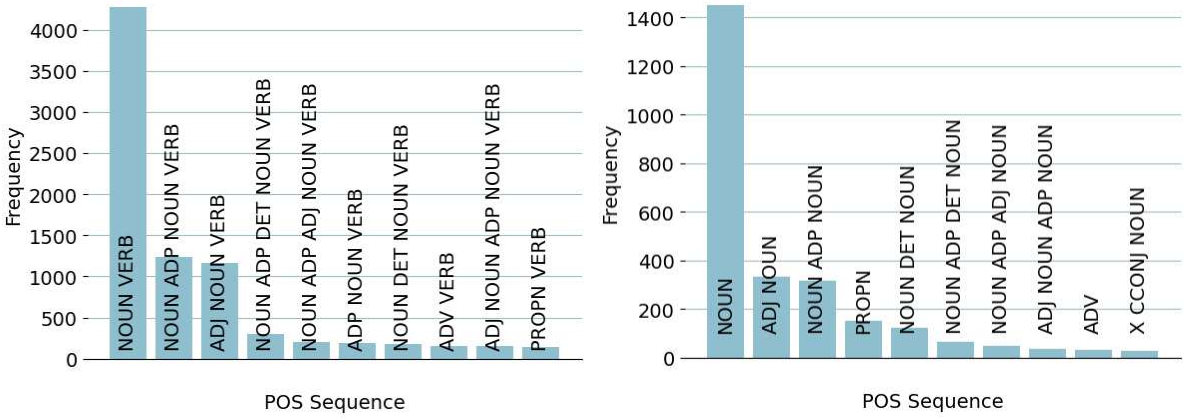

Figure 19: Part-of-speech sequences of skill (left) and knowledge (right) concepts

In conclusion, the analysis of word counts and POS sequences in skill and knowledge concepts provides critical insights into their semantic structure. For example, the most common label lengths range from one to five. Additionally, noun-verb sequences should be a focus in the identification of skills, whereas noun-only sequences should be prioritised for recognising knowledge concepts.

4.2 Creation of Annotated Data Set

As already mentioned in the problem definition, specifically P2, currently no publicly available data set for skill labelling exists. Because of this, a new labelled data set for validation and testing had to be created. The applied approach and methodology,

including the analysis of the resulting data set, will be discussed in the following chapter.

4.2.1 Data Source and Search Terms

To gather a set of unlabelled German job vacancies, the publicly accessible API of the German Federal Employment Agency[9] (German: Bundesagentur für Arbeit) was queried using three sets of five thematically related search terms each, see Table 6 for an overview of the search terms. The first set focuses on the maintenance and assembly domain. The second includes typical office jobs and tasks, and the third focuses on computer science-related job listings. This spread of jobs was chosen to compare the performance and applicability of the approach in different job domains. Additionally, gender-neutral and/or genderless search terms were used to prohibit potential biases in the job selection.

The maintenance and assembly domain included the following German search terms: Mechanik, Servicetechnik, Inbetriebnahme, Montage, Mechatronik (Engl.: Mechanics, service engineering, commissioning, assembly, mechatronics). For the office domain, the following search terms were queried: Verwaltung, Bürofachkraft, Corporate Services, Controlling, Rechnungswesen (Engl.: Administration, office specialist, corporate services, controlling, accounting). Finally, for the computer science domain, the following search terms were used: Programmierung, Developer, Data Scientist, Data Engineer, Data Analyst (Engl: Programmer, Developer, Data Scientist, Data Engineer, Data Analyst). Due to the prevalence of English job titles in the computer science domain, the German search terms are mostly interchangeable with their English counterparts.

Table 6: Overview of search terms for job listings in respective domains

	Maintenance and Assembly	Office	Computer Science
Search term 1	Mechanik	Verwaltung	Programmierung
Search term 2	Servicetechnik	Bürofachkraft	Developer
Search term 3	Inbetriebnahme	Corporate Services	Data Scientist
Search term 4	Montage	Controlling	Data Engineer
Search term 5	Mechatronik	Rechnungswesen	Data Analyst

[9] https://github.com/bundesAPI/jobsuche-api

4.2.2 Data Selection and Preparation

In order to obtain a diverse data set that is representative of the domain in question and is still feasible to annotate within the scope of this thesis, the objective is to create a data set comprising 20 job listings per domain, resulting in a total of 60 job listings. To create this data set, which will subsequently be employed for annotation purposes, the following data acquisition workflow was utilised (see Figure 20). The flow diagram illustrates the sequential process for the aggregation of job listings, beginning with the definition of five particular search terms, which are employed to query a database via an application programming interface (API).

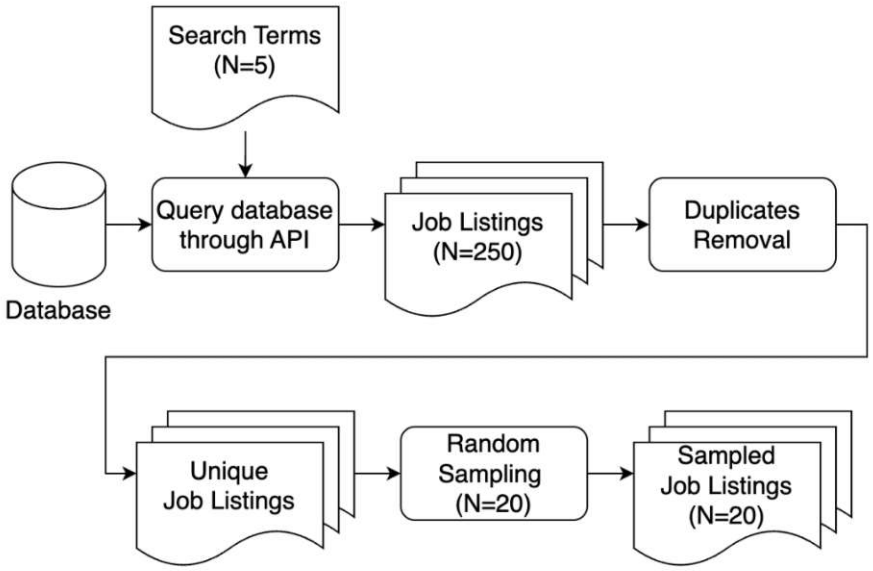

Figure 20: Overview of data acquisition and selection workflow for each domain

This query results in an initial collection of job listings, which likely also include duplicate entries that were found due to partial overlap between the search terms. To refine this data, a crucial step involves removing duplicates to ensure that the data set covers a diverse range of job listings, without some entries skewing the results and introducing a bias towards certain phrases and patterns. Besides duplicate entries, it was observed in the data that many companies post their job listings in batches, with each of the job listings being very similar in structure and content. To not introduce a bias towards any particular company that posted their batch close to the time of the query, the initial pool of job listings was further increased to include 250 job listings. This set of 250 job listings per domain served as the basis for the subsequent duplicate removal as well as a random sampling step. This is performed to further increase the diversity of job listings and to minimise the effect of the timing when the query is performed.

The impact of the random sampling on the diversity of employers can be seen in Table 7. While it was the case that some employers had more than 15 job listings included

in the original 250 job listings, this number was decreased to only 4 in the final data set. The final output of this workflow thus consists of these 20 sampled job listings.

Table 7: Employer diversity in acquired job listing data set

	Maintenance and Assembly		Office		Computer Science	
Size	250	20	250	20	250	20
Unique Employer	210	20	221	20	235	20
Min	1	1	1	1	1	1
Max	8	1	15	1	3	1
Mean	1,19	1	1,13	1	1,04	1

To get an initial overview of the resulting data set, the following word clouds were created using the Python *wordcloud* package[10]. Additionally, the data set was filtered using the German stopwords list provided by the Natural Language Toolkit (NLTK)[11]. Shared across all domains is the common usage of the word *team*. This is especially true for the office and computer science domains. In contrast, for the maintenance and assembly domain, the focus on *tasks* (German: Aufgaben), as well as the *customer* (German: Kunde), is much more pronounced.

Figure 21: WordClouds of frequent words within the job listings of the different domains (left: maintenance and assembly, middle: office, right: computer science)

4.2.3 Annotation Process and Guidelines

In the next phase, the acquired data set is manually labelled using annotation guidelines to create a reference data set with skill and knowledge labels. Additionally, personal or other identifying information in the job listings are also labelled, such as company names, contact details or specific city names. These sections are then removed from the data set for anonymity purposes before the data set is published. The annotation guidelines for the labelling process are based on the works of Zhang, Jensen, Sonniks, et al. (2022), which have been translated and, where necessary, adapted to better fit the German language use. The development of the annotation guidelines is an iterative process, requiring multiple stages of drafting, testing, and

[10] https://pypi.org/project/wordcloud/
[11] https://www.nltk.org/index.html

refinement to achieve clarity, consistency, and applicability, see Figure 22. This section explores the iterative improvements made to the annotation guidelines across the main iterations, providing examples to illustrate the changes made, as well as an overview of the metrics used for reliability testing. The complete guidelines can be found in Appendix Annotation Guidelines.

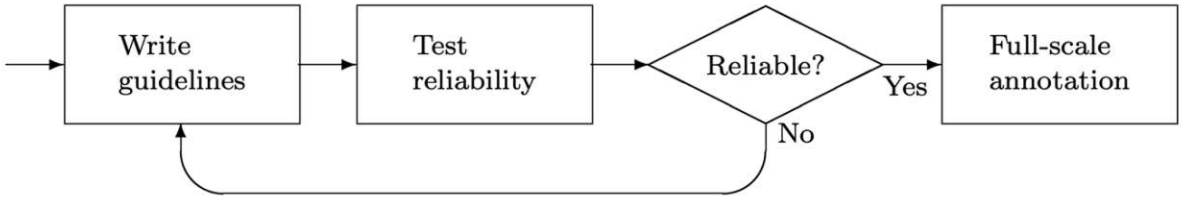

Figure 22: Schematics of iterative reliability testing process (Artstein, 2017, p. 3)

The preliminary annotation guidelines were developed based on the works of Zhang, Jensen, Sonniks, et al. (2022) and a small number of reference job listings from the previously obtained data set. First, the original guidelines were translated and extended with examples from the reference job listings to add additional context and catch potential uncertainties created through the language transfer. Figure 23 shows an annotated example sentence with skill (German: *Fähigkeit*) and knowledge (German: *Kenntnis*) labels from the created guidelines.

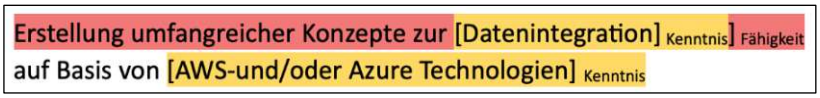

Figure 23: Annotation example from the guidelines (Engl.: "Creation of comprehensive concepts for data integration based on AWS and/or Azure technologies")

For the iterative improvement and the reliability testing, a set of nine job listings, three per domain, was annotated independently by two annotators using the open-source text annotation tool *doccano* (Nakayama et al., 2018), see Figure 24 for an annotated example.

After each round, the reliability of the annotation, and thus of the guidelines, was evaluated by calculating the inter-annotator agreement for each label individually (i.e.: "Skill" or "Knowledge") using Cohen's κ (Fleiss & Cohen, 1973). Cohen's κ is a statistical measure used to evaluate the level of agreement between two raters when classifying items into categorical outcomes. In our case, this means classifying each token as either labelled or unlabelled. The κ statistic accounts for the possibility of agreement occurring by chance, providing a more robust measure than simple percentage agreement. Cohen's κ is calculated as follows, see Formula 1, where:

- P_o represents the proportion of times that the raters agree in their observations
- p_e indicates the probability that the raters would agree by chance

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

Formula 1: Definition of Cohen's kappa

A value of $\kappa = 1$ indicates perfect agreement, whereas $\kappa = 0$ indicates no agreement beyond chance. Generally, values above 0.61 indicate substantial agreement, with values above 0.81 indicating almost perfect agreement. Conversely, values below 0.40 indicate poor agreement (Landis & Koch, 1977).

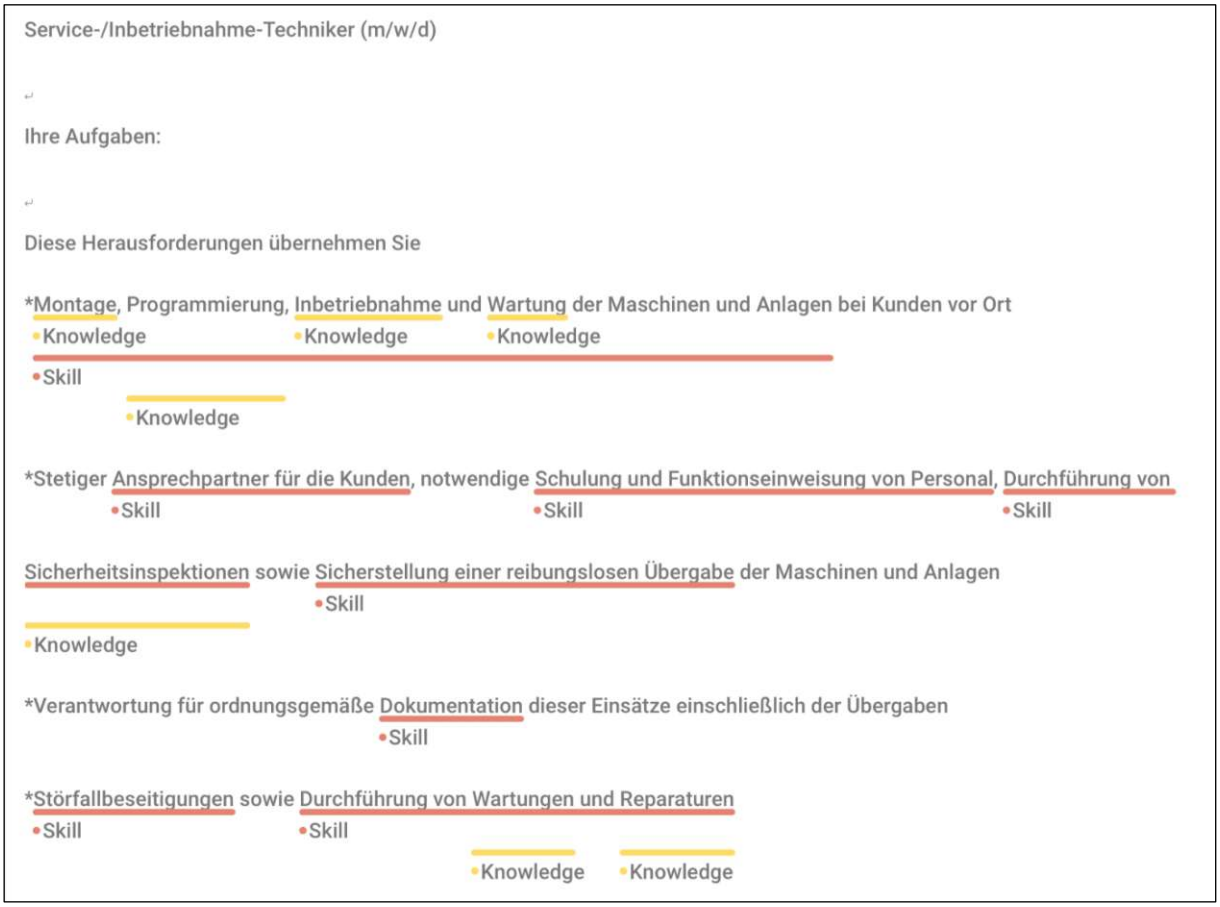

Figure 24: Example of an annotated job listing in doccano

Given the complexity of skill and knowledge span annotation, as outlined by (Zhang, Jensen, Sonniks, et al., 2022), the targeted reliability for the annotation guidelines was set to achieve substantial agreement, hence a value above 0.61. After each annotation iteration, the disagreement between the annotators was analysed, and the guidelines were adapted for the next round of iterations. After the fourth round of improvements, an inter-annotator agreement of at least $\kappa = 0.79$ for skills annotations and at least $\kappa = 0.73$ for knowledge annotations was achieved, with both values averaging 0.87 and 0.88 across domains in skill and knowledge annotation, respectively. Among the changes made during the iterative improvement process are, for example, the inclusion of completed apprenticeships as knowledge components, see Figure 25. This is

especially important in the maintenance and assembly domain since it is a very common requirement within the job listings.

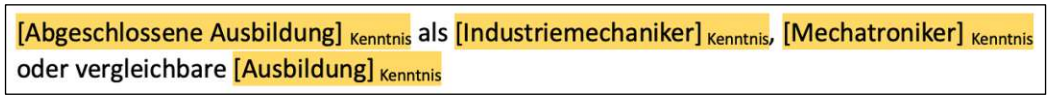

Figure 25: Example sentence for annotating apprenticeship requirements as knowledge components

Other additions include the handling of anaphors as well as compounded words, see Figure 26 for examples.

Figure 26: Examples for correct annotation of combinations of compounded words

After achieving sufficient reliability between annotators using the adapted guidelines, the full data set of 60 job listings was annotated by a single annotator. The final data set with the annotations is stored in the JSONL file format, with the spans being marked as offset arrays [span start, span end] for each individual label in each job listing.

4.2.4 Final Data Set Statistics

In the following, the final annotated data set will be analysed to get a better understanding of the underlying data distributions and the make-up of the skill and knowledge labels.

Starting with the summary statistics of the annotated dataset in Table 8, 60 job listings, consisting of 1188 sentences and 22,722 tokens were annotated. The computer science domain included the highest number of tokens and sentences per job listing on average. This reflects the at times much more detailed job listings in this domain, highlighting the effort put into the job listings to attract the right talent. Skill spans are most numerous in the office domain. In contrast, the computer science domain leads in knowledge spans, highlighting the listing of specialised knowledge, usually in the form of various programming languages and software frameworks required in these roles. The presence of overlapping spans, particularly in the maintenance and assembly domain, suggests that these roles often require interconnected skills and knowledge areas, reflecting the complexity and multi-disciplinary nature of tasks in this sector.

