Lappeenranta-Lahti University of Technology LUT

School of Business and Management

Master in Business Analytics

Riabchenko Alisa

TAXONOMY-BASED VACANCY – CV MATCHING

Examiners: Associate Professor Jan Stoklasa

Professor Pasi Luukka

ABSTRACT

University: Lappeenranta-Lahti University of Technology LUT

Faculty: School of Business and Management

Major: Degree in Business Analytics

Author: Riabchenko Alisa

Title: Taxonomy-based Vacancy – CV matching

Year: 2022

Masters's thesis: 94 pages, 25 figures, 7 tables and 2 appendices

Examiners: Associate Professor Jan Stoklasa and Professor Pasi Luukka

Keywords: information extraction, natural language processing, SBERT model, Sentence Transformer, NER, job descriptions analysis, skills taxonomy

The changes in the labor market accompanied by digitalization have influenced the approaches used for recruitment purposes. The development of machine learning and, specifically, natural language processing allowed new services for automated resume screening to appear. These services aim to reduce the time that HR specialists dedicate to sorting out unsuitable candidates, thus saving the costs per hire. This master thesis provides an analysis of the possible approach for matching CVs to JDs based on the taxonomy of competencies.

In the literature review, there are the description and comparison of the different ontologies and taxonomies, and the analysis of the previous research performed in vacancy–CV matching provided.

In the methodology part, there are the concepts of natural language processing and information extraction described, and the Sentence Transformer model and named entity recognition explained. Besides, evaluation approaches for taxonomy enrichment and vacancy–CV matching are suggested.

In the solution development part, the described concepts and models are applied to the JDs and CVs data to perform taxonomy enrichment and matching tasks. The assessment of the matching algorithm performance based on the expert's evaluation is presented. Based on the results obtained, the potential usage of the research and possible limitations are discussed.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to all staff from LUT University who were always ready to help me during my studies, it was a really great opportunity and unique experience to study in Finland. Also, I would like to thank my scientific advisors Jan Stoklasa and Pasi Luukka for the dedicated time and valuable comments. You really helped me to improve the quality of my thesis.

Lastly, I am also thankful to my dear parents Andrey Riabchenko and Svetlana Sakharova, my grandmother Larissa Sakharova, and all my friends, who provided their support throughout my study journey.

Kind regards,

Alisa Riabchenko

List of abbreviations

AI Artificial Intelligence

BERT Bidirectional Encoder Representations from Transformers

CNN Convolutional Neural Network

DL Deep Learning

IE Information Extraction

JD Job Description

ML Machine Learning

NER Named Entity Recognition

NLP Natural Language Processing

SBERT Sentence Bidirectional Encoder Representations from Transformers

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

This chapter provides the background for the research and the aims of the study including the problem statement and expected results of the thesis.

1.1. Background

The development of the labor market accompanied by digitalization in all spheres compels companies to implement new approaches to recruitment processes and skills and competencies management (Charles et al., 2022). The occurrence of new technologies, tools, and frameworks has changed the requirements for successful candidates (Bogush, 2022). This mostly can be observed in IT and spheres related to data science and analytics. Employers struggle to find employees with the perfect combination of skills that can allow them to fulfill the company's objectives and perform everyday tasks thus contributing to the development of the company (Hoff, 2022). It is obvious, that any job position implies the list of core competencies that characterize a successful candidate. So, the task of a recruiter is to analyze the job requirements and scan all the CVs of potential candidates, keeping these needed core competencies in mind, and then to make a judgment about the relevance of each candidate to the job position. Thus, the recruitment process tends to be difficult and time-consuming for both employers and employees.

Currently, there are occurring many services that aim to provide automated screening programs that allow reviewing many CVs at the same time. Even though these services do not allow for the analysis of the soft skills of each candidate, still they can make the recruitment process easier, more efficient, and less time-consuming for HR specialists by automation of resumes analysis. As an example of such a service, there can be named CVVIZ (https://cvviz.com/), which provides an opportunity for resumes screening, relative resumes matching, and automatic discovery of the best candidates. Freshteam software (https://www.freshworks.com/) also provides an automated applicants screening system that allows one to filter candidates, leverage customized assignments for candidates, schedule emails, and track interactions with possible employees. The existence of these services has become possible with the development of machine learning (ML), specifically natural language processing (NLP) techniques that allow for text recognition and extraction of information from vague unstructured texts (Harsha et al., 2022).

According to the research in the area of recruitment that was held in 2021, improving time to hire and quality of hire were named within the list of top five recruiting priorities (Human Resources Director, 2021), and using artificial intelligence (AI) as a must to overcome recruitment challenges (SHRM, 2021).

1.2. Problem statement

As it was previously stated, the work of an HR specialist in the area of recruitment has become tough and time-consuming (Hoff, 2022). Many factors determine a successful candidate, who will be competent enough to perform a certain role within a company. Especially, this can be observed in the IT-related spheres, which are developing very fast nowadays. This is also associated with the emergence of new tools, IT products, frameworks, knowledge, and experience which are important for employers.

According to the research (Farrugia, 2022) the average corporate job posting receives around 250 CVs from applicants. Some of the applications are completely not suitable for the job position, however, HR specialists spend time reviewing these CVs, instead of dedicating all the time to candidates with needed competencies. Systems, that allow matching CVs with vacancies and rating them with the respect to their suitability can potentially save the time and effort of HR specialists, thus saving the expenses of the company per one candidate employed. However, despite all the convenience of these services, they still have drawbacks. Experts mention that the algorithms that are used for screening the resumes may not always use the best criteria for matching which can result in the rejection of qualified candidates (Vickery, 2022). Also, there are ethical problems arising from using automated recruitment, such as privacy challenges while dealing with personal information (CVs are transferred to the new location (AI systems allowing for automated screening) that could increase the risk of unauthorized processing of data (Morrison, 2021)), companies profiting from the personal information of candidates even though they were rejected, algorithmic biases and negative effect on workforce diversity (Hunkenschroer & Luetge, 2022). However, according to the SHRM survey (there were 1688 HR professionals currently working in organizations within a wide range of industries across the United States interviewed) (SHRM, 2022):

- 85% of employers using automation stated that it helped them to save time and increase efficiency
- 64% of HR specialists stated that automation helped them to filter out underqualified candidates

Apart from traditional hiring methods that only use a vacancy and the list of possible candidates with their experiences mentioned, there is another possible way to perform the analysis of CVs is using competence maps – ontologies and taxonomies, that will be a basis for the vacancy – CV matching (Wang et al., 2021). The current thesis focuses on the analysis of such job positions as "business analyst", "data analyst", "data architect" and "enterprise architect". For the purpose of this thesis – vacancy – CV matching for the previously mentioned job positions, there will be used the taxonomy of competencies that was developed by the company, which provides consulting services in business analysis and management of corporate architecture. This taxonomy was primarily developed for such occupations as "business analyst" and "enterprise architect", so it can be used as a starting point for the further development and enrichment of the taxonomy by including the competencies relevant to the "data analyst" and "data architect" occupations and extending the list of competencies for "business analyst" and "business architect". The enrichment of the taxonomy also aims to make it more relevant to the European labor market by analyzing the job posting from the Monster.com website. The original taxonomy was developed by the company that operates in the Russian market, which is why the list of competencies could be more applicable to the business environment in Russia. The description and comparison of several widely used ontologies and taxonomies including the previously mentioned taxonomy are presented in the literature review. The enriched taxonomy of competencies will be further used as the basis for matching the vacancies with suitable CVs.

In the taxonomy enrichment task, the pre-trained Sentence Transformer model (SBERT model) will be used in this thesis to extract the competencies from the collected job descriptions based on the chosen taxonomy of competencies. This model aims to calculate the cosine similarity between 2 sentences (Devika et al., 2021), in this specific case – between the sentence or phrase from the job description and the competency from the taxonomy. For the matching task, except for the Sentence Transformer model, there will be applied NER pipe in order to extract the competencies that represent the experience with tools and software.

1.3. Research questions and goal of the thesis

Research questions:

- How accurately can the pre-trained Sentence Transformer model extract the competencies from JDs and CVs using the taxonomy of skills based on the F-measure?
- Are the rankings of the top 10 suitable candidates for certain job positions obtained by the matching algorithm accurate or not?

The current thesis aims to fulfill two main goals:

- Enrich the selected taxonomy of competencies by performing the analysis of job postings from the website Monster.com for the job positions of:
    - Business Analyst
    - Enterprise Architect
    - Data Analyst
    - Data Architect
- Create an algorithm used for matching job descriptions and resumes

There are several tasks within this thesis:

- Analyze the existing findings within the field examined and perform the literature review
- Collect the job descriptions for the positions of “business analyst”, “business architect”, “data analyst” and “data architect” by implementing the job descriptions parser component
- Collect the CVs for the positions of “business analyst”, “business architect”, “data analyst” and “data architect”
- Build the model to extract competencies from the job descriptions collected from the Monster.com website based on the enriched taxonomy of skills and assess its performance by F-score metric

- Assess the quality of the JD – CV matching algorithm by expert's evaluation

1.4. Structure of the thesis

The first chapter of this thesis is devoted to the review of the existing ontologies and taxonomies of competencies, the analysis of their advantages and disadvantages, and their potential applicability to the purpose of this thesis. Apart from that, the first part describes the previous research made in the field of the application of NLP to the task of matching JDs and CVs. Further, in the second chapter there are the main concepts of NLP, the models used, and also the methods of models' performance evaluation presented. In the third chapter, the focus of the thesis is on the solution development and description of the steps done to fulfill the purpose of the thesis. The final chapter is devoted to the managerial application of the solution presented, the limitations, and ideas for further research.

2. Literature review

The literature review performed within this chapter can be divided into three main parts, the first one describes the problems that HR specialists face while performing the recruitment and selection process and the need for automation of the resumes screening process. The second part is dedicated to the description of the existing skills ontologies and taxonomies, and the third one – to the examples of the implementation of NLP concepts and models to the task of JD-CV matching.

2.1. Challenges and opportunities for the recruitment process from the perspective of HR specialists

Nowadays, the HR management field experiences many pressures to change due to the changes in the economy, the effect of globalization, and also the new technologies created (Stone & Deadrick, 2015).

Sajwani (2022) describes five recruitment challenges that companies and HR specialists face. One of them, mentioned by the author is the difficulty in finding the right fit – 87% of HR specialists reported that for the positions they wanted to fill, there were either few or no qualified applicants. Another statistic (Todorov, 2022) provides the following information on the recruitment process:

- Average time to hire across the range of platforms is 41 days
- An average job posting gets 250 resumes
- On average, HR professionals spend a third of their week sourcing candidates
- Attracting qualified candidates is the biggest challenge for 76% of recruiters
- Bad hires can result in a 36% decline in productivity

Thus, it becomes obvious that recruitment is a very time-consuming and difficult process since it is very hard for HR specialists to find the most suitable qualified candidate for each position. These challenges should be resolved, and the quality of hires should be increased to avoid financial losses resulting from bad hires.

Technology has influenced the HR and recruitment field, which is now changing facing new demands (Reman, 2022). A CareerBuilder survey found that 72% of employers believe some roles within talent acquisition and human capital management will be fully automated within 10 years (Tarpey, 2022).

2.2. Overview of existing skills ontologies and taxonomies

Firstly, it is important to describe what ontologies and taxonomies are. Taxonomies represent the formal structure of different classes or types of objects within a specific domain (Knight, 2020). They have a hierarchic format and categorize objects in a domain and consider only parent-child ("is a part of") relationships without any additional links (Keyser, 2012). Taxonomies, for example, can be used to classify documents into categories.

Ontologies have a higher level of sophistication in comparison to taxonomies. The same as taxonomies, ontologies have a hierarchic format and organize structured and unstructured

information through entities and their properties, but they take into consideration more relations to categorize concepts ("is located in", "is based on", "works for" etc.) (Keyser, 2012).