Table 8: Overview of summary statistics of the final annotated data set

	Maintenance and Assembly	Office	Computer Science	Overall
Job Listings	20	20	20	60
Sentences	328	366	494	1188
Tokens	5825	7403	9494	22722
Skill Spans	162	253	205	620
Knowledge Spans	224	276	509	1009
Overlapping Spans	92	57	100	249

The word clouds in Figure 27 reveal significant differences in skill and knowledge spans associated with each domain. In the maintenance and assembly domain, common terms such as “Montage” (assembly), “Wartung” (maintenance), and “Inbetriebnahme” (commissioning) highlight the technical and operational focus of these jobs. Interestingly, “Programmierung” (programming) is also a highly requested skill, showing the increased importance of computer skills in traditionally hands-on environments. In contrast, the office domain is characterised by terms like “Rechnungswesen” (accounting) and “Controlling”, which indicate a focus on finance and administrative tasks. The term “Weiterentwicklung” (further development) additionally hints at the importance of financial management and continuous improvement in office-based roles, where precision and knowledge of business processes are key. In the computer science domain, the dominant terms are much more tool-centric and programming-language, such as “SQL”, “Python” and “Java” among others, which underscore the technical skills required for data management and software development. These terms point to a strong emphasis on programming languages and data analysis, essential for roles in the rapidly evolving field of technology.

Figure 27: Word clouds of annotated skill and knowledge spans across the domains (left: maintenance and assembly, middle: office, right: computer science)

Moving on to analysing the distribution of skill and knowledge span lengths, general trends can be discovered, see Figure 28. In the maintenance and assembly domain, the violin plot shows a broad distribution for skill span lengths, with a higher emphasis on skill spans of lengths one to three. The knowledge spans, however, are even more compact, with a concentration around length one and a much narrower distribution,

suggesting that the knowledge required is often expressed in shorter, more concise terms.

For the office domain, the distribution for skill span lengths is less wide compared to the maintenance and assembly domain, with the median moving to length three, while a length of five is also still relatively common. Additionally, the longest skill spans reach up to a length of 15 tokens. The knowledge spans are more widely distributed, with a significant number of longer spans. This implies that knowledge in office-related roles may be more varied and complex, often requiring longer phrases to fully articulate the necessary requirements.

In the computer science domain, the pattern for skill spans is somewhat similar to the office domain, with a relatively wide distribution and a focus on longer lengths. This could reflect the complex and technical nature of skills in the tech sector, where much detail is given. The knowledge spans in the tech domain show a narrower distribution, with many short spans, indicating that the knowledge required in these roles is often concise, likely due to the aforementioned focus on various programming languages and software frameworks.

Overall, the plot reveals that skill span lengths tend to be longer and more varied across all domains, while knowledge spans are generally more concise. The differences in span lengths across domains also highlight how the complexity and specificity of skills and knowledge vary depending on the sector.

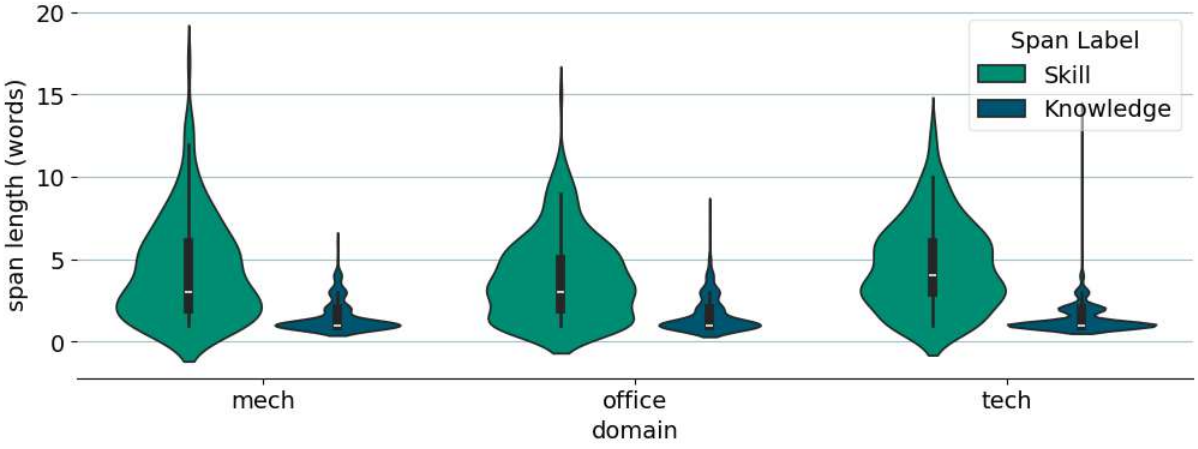

Figure 28: Distribution of span lengths of annotated skill and knowledge spans in the final data set (n=60)

The POS distribution for skill and knowledge annotations shows that skill annotations are predominantly noun-based, with frequent combinations like noun-adposition-noun and adjective-noun, see Figure 29. This contrasts with the findings in the analysis of the ESCO taxonomy, where noun-verb combinations were the most common POS tags for skill concepts and needs to be further analysed when assessing the model performances. On the other hand, knowledge annotations are also heavily noun-

based, which conforms with the findings in the analysis of the ESCO taxonomy. Additionally, a significant presence of proper nouns (PROPN) exists particularly in the computer science and office domains, indicating the use of specific terminologies and concepts unique to these fields.

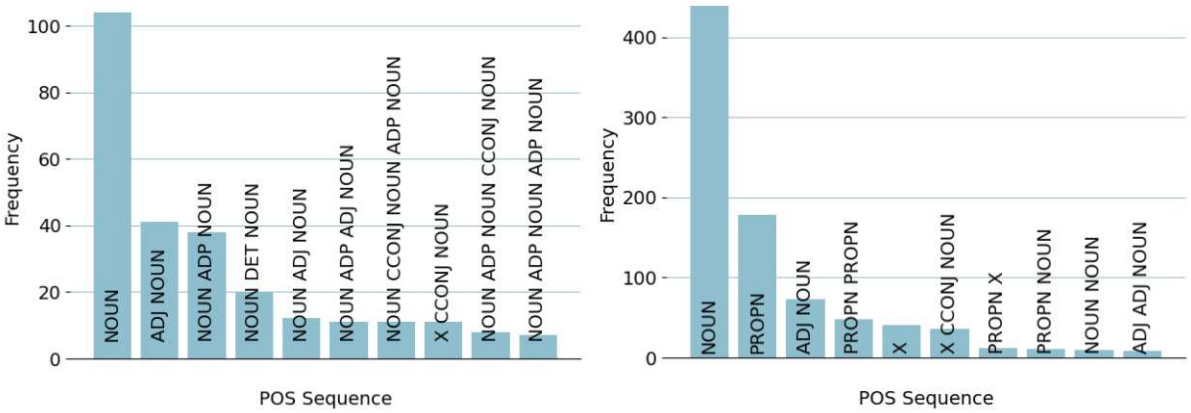

Figure 29: Distribution of part-of-speech tags of annotated skill and knowledge spans in the final data set (n=60)

4.3 Summary of Data Acquisition and Analysis

To summarise, the above chapter outlines the systematic approach to data acquisition and analysis, addressing key research questions and objectives (RQ2, O3, O1) and laying the foundation for benchmarking and evaluating the impact of language-specific preprocessing and pretraining on skill and knowledge extraction.

The chapter first gives an overview of the ESCO taxonomy, specifically its "skills pillar," which serves as the core reference for determining what constitutes a skill in job listings. The taxonomy's hierarchical and interconnected structure spans 13,890 skill concepts across four high-level categories: Knowledge, Skills, Transversal skills and competences, and Language skills and knowledge. Concepts are classified into 27 subgroups with further granularity, allowing nuanced categorisation. The analysis of the skill and knowledge concepts reveals a higher prevalence of skills (10,831) compared to knowledge concepts (3,059), reflecting their task-specific nature. Skill reusability is categorised into transversal, cross-sectoral, sector-specific, and occupation-specific levels. Sector-specific concepts dominate due to the specialised requirements of industries. Overlaps in skill groups highlight the multidimensional applicability of transversal and communication skills, which are essential for classification flexibility. Further analysing the semantic structure and distribution of skill concepts, the following patterns emerge: Skill concepts typically use multi-word labels (2–4 words), emphasising actions and processes, while knowledge concepts are more concise (often single words). Analysis of part-of-speech (POS) sequences identifies noun-verb combinations as dominant for skills, whereas knowledge concepts favour

noun-only structures. These patterns provide insights for designing skill extraction pipelines.

The lack of a public skill-labelling dataset necessitated the creation of a German-language dataset. A representative sample of 60 job listings was curated from three domains (Maintenance and Assembly, Office, and Computer Science), using predefined search terms and a publicly accessible job platform to ensure diversity and mitigate biases. Job listings were filtered to remove duplicates and batch postings, with random sampling ensuring varied employer representation. Annotations were guided by iteratively developed annotation guidelines, achieving substantial inter-annotator agreement ($\kappa \geq 0.79$ for skills and $\kappa \geq 0.73$ for knowledge). The dataset, comprising 60 annotated job listings with 22,722 tokens and 1,629 skill and knowledge spans, is stored in JSONL format. The analysis of the annotated dataset across the three domains reveals that skill spans are typically longer and more varied than knowledge spans across all domains, aligning with the complexity of skill requirements. POS analysis shows a predominance of noun-based annotations, with variations reflecting domain-specific terminology. Within each domain, the domain-specific terminology varies, along with the prevalence of overlapping skills and knowledge annotations, as well as the ratio of skill to knowledge concepts within each job listing.

This variance of different POS patterns, terminology and nested skill and knowledge concepts, highlights the importance of flexible classification systems to handle multidimensional skill concepts. Insights from the annotated dataset support the development of language-specific preprocessing techniques and serve as a benchmark for evaluating extraction models, which will be further detailed in the following chapter.

5 Model Architecture

In the following chapter, the developed model architecture is explained in detail. For this, first an overview of the general model is given, along with the reason behind the design choices. Then, the individual elements of the model architecture is discussed further, also highlighting the algorithms themselves, as well as their input and output parameters. The model architecture detailed in this chapter serves as the experimental basis for comparing the impact of language-specific pre-training and pre-processing techniques (O3) by introducing a novel POS-n-gram suggester and utilizing language and domain adapted transformer models, as well as comparing the performance of different pre-trained models and pre-processing steps on the German benchmarking data set (O4) by introducing a modular extraction and classification pipeline.

5.1 Overview

Building upon the findings of the state-of-the-art analysis, the applied model will also utilise the current state-of-the-art transformer architecture. In the context of this thesis, labelled data is only available in the form of the ESCO taxonomy and the annotated data set, as described above. Since only a limited number of job listings are available, fine-tuning and additional domain adaptation were ruled out, following the guidelines presented by Tunstall et al. (2022), see Figure 30. Further, domain adaptation for German job listings was already carried out (Gnehm, Bühlmann, & Clematide, 2022), and the findings will also be incorporated in the developed approach. Additionally, further training of the transformer models is not within the scope of this thesis (as is proposed in UMLFiT), thus embedding lookup and few-shot learning are the only viable options based on the decision tree. Embedding lookup has already been successfully applied (Khaouja, Mezzour, et al., 2021; Konstantinidis et al., 2022; Zhang, Jensen, van der Goot, et al., 2022) among others, albeit only on English data. To evaluate the performance of these state-of-the-art methods on German job listings (see RQ1), the chosen model architecture will also build upon embedding lookups.

The developed model consists of three main modules: the suggester, the matcher and the classifier, see Figure 31. The *suggester module* extracts potential skill phrases (chunks) from the job listings using a range of different chunking techniques: Nouns only, noun chunks, n-grams and language-adapted POS-n-grams. The POS-n-grams extract n-grams based on common part-of-speech tags, utilising the most common patterns of the German labels of the ESCO taxonomy.

The *matcher module* then takes in the chunks provided by the suggester, embeds them using one of three state-of-the-art transformer models, and matches the resulting embeddings against a precalculated set of ESCO skill embeddings using cosine

similarity and different matching thresholds. The goal is to identify semantically similar skills that will then be used as the basis for the following classification step.

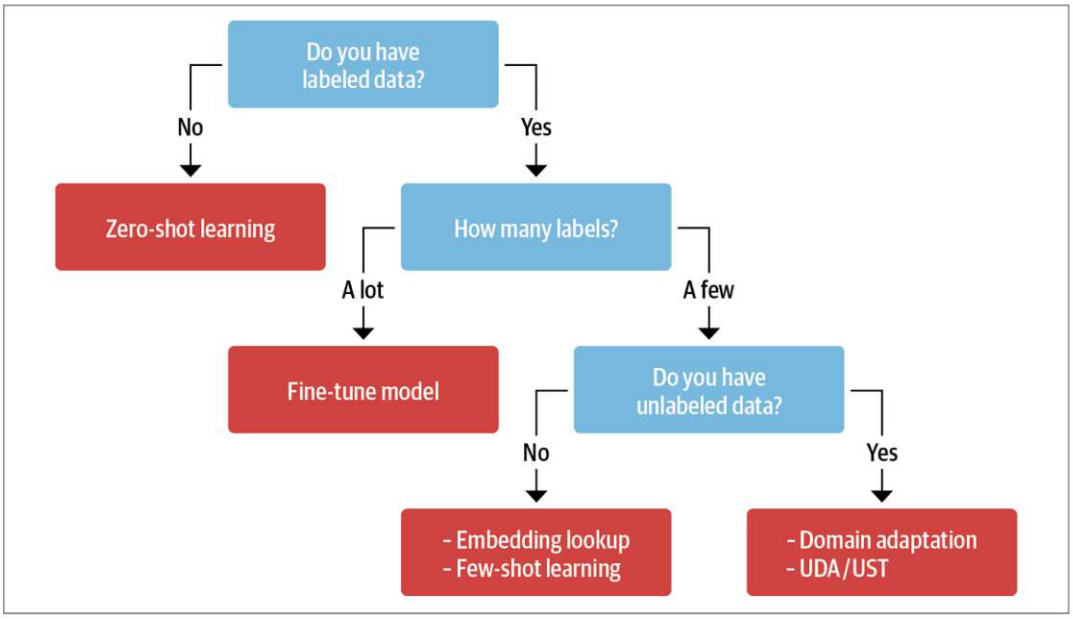

Figure 30: Decision tree for viable transfer learning techniques in NLP based on the availability of training data in the target domain (Tunstall et al., 2022, p. 250)

Finally, the *classifier module* classifies the suggested spans based on their semantic similarity to relevant ESCO skills, determined by the matcher component. For this different voting strategies are explored: confidence voting, based on the most semantic similar ESCO skill above the predefined matching threshold; majority voting, where the label of the top n most semantically similar labels is assigned to the span; and weighted majority voting, where the influence of each vote is weighted by its semantic similarity.

Overall, the pipeline is designed to enable automated extraction and classification of skills from unstructured job listings, applying various natural language processing techniques and transformer models to ensure accurate skill identification and alignment with a standardised taxonomy. The following subchapters will discuss the functionality and structure of the individual components in more detail.

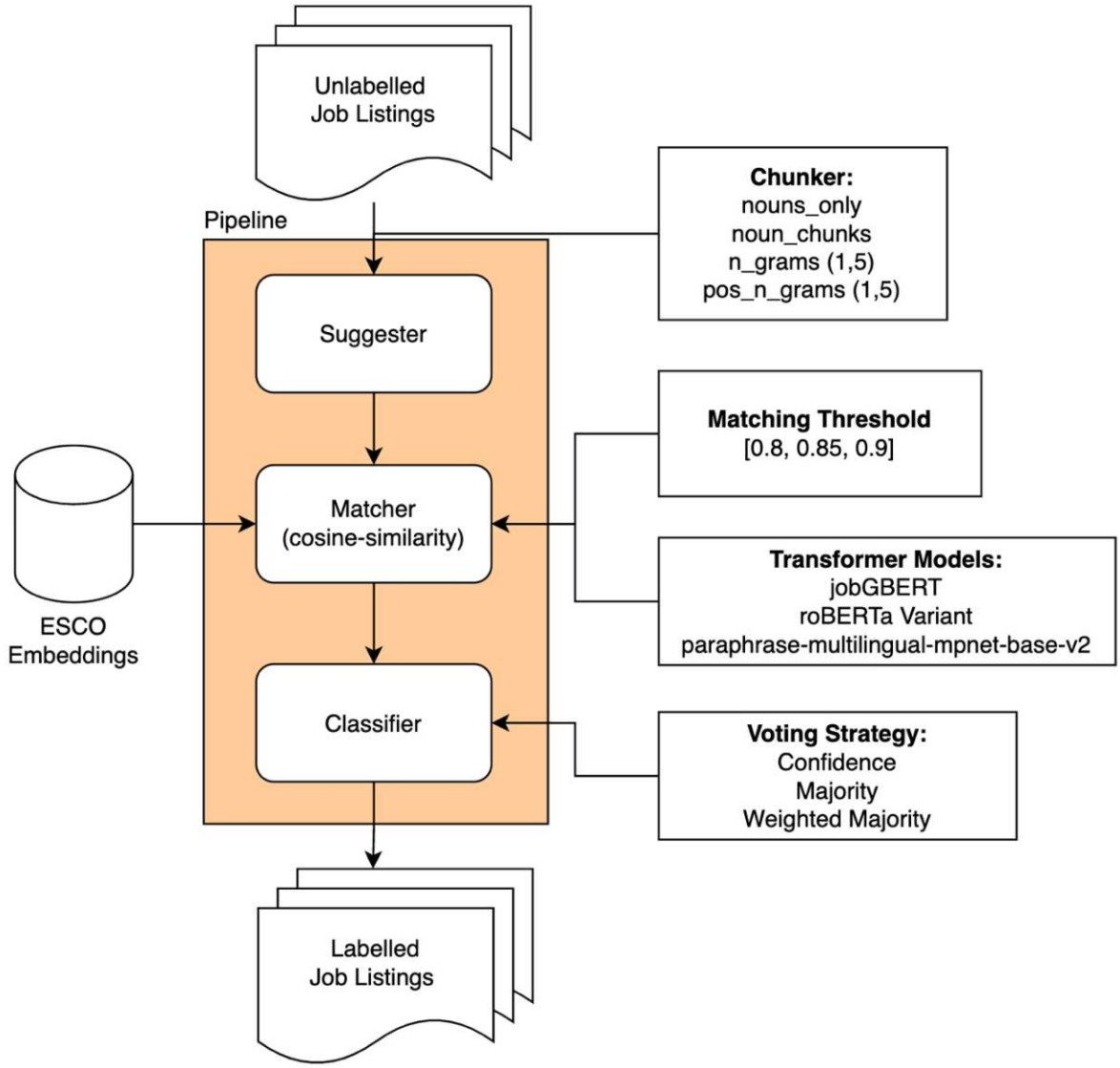

Figure 31: Schematics of the proposed model architecture

5.2 Suggester Module

The first major component in the pipeline is the *suggester module*. This module is responsible for identifying potential skill-related phrases from the unstructured text of the job listings. Within the suggester, the choice of the *chunker* plays a crucial role. The chunker applies various text chunking strategies to isolate relevant text segments. The strategies that have been implemented and will be compared within this thesis include extracting only nouns (*nouns_only*), identifying noun phrases (*noun_chunks*), generating n-grams (sequences of words) with lengths between 1 and 5 words (*n_grams*), and creating n-grams based on part-of-speech tags (*pos_n_grams*). The suggester's output is a set of candidate phrases that might represent skills mentioned in the job listings. The basis for the suggester module are the part-of-speech tagging[12]

[12] https://spacy.io/usage/linguistic-features#pos-tagging

and noun chunking[13] capabilities of the spaCy library. The key components of the suggester across all chunkers are as follows:

- **Document (doc)**: The input of the component is a spaCy object, which represents a preprocessed input job listing as the document. This document is already segmented into sentences and tokens.
- **Sentence iteration**: This method iterates over each sentence in the document. Analysing the job listing by passing it sentence by sentence to the chunker.
- **Chunker:** Each sentence is passed into the chunker, which then iterates over each word in the sentence and creates a set of potential skill chunks. For this, different methods are applied, namely extracting only nouns (*nouns_only*), identifying noun phrases (*noun_chunks*), generating n-grams (sequences of words) with lengths between 1 and 5 words (*n_grams*), and creating n-grams based on part-of-speech tags (*pos_n_grams*).
- **Return value**: The method returns a list of potential skill chunks, where each chunk includes its text as a string, its starting character position, and its ending character position in the overarching document.