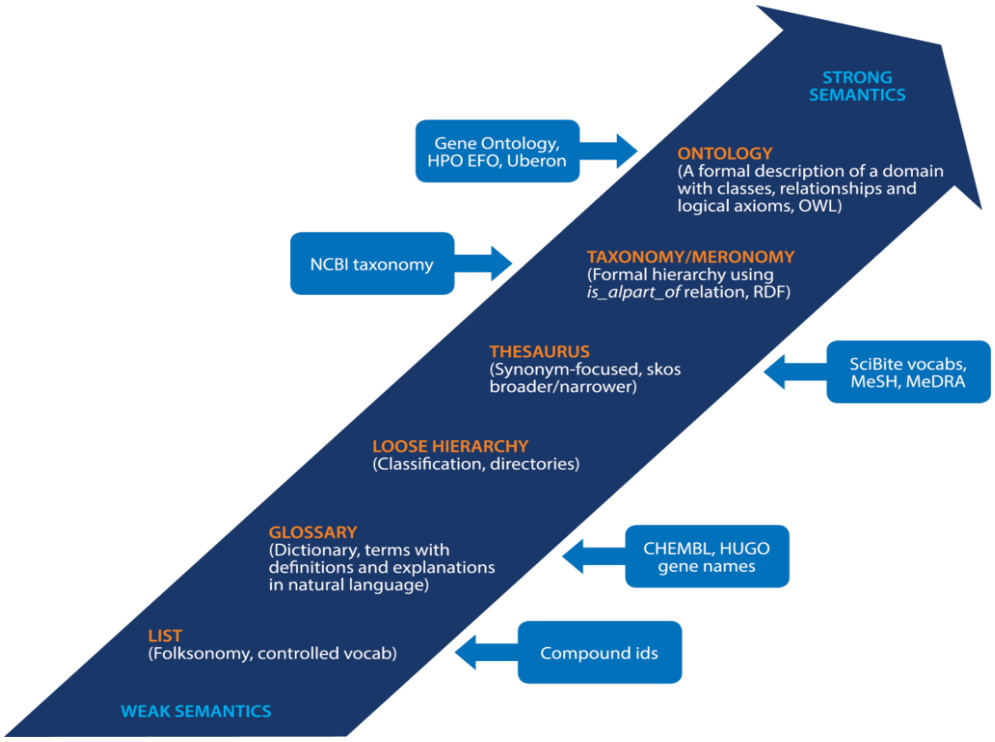

Figure 1. Taxonomies and ontologies (Tainter et al., 2022)

Several ontologies are existing that describe the skills and competencies of employees. These ontologies present various skills and competencies in a structured way. Usually, these skills ontologies are used by organizations to define and measure relationships between skills, jobs, and employees (Lazzareschi, 2022). Further in this chapter, there will be presented several most widely used ontologies.

2.2.1. ESCO ontology

ESCO ontology stands for "European Skills, Competencies Qualifications, and Occupation" ontology. This ontology was developed by European Commission to improve the supply of information on skills demand in the labor market (Chiarello et al., 2021). It contains around 13 500 skills in almost 3 000 occupations, but the number of occupations and skills is growing because the ontology is continuously updated (ESCO database and network design

and Administration, 2022). ESCO ontology is available in 28 languages, including English (ESCO, 2022). The structure of the ESCO classifier consists of three main pillars – occupations, skills and competencies, and qualifications (ESCO, 2022). All these three elements are interrelated and allow to organize common terminology for the European labor market (Rogushina et al., 2019). The "Occupations" element describes the professions that are important for the European labor market and organized by comparing them with the International Standard Classification of Occupations (Rogushina et al., 2019).

According to the information presented on the official website of ESCO, on the upper level, "Skills and competencies" are classified into 4 main groups – knowledge, skills, attitudes and values, language skills, and knowledge. The knowledge group of skills is divided into the groups based on the sector or domain, such as "agriculture, forestry, fisheries and veterinary", "arts and humanities", "health and welfare" etc. Further, these categories of sectors are subdivided into more concrete sector groups, that include the respective knowledge. For example, the "information and communication technologies" sector has a subsector "database and network design and administration" that includes knowledge in "data protection", "cloud technologies" etc. Language skills and knowledge group includes the list of existing languages. Skills also include different sub-categories, such as "communication, collaboration, and creativity", "information skills", "management skills" etc., that in their turn are also divided into categories. Thus, it can be stated that this ontology has many specification levels.

This ontology is used in many online platforms and applications, for example in job market analysis services, such as Skills Intelligence and Skills-OVATE. ESCO ontology is publicly available and can be downloaded from the official website of ESCO. One of the advantages of this ontology is that it analyzes a wide range of different sectors and occupations. However, at the same time, this can be a drawback because the ontology doesn't have any emphasis on a specific industry, which is why the list of skills can be not that precise. Also, the presence of such a great number of various skills can influence the speed of processing of these skills and also the memory dedicated to processing.

2.2.2. SARO ontology (Dadzie et al., 2018)

SARO ontology is Skills and Recruitment ontology, which represents the knowledge that is required to correctly define job descriptions on the basis of skills, competencies, and qualifications needed to fill a job role. This ontology was based on previously developed skills vocabularies and ontologies (including ESCO ontology) and enriched by the new best practices and open knowledge bases. Figure 2 demonstrates the general level of SARO ontology.

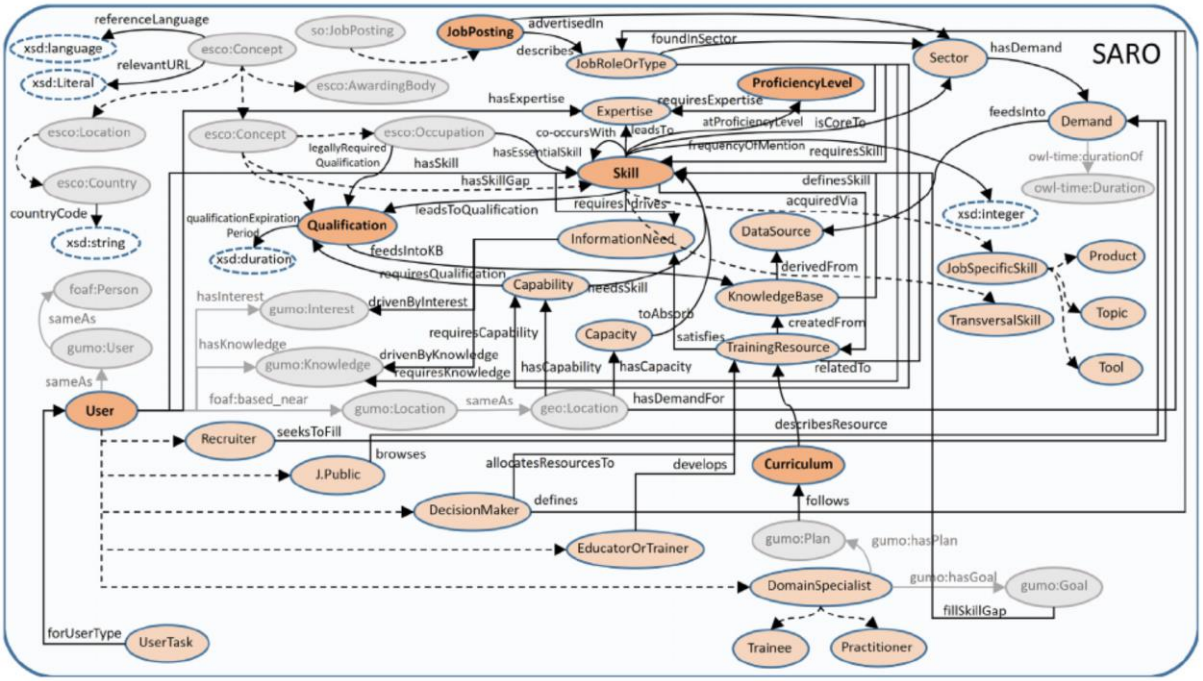

Figure 2. Upper-level view on SARO ontology (Dadzie et al, 2018)

The figure presents several main concepts – user, skill, job posting, qualification, curriculum, and proficiency level, and the relationships between them. These 6 main concepts are presented in orange color in figure 2. The job posting concept describes the job advert, specifically the role or type of job, and demand for it (location and date of posting, and also sector). The job posting concept leads to the skills concept, which refers to the ESCO ontology of skills classification, which divides the skills into job-specific and transversal. On the level of job-specific skills, SARO extends it to "Product" (competence related to using a particular product), "Topic" (knowledge within a specific domain or role), and "Tool" (competence in using a specific tool to perform technical tasks). Such lower lever concepts as "Product", "Topic" and "Tool" are presented in the light orange color in figure 2. Concepts that are

highlighted in gray color are related to other ontologies, such as, for example, ESCO. Solid black arrows represent “includes” relationships, and dashed black arrows represent subclasses in this ontology. The description of other elements of the legend is presented in figure 3.

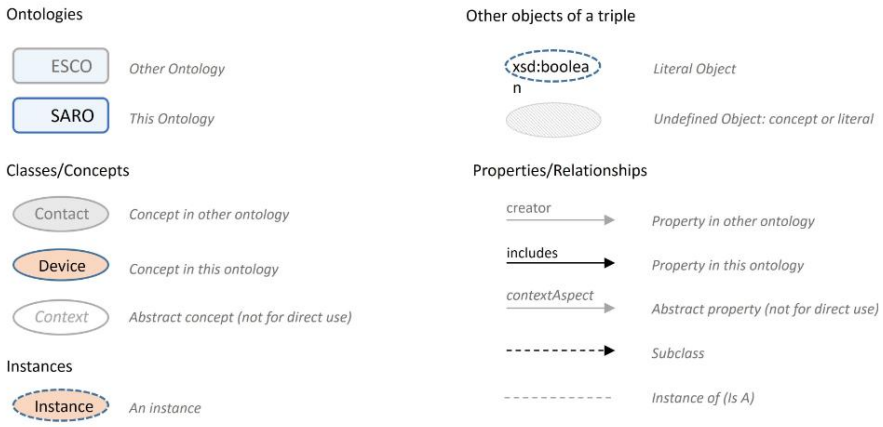

Figure 3. Legend for SARO ontology scheme (Sibarani et al., 2020)

SARO ontology as well as ESCO ontology contains a very wide range of professions and skills, which could be a disadvantage while analyzing just one specific industry. However, to one of the undoubted advantages can be attributed the division of job-specific skills into the categories such as products, tools, and topics. This can be extremely important when it comes to the analysis of IT-related occupations, where the criteria of employee success are determined mostly by the knowledge and experience in using specific tools and products.

2.2.3. O*NET taxonomy (The Occupational Information Network (O*NET) official website, 2022)

O*NET is an American service for job market analysis, that has its database of occupations and respective skills and knowledge relevant to these specific occupations.

Figure 4 presents the content model of O*NET. As can be seen, it includes the following major domains:

- Worker characteristics
    - Abilities (cognitive, physical, etc.)

    - Occupational Interests (personality types and working environments)
    - Work values (aspects of work related to job satisfaction)
    - Work styles (personal characteristics affecting the performance at work)
- Worker requirements
    - Basic skills (background structures that allow for more rapid knowledge acquisition in more specific domains)
    - Cross-Functional Skills (capacities to perform activities across various job roles)
    - Knowledge (facts and principles of general domains)
    - Education (educational experience)
- Experience requirements (typical experiential background needed to perform a job)
    - Experience and training
    - Basic skills – entry requirement
    - Cross-functional skills – entry requirement
    - Licensing
- Occupational requirements
    - Generalized work activities (work activities common across various occupations)
    - Intermediate work activities
    - Detailed work activities (occupational activities that are common to a small group of occupations)
    - Organizational context (characteristics of organization)
    - Work context (physical and social pecularities of the work)
- Workforce characteristics
    - Labor market information

- Occupational outlook

- Occupation-specific information
  - Title (title and code according to O*NET taxonomy)
  - Description (requited duties performed by workers)
  - Alternative titles
  - Tasks (specific to certain occupations)
  - Technology skills (IT and software skills)
  - Tools (machines, equipment, and tools)

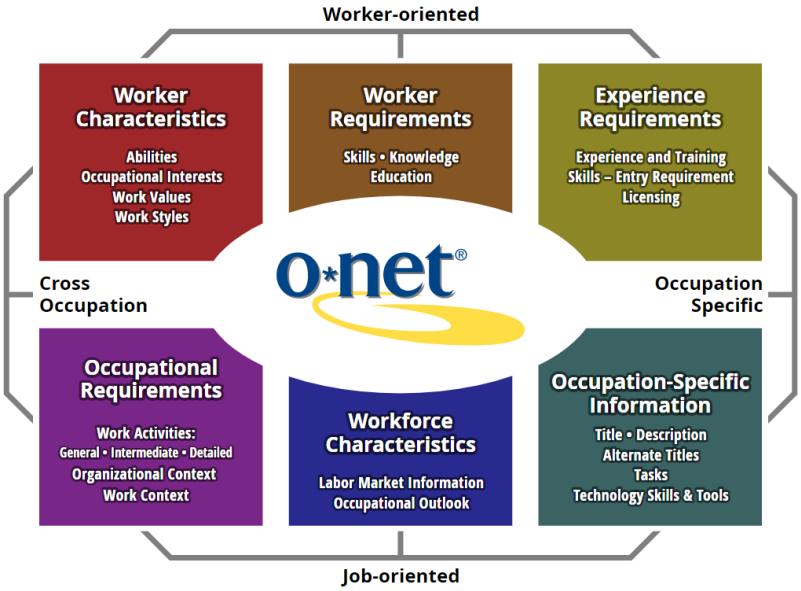

Figure 4. The content model of O*NET taxonomy (The Occupational Information Network (O*NET) official website, 2022)

The most relevant to this research are worker-oriented characteristics and requirements because they are more related to the skills and competencies of an employee and correspond to the requirements that employers formulate for a successful employee.

2.2.4. Taxonomy of competencies that is used in this thesis

A company that preferred to stay anonymous has developed the taxonomy of competencies and was willing to share it. A company is a consulting start-up company, that is working in the field of knowledge management and enterprise architecture.

The company has developed the reference model of competencies, the taxonomy of competencies, that is used to describe and estimate the competencies. The model was developed based on the third-party competency models (FEAPO competency model) and ontologies (ESCO ontology) and on the company's experience gained in the process of providing consulting services.

Based on the company's specifics the competencies included in the taxonomy describe the skills and knowledge needed for the IT and business architecture-related positions, such as "business analyst" and "enterprise architect". So, to be able to analyze and perform a matching task for the "data analyst" and "data architect" positions, it is needed to enrich the taxonomy with the skills that would be relevant for these job positions. Also, the range of competencies suitable for the positions of "business analyst" and "enterprise architect" should be extended for two reasons:

- adjusting the taxonomy of competencies to the realities of the European labor market (the taxonomy developed by a company is more relevant to the Russian labor market)
- updating the taxonomy of competencies by performing the analysis of job postings to make the taxonomy more up to date

The reference model of competencies consists of 416 competencies, which are divided into 4 main categories:

- Professional competencies (hard and close to hard skills)
    - Core enterprise architecture and business analysis activities
    - Management, support, and development of corporate architecture and business analysis activities
    - Related competencies in management
    - Related IT competencies
    - Basic interdisciplinary competencies
- Basic competencies (soft and close to soft skills)
    - Fundamentals of architectural thinking

    - Cognitive competencies
    - Competencies in communication and interaction with people
    - Managerial competencies
    - Personal competencies and characteristics
- Knowledge and experience with technologies and IT tools
    - Office tools and technologies (knowledge and work experience)
    - Collaboration tools (knowledge and work experience)
    - Charting tools (knowledge and work experience)
    - Tools and technologies for modeling and working with requirements (knowledge and work experience)
    - Project management tools (knowledge and work experience)
    - Tools for structuring information (knowledge and work experience)
    - Software development technologies (knowledge and work experience)
- Knowledge of methods, approaches, and standards
    - Related to business analysis and business planning
    - Methodologies, standards, and reference models in the field of enterprise architecture management
    - Knowledge and experience in the subject area
    - Organizational development and business transformation

The advantage of this taxonomy for the realization of the objectives of this thesis over the above-mentioned ontologies is that it was developed especially for the analysis of competencies within IT-related job positions, which is why it describes in detail the tools, products, frameworks, etc. that are associated with the IT field. Also, the emphasis on the specific field allows to save time and memory while performing matching tasks due to the limited quantity of only relevant to the field competencies. The structure of the taxonomy also can be named as suitable for IT-related occupations, because the grouping of the skills into

categories is done understandably, distinguishing hard and soft skills, and also knowledge in using various tools, technologies, software, and methodologies.

Thus, this taxonomy of competencies can be named a good starting point for the research proposed in this thesis. The updated and enriched taxonomy of competencies will be used to develop an algorithm to match JDs and CVs based on the competencies that are needed to perform a certain job position.

2.3. Applicability of natural language processing models in CV-JD matching

Bhatia et al. (2019) present their solution for the task of candidates ranking based on their suitability for certain job positions. The general scheme of the solution is presented in figure 5.

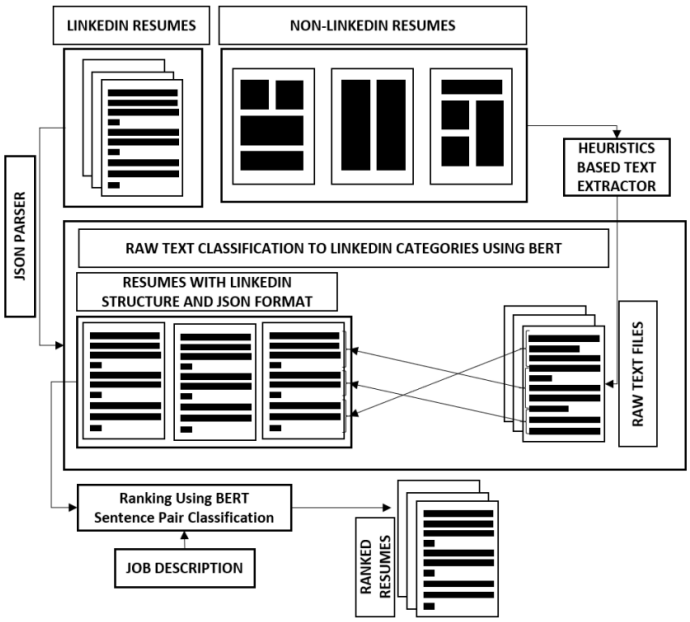

Figure 5. System of resumes ranking diagram (Bhatia et al., 2019)

As can be seen from figure 5, the method used by the authors includes several stages: resume parsing (both of non-fixed format and LinkedIn format), conversion of non-fixed format resumes to the LinkedIn format and finally ranking the candidates based on their suitability for the specific job position.

Parsing non-fixed format resumes turned out to be a complicated task, because of the diversity in resumes' structures (tabular, list, dual-list, and other formats). However, assuming that all resumes are presented in PDF format, authors have used such tools as "pdftohtml" and "Apache Tika" to convert resumes into raw text that can be further processed and structured. To process the raw text, there were used two different approaches – heuristics and vison-based techniques, however, the last one did not bring any sufficient results. Applying the heuristics approach included using various techniques to extract the text within its logical flow (that can appear to be difficult in the cases when resumes have two-column format) and then to cluster the text into categories such as previous work experience, soft and hard skills, etc. Further, the segmented texts were used to convert previously non-structured resumes into LinkedIn format resumes. Building the parser for LinkedIn format resumes also involves using the heuristic approach, which allowed one to extract separate details of personal information, career path, and also recommendations and transform it finally into a JSON or CSV file.

Later, the authors proceeded with performing the ranking task. One of the interesting details about their approach was that they used a set of LinkedIn format resumes that included previous work experience (specifically job responsibilities) to simulate the job descriptions. There were created samples with combinations of responsibilities that belong to one person, that were treated as positive ones, for example, if person P1 had job responsibilities E_{11}, E_{12}, and E_{13}, then all the possible combinations (E_{11}, E_{12}), (E_{11}, E_{13}), (E_{12}, E_{13}) are positive samples. Samples with a set of responsibilities that belong to two different people were treated as negative ones. Thus, it became possible to train the BERT model for sequence pair classification task, which would define whether two job descriptions belong to the same applicant, assigning each pair a score from 0 to 1. Using this approach was beneficial in terms of the opportunity to train the model having no pre-labeled dataset of job descriptions, which allows to save a lot of time and human resources. After training the model on these pairs of competencies, it became possible to use the trained model to perform the matching task on job descriptions and CVs. The accuracy equal to almost 73% that was achieved using this method can be named as a sufficient result for the sentence pair classification task. BERT models are powerful state-of-the-art instruments as they enable encoding of the full word sequence and learn the context of a word based on the words to the left and to the right of it (Mohhamad, 2022). The type of BERT training process that was used by the authors in this article is named "next sentence prediction" – after receiving two input sentences, the model tries to predict whether the second sentence could come after the first one in the original text (Horev, 2018).

Shakya & Paudel (2019) provide their solution for matching the candidate with the job position by the use of ESCO (European skills, competencies, qualifications, and occupations) ontology by taking into consideration several criteria (skills, qualifications, and experience). The prototype of their system was tested on the dataset of IT-related occupations.

The methodology that is described by the authors includes several main steps, such as:

- collection and pre-processing (basically a preparation of data and transforming it into a uniform format, consisting only of skillset, experience, education, and salary expectations) of CVs and JDs
- mapping CVs and JDs skillsets to ESCO ontology
- linking CVs and JDs based on their semantical meaning using ESCO ontology thus calculating the semantic distance between different terms used for the definition of the skillset
- filtering out unsuitable CVs using the calculated semantic distance score
- ranking the CVs left after filtering based on multiple criteria using the fitness score

The evaluation of the proposed solution performance was done by 7 IT specialists, working in the industry for at least 5 years, who were to assign the evaluation score to the CV and rank the CVs for job descriptions. After that, there was calculated the difference between the scores assigned by evaluators and by the system and provided the measurements for an absolute error and average error. In total, the validation of the proposed system was tested on 14 JDs and 15 CVs (out of 110 JDs and 1060 CVs in total).

Phan et al. (2021) describe the use of the CSO (Computer Science Ontology) classifier to automatically classify the JDs and CVs based on their content using ontology. CSO classifier is based on using two different modules – semantic and syntactic. As can be seen in figure 6, the semantic module is used to find the entities within the text, and identify, select, and rank the concepts. This is done via the use of word embeddings that were generated using the Word2Vec model. Out of words, Word2Vec model produces vectors, named word embeddings that numerically represent word features (Dutta, 2022). There exist two main types of Word2Vec models – Continuous Bag of Words (predicts a word by a context) and Skip-Gram (on the contrary, is aimed to predict a context by a given word) (Meyer, 2016). Another

algorithm is the syntactic one, it is needed to pre-process the text, by removing stop words, and to map n-gram chunks to the concepts. After that, for each n-gram, there is Levenstein distance similarity calculated with the labels of topics in ontologies. Levenstein distance measures the similarity between two strings, that is defined by the number of deletions, insertions, and substitutions that are needed to be performed to modify one string to another (Haldar & Mukhopadhyay, 2011).

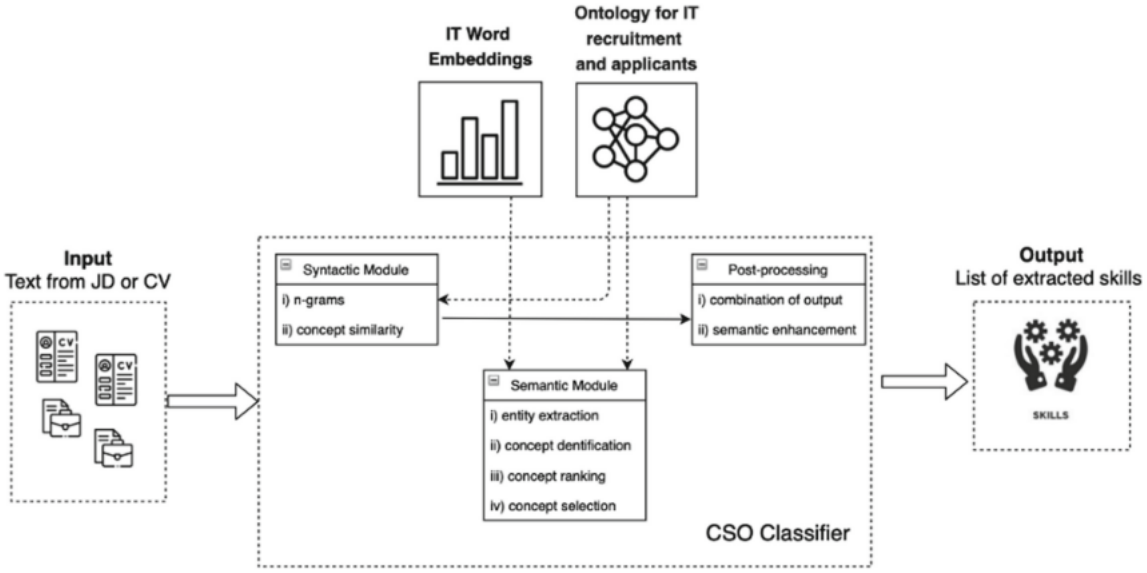

Figure 6. The architecture of CSO classifier (Phan et al., 2021)

The algorithm for the resume searching system included three main modules – resume parser module, extracting skills module, and matching module. The way of parsing resumes authors used is relatively the same as in the article described previously (Bhatia et al., 2019), they have got the resumes in many different forms such as PDF and Word and then transferred them into the raw text. However, there were different packages and techniques used to convert the text from pdf format, including optical character recognition. After that, the raw text was pre-processed, which included the following steps – removing HTML tags, separating the text into sentences, removing stop words and special characters, and lemmatization (an approach that is used to produce the root forms of the word) (Jabeen, 2018). Further, from the resumes, there were extracted the skills by four main domains – education, general technical skills, domain technical skills, and soft skills. These extracted skills were used to create a domain-specific ontology. The created ontology was "fed" to the CSO classifier and further, from the

skills extracted by the CSO classifier there were generated skill graphs for each CV and JD. The matching algorithm was based on calculating the graph edit distance score between each CV and JD, and then the ranking procedure was performed.