Chunking Method 1: Nouns Only

This method uses the spaCy POS-tagging function to extract only individual nouns and proper nouns from a document as potential skill phrases, see Figure 32.

```
Input:
    doc: A processed spaCy document containing sentences and
tokens
Result:
    pot_skills_chunks: A list of potential skill chunks containing
text and character positions

foreach sent ϵ doc.sents:
    foreach token ϵ sent:
        if token.pos_ ϵ {"NOUN", "PROPN"}:
            span ← token.text
            span_start ← token.idx
            span_end ← span_start + length of token
            pot_skills_chunks ← [span, span_start, span_end]

return pot_skills_chunks
```

Figure 32: Pseudo code for noun-only chunker of suggester component

[13] https://spacy.io/usage/linguistic-features#noun-chunks

Chunking Method 2: Noun Chunks

This method extracts phrases (noun chunks) from a document that represent potential skill phrases, see Figure 33. A noun chunk consists of a noun and its immediate modifiers, usually providing a more meaningful phrase than single-word tokens. For the noun chunk extraction, the built-in spaCy noun chunking function[14] is utilised.

```
Input:
    doc: A processed spaCy document containing sentences and
tokens Result:
    pot_skills_chunks: A list of potential skill chunks containing
text and character positions

foreach sent ϵ doc.sents:
    foreach chunk ϵ sent.noun_chunks:
        span ← chunk.text
        span_start ← chunk.start_char
        span_end ← chunk.end_char
        pot_skills_chunks ← [span, span_start, span_end]

return pot_skills_chunks
```

Figure 33: Pseudo-code for noun chunks chunker of suggester module

Chunking Method 3: N-grams

This method works by identifying continuous sequences of words (n-grams) within the text, excluding punctuation marks. The method generates n-grams of various lengths, determined by the predefined minimum (min_size) and maximum length (max_size) and returns them as a list. For each n-gram length, the code creates a span (a sequence of tokens), and if the span does not exceed the sentence boundary and does not end with punctuation it is added to a list of potential skill chunks, see Figure 34.

[14] https://spacy.io/usage/linguistic-features#noun-chunks

```
    Input:
        doc: A processed spaCy document containing sentences and tokens
        min_size: Minimum length of n-grams
        max_size: Maximum length of n-grams
        include_punct: Boolean indicating whether to include
    punctuation in n-grams
    Result:
    pot_skills_chunks: A list of potential skill chunks containing text
    and character positions

    sizes ← list of integers from min_size to max_size
    foreach sent ∈ doc.sents:
        foreach token ∈ sent:
            if token.pos_ == PUNCT:
                continue
            foreach size ∈ sizes:
                if token.i + size <= sent.end and (include_punct or
        doc[token.i + size - 1].pos_ ≠
        PUNCT):
                    span ← tokens from token.i to token.i + size
                    span_start ← start character position of span
                    span_end ← end character position of span
                    pot_skills_chunks ← [span.text, span_start,
    span_end]

    return pot_skills_chunks
```

Figure 34: Pseudo-code for n-grams chunker of suggester module

Chunking Method 4: Part-of-Speech n-grams

This method focuses on identifying sequences of words that may be relevant for skill extraction, based on common part-of-speech tags, utilising the most common patterns of the German labels of the ESCO taxonomy. Based on this, nouns, proper nouns and adjectives were used as the entry points for the n-gram creation. Like the n-gram implementation before, the method generates n-grams of various lengths, determined by the predefined minimum (min_size) and maximum length (max_size) text, excluding punctuation marks, and returns them as a list, see Figure 35.

```
Input:
    doc: A processed spaCy document containing sentences and
tokens
    min_size: Minimum length of n-grams
    max_size: Maximum length of n-grams
    include_punct: Boolean indicating whether to include
punctuation in n-grams
Result:
    pot_skills_chunks: A list of potential skill chunks containing
text and character positions

sizes ← list of integers from min_size to max_size

foreach sent ∈ doc.sents:
    foreach token ∈ sent:
        if token.pos_ == "PUNCT":
            continue
        if token.pos_ ∈ {"NOUN", "PROPN", "ADJ"}:
            span ← token.text
            span_start ← token.idx
            span_end ← span_start + length of token
            pot_skills_chunks ← [span, span_start, span_end]
            foreach size ∈ sizes:
                if token.i + size ≤ sent.end and (include_punct or
doc[token.i + size - 1].pos_ ≠ "PUNCT"):
                    span ← tokens from token.i to token.i + size
                    span_start ← start character position of span
                    span_end ← end character position of span
                    pot_skills_chunks ← [span.text, span_start,
span_end]

return pot_skills_chunks
```

Figure 35: Pseudo-code for POS-n-grams chunker of suggester module

5.3 Matcher Module

Following the suggestion module, the pipeline moves into the *matcher module*. The matcher uses a semantic similarity approach, based on cosine similarity as the similarity measure, to compare a list of suggested chunks against a predefined database of skill embeddings, specifically those from the ESCO taxonomy. This process involves calculating the similarity between the individual chunk candidates and the ESCO embeddings to identify potential matches. The matcher includes a configurable matching threshold, which can be adjusted to fine-tune the stringency of the matching process. Within this thesis, the different matching thresholds included are

[0.8, 0.85, 0.9]. This range has shown to be successful across the range of different transformer models. Higher thresholds result in stricter matching criteria, while lower thresholds may yield more matches but with less precision. Additionally, the module also incorporates different state-of-the-art transformer models to create the embeddings of the suggested chunks. The models included in the analysis are:

- *jobGBERT*[15]*,* a domain-adapted version of the German BERT variation GBERT, that has been trained specifically on job listings (Gnehm, Bühlmann, & Clematide, 2022)
- *cross-en-de-roberta-sentence-transformer*[16]**,** a RoBERTa variant specifically trained for sentence embeddings of German text
- *paraphrase-multilingual-mpnet-base-v2*[17], a model included in the *sentence-transformers* library, used for text embeddings and similarity search using Siamese BERT networks (Reimers & Gurevych, 2019)

These models were chosen based on the findings within the state-of-the-art anaylsis. JobGBERT showed the best performance for German and domain adapted models in for similar tasks on German job listings in the study conducted by Gnehm, Bühlmann, & Clematide (2022). The RoBERTa model showed some cases the best performance results in a similar weakly supervised skill extraction setting for English job listings, even without additional domain adaptation (Zhang, Jensen, van der Goot, et al., 2022), and the paraphrase-multilingual-mpnet-base-v2 is based SBERT networks (Reimers & Gurevych, 2019), being one of the most widely used models for computing semantic similarity.

The matcher module then uses these models to create the embeddings of the skill chunks. Each skill chunk is embedded using the specified transformer model and an array of embeddings is returned. Then, the cosine similarity between the encoded skill chunks and the reference embeddings is calculated, where the reference embeddings are a list of pre-embedded ESCO-skill labels using the same transformer model. Next, the top N (with the default being set to 5) matches for each skill chunk are identified and if the similarity exceeds the specified threshold. Finally, the module returns the list of length ≤N, which includes the matches above the specified threshold per given skill chunk from the suggester class. The pseudo-code of the matcher module is as follows, see Figure 36:

[15] https://huggingface.co/agne/jobGBERT
[16] https://huggingface.co/T-Systems-onsite/cross-en-de-roberta-sentence-transformer
[17] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

```
Data:
  T: pre-trained transformer model
  R: reference ESCO skill embeddings
  P: list of potential skill chunks
  n: number of top matches to consider
  t: similarity threshold
Result:
  M: list of matches for each skill chunk
Initialise:
  chunk_embeddings ← empty list

foreach chunk ϵ P:
    encoded_chunk ← T.encode(chunk[0])
    append encoded_chunk to chunk_embeddings

similarities ← calculate cosine similarity between
chunk_embeddings and R

foreach i, chunk ϵ enumerate(P):
    max_similarity ← find maximum value in similarities[i]
    if max_similarity > t:
        top_n_indices ← indices of the top n values in
similarities[i]
        top_n_values ← top n similarity values in similarities[i]
        append [chunk, top_n_indices, top_n_values] to M

return M
```

Figure 36: Pseudo-code of matcher module

5.4 Classifier Module

Once the potential skill matches are identified for each chunk from the job listing, the results are passed to the *classifier module*. This module is responsible for the final classification of the extracted chunks. For this, different voting strategies are explored:

- *confidence voting*, based on the label of the most semantic similar ESCO skill above the predefined matching threshold
- *majority voting*, where the label of the top N most semantically similar labels is assigned to the span, but only if at least one ESCO skill label is above the predefined matching threshold
- *weighted majority voting*, where similar to majority voting, the most similar labels are taken into account, but the influence of each vote is weighted by their semantic similarity. Again, only if at least one ESCO skill label is above the predefined matching threshold

The module returns a list of tuples, where each tuple contains the match identifier, the selected skill label, and the score associated with that label. The pseudo-code of the classifier module is as follows, see Figure 37:

```
Data:
  R: DataFrame containing reference skill types
  V: selected voting strategy ("majority", "confidence",
"weighted_majority")
  M: list of matches, where each match contains (match_id,
indices, confidences)
Result:
  L: list of tuples, each containing (match_id, label, score)

Initialise:
  L ← empty list

foreach match ϵ M:
    match_labels ← empty list
    foreach index ϵ match:
        label_temp ← R["skillType"][index]
            append label_temp to match_labels
    if V = "majority":
        label ← most frequent label in match_labels
        score ← frequency of label in match_labels
    else if V = "confidence":
        label ← label in match_labels with highest confidence
        score ← highest confidence value in match
    else if V = "weighted_majority":
        label_scores ← empty dictionary
        foreach label, confidence in zip(match_labels, match):
            if label in label_scores:
                label_scores[label] += confidence
            else:
                label_scores[label] = confidence
        foreach label ϵ label_scores:
            label_scores[label] = label_scores[label] / frequency
of label in match_labels

        label ← label with highest value in label_scores
        score ← highest value in label_scores

    append (match, label, score) to L

return L
```

Figure 37: Pseudo-code of classifier module

5.5 Summary of Model Architecture

To summarise, the above chapter details the developed model architecture, beginning with an overview of the design and its rationale, followed by an in-depth discussion of its components, algorithms, and parameters. The proposed model architecture builds upon state-of-the-art transformer architectures, constrained by the limited availability of labelled data. As fine-tuning and further domain adaptation are outside the scope, the approach relies on embedding lookups, combined with domain and language specific POS-patterns, utilising the ESCO taxonomy for alignment of identified skills with a standardised taxonomy. The architecture consists of three main modules, see Figure 31 on page 61:

1. **Suggester Module**: Extracts potential skill phrases (chunks) from job listings using techniques like noun-only chunks, n-grams, and a novel domain and language-adapted POS-n-gram suggester, designed to consider the specific wording patterns commonly present in German job listings.

2. **Matcher Module**: Embeds the extracted chunks using one of three transformer models and matches them to precalculated ESCO skill embeddings. Matching is performed using cosine similarity and different thresholds to identify semantically similar skills for classification.

3. **Classifier Module**: Categorises suggested spans based on semantic similarity to ESCO skills. It employs strategies like confidence voting (top semantic similarity above a threshold), majority voting (top n similar labels), and weighted majority voting (weights based on similarity).

The proposed architecture serves as the experimental basis for evaluating and comparing different language-specific pre-trained models and pre-processing techniques. The experimental evaluation thus also serves as a cornerstone to refine the pipeline, address potential shortcomings, and validate its applicability in real-world scenarios and will be detailed in the following chapter.

6 Experiment Design and Evaluation

This chapter outlines the experimental design and methodology employed to evaluate the performance of different models in extracting skills and knowledge from job listings using the developed pipeline. Firstly, the structured experimental design is detailed, including the selection of models, the tuning of hyperparameters and the assessment of performance. Subsequently, the employed evaluation metrics are explained. The objectives of this chapter are twofold: firstly, compare the impact of language-specific preprocessing techniques, such as POS-n-grams (O3) and secondly, compare the performance of different pre-trained models and pre-processing steps on the German benchmarking data set (O4). By following a systematic approach, the experiment aims to provide insights into the applicability of advanced NLP models for automated skill extraction in German-speaking job markets. The results of this chapter then serve as the basis for conclusively answering the research questions RQ1-RQ3.

6.1 Experiment Design

The following subchapter presents the experimental design and methodology applied in this thesis to evaluate the performance of different models in skill and knowledge extraction from job listings. The workflow of the empirical part is structured into several key phases, which are depicted schematically in Figure 38. This workflow ensures a systematic approach to model selection, hyperparameter tuning, and performance evaluation, aiming to investigate the applicability of state-of-the-art methods on German job listings (O2), compare the impact of language-specific pre-processing (O3) in the form of POS-n-grams and finally compare the performance of different pre-trained models and pre-processing steps on the German benchmarking data set (O4).

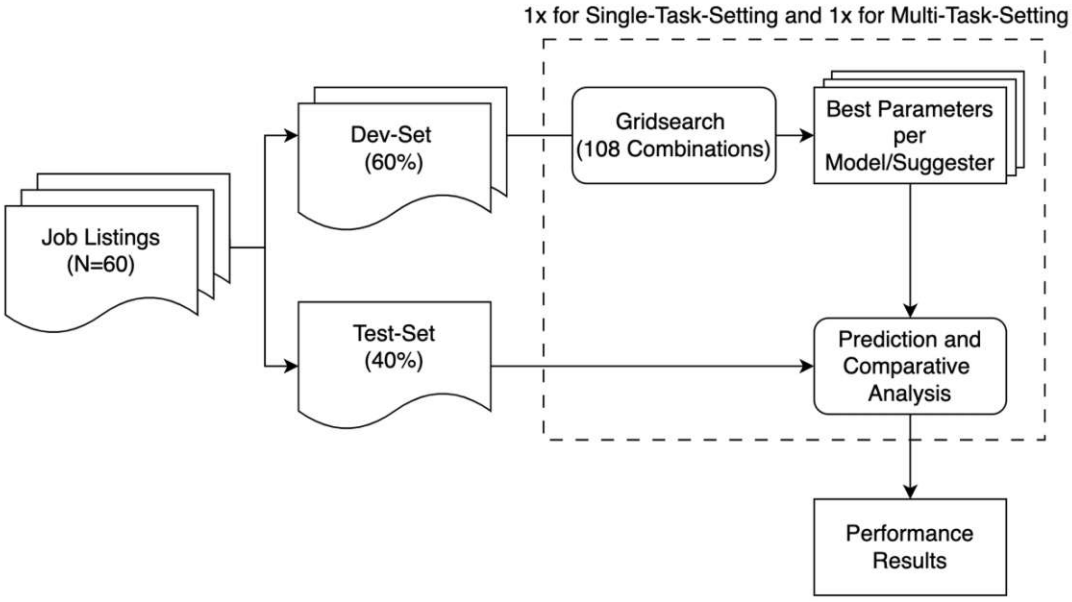

Figure 38: Schematics of experiment design

The experiment begins with the collection and annotation of job listings, see Chapter Creation of Annotated Data Set for a detailed description of the annotation process. A total of 60 annotated job listings are used, representing the ground truth for evaluating different models. These job listings are divided into two distinct sets:

1. **Development-Set (Dev-Set, 60%)**: Comprising 60% of the total data, this set is utilised for initial model training and hyperparameter tuning.
2. **Test-Set (Test-Set, 40%)**: The remaining 40% of the data is reserved for evaluating the performance of the trained models.

The division into these sets ensures that the model tuning and evaluation processes are not biased, thereby allowing for a fair comparison of performance metrics.

Using the dev-set, a grid search is performed to identify the optimal hyperparameters for each of the three examined embedding models as well as for the implemented chunker variations. For a detailed description of the model architecture see Chapter Model Architecture. The grid search covers a total of 108 different hyperparameter combinations, see Table 9.

Table 9: Overview of grid search parameters

Module	**Suggester**	**Matcher**		**Classifier**
Parameter	**Chunker**	**Embedding Model**	**Matching Threshold**	**Voting Strategy**
Values for Grid Search	Nouns	jobGBERT	0.8	Confidence
	Noun chunks	cross-en-de-roberta-sentence-transformer	0.85	Majority
	n-grams (1,5)	paraphrase-multilingual-mpnet-base-v2	0.9	Weighted majority
	POS-n-grams (1,5)			

After identifying the best hyperparameters through the grid search, these parameters are applied to the test-set. The model's predictions are then subjected to a comparative analysis, where the performance of different models or suggesters is compared. This comparison is critical for understanding the strengths and weaknesses of each approach.

The performance of each model is evaluated using precision, recall and F1 score across two settings:

- **Single-Task**: The evaluation focuses on extracting spans only, without considering the associated labels (Skill/Knowledge).
- **Multi-Task**: The evaluation not only considers correctly extracted spans but also the assigned labels; the individual label performance is then averaged across labels.

This dual setting follows the evaluation presented by Zhang, Jensen, Sonniks, et al. (2022) and allows for the assessment of each model's capabilities in handling both simple and more complex extraction tasks. The performance evaluation metrics will be explained in more detail in Chapter Evaluation Metrics.