Lad et al. (2022) propose using the KNN algorithm to define the candidates with the CVs that are the closest to the analyzed job position. The k-nearest neighbors (KNN) algorithm is a supervised machine-learning algorithm that is used to solve both regression and classification tasks (Harrison, 2019).

Figure 7 demonstrates the diagram of the proposed system presented. The system is a solution adapted for recruiters, that allows them to select a specific job description and scan the resumes, which will be automatically converted into text and pre-processed. Also, the algorithm for removing stop-words (words that do not add additional meaning to the context) was applied. To extract the main information from the resumes the authors have used Tf-Idf (term frequency and inverse document frequency) method (Mansour, 2021), which allows to understand which words are the most important and relevant within the text.

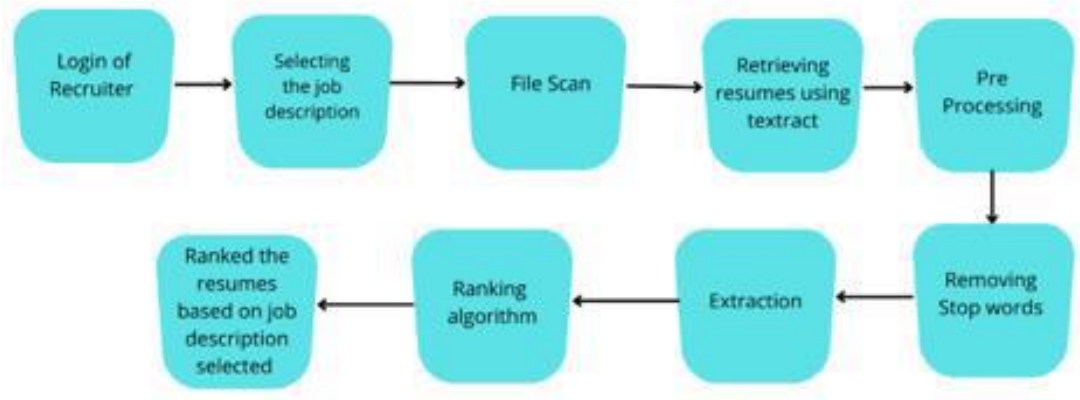

Figure 7. Block diagram of the proposed system (Lad et al., 2022)

Moving to the ranking part, let us discuss the peculiarities of using a KNN algorithm for this task. The k-NN algorithm was used to find the CVs that are the most suitable for the specified job description based on their (CVs) vector distance from the ideal vector (that is a job description for which the perfect candidate is needed to be found). So, for a given vector of job description, k closest points (resumes vectors) need to be found, and then sorted based on the distances. For this purpose, there is the square Euclidian distance between two vectors

calculated, so that if vector $v_1(w_1, x_1, y_1, z_1)$ and vector $v_2(w_2, x_2, y_2, z_2)$ are as stated, then squared Euclidian distance = $(w_1-w_2)^2 + (x_1-x_2)^2 + (y_1-y_2)^2 + (z_1-z_2)^2$.

The authors have discussed and used three alternative algorithms for computing nearest neighbors – brute force, KD tree method, and Ball tree method, that were analyzed in terms of their usage for the specific dataset size, data structure, k (number of closest points), and number of query points. For example, the brute force method (that is a very general problem-solving technique that implies systematic enumeration of all possible candidates and checking whether they satisfy the problem statement (Rasool et al., 2012)) can be only applied when the training dataset is less than 30 in size. KD Tree method is more efficient in comparison to brute force because it is aimed to decrease the number of calculations of the distance, based on the principle that, if point A is far from point B, and point B is close to point C, then point A is also far from point C. This allows not to calculate the distance between points A and C. This improvement in calculation time allows using this method for bigger datasets. The Ball trees method in its turn can be named as an improvement of the KD tree method because it is more efficient in terms of working with high dimension vectors.

3. Methodology

This chapter discusses the applicability of natural language processing concepts and models to taxonomy enrichment and vacancy – CV matching tasks.

3.1. Natural Language processing and text data mining

The study of natural language processing looks into how computers can process or comprehend human languages to carry out useful tasks (Deng & Liu, 2018). Human languages in this case can be both in the format of text and speech. There are various examples of application of NLP models (Zong et al., 2021):

- Text classification – means dividing a text into pre-defined text categories based on the content
- Text clustering – means dividing a text into categories that are not pre-defined based on different perspectives

- Topic model – means defining a topic of a text based on its content
- Text sentiment analysis and opinion mining – means defining the subjective information expressed in the test, analysis of viewpoints and attributes within a text
- Topic detection and tracking – means screening various text topics in order to, for example, track the hot topics
- Information extraction – means the extraction of information (entities, relationships between them) from semi-structured and unstructured texts. This current thesis is related to performing the task of text data mining
- Automatic text summarization – means generating a summary from a text

The models that perform all the text data mining tasks mentioned consist of the number of "pipes", each of them performs a specific task within language processing, such as "reading" the raw text, splitting the text into sentences, gaining the semantic meaning, capturing specific words within text and others.

3.2. Information Extraction

As it was mentioned previously, information extraction is text data mining task aimed to extract such information as entities, entity attributes, relationships between entities, and events from unstructured and semi-structured natural language texts (Zong et al., 2021). Russell & Norvig (2003) claim that IE systems are different from information retrieval ones that consider the text as a bag of words and try to find relevant to users' queries texts and full-text parsers that are needed to extract the semantic meaning of the text.

One of the subfields of Information Extraction is Ontology-based information extraction (OBIE) (Konys, 2015). As well as IE systems, ontology-based IE system processes natural language and extracts needed types of information, however, this system is guided by an ontology and present the output using this ontology (Wimalasuriya & Dejing, 2010).

3.3. BERT models

BERT (Bidirectional Encoder Representations from Transformers) – is an open-source machine-learning model for natural language processing (Lutkevich, 2020). Contextual relations between words in a text can be learned using this model. The model is used to pre-train deep bidirectional representations from the unlabeled text by considering both the left and right context simultaneously (Devlin et al., 2019). So, the model learns the context of the word using all the words around it. Usually, transformers have two mechanisms, the encoder (Moses (2021) states that the role of the encoder is to process each token in the input sequence into a vector of a fixed length – context vector), and the decoder (the decoder reads the context vector defined by the encoder and predicts the sequence by context vectors of tokens (Moses, 2021)), however, BERT models use just encored. This encoder allows to convert the input text tokens (building blocks of natural language – words, characters, or subwords (Pai, 2022)) into vectors, which are later processed by the neural network to obtain the sequence of vectors, that would determine the meaning of words within the context (Sabharwal et al., 2021). These generated vectors that represent the words are named word embeddings (Yang & Tao, 2017). Now, let us discuss more how these embeddings are generated. The representation of the token is constructed by using token, segment, and positional embeddings (fig. 8).

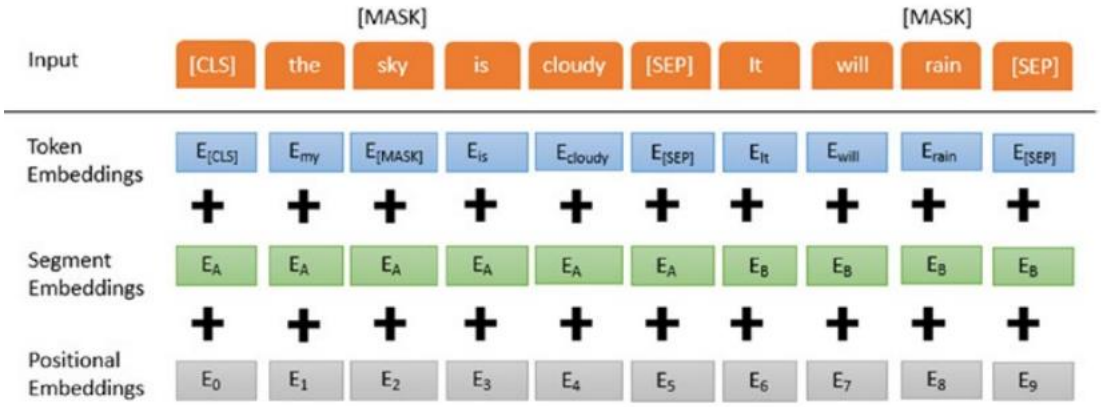

Figure 8. Embeddings calculation with the BERT model (Sabharwal et al., 2021)

Token embeddings mentioned in figure 8 refer to the embedding obtained from the WordPiece vocabulary. WordPiece is a single-word tokenization algorithm used in NLP.

Vocabulary initialization begins with individual characters of the language, followed by iterative adding of the most frequently occurring combinations of symbols to the vocabulary (Schuster & Nakajima, 2012). The meaning of segment embeddings lies in the calculation of embeddings for a pair of sentences and understanding the difference between them. Position embeddings are needed to calculate the unique positional embedding to take into account the position of words within a sentence.

3.4. SBERT models

The BERT model has become a base for various other models, one of them being the Sentence BERT model. The Sentence BERT model is a specialized BERT for building sentence embeddings. Previously described BERT model while calculating the sentence's embedding used the simple average of output embeddings for each word in the sentence, however, this approach was too straightforward and did not show sufficient results. A newly developed SBERT model overcomes this issue, by using Siamese and Triplet network architecture, that calculates fixed-sized vectors for input sentences (Reimers & Gurevych, 2019). Siamese neural networks are composed of two identical subnetworks that output two embeddings, that are further used as inputs to the loss function (Mokhtari, 2022). The loss function describes the minimization task for the distance between similar inputs or the maximization task for the distance between dissimilar inputs (Mokhtari, 2022). So, the loss function is measured by the discrepancy between model predictions and actual observed data (Jung, 2022). Training and fine-tuning of the model are performed via annotating the pair of sentences with a similarity score between them thus "telling" the network which pairs of sentences are similar. Further, the sentences can be given to the neural network for it to obtain the embeddings for each sentence in the pair and calculate the cosine similarity score (fig.9). This score obtained by the neural network is compared with the "golden" cosine similarity, thus allowing for calculation of loss function.

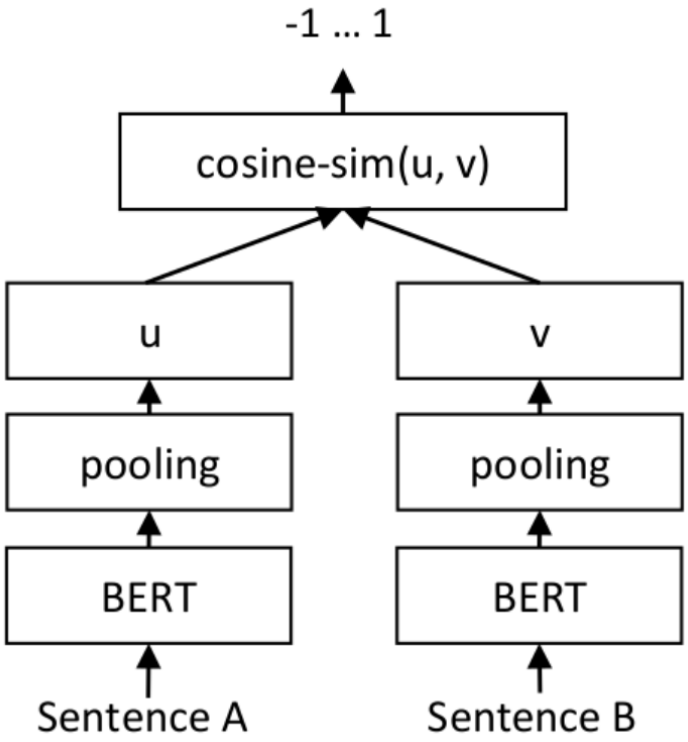

Figure 9. The general architecture of the SBERT model (Devlin et al., 2019)

The tasks that can be named as the most appropriate for the SBERT model are searching for similar sentences and judging the similarity between two sentences based on the cosine-similarity measure (Reimers & Gurevych, 2019).

There are many open-source pre-trained and fine-tuned Sentence Transformer models based on the SBERT model, that are offered by sbert.net. One of the models is "all-MiniLM-L6-v2", that is an all-rounded model tuned for many use cases. The training set for the model was large and diverse, and consisted of over 1 billion training pairs, and also the model allows to generate embeddings for the sentences or phrases using the "model.encode()" function (Sbert.net, n.d.). The model "all-MiniLM-L6-v2" is well-balanced in terms of quality and speed and suits the task, performed in this thesis, which is why it was decided to use this model.

The model "all-MiniLM-L6-v2" was used by Casadio et al. (2022) to produce 384-dimensional embeddings to complete the task of verification specification for natural language understanding classification. This model was also discussed by Jayanthi et al. (2021) while evaluating different pre-trained transformer models for the task of unsupervised Entity Linking (the process of linking a textual name of an entity to a knowledge base entry (Dai et al., 2012)) in task-oriented dialog across five characteristics – syntactic, semantic, short- forms, numeric and phonetic. To evaluate the performance, the authors have annotated 1300 queries from the

dataset and divided them into five pre-defined categories mentioned previously. According to the results they got, the "all-MiniLM-L6-v2" model performed superior to the defined baseline by 5-13% based on the precision criteria (P@1) on these five datasets (figure 10) and particularly showed good results within the semantic subset.

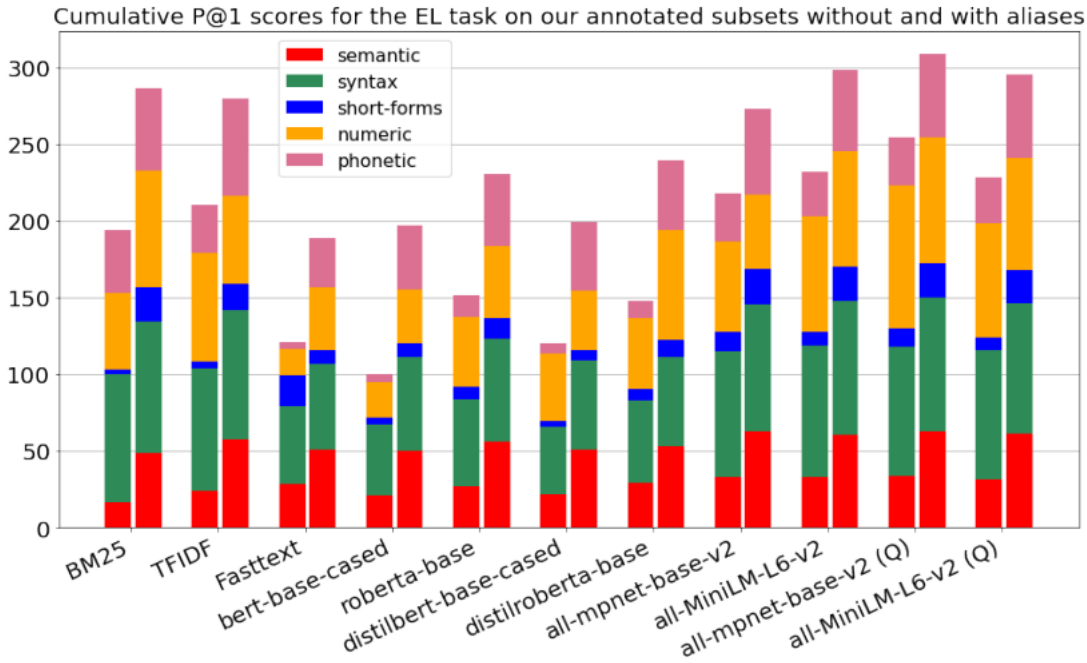

Figure 10. Precision scores of different models tested to perform the Entity Linking task (Jayanthi et al., 2021)

3.5. RAKE algorithm

There are plenty of various algorithms to obtain keywords from vague texts. One of these algorithms is RAKE – Rapid Automatic Keyword Extractor. RAKE is an individual document-oriented dynamic information retrieval method (Sanyal, 2021). This method uses words' frequencies, work collocations, co-occurrences, and linguistic approaches to extract key phrases. RAKE algorithm takes the raw text, splits it into the list of words, removes stop words, and thus presents the Content Words list (list of content-bearing words). Stop words are those uninformative words, that do not add meaning to the sentence, for example, "are", "the", "and" etc., that is why it is better to drop them to make the text clean (Rose et al., 2010). Further, this Content Words list is used to extract keyword phrases. To extract them, the graph of word co-occurrences is generated that helps to evaluate the metric for calculating the scores for the

words based on the word degree (how often the word occurs and how long is the phrase it occurs in), and word frequency (how often the word occurs regardless the number of words with which it co-occurs) (Agarwal, 2022).

The performance of the RAKE algorithm was evaluated as highly accurate by many authors. One of the experiments on the accuracy of the RAKE algorithm describes the benchmark of 7 different keyword extraction algorithms in Python on a corpus of 2000 documents (D'Agostino, 2022). For each of the algorithms used, the author has calculated the average number of extracted keywords, the average number of matched keywords, and the score, which depicts the time needed to find one matched keyword. The results of the benchmark are presented below in figure 11, as can be seen, the percentage of matched keywords is almost the highest for the RAKE algorithm. KeyBERT algorithm, which showed the best result in terms of average keywords matched per document, is however slow in terms of execution time (almost 7 minutes in comparison to 2 seconds for RAKE) and could be not the optimal solution while processing many large documents.

	algorithm	elapsed_time	avg_keywords_per_document	avg_matched_keywords_per_document	avg_percentage_matched_keywords	performance_score
0	rake_extractor	00:00:02	3.8155	2.0720	0.54	0.99
1	yake_extractor	00:00:40	4.6130	2.0655	0.45	0.05
2	topic_rank_extractor	00:45:03	3.3835	1.1665	0.34	0.00
3	position_rank_extractor	00:43:57	3.4035	1.8010	0.53	0.00
4	single_rank_extractor	00:43:47	3.6245	1.9710	0.54	0.00
5	multipartite_rank_extractor	00:43:48	3.4205	1.2165	0.36	0.00
6	keybert_extractor	00:06:57	4.0135	3.8580	0.96	0.01

Figure 11. Benchmark of keyword extraction algorithms in Python (D'Agostino, 2022)

Baruni & Sathiaseelan (2020) present the comparison of using RAKE and TextRank algorithms for key phrase extraction from documents. TextRank is a graph-based ranking algorithm, that extracts words from a document and plots relationships between them on the graph, and then identifies the most crucial ones by calculating the importance scores recursively from the graph (Mihalcea, 2004). To test the performance of the algorithms, the authors have used literature abstracts extracted from Arvix NLP papers, and the results showed that in terms of computations the RAKE algorithm is more efficient than the TextRank algorithm and also achieved higher precision and recall scores.

3.6. Named Entity Recognition

According to Marrero et al. (2013) Named Entity Recognition is a task in Information Extraction that is aimed to identify and classify different types of information elements, called Named Entities. As common types of named entities, there could be named organizations, places, dates, persons, etc.

There are several types of NER systems, that differ in the approach that they use to detect and categorize entities (Goyal, 2021). The first approach is the dictionary-based approach, which relies on a list called a gazetteer that contains named entities (Song et al., 2018). It is the simplest approach, which implies having a dictionary, containing a vocabulary, that is used to match the entities that occur in the text with the items in the vocabulary (Goyal, 2021). So, in order to apply the dictionary-based NER system, one should define the classes of entities and also named entities, that belong to those classes; these named entities and respective classes can be further found in texts. The second approach is a rule-based one, which uses regular expressions (rules and patterns) combining information from terminological resources and the entities' attributes (Eftimov et al., 2017). For example, there could be used a pattern "[{"LOWER": "hello"}, {"IS_PUNCT": True}, {"LOWER": "world"}]", that would mean that there should be found the combination of three tokens in text – the first token, which lowercase form matches "hello", the second one – any punctuation sign, and the third one – token, which lowercase form matches "world" (SpaCy, n.d.). As it can be seen, both of the approaches mentioned are based on manually created lists of entities or rules, and that's why, even though showing good results on similar texts, they could be not that precise when the data changes (Goyal, 2021). In contrast to these approaches, the third approach is based on using ML techniques and allows to tag the named entities even when the terms are not mentioned in the dictionary and the context is not specified in the rule set (Song et al., 2018). Li et al. (2020) describe feature-based supervised learning approach to NER. The essence of this approach is that after obtaining the data annotated with the features representing all possible examples of entities, ML algorithms are used to recognize similar patterns from unseen data (Li et al., 2020).

It is not needed to develop NER for each task, because there are several pre-trained NER models that are implemented in different programming libraries. NER models are provided by open-source NLP libraries, such as SpaCy, NLTK, Stanford Core NLP, etc. (Terry-Jack, 2019). The SpaCy package according to Jugran et al. (2021) has advantages over some other packages, for example, NLTK, because the selection of the best algorithm is

automatic and is based on using an object-oriented approach, thus saving time and increasing accuracy. Naseer et al. (2022) analyzed the performance of different libraries serving NER models and obtained the best result in terms of accuracy and F1-score using SpaCy NER. However, while using dictionary-based NER, there is not much difference in what packages to use, because dictionary-based NER will just look for the exact matches with the concepts stated in the dictionary.

SpaCy is the package, designed to build information extraction of NLP systems, that has 80 trained pipelines for 24 languages, including components for named entity recognition (SpaCy, n.d.). The default NER pipe within SpaCy NLP pipelines is able to tag various named and numeric entities, including companies, locations, organizations, and products (SpaCy, n.d.). The example of extracted entities and their visual representation are presented in figure 12. In the figure it can be seen that three named entities were found – "Apple", that was identified as organization name, "U.K.", identified as geo-political entity, and "$1 billion", identified as money.

Figure 12. Visualization of in-built SpaCY NER (SpaCy, n.d.)

However, apart from the default NER, there could be created custom NER. The custom entity recognition (pipeline component) uses token-based rules or exact matches (SpaCy, n.d.). This custom pipeline component can be added to the trained SpaCy pipeline, for example, using the desired dictionary, by specifying the named entities and also the corresponding label (type of entities).

3.7. Performance measures used to evaluate IE systems (Zong et al., 2021)

To be able to assess the performance of a model it is needed to have a golden standard – a set of correct answers for a sample of typical inputs (Wissler, 2014). In NLP this kind of standard is usually named "golden corpus", and it is a manually annotated collection of text. Later, the outputs of the model are compared to the ones, that are mentioned in the golden

corpus and there is a confusion matrix to be built. This matrix is a two-dimensions table, that consists of combinations of the actual and predicted values. The elements of the matrix are presented in the table with the clarification of their meaning in this specific research.