The experiments are conducted in a controlled environment to ensure reproducibility. The following software and hardware configurations were used:

- **Python Version**: 3.11.4
- **SpaCy Version**: 3.6.1 (de_core_news_lg version 3.6.0 for POS tagging)
- **Operating System**: macOS 13.4 (ARM64 architecture)
- **GPU**: Apple M2 Max

6.2 Evaluation Metrics

Based on the findings of the state-of-the-art analysis, the evaluation metrics Precision, Recall, and F1 score will be used to evaluate and compare the performance of the different parameter combinations on the dev- and the test-set. See Formulas x and y for their formal definitions. In the context of span labelling, the metrics are defined using the following variables:

- True Positives (TP): The number of instances that are correctly identified by the algorithm as belonging to the correct class.
- False Positives (FP): The number of instances that are incorrectly identified as belonging to a class when they do not.
- False Negatives (FN): The number of instances that belong to a class but were missed by the algorithm.
- True Negatives (TN): The number of spans that are correctly identified as not belonging to a class.

Precision measures the accuracy of the positive predictions made by the model. It is defined as the ratio of true positive spans to the total number of spans predicted as positive. A higher precision indicates a lower rate of false positives. On the other hand, **Recall** measures the model's ability to correctly identify all relevant spans. It is defined as the ratio of true positive spans to the total number of actual positive spans. A higher recall indicates a lower rate of false negatives.

$$Precision = \frac{TP}{TP + FP}, \qquad Recall = \frac{TP}{TP + FN}$$

Formula 2: Definition of Precision and Recall

The **F1 score** is the harmonic mean of precision and recall, offering a balance between the two metrics. It is particularly useful when the class distribution is imbalanced, as it provides a single metric that considers both precision and recall.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Formula 3: Definition of F1 score

For the span labelling task, the metrics will be calculated similarly as was proposed by (Da San Martino et al., 2019; Pavlopoulos et al., 2021). The individual variables such as TP and FN are evaluated on the character level. Precision, Recall and F1 are then calculated for each job listing using the formulas above and then averaged across all listings. Following Pavlopoulos et al. (2021), in the case that there are no instances of a class in the job listing, the F1 score for that class will be set to 1 if also none are predicted, and to 0 if the prediction is not empty.

Additionally, the performance of the proposed models will be evaluated in two different settings (Zhang, Jensen, Sonniks, et al., 2022): a single-task and a multi-task setting, see Table 10 for a detailed example. For the single-task setting, the evaluation focuses on extracting spans only, without considering the associated labels. Because of this, a prediction of a knowledge span label for an annotated skill span, will still be counted as a TP.

Table 10: Overview of multi-task and single-task evaluation settings using example labels, S/K denotes a nested span

Evaluation

Multi-Task (Knowledge):	TN	TN	TN	TN	TN	TN	TN	TN	FN	TP	FP
Multi-Task (Skill):	TN	TN	FP	TP	TP	FN	TN	TN	FP	FP	TN
Single-Task:	TN	TN	FP	TP	TP	FN	TN	TN	TP	TP	FP

Labels

Prediction:	O	O	S	S	S	O	O	O	S	S/K	K
Annotation:	O	O	O	S	S	S	O	O	K	K	O

In the multi-task setting, the evaluation not only considers correctly extracted spans but also the assigned labels when calculating the performance metrics. The individual label performance is then averaged across labels using **macro-averaging** for each job listing. Macro-averaging is used in multi-class scenarios and calculates the precision,

recall, and F1 score for each label independently and then takes the arithmetic mean across all N-labels, where N is the number of labels:

$$Metric_{Macro\ Avg} = \frac{1}{N}\sum_{i=1}^{N} Metric_i$$

Formula 4: Definition of macro-average

Macro-averaging treats each label equally, regardless of its frequency in the dataset, making it particularly valuable when the class distribution is uneven. This approach ensures that the model's performance on less frequent labels is adequately represented in the overall metric.

To evaluate the impact of different language-specific chunking methods (O3) and the selection of different pre-trained transformer models (O4), a **one-sided ANOVA** (Analysis of Variance) is performed to test for statistical significance. The ANOVA test assesses whether the means of two or more groups defined by the independent variable are statistically different from the overall mean of the dependent variable. The null Hypothesis H_0 states that all group means are equal and there are no differences. If any of the group means significantly differ from the overall mean, the null hypothesis is rejected, indicating that at least one group is different from the others.

The ANOVA test uses the *F*-test to assess statistical significance. The *F*-test is a ratio of the variance between the group means (between-group variability) to the variance within the groups (within-group variability). If the variance within groups is smaller than the variance between groups, the *p*-value of the F-test will be higher, suggesting a greater likelihood that the observed differences between the group means are real and not due to chance. A p-value $p < 0.05$ is set as the benchmark for statistical significance.

6.3 Evaluation Results

The performance of the various parameter combinations of the skill labelling pipeline is evaluated using precision, recall and F1 score across two settings. In the initial phase, the evaluation is conducted within the context of a single task. This entails the extraction of spans, without reference to the associated labels (skill/knowledge). Subsequently, in the multi-task setting, the evaluation considers not only correctly extracted spans but also the assigned labels. The individual label performance is then macro-averaged across labels. In conclusion, the results are presented and discussed.

Additionally, to increase the readability of this chapter, the following abbreviations for the included transformer, chunker methods and short forms for the different domains are introduced models:

- Domains: Maintenance and Assembly (short: mech), Office (short: office), Computer Science (short: tech)
- Models: cross-en-de-roberta-sentence-transformer (CRS), jobGBERT (JGB), paraphrase-multilingual-mpnet-base-v2 (PM2),
- Chunker: POS-n-grams (POS), n-grams (NG), noun chunks (NC), noun-only (NO)

6.3.1 Single-Task Performance

The single-task performance is evaluated in accordance with the flowchart illustrated in Figure 38. In the single-task setting, the correct assignment of labels is not considered when calculating performance measures. The process is as follows: First, the optimal parameters for each chunker and embedding model are identified through a grid search of 108 parameter combinations, utilising the development set. Subsequently, the optimal parameter combination for each chunker and embedding model is used to predict the span labels on the test-set and the performance discussed.

Grid Search on Dev-Set

Figure 39 illustrates the outcomes of the grid search for each chunker, employing the combinations of models, matching thresholds, and classifier strategies. In terms of macro F1 score, the POS demonstrated the most favourable performance overall, with the NG chunker exhibiting a similarly high level of proficiency. It is unsurprising that the NO and NC methods exhibited a higher macro precision on average, given that only the statistically most likely POS combination (mainly nouns) was presented to the matching algorithm. The POS and the NG chunker demonstrated a high recall in certain settings, although at a lower precision, indicating an optimistic matching algorithm.

Table 11 shows the various methods of chunking, along with the parameter combinations that yielded the best results and the associated performance metrics. All variants employ the PM2 embedding model with distinct matching thresholds, either 0.8 or 0.85, and voting strategies, either confidence or majority voting. In the development set, the POS chunker once again demonstrated the highest macro F1, followed by the NC chunker. In general, the longer span lengths of the POS and NG chunker have a higher recall, paired with a lower precision, compared to the NO and NC chunker. This indicates an optimistic matching process. Nevertheless, the POS chunker exhibits higher precision than the NG chunker, indicating a potential advantage of balancing the inclusion of the most prevalent POS patterns (noun-only) with n-grams.

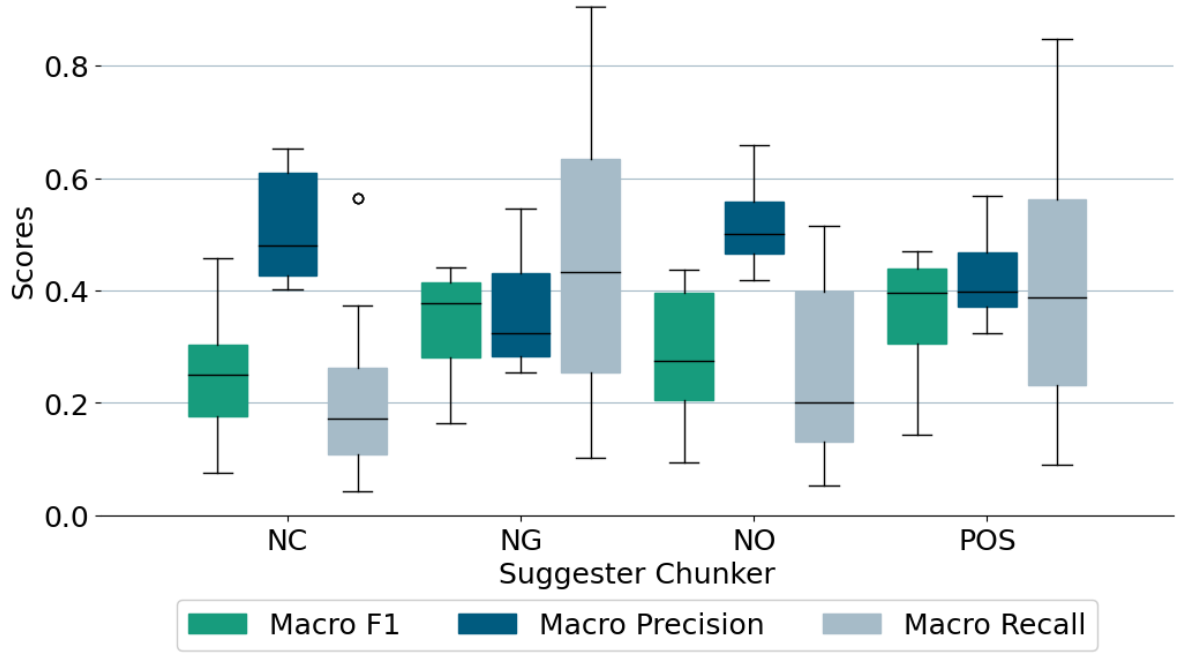

Figure 39: Performance of chunkers on the dev-set across all grid search parameters in the single task setting

Table 11: Best hyperparamter combinations for each chunker on the dev-set in the single-task setting

Suggester	Matcher		Classifier	Performance Metrics		
Chunker	**Model**	**Threshold**	**Voting Strategy**	**Macro F1**	**Macro Precision**	**Macro Recall**
POS	PM2	0.85	Confidence	**0.47**	0.37	0.71
NG	PM2	0.8	Confidence	0.44	0.33	**0.77**
NO	PM2	0.8	Majority	0.44	**0.42**	0.51
NC	PM2	0.85	Majority	0.46	**0.42**	0.56

The results of the grid search for the various models included in the analysis can be seen in Figure 40. While the PM2 model demonstrates the highest overall performance on the test-set, the CRS based on roBERTa exhibits the most consistent precision across all parameter settings. Conversely, the macro recall is typically inferior to that of the other two models. This is consistent with the observation of comparatively strict matching, which may favour shorter chunks provided by the suggester.

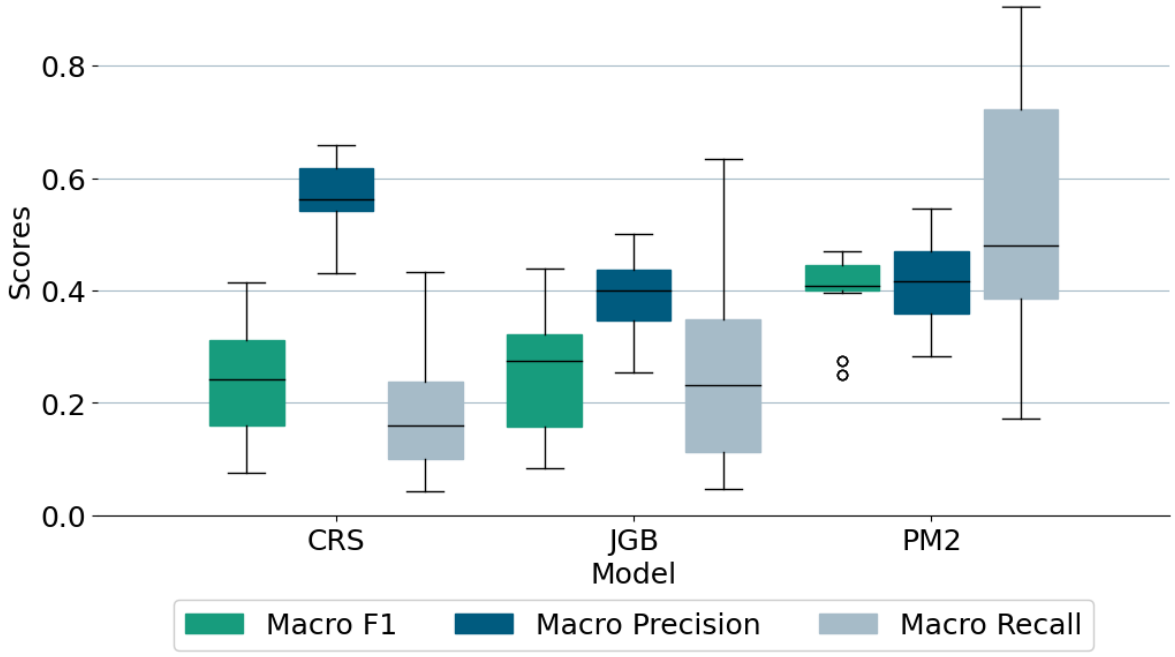

Figure 40: Performance of models on dev-set across all grid search parameters in single task setting

The optimal parameter configurations for each model are presented in Table 12. It is noteworthy that the majority of models achieve their optimal results when using the language-specific POS chunker. The more conservative CRS is the only model that is able to utilise the wider range of chunk suggestions provided by the NG chunker. The optimal matching threshold for the PM2 model is higher than for the other models, with a value of 0.85 compared to 0.8. This further emphasises the rather optimistic matching characteristics of the model. The CRS exhibits the lowest F1 score among the models, while simultaneously demonstrating the highest precision among the models in the development set. This further supports the hypothesis that the matching parameters are particularly strict. Once more, the combination of the PM2 model and POS chunker demonstrated the greatest performance in terms of F1 score and recall.

Table 12: Best hyperparameter combinations for each model on dev-set in single-task setting

Suggester	**Matcher**		**Classifier**	**Performance Metrics**		
Chunker	**Model**	**Threshold**	**Voting Strategy**	**Macro F1**	**Macro Precision**	**Macro Recall**
NG	CRS	0.8	Weighted majority	0.42	**0.43**	0.43
POS	JGB	0.8	Majority	0.44	0.38	0.56
POS	PM2	0.85	Confidence	**0.47**	0.37	**0.71**

Performance Evaluation on Test-Set

Subsequently, the optimal parameter combinations for each chunker and model are employed to predict the spans on the test-set. The results for the various chunker options are presented in Figure 41. The initial differences between the various chunker variants remain evident. Notably, the POS chunker once again achieved the highest macro F1 score (0.49), with the NC chunker following closely behind at 0.46. The longer span lengths of the POS and NG chunker have a higher recall, paired with a lower precision, compared to the NO and NC chunker. This indicates an optimistic matching process. With the POS chunker again showing higher precision than the NG chunker, it can be suggested that there is a benefit to be gained from a trade-off between suggesting only the most common POS pattern (nouns-only) and n-gram.

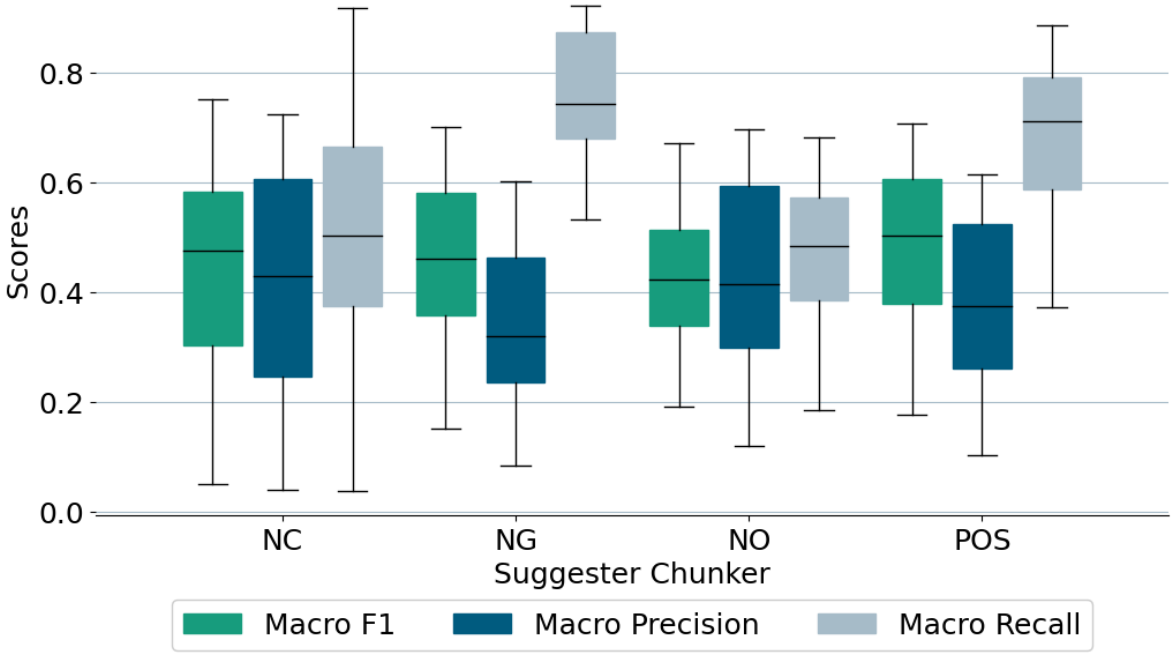

Figure 41: Performance of best-performing chunker settings on test-set in single-task setting

The detailed performance of the optimal chunking parameters on the test-set across different domains is presented in Table 13. The POS and NG chunker exhibited the highest average F1 scores. POS demonstrate a balanced performance across all domains, exhibiting particularly strong precision in the mechanical domain. Although the NG chunker has lower precision overall, it compensates with the highest recall scores, averaging 0.76. This makes it highly effective when the goal is to capture more relevant instances rather than to maximise precision. The Office domain consistently demonstrates the highest performance scores across all chunkers, particularly in terms of F1 and precision. This indicates that the pipelines are particularly well-suited to the linguistic characteristics and structural patterns observed in texts pertaining to the office domain. The Tech domain demonstrates the lowest performance across the majority of metrics, particularly in terms of precision. This suggests that the pipeline is

unable to effectively process the specific language and frequently changing terminology that is characteristic of the computer science domain, particularly in regard to the various programming languages and software frameworks that are required.

Table 13: Evaluation results of best chunker hyperparameters on test-set in single-task setting

Chun-ker	Macro F1				Macro Precision				Macro Recall			
	Mech	Office	Tech	Avg.	Mech	Office	Tech	Avg.	Mech	Office	Tech	Avg.
POS	**0.52**	**0.57**	0.36	**0.48**	**0.51**	0.47	0.30	0.39	0.75	0.73	0.55	0.68
NG	0.48	0.54	**0.37**	0.46	0.36	0.42	0.28	0.35	**0.80**	**0.80**	**0.68**	**0.76**
NO	0.47	0.49	0.32	0.43	0.43	**0.52**	**0.36**	**0.43**	0.54	0.49	0.38	0.47
NC	0.43	0.54	0.33	0.44	0.39	0.51	0.35	0.42	0.50	0.61	0.41	0.51

The impact of differently pretrained models, utilising their optimal parameter combinations identified through grid search, is evident, particularly in terms of recall, see Figure 42. While the F1 performance of the three models is distributed similarly, the combination of the POS chunker and PM2 once again demonstrates the highest F1 score (0.48) and particularly high recall (0.68). The CRS once more demonstrates the highest average precision (0.46) in comparison to JGB (0.41) and PM2 (0.39).

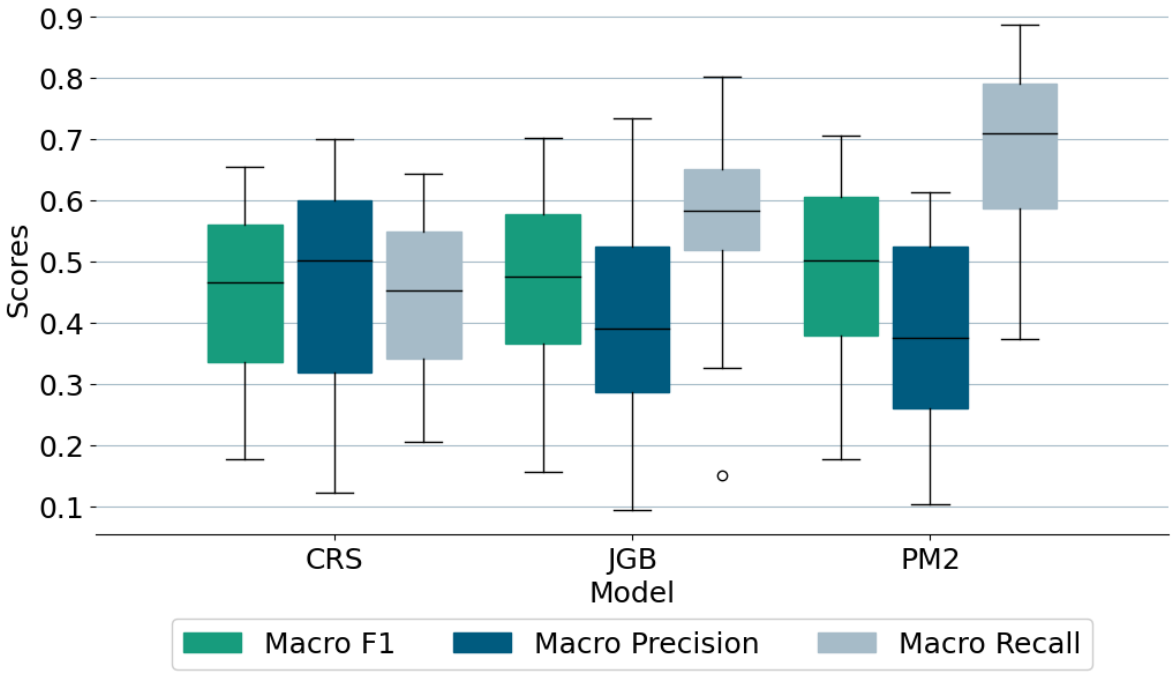

Figure 42: Performance of best-performing model settings on test-set in single-task setting

The detailed performance of the various models on the test-set across the different domains is presented in Table 14. Among the models, PM2 demonstrates consistent superiority in terms of Macro F1 (0.48) and Macro Recall (0.68), driven by particularly robust recall scores in the Mech (0.75) and Office domains (0.73). However, its precision remains lower across all domains, with an average of 0.39. JGB

demonstrates the most balanced performance between precision and recall, attaining an F1 score of 0.44, which falls between the scores of CRS (0.44) and PM2 (0.48). CRS once more demonstrates the highest precision (0.46), although this is accompanied by the lowest recall performance of the models (0.44). The Office domain consistently demonstrates the highest performance scores, particularly in terms of F1 and precision, indicating that the models are well-suited for the structured language found in office-related texts. Conversely, the Tech domain again exhibits the lowest performance across the majority of metrics, underscoring the difficulties associated with addressing specific and evolving terminology.

Table 14: Evaluation results of best model hyperparameters on test-set in single-task setting

Model	Macro F1				Macro Precision				Macro Recall			
	Mech	Office	Tech	Avg.	Mech	Office	Tech	Avg.	Mech	Office	Tech	Avg.
CRS	0.46	0.49	**0.39**	0.44	**0.47**	**0.53**	**0.38**	**0.46**	0.46	0.46	0.41	0.44
JGB	**0.52**	0.52	0.35	0.46	0.43	0.48	0.31	0.41	0.67	0.57	0.50	0.58
PM2	**0.52**	**0.57**	0.36	**0.48**	0.41	0.47	0.30	0.39	**0.75**	**0.73**	**0.55**	**0.68**

Significance Testing

Despite the POS chunker and the PM2-model attaining the highest F1 scores on both the development and test-sets, these results were not statistically significant when evaluated using one-way ANOVA. In contrast, significant differences were observed in recall between the models, particularly between POS and NG chunker in comparison to NC and NO methods. Furthermore, notable discrepancies were observed between the JGB and PM2 models and the CRS model. The null hypothesis of no difference was rejected, indicating that these models exhibited differential recall performance.

6.3.2 Multi-Task Performance

The performance of the multi-task setting is once more evaluated in accordance with the flowchart illustrated in Figure 39. In the context of the multi-task setting, the correct assignment of labels is a necessary component in the calculation of performance measures. Table 10, on page 73, provides an overview of the evaluation method employed for the multi-task setting. Firstly, the optimal parameters for each chunker and embedding model are identified through a grid search of 108 parameter combinations, utilising the development set. Subsequently, the optimal parameter combination for each chunker and embedding model is employed to predict the span labels on the test-set.

Grid Search on Dev-Set

Figure 43 illustrates the outcomes of the grid search for distinct chunkers in conjunction with diverse models, thresholds, and classifier strategies. The general trend observed in the single-task setting persists in the multi-task setting. The POS chunker, on average, achieves the highest macro F1 score, closely followed by the NG chunker. The NO and NC methods demonstrate superior macro precision, as they present the most probable POS combinations (predominantly nouns) to the matching algorithm. The NG and POS chunker demonstrate high recall in specific contexts, although this is accompanied by lower precision, indicating an optimistic matching approach.

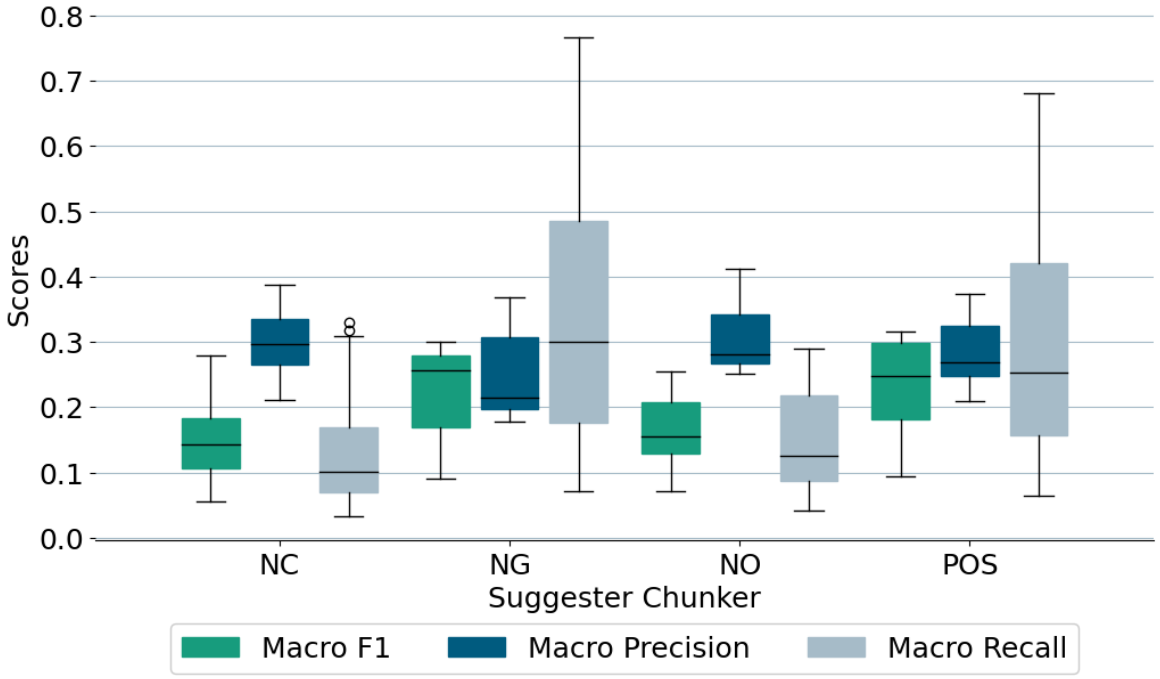

Figure 43: Performance of chunkers on dev-set across all grid search parameters in multi-task setting

The optimal parameter configurations for each chunker in terms of macro F1 score are presented in Table 15. All chunker utilise the PM2 embedding model in their optimal configurations, with varying matching thresholds between 0.8 and 0.85. The NG and POS chunker employ confidence voting, indicating that longer contexts facilitate more precise labelling than shorter variants. In the development set, the POS chunker once more attained the highest macro F1, with the NC chunker following closely behind. As in the single-task setting, the longer span lengths of the POS and NG chunker result in lower precision accompanied by higher recall. The POS chunker exhibits the highest recall and the lowest precision, again indicating a potential benefit of balancing the inclusion of the most common POS pattern (noun-only) with those of n-grams.

Table 15: Best hyperparameter combinations for each chunker on dev-set in multi-task setting

Suggester	Matcher		Classifier	Performance Metrics		
Chunker	Model	Threshold	Voting Strategy	Macro F1	Macro Precision	Macro Recall
POS	PM2	0.85	Confidence	**0.31**	0.25	**0.50**
NG	CRS	0.8	Confidence	0.30	**0.31**	0.32
NO	PM2	0.8	Weighted majority	0.26	0.26	0.29
NC	PM2	0.8	Weighted majority	0.28	0.27	0.33

The results of the grid search for the various models included in the analysis can be seen in Figure 44. The PM2 model demonstrates consistent F1 score performance across settings, with a notable high recall, which aligns with its optimistic matching tendency. In contrast, the roBERTa-based CRS model demonstrates the highest precision overall, exhibiting minimal sensitivity to the diverse parameters employed in the pipeline.

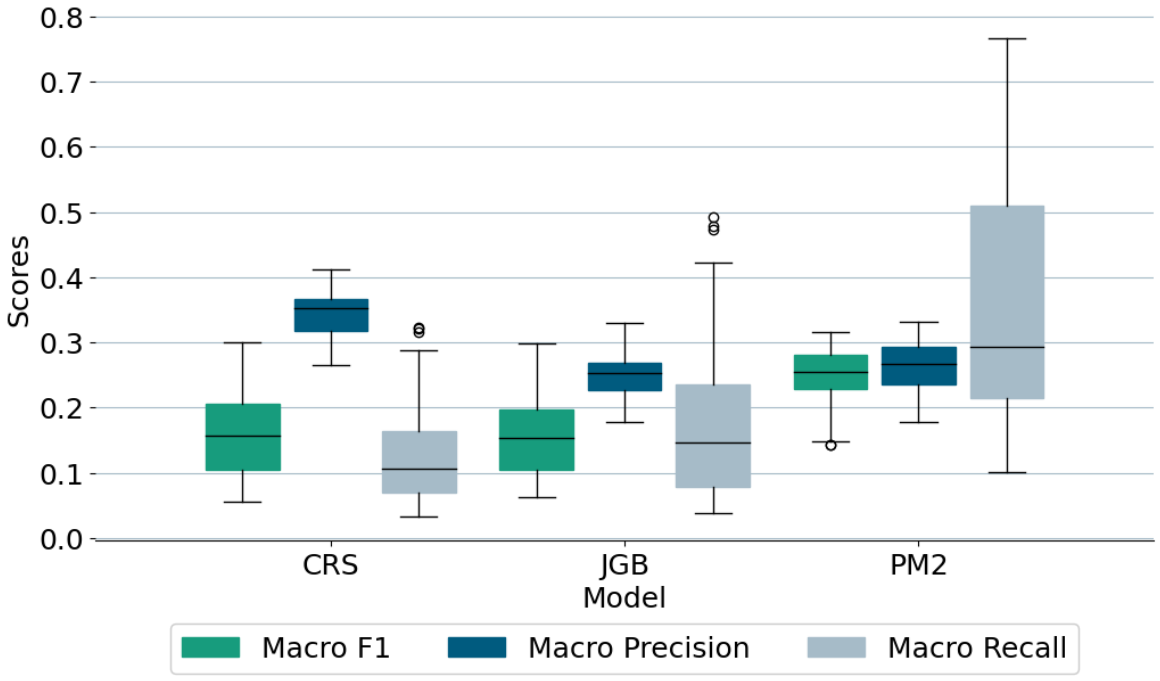

Figure 44: Performance of models on dev-set across all grid search parameters in multi-task setting

The optimal parameter configurations for each model are shown in Table 16. It is noteworthy that the majority of models achieve their optimal results when utilising the language-specific POS chunker. Similarly, as observed in the single-task setting, the CRS employs the broader chunk suggestions provided by the NG chunker, resulting in a macro F1 score of 0.30 with a threshold of 0.8. The optimal matching threshold for the PM2 model is higher than for the other models, at 0.85 compared to 0.8. This highlights the model's optimistic matching characteristics. The CRS exhibits the lowest

F1 score among the models (0.30), while achieving the highest precision (0.31) on the development set, indicating strict matching parameters. In contrast, the PM2 model, in conjunction with the POS chunker, exhibits the highest F1 score (0.31) and recall (0.50), thereby corroborating its robust performance in less stringent matching scenarios.

Table 16: Best hyperparameter combinations for each model on dev-set in multi-task setting

Suggester	Matcher		Classifier	Performance Metrics		
Chunker	**Model**	**Threshold**	**Voting Strategy**	**Macro F1**	**Macro Precision**	**Macro Recall**
NG	CRS	0.8	Confidence	0.30	**0.31**	0.32
POS	JGB	0.8	Majority	0.30	0.26	0.42
POS	PM2	0.85	Confidence	**0.31**	0.25	**0.50**

Performance Evaluation on Test-Set

Figure 45 illustrates the performance of the optimal parameter combinations for each chunker in predicting the spans on the test-set. In contrast to the single-task setting, where the initial differences between the chunker variants were more pronounced, the multi-task setting demonstrates a more equal distribution of macro F1. It is noteworthy that while the POS and NG chunker exhibit a similar F1 score, with the NG chunker demonstrating higher precision but lower recall than the POS chunker. This is due to the fact that the NG chunker utilises the CRS, which has demonstrated the capacity to produce higher precision and lower recall overall, in comparison to the PM2 employed by the POS chunker.

Upon detailed examination of the performance of the optimal chunking parameters across diverse domains, as illustrated in Table 17, it becomes evident that POS and NG chunking consistently demonstrate the highest F1 scores in the multi-task settings. The POS chunker achieves the highest average F1 score (0.31), demonstrating balanced performance across all domains and particularly strong precision in the Office domain (0.30). This aligns with their single-task results, where they exhibited superior precision, especially in the mechanical domain. However, while the NG chunker achieves the same average F1 score (0.31) in the multi-task setting, it displays a more balanced precision (0.33) and recall (0.33) across all domains. This contrasts with its single-task performance, where it had lower precision but compensated with the highest recall scores (averaging 0.76).

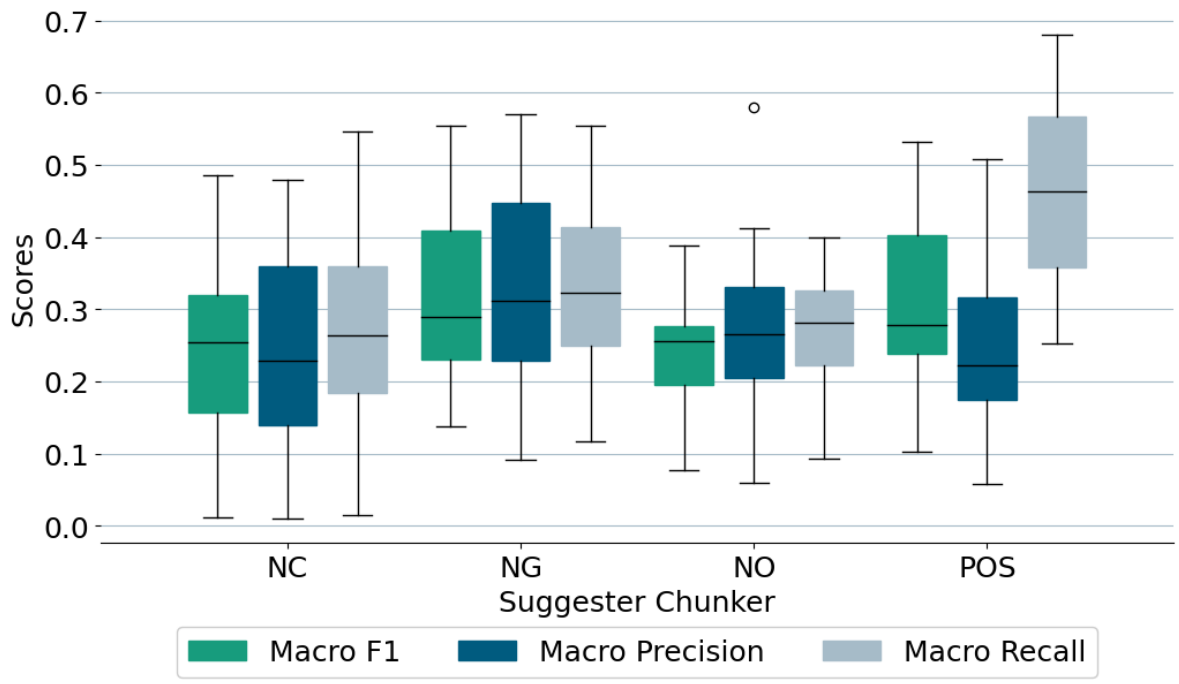

Figure 45: Performance of best-performing chunker settings on test-set in multi-task setting

As in the single-task setting, the Office domain consistently yields the highest performance, particularly in terms of F1 and precision. This suggests that the pipelines are particularly well-suited to the structured language of office-related texts. In contrast, the Tech domain consistently exhibits the lowest performance across most metrics in both settings, particularly in precision (0.20 in the multi-task setting), highlighting the challenges the pipelines face in dealing with the rapidly evolving terminology of the computer science field. The results demonstrate that, while both chunker perform well across settings, their relative strengths exhibit slight variation between the single-task and multi-task approaches, particularly with regard to their respective handling of precision and recall balances.