Table 1. Elements of the confusion matrix (Kulkarni et al., 2020)

Element of the confusion matrix	**Description**
True positive (TP)	A positive example that was classified correctly
True negative (TN)	A negative example that was classified correctly
False positive (FP)	An actual negative example that was classified as positive
False negative (FN)	An actual positive example that was classified as negative

Based on the elements of the confusion matrix it is possible to calculate several metrics that are commonly used to evaluate the quality of information extraction models. These are accuracy, precision, recall, and F-measure.

$$Accuracy = \frac{TN+TP}{TN+FP+FN+TP} \quad (1)$$

Accuracy metric shows in general how well the model can detect items, so basically, it calculates the ratio of all correctly classified cases to the total number of cases. However, this metric can be misleading if the dataset is imbalanced (Kulkarni et al., 2020).

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

As can be seen from the formula, precision depicts the proportion of predicted positive cases that are in fact positive.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Recall in its turn shows the proportion of real positive cases that are correctly predicted as positive.

It is obvious that these two metrics – precision and recall, are focused only on detecting positive cases, not considering real negative cases. Also, based on the formulas, there is a trade-off between these two metrics, which is why there was another metric suggested F-measure, that serves as a harmonic mean between precision and recall (Powers & Ailab, 2011).

$$F - measure = \frac{2 * precision * recall}{precision + recall} \quad (4)$$

These metrics mentioned will be used in this thesis to evaluate the performance of the Sentence Transformer (SBERT) model applied to the task of extracting the competencies from the sentences within job descriptions. The golden corpus made for the evaluation of the performance of the model consists of 100 sentences randomly chosen from job descriptions and the set of 5 competencies identified for each of the sentences. The choice of exactly 5 competencies is explained by the fact that none of the sentences from the golden corpus have more than 5 competencies identified from the sentence. This allows us to assume, that such situations where more than 5 competencies could rarely or never can be met in other sentences within the JDs dataset. If there are less than 5 competencies from the taxonomy that could be derived from the sentence, then the competency is marked as "new". So, the set of competencies for 1 sentence could, for example, look the following way: S_{GC} = {"new", "C1", "C2", "C3", "C4"}, where "C1", "C2", "C3", "C4" represent the competencies from the taxonomy, that were mentioned in this sentence, and "new" represents the competency that is not present in the taxonomy. The same as for the golden corpus, if the model can extract less than 5 taxonomy competencies from the sentence, then the competency is also marked as "new". For example, the set of competencies extracted from the sentence by the model could be presented in the following way: S_M = {"new", "new", "C1", "C2", "C3"}, which means that there are 3 competencies from the taxonomy that were detected by the model, and 2 competencies, that are not present in the taxonomy. Thus, to obtain the elements of the confusion matrix, for each of 100 sentences within the golden corpus, these two previously mentioned sets (set S_{GC} of competencies stated in the golden corpus and set S_M of competencies defined by the model) are compared. So, for each sentence "i" within the golden corpus the number of elements of the confusion matrix identified is equal to 5: $TP_i + TN_i + FP_i + FN_i = 5$. The description of the elements of the confusion matrix for the evaluation will be the following as in table 2.

Table 2. Description of the elements of the confusion matrix to evaluate the performance of the Sentence Transformer model

Element of the confusion matrix	**Description**
True positive (TP)	The competency identified by the model is present in the set of competencies in the golden corpus
True negative (TN)	The competency identified by the model is "new" and in the set of competencies in the golden corpus it is also "new"
False positive (FP)	The competency identified by the model is not present in the set of competencies in the golden corpus
False negative (FN)	The competency identified by the model is "new" and "new" is not present in the set of competencies in the golden corpus

3.8. Similarity measures between sets of competencies for JDs and CVs

After extracting competencies from JDs and CVs using the chosen taxonomy, it is needed to compare them to one another in order to obtain the ranking of the CVs based on their suitability for each of the job positions.

Vijaymeena & Kavitha (2016) suggest several term-based similarity measures:

- Cosine similarity, that measures the angle between vector representations of documents d_1 and d_2:

$$S_{Cos}(d_1, d_2) = \frac{d_1 * d_2}{(d_1 * d_1)^{1/2}(d_2 * d_2)^{1/2}}, \quad (5)$$

where $d_1 * d_2 = \sum_{i=1}^{n} d_{1,i}\, d_{2,i}$, $(d_1 * d_1)^{1/2} = \sqrt{\sum_{i=1}^{n} d_{1,i}^2}$, $(d_2 * d_2)^{1/2} = \sqrt{\sum_{i=1}^{n} d_{2,i}^2}$

Thus, for example, vector representation of document 1 being d_1=[1,1,0,2,1] and vector representation of document 2 being d_2=[1,0,1,1,0], cosine similarity is equal to $S_{cos}(d_1, d_2) = \frac{(1*1)+(1*0)+(0*1)+(2*1)+(1*0)}{(1+1+0+4+1)^{1/2}*(1+0+1+1+0)^{1/2}} = 0{,}65$

- Dice's coefficient, that calculates as double number of terms common in two documents divided by total number of terms present in both documents:

$$S_{Dic}(d_1, d_2) = \frac{2* d_1 * d_2}{d_1 * d_1 + d_2 * d_2} \quad (6)$$

- Euclidean distance, calculated by the square root of the sum of squared differences between two documents elements:

$$S_{Euc}(d_1, d_2) = [(d_1 - d_2) * (d_1 - d_2)]^{1/2} \quad (7)$$

- Jaccard similarity, that shows the proportion of shared terms in the set of all unique terms from both documents (basically, size of intersection divided by the size of union):

$$S_J(d_1, d_2) = \frac{d_1 * d_2}{d_1 * d_1 + d_2 * d_2 - d_1 * d_2} \quad (8)$$

Fletcher & Islam (2018) used Jaccard similarity to calculate the similarity between sets of patterns/criteria, that were treated as single elements within the set (the same can be applied to the sets of competencies, which can be considered as single terms or elements within the sets). This choice is explained by conceptual simplicity – this measure is very straight-forward in terms of its interpretation – it can be intuitively understood; and by computational simplicity – the intersection and union can be easily and rapidly calculated.

When it comes to the analysis of competencies, the measures that are clearly understandable and can be easily interpreted are preferable. Let's discuss the applicability of Jaccard similarity measure to the JDs and CVs data. Jaccard similarity could be calculated by dividing the number of intersections between JD's set of competencies and also the set of competencies from CV, by the number of all unique competencies in these two sets. However, in terms of meaning, this measure can be to some extent inappropriate. As it is needed to understand how suitable each candidate is to the specific job position in terms of skillset they have, it is important to consider not the union of both sets, but only the set of competencies from the job description. This can allow us to understand, how many competencies specified in the JD does each candidate have. So, the formula can be modified the following way:

$$Sim(s_1, s_2) = \frac{s_1 * s_2}{s_1 * s_1}, \qquad (9)$$

$$where\ s_1\ is\ a\ set\ of\ competencies\ from\ a\ JD, and\ s_2\ is\ a\ set\ of\ competencies\ from\ a\ CV$$

So, for example, if in the job description there are 17 competencies mentioned, and the CV's skillset contains 8 competencies from the JD competency list, then the similarity of JD and CV is equal to 8 (intersection between JD and CV sets of competencies) divided by 17 (number of competencies in JD set), so equal to 0.47.

This measure can be easily calculated, it is correct in terms of interpretation, however, it still has some drawbacks:

- It treats all the competencies, mentioned in the set from JD equally. So, the weight of the hard skill related to the use of Python and the weight of the soft one, for example, "Dedication" will be the same, which is not really true in the reality
- It does not allow to choose the core competencies from the JD, so for example, if the employer states in the JD that the competency in developing implementation strategies for the new systems is a must and experience in using MS SQL can be a plus, these competencies will be treated the same

3.9. The description of the algorithm used to extract competencies using the Sentence Transformer model

The research that I have previously performed is related to the competency demand analysis (Riabchenko & Zheleiko, 2022). The goal of the research was to determine core competencies for a couple of specified job positions (business analyst and business architect) based on the analysis of job descriptions. The goal was fulfilled by using the same taxonomy of competencies that is used in this thesis as well. The current thesis is partially built upon the findings of the previous research in terms of the models and algorithms used. Specifically, the keyword extraction algorithm and, also the model used to extract the competencies from the job descriptions based on the taxonomy using the cosine similarity score. In this sub-chapter, there will be provided the comparison of models and algorithms used in this research.

In terms of keyword extraction, there were used two different approaches. The first method was using Rule-Based Matching. This algorithm is pre-built in the SpaCy package in Python. As it is clear from the name, this method involves using a system of rules, that would define the keyword phrases. As it is needed to extract the phrases that would likely be competencies, it was needed to understand what linguistic constructions are used to define the competency. Thus, there were defined several rules, for example, a phrase that should be extracted contains the word with the ending "-ing" ("Performing stakeholders' analysis"). Also, another approach takes into account different parts of speech, so the rule could be the following – a phrase that should be extracted has the construction "Verb + Conjunction + Verb + Noun" ("Analyzes and communicates solutions"). In total, there were 12 rules defined to capture the possible phrases containing competencies. Even though this method is beneficial, because it allows for a particular tailor-made solution, it did not show sufficient results – for some sentences, there were no phrases extracted. Thus, to extract keyword phrases there was chosen RAKE algorithm described previously in the Methodology chapter. The advantage of the algorithm is that there is no need to specify all the rules for the extraction of key phrases and also, there is a deep analysis of the text behind the algorithm, that allows to extract meaningful and semantically correct phrases. Thereby, the RAKE algorithm proved its efficiency while performing a similar task on similar data and it can be used for the purposes stated in this thesis.

In terms of extraction of the competencies from the job descriptions based on the taxonomy, there were used several methods – similarity tool from SpaCy package, custom Word2Vec model and, also SBERT (Sentence Transformer "all-MiniLM-L6-v2") model. The SBERT model was tested on the whole sentence and also on the phrases extracted from the sentence. All these methods are based on the calculation of cosine similarity between two phrases/sentences, which would depict whether these two phrases/sentences are semantically alike or different. For each of the models, there were tested several threshold values – minimum similarity scores that would mean that two phrases/sentences can be named semantically alike. The threshold values were tested in the range from 0.4 to 0.85, and there were chosen the one, that allow for the best performance of each model.

The estimations of models' performances with the optimal threshold are presented in table 3. As can be seen, the similarity tool from the SpaCy package (threshold value equal to 0.7) showed the worst result with F-score equal to just 6%, which can be explained by the fact that it is a pre-built function within the package, that was first of all developed to calculate the similarity between words, not phrases. Custom Word2Vec model (threshold value equal to

0.7), which was trained on job descriptions data, also showed insufficient performance with an F-score equal to 28%, one of the reasons behind it can be that embeddings (vector representations) for the whole phrase were calculated by just averaging the embeddings for the words within the sentence. The SBERT model on sentences (threshold value equal to 0.45) has performed better than the models described previously with an F-score equal to 62%. SBERT model on phrases (threshold value equal to 0.55) demonstrated the best result based on evaluation criteria, which can be explained by its focus on calculating sentence or phrase embeddings and the use of sophisticated algorithms, that are based on using Siamese and Triplet neural networks, and also by the fact that extracting key meaningful phrases can help in matching them with the competencies from the taxonomy.

Table 3. Comparison of the performances of different NLP models

Model	**Accuracy**	**Precision**	**Recall**	**F-score**
SpaCy similarity tool (threshold – 0.7)	0.21	0.09	0.04	0.06
Custom Word2Vec model (threshold – 0.7)	0.29	0.63	0.18	0.28
SBERT model on phrases (threshold – 0.55)	0.60	0.73	0.59	0.65
SBERT model on sentences (threshold – 0.45)	0.55	0.80	0.51	0.62

Based on the comparison of the models used, it is suggested to use the SBERT model with a threshold value equal to 0.55 for the taxonomy enrichment task in this thesis.

Thus, the algorithm consisting of extracting the key phrases from the sentences from JDs using RAKE and applying SBERT, particularly the Sentence Transformer "all-MiniLM-L6-v2" model, can be named the best among all the tested algorithms.

3.10. Taxonomy enrichment

Taxonomies represent the hierarchical relationships between concepts or entities (Takeoka et al., 2021). They are widely used in tasks that are related to information retrieval, recommendation, and classification (Huang et al., 2019).

The taxonomy enrichment task is in the discovery of emerging concepts and the attachment of them to the existing taxonomy (Mao et al., 2020). Tikhomirov & Loukachevich (2021) define the task of taxonomy enrichment as finding an appropriate higher-level concept from a given taxonomy for a new word that can be considered as a class for this word. Nikishina et al. (2022) state that taxonomy enrichment is a task, that aims to associate each new word that is not yet present in the taxonomy with the respective hypernyms of it. Usually, it is more beneficial to assist manual labeling with automatic annotation systems to make the process of enrichment less time-consuming (Nikishina et al., 2020). Takeoka et al. (2021) propose using a framework that uses a classifier based on pre-trained language models to determine whether an inputted term pair have hierarchical relationships. Arslan & Cruz (2022) have suggested performing semantic taxonomy enrichment by applying BERTopic that is a Neural topic modeling using contextualized BERT to add additional concepts into the given business taxonomy by analyzing online news documents data. However, the methods mentioned were used on the large scope of data and sophisticated taxonomies.

Enrichment of existing taxonomies with new concepts, in general, allows to better capture the dynamic world (Arslan & Cruz, 2022). Whenever new data and new concepts from the same domain of knowledge become available, it becomes possible to enrich and update existing taxonomy (Biswas, 2020).

3.11. Use of Python as a programming language for performing NLP tasks

The worldwide use of Python is explained by its numerous advantages. Apart from the general ones, which include interpretability (no need to compile the code before executing), interactivity, and object orientation (the easier process of writing the code, because it is encapsulated within objects), Python is also very convenient and easy to use for Machine Learning tasks (Charatan & Kans, 2022). Particularly, Python can be very useful in terms of using it to solve NLP tasks. There are many libraries, models, and tools developed for solving the problems of language processing. These libraries provide easy-to-implement ready-to-use solutions, that allow for a convenient and fast process of writing code. The examples of NLP Python packages that were used in this thesis are provided in table 4.

Table 4. Python libraries for NLP

Name of Python package	**Use of the package**
Selenium	Web scraping (navigating to web pages, loading elements, buttons clicking, page scrolling), parsing pages contents
SpaCy	Sentence detection, tokenization, pre-processing functions (Lowercase, lemmatization, removing punctuation, removing stop words), NER
Rake_nltk	Keyword phrases extraction
Sentence_transformers	SBERT model
WebDriverManagement	Carrying out the management of the drivers required by Selenium WebDriver

3.12. Monster.com as a source of JDs for taxonomy enrichment and matching tasks

On the official website of Monster Worldwide, Inc. (Monster, n.d.) it is stated that "Monster is a global online employment solution for people seeking jobs and the employers

who need great people". Nowadays, the company has been operating on the market for more than 20 years and offers services in more than 40 countries.

According to Semrush traffic statistics, in October 2022 monster.com received 16.2 million visits (Semrush, n.d.). Based on the information published on the official website, every minute on monster.com there are 7,900 jobs searched and 2,800 jobs reviewed. All of the mentioned allowed the service to appear in the top 10 best job search websites according to the research held (Polner, 2022). That is why, this website can be considered a good source of JDs for this thesis.

4. Solution development

Based on the analysis of literature and possible ways to implement the goal of the thesis, there was developed the algorithm, that includes several steps:

- Collect the job descriptions for the job positions of "business analyst", "enterprise architect", "data analyst" and "data architect" from the website Monster.com
- Collect CVs for the job positions of "business analyst", "enterprise architect", "data analyst" and "data architect"
- Build the model to extract the competencies from job descriptions based on the taxonomy of skills using the SBERT model and RAKE algorithm
- Calculate accuracy, precision, recall, and F-score metrics and assess the performance of the SBERT model based on evaluation criteria (F-score)
- Enrich the taxonomy of skills with new competencies extracted from job descriptions by adding the competencies in the corresponding section of the taxonomy or creating a new section of the taxonomy
- Build an algorithm to match the job descriptions and CVs using NER and Sentence Transformer model
- Assess the quality of the algorithm by comparing the results on resume ranking with the expert's evaluations

So, on the higher level two main tasks should be performed, firstly, the enrichment of the taxonomy and secondly, matching the job description with the most suitable CV based on the enriched taxonomy.

4.1. Data collection and pre-processing

Firstly, it is needed to collect job descriptions and resumes for taxonomy enrichment and matching tasks.

4.1.1. Job descriptions

The first step in this research is data collection and pre-processing. To collect job descriptions there was used previously mentioned Python library "Selenium". This library allows for web scraping and storing the data from web pages in a suitable format. So, firstly there was done the installation of "Selenium" and "WebDriverManager" packages. Further, it is possible to create a web driver instance and open the needed page in the used browser (a page with the job descriptions on the specified job position).

Later, there was performed an analysis of the code of the page to understand the paths to the needed data. The data, that is needed to be stored includes the name of the job position (it is also important to later check whether all the collected job descriptions describe the specified job position) and the part of the job description where required skills and qualifications are mentioned. The procedure of finding the needed information is the following – it is needed to find the job description title on the page (for example, "Data Architect"), store this title, click on the title, find the sections that describe required skills and qualifications within the job description, store the information from these sections, proceed to the next job description and scroll the page to obtain more job descriptions. The task of finding the title of the job description is performed by using the "driver.find_elements" function and specifying the class name of the object. Thus, it is possible to return all the elements with the matching class name, an example is provided in figure 13.

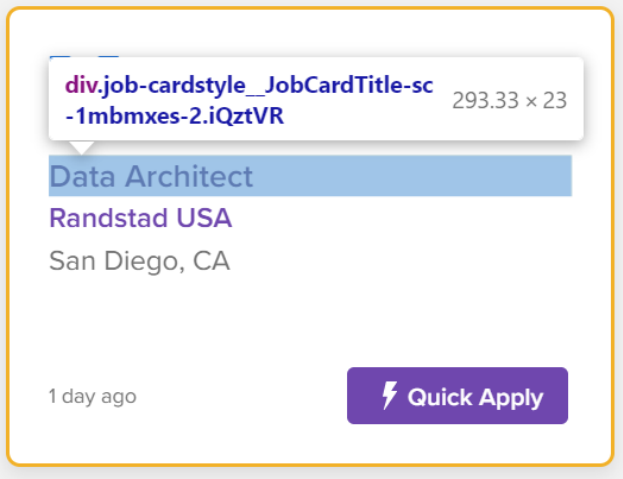

Figure 13. Web scraping “Monster.com”, finding elements by class name

Further, it is needed to store the title of the job position and click it to get access to the corresponding job posting by using the function “click ()”. After reviewing the job posting, it is needed to find specific sections of it. This can also be done by the “driver.find_elements” function, however, the search will be performed using the tag name. In the overwhelming majority of job descriptions, required skills and qualifications, and responsibilities are organized as a bulleted list, that always has the tag “<ul>”. Every element of this list in its turn has the tag “<li>”. Thus, concrete qualifications can be found within the job description (figure 14). The information on competencies needed is stored in the same Pandas DataFrame as the title of the job description.

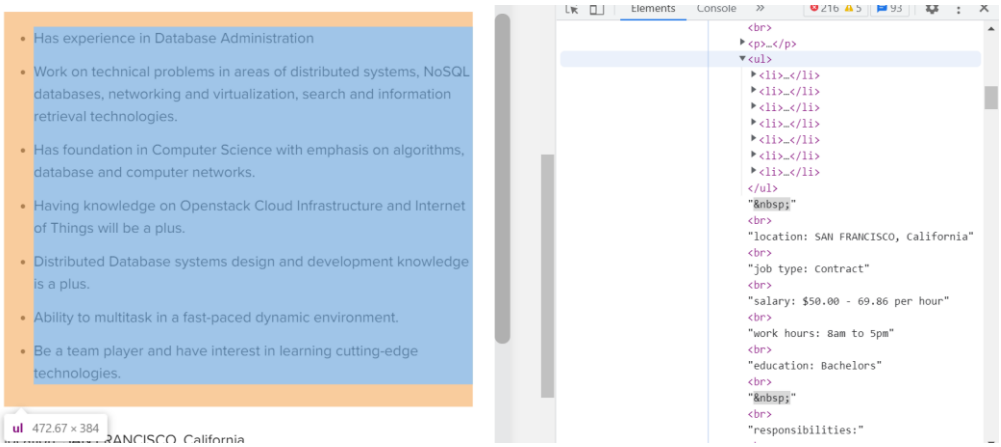

Figure 14. Web scraping “Monster.com”, finding elements by tag name

The procedure described was performed for all job positions – “data architect”, “data analyst”, “business analyst”, and “enterprise architect”.

The initial number of job descriptions collected from Monster.com:

- Data architect – 95 job descriptions
- Data analyst – 198 job descriptions
- Business analyst – 288 job descriptions
- Enterprise architect – 93 job descriptions

After that, it was important to check whether all the job descriptions do correspond to the specified job title (sometimes in the search irrelevant job descriptions can appear), and irrelevant job descriptions were deleted from the dataset. Also, some of the job descriptions appear several times in the search results, so the duplicates were deleted using the function “drop.duplicates()”. The example of the Pandas DataFrame for the “business analyst” job position is presented in figure 15.

	Job title	Skills and qualifications
0	Business Analyst, Allocation - Remote	['1+ years experience in technical support or ...
1	Business Analyst- Oracle RMS	['Identify, understand, and develop ongoing en...
2	Business Analyst II	['Leads the on-boarding and off-boarding tasks...
3	Principal Business Analyst	['Develops and documents workflow, systems req...
4	Sr Business Analyst, Forecasting	['5+ years’ experience as a Sr Business Analys...
...	...	...
148	Technical Business Analysts - Education Experi...	['Supporting external customers with analyzing...
149	Technical Business Analysts - Education Experi...	['Support bi-annual road mapping / planning ef...
150	Business Analyst (Oracle HCM - Benefits)	['Translate the products vision into detailed ...
151	Salesforce Business Analyst	['Gather operational and workflow requirements...
152	Business Analyst	['Learning & understanding our database struct...

153 rows × 2 columns

Figure 15. Example of data collected for “Business analyst” job position

Thus, the input data for the taxonomy enrichment task include recent job descriptions collected from the website Monster.com (skills required and responsibilities sections):

- Data architect – 57 vacancies
- Data analyst – 153 vacancies
- Business analyst – 153 vacancies
- Enterprise architect – 51 vacancies

The text of job descriptions, however, is not clean, because it was parsed directly from the websites, so that includes different redundant punctuation and cases.

The example of the part of the text of the job description for data architect is presented in the table below. As can be seen, the text includes quotes for some of the sentences within a job description, square brackets, and also the "\n" sign that depicts the beginning of the new string. Removal of these symbols was performed by using the "replace ()" method in Python, which allows to replace the specific phrase with another one that is specified. The text after pre-processing looks in the following way (table 5).

Table 5. Example of JD text from Monster.com website before and after pre-processing

The original text of JD parsed	**Pre-processed text of JD**
"['Leads data analysis, database design, develop Logical and Physical Data Models, including Relational and Dimensional Models, prepare Mapping documents with all the transformation logics, BI Objects, Data Quality Rule creation/ enforcement, etc. ', 'Write DDL script to create/update database structures supporting Oracle and Redshift either manually or through erwin, ability to tune database performance.\nAnalyzes multi-data mart system integration opportunities and challenges. Proposes and communicates	"Leads data analysis, database design, develop Logical and Physical Data Models, including Relational and Dimensional Models, prepare Mapping documents with all the transformation logics, BI Objects, Data Quality Rule creation/ enforcement, etc. Write DDL script to create/update database structures supporting Oracle and Redshift either manually or through erwin, ability to tune database performance. Analyzes multi-data mart system integration opportunities and challenges. Proposes and communicates

recommended solutions. Recommends testing methodologies for the proposed design. \nWorks individually and with peers to perform metadata management across erwin, Oracle, Redshift, Informatica and MicroStrategy.\n]"	recommended solutions. Recommends testing methodologies for the proposed design. Works individually and with peers to perform metadata management across erwin, Oracle, Redshift, Informatica and MicroStrategy."

4.1.2. Resumes

The resumes that need to be collected should represent the analyzed list of job positions: "business analyst", "data analyst", "data architect" and "enterprise architect" and also some job positions that are not related to this list – "Data scientist", "HR specialist" and "Sales manager", etc. Resumes on irrelevant job positions are needed in order to assess the quality of the algorithm – whether irrelevant resumes do not appear in the list of the top-matched resumes for each of the analyzed job positions. This type of performance evaluation will be used as an extra confirmation of the quality of the algorithm in addition to expers's evaluation.