Table 17: Evaluation results of best chunker hyperparameters on test-set in multi-task setting

Chunker	Macro F1				Macro Precision				Macro Recall			
	Mech	Office	Tech	Avg.	Mech	Office	Tech	Avg.	Mech	Office	Tech	Avg.
POS	**0.33**	**0.36**	0.25	**0.31**	0.26	0.30	0.20	0.25	**0.51**	**0.49**	**0.41**	**0.47**
NG	0.30	0.33	**0.30**	**0.31**	**0.33**	**0.37**	**0.29**	**0.33**	0.33	0.33	0.33	0.33
NO	0.27	0.27	0.17	0.24	0.28	0.31	0.23	0.27	0.29	0.26	0.20	0.25
NC	0.20	0.30	0.24	0.25	0.21	0.31	0.24	0.25	0.22	0.32	0.27	0.33

A comparison of the multi-task performance of differently pretrained models on the test-set, using their optimal parameter combinations identified through grid search, reveals notable differences in terms of recall and precision, see Figure 46. The F1-performance of the three models is distributed similarly, but the precision of CRS is

significantly higher than that of the other two models. With regard to recall, the PM2 and JGB demonstrate a comparable degree of performance.

In general, the performance trends of the individual models are in close alignment with those observed in the single-task setting, although they are slightly lower on average. This suggests that there are no notable discrepancies in the manner in which labels are assigned between the models with regard to skill and knowledge labels. The primary discrepancies emerge during the matcher phase, wherein the determination of whether a label should be assigned is made, as opposed to the classifier phase, where the decision regarding the assignment of a skill or knowledge label is made.

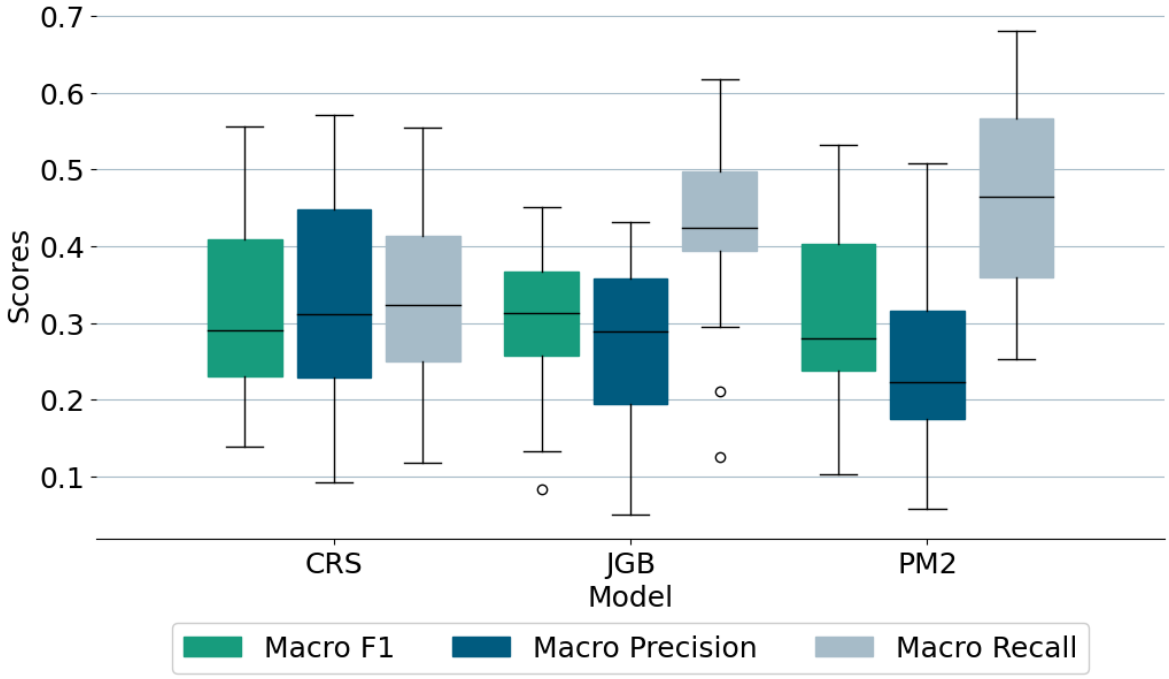

Figure 46: Performance of best-performing model settings on test-set in multi-task setting

The detailed performance of the various models across the three domains is shown in Table 18. In the multi-task setting, CRS and PM2 demonstrate the most optimal performance on average in terms of Macro F1 (0.31), driven by a high average recall for PM2 and a balanced performance between precision (0.33) and recall (0.33) for CRS. This is consistent with the results obtained in the single-task setting, where a similar balance was achieved with an F1 score of 0.48 and 0.44, respectively. Once more, JGB demonstrates the most balanced performance between precision (0.27) and recall (0.43). In the multi-task setting, PM2 demonstrates robust recall, particularly in the Mech (0.51) and Office (0.49) domains. This aligns with its single-task results, where it also exhibited superior recall, achieving an average of 0.68. However, in both settings, PM2's precision remains lower across all domains, with an average of 0.25 in the multi-task setting and 0.39 in the single-task setting. This highlights a consistent trade-off between precision and recall. CRS maintains the highest precision in the multi-task setting (0.33), although this is at the cost of lower recall (0.33). This is similar

to its single-task performance, where it also achieved the highest precision (0.46), but the lowest recall (0.44). The trends observed in the different domains remain consistent, although in the multi-task setting, the differences between the Mech and Office domains become much smaller, indicating a greater degree of label misassignment in the Office context compared to the Mech domain.

Table 18: Evaluation results of best model hyperparameters on test-set in multi-task setting

Model	Macro F1				Macro Precision				Macro Recall			
	Mech	Office	Tech	Avg.	Mech	Office	Tech	Avg.	Mech	Office	Tech	Avg.
CRS	0.30	0.33	**0.30**	**0.31**	**0.33**	**0.37**	**0.29**	**0.33**	0.33	0.33	0.33	0.33
JGB	**0.33**	0.33	0.24	0.30	0.28	0.32	0.21	0.27	0.48	0.41	0.39	0.43
PM2	**0.33**	**0.36**	0.25	**0.31**	0.26	0.30	0.20	0.25	**0.51**	**0.49**	**0.41**	**0.47**

Significance Testing

As observed in the single-task case, the POS chunker and the PM2-model achieved the highest F1 scores on the development set. However, in the multi-task setting, their F1 performance results were similar to those of the combination of CRS and the NG chunker on the test-set. As a consequence, no definitive statistical significance can be attributed to the approaches in question with regard to the macro F1 results when evaluated using one-way ANOVA. In contrast to the single-task setting, significant differences were observed in recall between the chunking methods, particularly between POS and the other chunking methods, namely NG, NC and NO chunker. Furthermore, notable discrepancies were observed between the recall performance of JGB and PM2 models in comparison to the CRS model. The null hypothesis of no difference was thus rejected, suggesting that these models exhibited differential recall capabilities. It is important to note that the differences in recall between the POS and the NG chunker in the multitask setting are likely due to the different models (CRS vs. PM2) employed with varying parameter combinations.

6.3.3 Summary of Evaluation Results

This chapter assessed the performance of the skill labelling pipeline using precision, recall, and F1 score in both single-task and multi-task settings. In the single-task setting, the objective is to extract spans without considering the associated labels. In contrast, the multi-task setting encompasses both correctly extracted spans and their assigned labels, with performance macro-averaged across individual labels.

In the single-task evaluation, a grid search over 108 parameter combinations has been employed to identify the optimal hyperparameters within the skill extraction pipeline for each chunker and embedding model. The POS chunker demonstrated the highest F1

score, indicating a greater emphasis on recall than precision. In contrast, models such as PM2 exhibited superior recall but lower precision. Domain-specific performance revealed that the Office domain demonstrated the highest precision, whereas the Tech domain exhibited the lowest due to the rapid evolution of terminology and intricate linguistic nuances, particularly in the context of programming languages and software frameworks. This illustrates a significant drawback of the pipeline, which is dependent on a fixed ontology that is unable to adapt to the changing terminology that is characteristic of evolving domains.

The multi-task evaluation exhibited comparable performance trends to those observed in the single-task setting, with POS and PM2 demonstrating high recall rates. However, all performance measures, including precision and recall, exhibited reduced levels of achievement in the multi-task setting. This indicates that there are no substantial discrepancies in the manner of labelling between the models, whether pertaining to skills or knowledge. It can be inferred that the primary challenges lie in the initial span extraction process rather than in the labelling itself. The uniformity of label assignment suggests that once a label is assigned, it is likely to be correct, indicating that the primary challenge lies in distinguishing between spans with no label and those that are labelled.

The results of the statistical significance tests indicated that there were no significant differences in F1 scores between the various chunker and model combinations. This suggests that the overall performance was stable across settings. However, significant differences in recall were observed, particularly between the POS and noun-based chunkers, which highlighted their differential capabilities in span extraction.

In conclusion, the chapter demonstrates that while the language-specific POS chunker and PM2 model consistently achieved the best performance across both single-task and multi-task settings, there are evident trade-offs between precision and recall, influenced by domain-specific language. The statistical tests highlighted the limitations of the pipeline in handling dynamic fields such as technology, where the reliance on static ontologies hinders adaptability. This reinforces the importance of balancing these metrics in practical applications of the skill labelling pipeline.

7 Conclusion and Outlook

The final chapter of this thesis summarises the results, focusing on conclusively answering the research questions. It also offers a critical analysis of the limitations of the approaches and results and suggests further possible steps for future research.

7.1 Summary of Findings

Based on the current state-of-the-art, there were two main problem statements to be solved with the developed artefacts of this thesis:

- P1: There are currently no works, to the best of the author's knowledge, that extract skills from German job listings, without task specific pre-training, while also creating a mapping to a standardised skill taxonomy.
- P2: At the same time, no publicly available benchmarking data set exists for evaluating and comparing the performance of different skill extraction and classification approaches.

The first problem statement, P1, is directly addressed by implementing a skill extraction and classification pipeline specifically adapted for German job listings. The pipeline employs preexisting state-of-the-art transformer models such as jobGBERT (Gnehm, Bühlmann, & Clematide, 2022), which were fine-tuned to handle German language intricacies and job-specific terminology. Furthermore, the pipeline integrates the ESCO taxonomy as the reference framework for mapping extracted skills, creating a standardised linkage between identified skills and a recognised taxonomy. This solves the gap identified in P1 by providing a systematic approach to extracting and classifying skills in German job listings with proper taxonomy alignment and without the need for task-specific pretraining.

To address P2, the thesis developed and annotated a new benchmarking dataset of 60 German job listings. This dataset was created through a rigorous data acquisition process involving a diverse collection of job listings across multiple domains (e.g., Maintenance and Assembly, Office, Computer Science). The annotation process followed meticulously developed guidelines that ensured consistency and reliability in labelling skills and knowledge components. The resulting dataset provides an annotated German dataset specifically designed for skill extraction and classification tasks. This dataset serves as a valuable resource for evaluating and comparing the performance of different extraction methods, thus filling the critical gap of P2 by enabling reproducibility and benchmarking of future approaches. Additionally, the adapted annotation guidelines serve as the basis for a future extension of the existing data set, further supporting the scientific process.

Following the problem statements, three research questions were examined in the context of this thesis:

- RQ1: To what extent do current state-of-the-art skill extraction and classification methods perform on German job listing data sets in terms of the performance measures?
- RQ2: What is the impact of language-specific pre-processing on the performance measures, compared to the current state-of-the-art?
- RQ3: How does the selection of different pre-trained models impact the overall extraction and classification performance?

In order to respond to the initial research question (RQ1), a comprehensive and systematic analysis of the state-of-the-art was undertaken. This was done in order to identify the methods currently in use and which could be employed as a basis for the performance of skill extraction and classification methods on German job listings. In view of the paucity of available training data, embedding look-ups were employed and integrated into the skill extraction and classification pipeline that was developed. A benchmarking dataset was constructed using specifically adapted annotation guidelines to evaluate the performance of different state-of-the-art transformer and chunking methods. The evaluation was conducted in two settings: single-task and multi-task. It was demonstrated that the performance on the single-task dataset was comparable to that of other non-English skill extraction and classification methods, with an F1 score of 0.48. In the multi-task setting, an F1 score of 0.31 was achieved. The experiments demonstrated that these models exhibited consistent performance in span extraction tasks. However, the results varied by domain due to the evolving nature of job-specific terminology, particularly in technology sectors.

To address RQ2, the structure of skill labels in the ESCO taxonomy and the annotated data set were examined. A custom chunker, the POS chunker, was designed to integrate the most prevalent skill and knowledge patterns into the chunking process. The language-specific chunker exhibited the highest F1 score across both settings, demonstrating its potential compared to other traditional chunking methods. The POS chunker demonstrated superior recall rates, indicating that it captured a more expansive set of relevant skill spans than more general methods. This was particularly evident in the higher F1 scores achieved during the single-task evaluations, where language-specific adaptations were beneficial in the German context. The thesis emphasises the importance of language-specific models and pre-processing techniques to optimise performance when dealing with non-English texts.

Finally, to address RQ3, three state-of-the-art transformer and embedding models were employed to assess their influence on the pipeline's overall extraction and classification performance. It was demonstrated that the distinctive attributes of the models had a considerable impact on the balance of precision and recall, as well as their sensitivity to the various hyperparameters of the pipeline. This is a crucial insight to consider when balancing the extraction process for either. It was not established

that the domain-adapted JobGBERT model outperformed the other examined multilingual models that had not been trained explicitly on German job listing data.

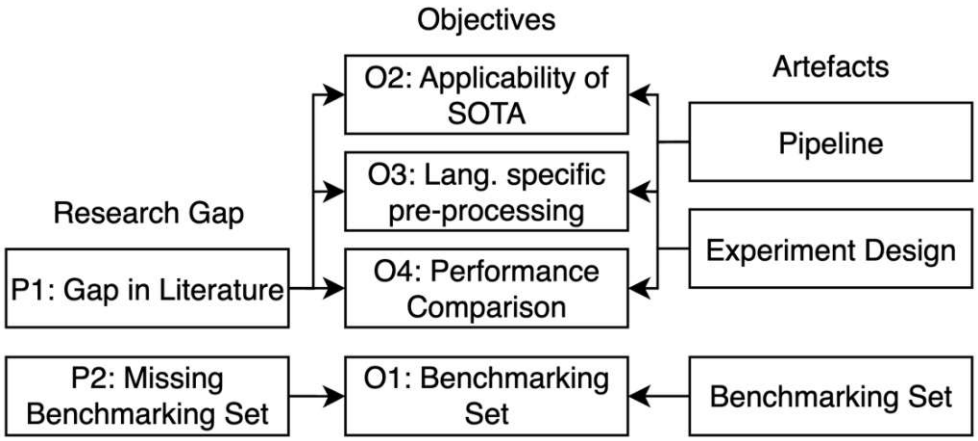

Figure 47: Schematic overview of developed artefacts and their corresponding objectives

The artefacts developed in the thesis—namely, the customised skill extraction pipeline using German-specific adaptations, the experiment design, and the creation of an annotated benchmarking dataset—effectively address the identified gaps by providing tools and resources that did not previously exist for German job listings. These contributions facilitate more accurate skill extraction and classification, benchmark performance, and advance the field toward standardised and reproducible research in this area.

7.2 Limitations

In the following, the limitations of each developed artefact and their impact on the results will be discussed.

Benchmarking Dataset and Fine Tuning

The primary limitation is the size of the annotated dataset, consisting of only 60 job listings across three domains. Although carefully curated, the limited size affects the generalizability and robustness of the evaluation of the state-of-the-art methods, as performance metrics might vary with a larger or more diverse dataset. This limitation also meant that no additional training or fine-tuning of pre-trained models was conducted due to the limited labelled training data and computational constraints. This decision restricts the potential performance improvements that could have been achieved through domain-specific adaptations of the models.

Experimental Design

The experimental design involved grid searches over selected hyperparameters and model configurations, but this also meant not all possible combinations or a larger selection of models could be explored. Consequently, the reported results represent

only a subset of potential outcomes, and further optimisation could yield better-performing configurations. The thesis did not implement local optimisation techniques, such as optimising the matching threshold, but instead relied on standard grid search methods that may not capture the finer nuances required for optimal performance.

Computational time, which was significant, especially during extensive grid searches and the embedding processes involving multiple models, was not factored into the approach's evaluation. This limits the practical applicability of the approach in real-time settings, which is crucial for operational environments like job-matching platforms.

Developed Extraction and Classification Pipeline

Finally, the approach relies heavily on a predefined static taxonomy (ESCO) for skill classification, which poses challenges in rapidly evolving fields. This dependency on a fixed taxonomy limits the model's ability to adapt to new or domain-specific terms, affecting the relevance and accuracy of extracted skills. The models showed performance variability across different domains, particularly in fast-evolving sectors such as IT, where outdated or overly static taxonomies contribute to performance degradation. The current approach lacks adaptability, as it cannot dynamically update or learn from new data without an update to the ESCO taxonomy, which is a significant limitation in fields with frequently changing skill requirements. It also means that the approach and the results may not directly translate to other taxonomies.

These limitations underscore the need for larger datasets, domain-specific training, and enhanced optimisation strategies to improve the effectiveness of skill extraction methods on German job listings. Addressing these constraints in future works would lead to more robust, adaptable, and computationally efficient models, enhancing their applicability in diverse, real-world settings.

7.3 Outlook and Future Work

This thesis lays the groundwork for utilising new technologies in the skill extraction and classification for the German language. These findings can be applied in field of job market monitoring, uncovering new trends in skill demand trends over time, or other industry and real-world applications, such as utilising the skill extraction process for providing better suitable job offers to job applicants or vice versa. Additionally, the applied methodology can be scaled to new languages or new document types, such as work instructions or internal job descriptions. This has the potential offer new insights into current and future skill demands in the industry and help HR departments in their internal and external hiring process, as well as provide a basis for upskilling programs, aligning strategic and operational competency management practices.

The thesis also identifies several areas for future research that could enhance the performance and applicability of the proposed pipeline for skill extraction and classification from German job listings. A primary objective is to expand the benchmarking dataset to include a more extensive and diverse range of job listings from various industries. This will enhance the generalisability of the findings and facilitate more comprehensive evaluations across different domains. The robustness and reliability of the dataset could be further enhanced by increasing the number of annotation rounds and involving a greater number of annotators.

A larger data set would also open up new avenues for model training, for example, by employing a pre-trained classifier within the classification module or introducing a classification layer to the utilised transformer models.

Another crucial avenue for future research is the integration of adaptive or dynamic taxonomies to overcome the constraints of static ontologies like ESCO. These ontologies often prove inadequate in keeping pace with the evolving skills required in dynamic fields such as IT and data science. Further work could investigate the incorporation of other taxonomies, the utilisation of new data sources, or the deployment of crowdsourced updates with a view to more accurately reflecting emerging skill trends.

An important next step to increase the impact of the proposed methodology would be to evaluate the developed extraction and classification pipeline in real-world applications, such as job-matching platforms, HR management tools, or educational recommendation systems. Such evaluations would provide valuable insights into the methods' practical usability, allow for refinements and validation based on human-in-the-loop user feedback, and assess the pipeline's performance in operational settings.

Finally, the incorporation of other ML models, such as a pre-classification step, has the potential to refine input processing, reduce false positives, and guarantee that only sentences deemed likely to contain pertinent skills are taken into consideration.

In conclusion, addressing these future directions—namely, exploring applicability to new languages and input data, expanding datasets, performing additional model training, integrating adaptive taxonomies, and evaluating in real-world scenarios—would enhance the robustness, adaptability, and applicability of the automated skill extraction and classification pipeline, particularly in the context of German job listings.

8 Bibliography

Alyafeai, Z., AlShaibani, M. S., & Ahmad, I. (2020). *A Survey on Transfer Learning in Natural Language Processing* (arXiv:2007.04239). arXiv. http://arxiv.org/abs/2007.04239

Ansari, F., Kohl, L., & Sihn, W. (2023). A competence-based planning methodology for optimizing human resource allocation in industrial maintenance. *CIRP Annals*, *72*(1), 389–392. https://doi.org/10.1016/j.cirp.2023.04.050

Ao, Z., Horváth, G., Sheng, C., Song, Y., & Sun, Y. (2023). Skill requirements in job advertisements: A comparison of skill-categorization methods based on wage regressions. *Information Processing & Management*, *60*(2), 103185. https://doi.org/10.1016/j.ipm.2022.103185

Artstein, R. (2017). Inter-annotator Agreement. In N. Ide & J. Pustejovsky (Eds.), *Handbook of Linguistic Annotation* (pp. 297–313). Springer Netherlands. https://doi.org/10.1007/978-94-024-0881-2_11

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). *Enriching Word Vectors with Subword Information* (arXiv:1607.04606). arXiv. http://arxiv.org/abs/1607.04606

Chan, B., Schweter, S., & Möller, T. (2020). *German's Next Language Model* (arXiv:2010.10906). arXiv. http://arxiv.org/abs/2010.10906

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step data mining guide*. https://www.semanticscholar.org/paper/CRISP-DM-1.0%3A-Step-by-step-data-mining-guide-Chapman-Clinton/54bad20bbc7938991bf34f86dde0babfbd2d5a72

Cheng, Q., Zhu, Y., Song, J., Zeng, H., Wang, S., Sun, K., & Zhang, J. (2021). Bert-Based Latent Semantic Analysis (Bert-LSA): A Case Study on Geospatial Data Technology and Application Trend Analysis. *Applied Sciences*, *11*(24), 11897. https://doi.org/10.3390/app112411897

Chung, H. W., Févry, T., Tsai, H., Johnson, M., & Ruder, S. (2020). *Rethinking embedding coupling in pre-trained language models* (arXiv:2010.12821). arXiv. http://arxiv.org/abs/2010.12821

Da San Martino, G., Yu, S., Barrón-Cedeño, A., Petrov, R., & Nakov, P. (2019). Fine-Grained Analysis of Propaganda in News Article. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5635–5645. https://doi.org/10.18653/v1/D19-1565

Decorte, J.-J., Van Hautte, J., Deleu, J., Develder, C., & Demeester, T. (2022). Design of Negative Sampling Strategies for Distantly Supervised Skill Extraction. In *CEUR Workshop Proceedings* (Vol. 3218). https://www.scopus.com/inward/record.uri?eid=2-s2.0-85139595469&partnerID=40&md5=7fa5e0b961e3c4990b31e4ea5db5ddcd

Decorte, J.-J., Van Hautte, J., Demeester, T., & Develder, C. (2021). *JobBERT: Understanding Job Titles through Skills* (arXiv:2109.09605). arXiv. http://arxiv.org/abs/2109.09605

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

European Commission. (2017). *ESCO handbook: European skills, competences, qualifications and occupations.* Publications Office. https://data.europa.eu/doi/10.2767/934956

European Commission. (2022a). *About ESCO*. https://esco.ec.europa.eu/en/about-esco

European Commission. (2022b). *ESCO version v1.1.1* [Dataset]. https://esco.ec.europa.eu/en/use-esco/download

European Commission. (2023a). *Definition Competence*. ESCOpedia. https://esco.ec.europa.eu/en/about-esco/escopedia/escopedia/competence

European Commission. (2023b). *Definition Knowledge*. ESCOpedia. https://esco.ec.europa.eu/en/about-esco/escopedia/escopedia/knowledge

European Commission. (2023c). *Definition Skill*. ESCOpedia. https://esco.ec.europa.eu/en/about-esco/escopedia/escopedia/skill

European Commission. (2023d). *Occupations pillar*. ESCOpedia. https://esco.ec.europa.eu/en/about-esco/escopedia/escopedia/occupations-pillar

European Commission. (2023e). *Qualifications and ESCO*. ESCOpedia. https://esco.ec.europa.eu/en/about-esco/escopedia/escopedia/qualifications-and-esco

European Commission. (2023f). *Skills pillar*. ESCOpedia. https://esco.ec.europa.eu/en/about-esco/escopedia/escopedia/skills-pillar

Fareri, S., Melluso, N., Chiarello, F., & Fantoni, G. (2021). SkillNER: Mining and mapping soft skills from any text. *Expert Systems with Applications*, *184*, 115544. https://doi.org/10.1016/j.eswa.2021.115544

Ferrone, L., & Zanzotto, F. M. (2020). Symbolic, Distributed, and Distributional Representations for Natural Language Processing in the Era of Deep Learning: A Survey. *Frontiers in Robotics and AI*, *6*, 153. https://doi.org/10.3389/frobt.2019.00153

Firth, J. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis*, 10–32.

Fleiss, J. L., & Cohen, J. (1973). The Equivalence of Weighted Kappa and the Intraclass Correlation Coefficient as Measures of Reliability. *Educational and Psychological Measurement*, *33*(3), 613–619. https://doi.org/10.1177/001316447303300309

Fu, J., Huang, X., & Liu, P. (2021). *SpanNER: Named Entity Re-/Recognition as Span Prediction* (arXiv:2106.00641). arXiv. https://doi.org/10.48550/arXiv.2106.00641

Gnehm, A.-S., Bühlmann, E., Buchs, H., & Clematide, S. (2022). Fine-Grained Extraction and Classification of Skill Requirements in German-Speaking Job Ads. In *NLPCSS 2022—5th Workshop on Natural Language Processing and Computational Social*

Science ,NLP+CSS, Held at the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022 (pp. 14–24). https://www.scopus.com/inward/record.uri?eid=2-s2.0-85154569390&partnerID=40&md5=749bddd84e60db193aad9f4d9e8882bd

Gnehm, A.-S., Bühlmann, E., & Clematide, S. (2022). Evaluation of Transfer Learning and Domain Adaptation for Analyzing German-Speaking Job Advertisements. *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 3892–3901. https://aclanthology.org/2022.lrec-1.414

Gnehm, A.-S., & Clematide, S. (2020). Text Zoning and Classification for Job Advertisements in German, French and English. *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, 83–93. https://doi.org/10.18653/v1/2020.nlpcss-1.10

Goyal, A., Gupta, V., & Kumar, M. (2018). Recent Named Entity Recognition and Classification techniques: A systematic review. *Computer Science Review*, *29*, 21–43. https://doi.org/10.1016/j.cosrev.2018.06.001

Grüger, J., & Schneider, G. (2019). Automated Analysis of Job Requirements for Computer Scientists in Online Job Advertisements: *Proceedings of the 15th International Conference on Web Information Systems and Technologies*, 226–233. https://doi.org/10.5220/0008068202260233

Gurcan, F., & Cagiltay, N. E. (2019). Big Data Software Engineering: Analysis of Knowledge Domains and Skill Sets Using LDA-Based Topic Modeling. *IEEE Access*, *7*, 82541–82552. https://doi.org/10.1109/ACCESS.2019.2924075

Harris, Z. S. (1954). Distributional Structure. *WORD*, *10*(2–3), 146–162. https://doi.org/10.1080/00437956.1954.11659520

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition* (arXiv:1512.03385). arXiv. http://arxiv.org/abs/1512.03385

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*(6245), 261–266. https://doi.org/10.1126/science.aaa8685

Howard, J., & Ruder, S. (2018). *Universal Language Model Fine-tuning for Text Classification* (arXiv:1801.06146). arXiv. http://arxiv.org/abs/1801.06146

Khaouja, I., Kassou, I., & Ghogho, M. (2021). A Survey on Skill Identification From Online Job Ads. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2021.3106120

Khaouja, I., Mezzour, G., & Kassou, I. (2021). Unsupervised Skill Identification from Job Ads. *2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI)*, 147–151. https://doi.org/10.1109/IRI51335.2021.00026

Konstantinidis, I., Maragoudakis, M., Magnisalis, I., Berberidis, C., & Peristeras, V. (2022). Knowledge-driven Unsupervised Skills Extraction for Graph-based Talent Matching. *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*, 1–7. https://doi.org/10.1145/3549737.3549769

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text Classification Algorithms: A Survey. *Information*, *10*(4), Article 4. https://doi.org/10.3390/info10040150

Landis, J. R., & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, *33*(1), 159. https://doi.org/10.2307/2529310

Le, Q. V., & Mikolov, T. (2014). *Distributed Representations of Sentences and Documents* (arXiv:1405.4053). arXiv. http://arxiv.org/abs/1405.4053

le Vrang, M., Papantoniou, A., Pauwels, E., Fannes, P., Vandensteen, D., & De Smedt, J. (2014). ESCO: Boosting Job Matching in Europe with Semantic Interoperability. *Computer*, *47*(10), 57–64. Computer. https://doi.org/10.1109/MC.2014.283

Lovaglio, P. G., Cesarini, M., Mercorio, F., & Mezzanzanica, M. (2018). Skills in demand for ICT and statistical occupations: Evidence from web-based job vacancies. In *Statistical Analysis and Data Mining* (Vol. 11, Issue 2, pp. 78–91). https://doi.org/10.1002/sam.11372

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space* (arXiv:1301.3781). arXiv. http://arxiv.org/abs/1301.3781

Moghabghab, R., Tong, A., Hallaran, A., & Anderson, J. (2018). The Difference Between Competency and Competence: A Regulatory Perspective. *Journal of Nursing Regulation*, *9*(2), 54–59. https://doi.org/10.1016/S2155-8256(18)30118-2

Nakayama, H., Kubo, T., Kamura, J., Taniguchi, Y., & Liang, X. (2018). *doccano: Text Annotation Tool for Human*. https://github.com/doccano/doccano

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2024). *GPT-4 Technical Report* (arXiv:2303.08774). arXiv. http://arxiv.org/abs/2303.08774

Pais, S., Cordeiro, J., & Jamil, M. L. (2022). NLP-based platform as a service: A brief review. *Journal of Big Data*, *9*(1), 54. https://doi.org/10.1186/s40537-022-00603-5

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Pavlopoulos, J., Sorensen, J., Laugier, L., & Androutsopoulos, I. (2021). SemEval-2021 Task 5: Toxic Spans Detection. *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, 59–69. https://doi.org/10.18653/v1/2021.semeval-1.6

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). *Deep contextualized word representations* (arXiv:1802.05365). arXiv. http://arxiv.org/abs/1802.05365

Petukhova, A., Matos-Carvalho, J. P., & Fachada, N. (2024). *Text Clustering with LLM Embeddings* (arXiv:2403.15112). arXiv. http://arxiv.org/abs/2403.15112

Ramshaw, L., & Marcus, M. (1995). Text Chunking using Transformation-Based Learning. *Third Workshop on Very Large Corpora*. https://aclanthology.org/W95-0107

Ratinov, L., & Roth, D. (2009). Design challenges and misconceptions in named entity recognition. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning - CoNLL '09*, 147. https://doi.org/10.3115/1596374.1596399

Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks* (arXiv:1908.10084). arXiv. http://arxiv.org/abs/1908.10084

Ruder, S. (2019). *Neural transfer learning for natural language processing* [PhD Thesis]. NUI Galway.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620. https://doi.org/10.1145/361219.361220

Sayfullina, L., Malmi, E., & Kannala, J. (2018). *Learning Representations for Soft Skill Matching* (arXiv:1807.07741). arXiv. http://arxiv.org/abs/1807.07741

Shakina, E., Parshakov, P., & Alsufiev, A. (2021). Rethinking the corporate digital divide: The complementarity of technologies and the demand for digital skills. *Technological Forecasting and Social Change*, *162*, 120405. https://doi.org/10.1016/j.techfore.2020.120405

Shearer, Colin. (2000). The CRISP-DM model: The new blueprint for data mining. *Journal of Data Warehousing*, 13–22.

Smith, N. A. (2020). Contextual word representations: Putting words into computers. *Communications of the ACM*, *63*(6), 66–74. https://doi.org/10.1145/3347145

Sowmya V. B, Majumder, B., Gupta, A., & Surana, H. (2020). *Practical natural language processing: A comprehensive guide to building real-world NLP systems* (First edition). O'Reilly Media.

Sprark Jones, K. (1972). A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL. *Journal of Documentation*, *28*(1), 11–21. https://doi.org/10.1108/eb026526

Steiber, A., Alänge, S., Ghosh, S., & Goncalves, D. (2021). Digital transformation of industrial firms: An innovation diffusion perspective. *European Journal of Innovation Management*, *24*(3), 799–819. https://doi.org/10.1108/EJIM-01-2020-0018

Taylor, W. L. (1953). "Cloze Procedure": A New Tool for Measuring Readability. *Journalism Quarterly*, *30*(4), 415–433. https://doi.org/10.1177/107769905303000401

Teodorescu, T. (2006). Competence versus competency: What is the difference? *Performance Improvement*, *45*(10), 27–30. https://doi.org/10.1002/pfi.4930451027

Tunstall, L., Werra, L. von, & Wolf, T. (2022). *Natural language processing with transformers: Building language applications with Hugging Face* (First edition). O'Reilly Media.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.

Vermeer, N., Provatorova, V., Graus, D., Rajapakse, T., & Mesbah, S. (2022). *Using RobBERT and eXtreme Multi-Label Classification to Extract Implicit and Explicit Skills From Dutch Job Descriptions*.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*(1), 9. https://doi.org/10.1186/s40537-016-0043-6

Wu, J., Son, G., & Wang, S. (2020). A Competency Mining Method Based on Latent Dirichlet Allocation (LDA) Model. *Journal of Physics: Conference Series*, *1682*(1), 012059. https://doi.org/10.1088/1742-6596/1682/1/012059

Yang, Q., Zhang, Y., Dai, W., & Pan, S. J. (2020). *Transfer Learning*. Cambridge University Press. https://doi.org/10.1017/9781139061773

Zhang, M., Jensen, K. N., & Plank, B. (2022). *Kompetencer: Fine-grained Skill Classification in Danish Job Postings via Distant Supervision and Transfer Learning* (arXiv:2205.01381). arXiv. http://arxiv.org/abs/2205.01381

Zhang, M., Jensen, K. N., Sonniks, S. D., & Plank, B. (2022). *SkillSpan: Hard and Soft Skill Extraction from English Job Postings* (arXiv:2204.12811). arXiv. http://arxiv.org/abs/2204.12811

Zhang, M., Jensen, K. N., van der Goot, R., & Plank, B. (2022). *Skill Extraction from Job Postings using Weak Supervision* (arXiv:2209.08071). arXiv. http://arxiv.org/abs/2209.08071

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, *109*(1), 43–76. Proceedings of the IEEE. https://doi.org/10.1109/JPROC.2020.3004555

Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, *150*, 106889. https://doi.org/10.1016/j.cie.2020.106889

9 List of Figures

10 List of Formulas

11 List of Tables

12 Abbreviations

API	Application Programming Interface
ANOVA	Analysis of Variance
BERT	Bidirectional Encoder Representations from Transformers
BILOU	Beginning, Inside, Last, Outside, Unit
BIO	Begin, Inside, Outside
BIS	Berufsinformationssystem
BoN	Bag of N-grams
BoW	Bag of Words
CNN	Convolutional Neural Network
CRISP-DM	Cross-Industry Standard Process for Data Mining
CRS	cross-en-de-roberta-sentence-transformer
DISCO	European Dictionary of Skills and Competences
DSRP	Design Science Research Process
e.g.	exempli gratia (for example)
EOS token	End-of-Sequence Token
EQF	European Qualifications Framework
ESCO	European Skills, Competences, Qualifications and Occupations
FN	False Negative
FP	False Positive
GPU	Graphics Processing Unit
i.e.	id est (that is)
IE	Information Extraction
IS	Information System
IT	Information Technology
JDCO	Jobdigger Classification of Occupations
JGB	JobGBERT
LDA	Latent Dirichlet Allocation
LLM	Large Language Model
LSTM	Long Short-Term Memory
ML	Machine Learning
MLM	Masked Language Model
MRR	Mean Reciprocal Rank
nDCG	Normalized Discounted Cumulative Gain
NER	Named Entity Recognition
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
OOV	Out of Vocabulary
PLSA	Probabilistic Latent Semantic Analysis
PM2	paraphrase-multilingual-mpnet-base-v2
POS	Part of Speech

PROPN	Proper Nouns
RNN	Recurrent Neural Network
RQ	Research Question
SOTA	State-of-the-art
TF-IDF	Term Frequency-Inverse Document Frequency
TN	True Negative
TP	True Positive
ULMFiT	Universal Language Model Fine-tuning
VSM	Vector Space Model

13 Overview of Additional Tools Used

Tool	Usage
DeepL Write	Throughout this work: Phrasing support for a scientific writing style
Grammarly	Throughout this work: Grammar and spellchecking
ChatGPT	Throughout this work: Brainstorming and phrasing support for a scientific writing style

14 Appendix

14.1 Detailed Search Strings for Scientific Databases

Web of Science (EN):

ALL=("Job" AND ("ad" OR "advertisement*" OR “posting” OR “description” OR “listing”) AND (“skill” OR “competency”) AND (“extraction” OR “identification” OR “mining” OR “classification”) AND (“weak supervision” OR “unsupervised” OR “distant supervision” OR "transfer learning"))

Web of Science (DE):

ALL=("Job*" AND ("Fähigkeit*" OR "Kompetenz*") AND ("extraktion" OR "identifizierung" OR "Mining" OR "klassifizierung") AND (“weak supervision” OR “unsupervised” OR “distant supervision” OR “transfer learning”))

IEEE (EN):

("All Metadata":„Job” AND "All Metadata":("ad" OR "All Metadata":"advertisement*" OR "All Metadata":“posting” OR "All Metadata":“description” OR "All Metadata":“listing”) AND "All Metadata":(“skill” OR "All Metadata":“competency”) AND "All Metadata":(“extraction” OR "All Metadata":“identification” OR "All Metadata":“mining” OR "All Metadata":“classification”) AND "All Metadata":(“weak supervision” OR "All Metadata":“unsupervised” OR "All Metadata":“distant supervision” OR "All Metadata":“transfer learning”))

IEEE (DE):

("All Metadata":"Job*" AND "All Metadata":("Fähigkeit*" OR "All Metadata":"Kompetenz*") AND "All Metadata":("*extraktion" OR "All Metadata":"*identifizierung" OR "All Metadata":"Mining" OR "All Metadata":"*klassifizierung") AND "All Metadata":(“weak supervision” OR "All Metadata":“unsupervised” OR "All Metadata":“distant supervision” OR "All Metadata":“transfer learning”))

Scopus (EN):

("Job" AND ("ad*" OR "posting" OR "description" OR "listing") AND ("skill" OR "competency") AND ("extraction" OR "identification" OR "mining" OR "classification") AND ("weak supervision" OR "unsupervised" OR "distant supervision" OR "transfer learning"))

Scopus (DE):

"Job*" AND ("Fähigkeit*" OR "Kompetenz*") AND ("*extraktion" OR "*identifizierung" OR "Mining" OR "*klassifizierung") AND (“weak supervision” OR “unsupervised” OR “distant supervision” OR “transfer learning”)

14.2 ESCO Skill Group Codes

Table 19: Overview of ESCO group codes, their title and their description

Label	Title	Definition
K00	Generic programmes and qualifications	Generic programmes and qualifications are those providing fundamental and personal skills education which cover a broad range of subjects and do not emphasise or specialise in a particular broad or narrow field.
K01	Education	No Definition
K02	Arts and humanities	No Definition
K03	Social sciences, journalism and information	No Definition
K04	Business, administration and law	No Definition
K05	Natural sciences, mathematics and statistics	No Definition
K06	Information and communication technologies	No Definition
K07	Engineering, manufacturing and construction not elsewhere classified	No Definition
K08	Agriculture, forestry, fisheries and veterinary	No Definition
K09	Health and welfare	No Definition
K10	Services	No Definition
K99	Field unknown	No Definition
L1	Languages	Ability to communicate through reading, writing, speaking and listening in the mother tongue and/or in a foreign language.
L2	Classical Languages	All dead languages, no longer actively used, originating from various periods in history, such as Latin from Antiquity, Middle English from the Middle Ages,

		Classical Maya from the Pre-colonial Americas, and Renaissance Italian from the Early Modern Period.
S1	Communication, collaboration and creativity	Communicating, collaborating, liaising, and negotiating with other people, developing solutions to problems, creating plans or specifications for the design of objects and systems, composing text or music, performing to entertain an audience, and imparting knowledge to others.
S2	Information skills	Collecting, storing, monitoring, and using information; Conducting studies, investigations and tests; maintaining records; managing, evaluating, processing, analysing and monitoring information and projecting outcomes.
S3	Assisting and caring	Providing assistance, nurturing, care, service and support to people, and ensuring compliance to rules, standards, guidelines or laws.
S4	Management skills	Managing people, activities, resources, and organisation; developing objectives and strategies, organising work activities, allocating and controlling resources and leading, motivating, recruiting and supervising people and teams.
S5	Working with computers	Using computers and other digital tools to develop, install and maintain ICT software and infrastructure and to browse, search, filter, organise, store, retrieve, and analyse data, to collaborate and communicate with others, to create and edit new content.
S6	Handling and moving	Sorting, arranging, moving, transforming, fabricating and cleaning goods and materials by hand or using handheld tools and equipment. Tending plants, crops and animals.
S7	Construction	Building, repairing, installing and finishing interior and exterior structures.
S8	Working with machinery and specialised equipment	Controlling, operating and monitoring vehicles, stationary and mobile machinery and precision instrumentation and equipment.
T1	Core skills and competences	Skills and competences representing the foundation for interacting with others and for developing and learning as an individual. They comprise the ability to understand, speak, read and write language(s), to work with numbers and measures and to use digital devices and applications.
T2	Thinking skills and competences	Skills and competences relating to the ability to apply the mental processes of gathering, conceptualising, analysing, synthesising, and/or evaluating information gathered from, or generated by, observation, experience, reflection, reasoning, or communication. They include the ability to evaluate and use information of different kinds to plan activities, achieve goals, solve problems, deal with issues and perform complex tasks in routine and novel ways.

T3	Self-management skills and competences	Skills and competences requiring individuals to understand and control their own capabilities and limitations and use this self-awareness to manage activities in a variety of contexts. They include the ability to act reflectively and responsibly, to accept feedback, adapting to change and to seek opportunities for personal and professional development.
T4	Social and communication skills and competences	Skills and competences relating to the ability to interact positively and productively with others. This is demonstrated by communicating ideas effectively and empathetically, coordinating one's own objectives and actions with those of others and acting in ways which are structured according to values, ensuring the well-being and progress of others, and offering leadership.
T5	Physical and manual skills and competences	Skills and competences relating to the ability to perform tasks and activities requiring manual dexterity, agility and/or bodily strength. This is demonstrated by carrying out tasks and activities in demanding or hazardous environments requiring endurance or stamina. These tasks and activities may be carried out by hand, with other direct physical intervention, or by using equipment, tools or technology (such as ICT devices, machinery, craft or musical instruments) which requires guidance, movement or force.
T6	Life skills and competences	Skills and competences relating to the ability to process and use knowledge and information which has transversal significance and facilitates active citizenship. They comprise the areas of health, environment, civic engagement, culture, finance and the application of general knowledge.

14.3 Annotation Guidelines

Legende:

- Rot markiert eine Fähigkeit
- Gelb markiert eine Kenntnis
- Ein Stichpunkt („●“) signalisiert einen Beispielsatz
- Richtlinien bauen auf jenen aus (Zhang, Jensen, Sonniks, et al., 2022) auf und wurden im Rahmen dieser Abschlussarbeit ins Deutsche überführt und mit neuen Beispielsätzen versehen

14.3.1 Richtlinien für Fähigkeiten

1. Eine Fähigkeit startet mit einem **Verb**, oder mit einem **(Adjektiv) + Nomen**

    - Sie bringen [handwerkliches Geschick] Fähigkeit und [technisches Verständnis] Fähigkeit mit
    - [Durchführen von Controlling Prozessen] Fähigkeit

1.1 Modalverben (müssen, können, dürfen, sollen, wollen, mögen, möchten) werden nicht mit markiert

2. Satzteile mit Präpositionen und/oder Konjunktionen werden aufgetrennt

2.1 Es sei denn, die Konjunktion verbindet zwei Nomen als ein Argument

- [Konzeptionierung und Implementierung von Datenmanagementlösungen] Fähigkeit
- [selbstständige und eigenverantwortliche Arbeitsweise] Fähigkeit

2.2 Keine Fähigkeiten mit anaphorischen Pronomen kennzeichnen, sondern nur die vorangehende Fähigkeit:

- [Priorisierung von Aufgaben] Fähigkeit und Identifizierung der Wichtigsten

2.3 Trennung von zusammenhängendem Nomen und Adjektiven, wenn sie nicht mit einem Verb verbunden sind:

- Seien sie [neugierig] Fähigkeit und [proaktiv] Fähigkeit
- Erste Erfahrung in der [IT-Administration] Kenntnis, im [System Engineering] Kenntnis oder im [IT-Consulting] Kenntnis und mit der [Datenplattform Splunk] Kenntnis

3. Wenn relevante Informationen an irrelevante Informationen angehängt werden, versuchen wir, die Fähigkeit **so kurz wie möglich** zu halten. Bestimmte und unbestimmte Artikel werden weggelassen:

- ~~Wir schätzen eine~~ [eigenverantwortliche Arbeitsweise] Fähigkeit ~~und bieten Dir den Raum […]~~
- [Sammlung und Auswahl von strukturierten und unstrukturierten Daten] Fähigkeit ~~aus internen sowie externen Quellen~~

4. Wenn das Auslassen von Indikatorworten wie „Fähigkeiten“ und „Kenntnisse“ die Phrase unvollständig machen würde, werden diese mitmarkiert, ansonsten ausgelassen:

- [Zwischenmenschliche Fähigkeiten] Fähigkeit → nur [zwischenmenschliche] würde hier keinen Sinn ergeben

5. Klammern nach einer Fähigkeit/Kenntnis werden auch markiert, sofern sie die Fähigkeit näher beschreibt oder eine Abkürzung beinhaltet.

- Fundierte Kenntnisse in [Google Cloud (GCP)] Kenntnis

6. Es werden Adverbien nur eingefügt, wenn die Art und Weise beschreiben, **wie** etwas gemacht wird, für die Fähigkeit von Relevanz ist. Alle anderen werden ausgeschlossen:

- Sie [kommunizieren offen] Fähigkeit
- unseren Gästen einen ~~erstklassigen~~ [Kundenservice zu bieten] Fähigkeit

7. Eigenschaften werden als Fähigkeit eingeordnet:

- eine [Hands-On Mentalität] Fähigkeit

7.1 Eigenschaften werden nicht markiert, wenn sie Fähigkeiten oder Kenntnisse beinhalten – in dem Fall wird nur die Fähigkeit oder Kenntnis getaggt:

- ~~Leidenschaft für~~ [Automatisierung] Kenntnis

- ~~Freude an der~~ [Arbeit im Team] Fähigkeit

8. Sonstiges

8.1 Keine ironischen Fähigkeiten markieren (zb.: faul)

8.2 Verschachtelung von Fähigkeiten vermeiden, als eine Fähigkeit markieren

8.3 Wir vermerken alle Fähigkeiten, die Teil von Abschnitten wie „Anforderungen“, „Gut-zu-wissen“, „Optional“, „Nach x-monatiger Ausbildung werden Sie in der Lage sein, …“, „Bei der Arbeit werden Sie…“.

8.4 Wenn es einen allgemeinen Standard gibt, der der Fähigkeit hinzugefügt werden kann, fügen wir diese hinzu. Der Standard wird als Kenntnis markiert.

- [Verarbeitung von Zahlungen] Fähigkeit gemäß den […] [Standards] Kenntnis

14.3.2 Richtlinien für Kenntnisse

1. **Faustregel:** Kenntnisse sind etwas, welche man über ein Thema besitzt, aber (normalerweise) nicht physisch ausführen kann:

- [Python] Kenntnis
- [Instandhaltungsmaßnahmen] Kenntnis

2. Gibt es eine Komponente in einer Klammer, die zu der Kenntnis gehört, fügen wir sie hinzu:

- [(nicht-) relationale Datenbanken] Kenntnis
- [Führerschein der Klasse B (alte Klasse 3)]Kenntnis

3. **Lizenzen und Bescheinigungen:** Wenn erforderlich fügen wir der Kenntnis die zusätzlichen Wörter „Zertifikat“, „Karte“, „Lizenz“, etc., hinzu.

4. Sieht die Kenntnis aus wie eine Fähigkeit, das vorangehende Verb ist allerdings sehr allgemein gehalten (wie z.B., befolgen, anwenden, einhalten, arbeiten (mit)), markieren wir nur die Kenntnis:

- arbeiten mit [SQL-Datenbanken] Kenntnis

5. Verschiedenen Kenntnisse werden **einzeln** markiert

- deine Leidenschaft sind moderne [CMS-Architekturen] Kenntnis, [APIs] Kenntnis sowie [Domain-Driven-Design] Kenntnis

6. Kenntnisse können in Fähigkeiten verschachtelt sein:

- Erstellung umfangreicher Konzepte zur [Datenintegration] Kenntnis] Fähigkeit auf Basis von [AWS-und/oder Azure Technologien] Kenntnis
- [[Inbetriebnahme] Kenntnis, [Wartung] Kenntnis und [Instandhaltung] Kenntnis teil- oder vollautomatisierter Maschinen und Fertigungsanlagen] Fähigkeit

6.1 Einleitende Worte, welche die Anwendung einer Kenntnis implizieren, wie „Durchführung“, „Anwendung“ und „Umsetzung“, werden als Fähigkeit und Kenntnis markiert:

- [Durchführung von [Sicherheitsinspektionen] Kenntnis] Fähigkeit

7. Wenn **alle Kenntnisse auf ein Wort referenzieren**, markieren wir sie als eine Kenntnis

- Fachwissen im Design und dem Betrieb von [Analytics und Monitoring Plattformen] Kenntnis → referenzieren auf „Plattform“

7.1 Zusammenhängende Wörter als eine Kenntnis markieren

- [Service – und Wartungsarbeiten] Kenntnis
- [Deutsch- und Englischkenntnisse] Kenntnis
- [Masterstudium] Kenntnis im [technischen, wirtschaftlichen oder IT-Bereich] Kenntnis

8. Bei eine Auflistung von Kenntnissen, markieren wir alle Kenntnisse separat:

- [Abgeschlossenes Studium] Kenntnis im Bereich [Data Science] Kenntnis, [Informatik] Kenntnis, [Mathematik] Kenntnis, [Statistik] Kenntnis oder einem verwandten Feld

9. **Berufsbezeichnungen und Studien** werden ebenso als Kenntnis markiert

- [Abgeschlossene Ausbildung] Kenntnis als [Industriemechaniker] Kenntnis, [Mechatroniker] Kenntnis oder vergleichbare [Ausbildung] Kenntnis

14.3.3 Sonstige Richtlinien

1. **Faustregel**: Im Zweifelsfall als Fähigkeit markieren.

2. Wir präferieren **Fähigkeiten vor Kenntnissen**.

3. Fähigkeiten haben Vorrang vor Einstellungen/inneren Emotionen. Ist in der Eigenschaft eine Fähigkeit enthalten, markieren Sie nur die Fähigkeit

- ~~Leidenschaft für~~ [Erstellung anschaulicher Grafiken] Fähigkeit für die [Qualitätssicherung] Kenntnis

4. Wir versuchen die Fähigkeiten/Kenntnisse so kurz wie möglich zu halten (d.h. wir lassen zu spezifische Information weg)

5. Wir verzichten auf Füllwörter und „Trigger“ (d.h. Wörter, die darauf hinweisen, dass eine Fähigkeit oder Kenntnis folgen wird: „~~fortgeschrittene Kenntnisse in~~ […] Kenntnis“), um die Komponente herum.

- [Deutsch] Kenntnis und [Englisch] Kenntnis ~~sehr gut in Wort und Schrift~~
- ~~Gutes Verständnis von~~ [Data Warehousing/Business Intelligence (DWH/BI)] Kenntnis
- ~~Erfolgreich~~ [abgeschlossenes Studium] Kenntnis in […]
- ~~Erfahrungen im Bereich~~ [Maschinelles Lernen] Kenntnis
- ~~Erfahrung in der~~ [Kombination von Prozess und Technologie] Fähigkeit

6. Achten Sie auf Ausdrücke wie „Teilnahme an …“, „Beitrag“ und „Transfer“. Diese werden in der Regel nicht als Fähigkeiten betrachtet:

- ~~Beitrag zur Zufriedenheit unserer Kunden leisten~~

7. Erforderliche Fähigkeiten und Kenntnisse, die nicht an eindeutigen Stellen zu finden sind (z.B. in Projektbeschreibungen), werden ebenfalls markiert

8. Achten Sie auf das Muster „Fähigkeit gefolgt von einer Erklärung“. Dies kann meistens als Fähigkeit und Kenntnis markiert werden.

- Gezielte [Abfrage von Daten aus [Datenbanksystemen] Kenntnis und [Distributed Systems] Kenntnis] Fähigkeit mithilfe von [SQL] Kenntnis und anderen [Programmiersprachen] Kenntnis
- [Erstellung von Monats-, Quartals-, und Jahresabschlüssen] Fähigkeit nach [Steuer- und Handelsrecht] Kenntnis

10. Geben Sie nur Fähigkeiten an, die **für die Stelle relevant** sind.

    10.1. Dazu gehören auch Fähigkeiten, die in Zukunft erwartet werden.

    10.2. Dazu gehören keine Fähigkeiten, Kenntnisse oder Einstellungen, die **nur** das Unternehmen, die Gruppe, der Sie in der Abteilung angehören werden, beschreiben.

11. Wir geben Branchen und Bereiche (in denen der Arbeitnehmer arbeiten wird) als Kenntnis an.

12. Sollten Schlagwörter gegeben sein, werden die Kenntnisse einzeln markiert.

- Schlagworte: [IT] Kenntnis, [Python] Kenntnis, [Informatik] Kenntnis, [SQL] Kenntnis,[…]