In terms of resumes, there were 3 sources of data collection. The first one is an open-source dataset of resumes from Kaggle.com (Bhawal, 2021). This dataset consists of more than 2400 resumes from 24 different categories, such as:

- HR – 110 resumes
- Designer – 107 resumes
- IT – 120 resumes
- Teacher – 102 resumes
- Advocate – 118 resumes
- Business development – 120 resumes
- Healthcare – 115 resumes
- Fitness – 117 resumes

- Agriculture – 63 resumes
- BPO – 22 resumes, etc.

As for the job positions, analyzed in this thesis, the dataset includes 4 CVs for "Business Analyst" and 2 CVs for "Data Analyst".

Another Kaggle dataset used in this thesis is "Update Resume Dataset" (Kaggle, 2022), which consists of more than 900 resumes, for such job positions as "Data Science", "HR", "Web design", "Business Analyst", "Sales" etc. However, this dataset contains the resumes for only one job position from the list of analyzed job positions – "Business analyst" and there were only 6 unique resumes for this job position.

In order to obtain the resumes for the job positions, analyzed in this thesis, there was used another source of resumes – "resumeworded.com" website. Resume Worded (Resume Worded, n.d.) provides resume review services and examples of possible CVs for the list of job positions. Resume Worded has compiled hundreds of resumes across various industries, that were analyzed by recruiters to provide resume samples (Resume Worded, n.d.). So, these resumes can be presented as examples of how appropriate and suitable resumes should look. So, the number of resumes collected from Resume Worded official website is the following:

- "Business analyst" – 6 CVs
- "Data analyst" – 11 CVs
- "Data architect" – 10 CVs
- "Enterprise architect" – 10 CVs

Thus, in total, the dataset of resumes consists of almost 3500 resumes, including:

- "Business analyst" – 16 CVs
- "Data analyst" – 13 CVs
- "Data architect" – 10 CVs
- "Enterprise architect" – 10 CVs

As well as job descriptions text, the text of CVs needs to be pre-processed, because it includes some meaningless symbols (“â€ ¢”, “â□¦” etc.), “\n” sign that depicts the beginning of the new string, “\r” sign – a carriage return character and “\t” sign that depicts a tab character. Removal of these symbols was performed by using the “replace ()” method in Python, which allows to replace the specific phrase with another one that is specified. Pre-processed CVs are stored in the Pandas DataFrame format (fig.16).

	Candidate №	Occupation	CV
0	Candidate 1	data analyst	professional experience.resume worded, new yor...
1	Candidate 2	data analyst	resume worded university.boston, ma.bachelor o...
2	Candidate 3	senior data analyst	experience.2016-2022.resume worded & co.san fr...
3	Candidate 4	marketing data analyst	resume worded, new york, nyjun 2020 –2022.mark...
4	Candidate 5	financial data analyst	resume worded & co.oct 2017 –2022.financial da...
5	Candidate 6	data analyst	resume worded, london, united kingdom.educatio...
6	Candidate 7	junior data analyst	junior data analyst with 10 years of experienc...
7	Candidate 8	healthcare data analyst	healthcare data analyst with 10 years of exper...
8	Candidate 9	business data analyst	business data analyst with 10 years of experie...
9	Candidate 10	power bi analyst	power bi data analyst with 10 years of experie...

Figure 16. Extract from the CVs dataset

4.2. Taxonomy enrichment

After pre-processing the job descriptions dataset, it is possible to proceed with building the model to extract the competencies from the job descriptions based on the existing taxonomy of skills and enrich the taxonomy with new competencies. The general algorithm is the following:

- Extract key phrases from each sentence within each job description in the dataset using the RAKE algorithm
- Apply the Sentence Transformer (SBERT) model to the extracted key phrases and competencies from the taxonomy and obtain cosine similarity scores for each pair of key phrase and competency

- For each key phrase rank the competencies from the taxonomy based on the cosine similarity score and obtain the top five most alike competencies from the taxonomy
- Aggregate the extracted competencies from the key phrases with the corresponding cosine similarity scores within the sentence they belong to
- Rank the competencies within the sentence based on the cosine similarity score and obtain the top five competencies
- For each sentence mark the corresponding competency from the taxonomy as "new" if the cosine similarity between the sentence and the competency from the taxonomy is less than the stated threshold value
- Analyze the sentences with the competencies extracted as "new" to enrich the taxonomy
- Build the golden corpus to estimate the performance of the model and assess it using accuracy, precision, recall, and F-score metrics

4.2.1. Key phrases extraction

As was stated, the first step is to extract the key phrases from each sentence within each job description. For that, it is needed to import the RAKE algorithm from the "rake_nltk" package. Further, it is possible to use the algorithm on the job descriptions. The extracted phrases are stored in the nested list, on the upper level of which there is the job description, on the middle level – a sentence within job the description, and on the bottom level – key phrases within the sentence. The key phrases that contain just one word were excluded from the list due to the fact, that one word could barely have any semantic meaning and represent competency.

For example, for two job descriptions containing two sentences each, and two key phrases for each sentence, the nested list can look the following way – "[[['maintains data architecture', 'leads data analysis'], ['supporting planning activities', 'recommends testing methodologies']], [['problem-solving capabilities', 'communicates recommended solutions'], ['continuously learning new data technologies', 'data architecture – data models']]]".

It is important to mention, that most of the key phrases that were extracted are semantically correct and meaningful and could depict competencies, as can be seen from the example above.

4.2.2. Application of Sentence Transformer model to the extracted key phrases

After obtaining key phrases extracted from the job descriptions it is possible to proceed with running the chosen Sentence Transformer model on the phrases and competencies from the taxonomy.

Firstly, it is needed to import the model in the current workspace in Python. It is done via simple commands:

```
"import sentence_transformers

from sentence_transformers import SentenceTransformer, util

model = SentenceTransformer('all-MiniLM-L6-v2')".
```

After that, the model can be used to generate the embeddings for key phrases and competencies from the taxonomy. For this purpose, there was used `"model.encode(keyphrase/competency, convert_to_tensor=True)"` command. The generated embeddings were used to calculate cosine similarity between each key phrase and each competency by using the in-built `"utils.cos_sim ()"` command. For each key phrase, the competencies were ranked based on the cosine similarity, and the top five competencies for each phrase were chosen. The usage of cosine similarity measure is explained by the fact, that this measure calculates the angle between two vectors (embeddings), thus accounting for the direction of the vectors, which place a grear role in identifying the semantic meaning.

Further, these phrases and competencies are aggregated within the sentence the key phrase belongs to. So, for each sentence, the number of competencies assigned is equal to five multiplied by a number of key phrases in the sentence. These competencies are ranked again based on similarity score to obtain 5 more alike competencies for each sentence. Further, calculated cosine scores are compared with the threshold value, equal to 0.55, and if the cosine

score is less than 0.55, the competency is marked as “new”. The example of the result for 1 sentence is presented in figure 17. It can be seen from the figure that skills, that have cosine similarity equal to 0.47 and 0.51 are marked as “new”.

Score	Sentence	Competency
[[tensor(0.4719)]]	Microsoft Office software (e.g., Excel, Word, ...	new
[[tensor(0.5126)]]	Microsoft Office software (e.g., Excel, Word, ...	new
[[tensor(0.5666)]]	Microsoft Office software (e.g., Excel, Word, ...	MS Powerpoint
[[tensor(0.5856)]]	Microsoft Office software (e.g., Excel, Word, ...	MS Word
[[tensor(0.5897)]]	Microsoft Office software (e.g., Excel, Word, ...	Office tools and technologies (knowledge and w...

Figure 17. Example of the skills extracted from the sentence by the SBERT model

4.2.3. Analysis of new competencies to enrich the taxonomy

To enrich the existing taxonomy with the new skills from the job descriptions there were analyzed the sentences, that contain 50% or more skills marked as new (3 or more “new” skills). These sentences were collected separately, including the sentence itself and the competencies from the taxonomy, that were found in the sentence. This was done in order to understand for which skills in the sentence there were found the corresponding taxonomy’s skills, and for which were not. Apart from just finding these skills, it is important to build them into the correct section of the taxonomy.

Thus, based on the analysis of job descriptions, the following competencies were added to the existing taxonomy:

- Professional competencies (hard and close to hard skills)
    - Core enterprise architecture and business analysis activities
        - **Definition of requirements for data quality**
        - **Defining glossary definitions**

        - **Defining and profiling data sources**
        - **Aligning data architecture roadmaps with corporate strategy**
        - **Analyzing and optimizing different data models**
        - **Developing implementation strategies for new systems**
    - Management, support, and development of corporate architecture and business analysis activities
        - **Maintaining data architecture-related standards**
        - **Maintaining data definitions and business rules**
    - Related competencies in management
        - **Communicating recommended solutions**
        - **Supporting on-schedule delivery of milestones and deliverables**
        - **Assisting teams in data understanding and usage**
        - **Working in an Agile environment**
        - **Working in the Product team's environment**
    - Related IT competencies
        - **Database design**
        - **Creating Data Quality Rules**
        - **Enhancing and tuning database performance**
        - **Working with data mart systems**
        - **Gathering metadata**
        - **Measuring the quality of data**
        - **Implementing data remediation**
        - **Experience with DBMS**

        - **Designing data models for application**
        - **Data pre-processing (filtering, cleaning)**
        - **Develop data collection systems**
        - **Evaluate new system software**
        - **Review software updates**
        - **Review operational documentation on system software**
    - Basic interdisciplinary competencies
        - **Providing customer support for digital products**
        - **Experience in digital business systems analysis, digital platforms, digital product management**
        - **Knowledge of statistics and statistical packages**
        - **Writing documentation on detailed work performed**
- Basic competencies (soft and close to soft skills)
    - Fundamentals of architectural thinking
    - Cognitive competencies
        - **Detail-oriented**
    - Competencies in communication and interaction with people
        - **Teambuilding**
        - **Communication with the organization's leaders**
    - Managerial competencies
        - **Assess each employee's progress**
        - **Maintaining Best Practices sharing culture**
    - Personal competencies and characteristics
        - **Working in a dynamic environment**
        - **Working in an inclusive environment**

        - **Multitasking**

- Knowledge and experience with technologies and IT tools
    - Office tools and technologies (knowledge and work experience)
        - **Google Sheets**
    - Collaboration tools (knowledge and work experience)
    - Charting tools (knowledge and work experience)
    - Tools and technologies for modeling and working with requirements (knowledge and work experience)
    - Project management tools (knowledge and work experience)
    - Tools for structuring information (knowledge and work experience)
    - Software development technologies (knowledge and work experience)
        - **DDL scripts**
        - **Oracle**
        - **AWS Redshift**
        - **Erwin Data Modeler**
        - **Informatica**
        - **MicroStrategy**
        - **Teradata**
        - **Databricks**
        - **Jupiter**
        - **ER/Studio**
        - **Sparx Systems Enterprise Architect**
        - **PySpark**
        - **XML**
        - **SPSS**

        - **SAS**
        - **Shell scripts**
        - **VB scripts**
        - **Matlab**
        - **Tableau**
        - **Javascript**
        - **PHP**

- Knowledge of methods, approaches, and standards
    - Related to business analysis and business planning
        - **Relational Data model**
        - **Dimensional Data model**
        - **Developing data modeling standards**
        - **Error mapping**
    - Methodologies, standards, and reference models in the field of enterprise architecture management
    - Knowledge and experience in the subject area
        - **Degree in computer science**
        - **Degree in data analytics**
        - **Degree in mathematics**
        - **Degree in economics**
        - **Degree in information management**
        - **Degree in statistics**
    - Organizational development and business transformation
    - **Related to software development**

- **Knowledge of the SDLC (Software development life cycle)**

4.2.4. Assessment of the SBERT model based on the golden corpus

As previously mentioned in the methodology-related chapter of this thesis, in order to assess the quality of the model there should be developed a golden corpus. To build the golden corpus for this task, there were taken 100 sentences from the job descriptions, these sentences were manually labeled by me with the corresponding competencies from the taxonomy. The maximum number of competencies for each sentence is equal to five. If certain skills are not reflected by the taxonomy, then they are labeled as "new". In Appendix 1, there can be seen an extract from the golden corpus.

Consistent with the literature review performed previously in this thesis, the model is assessed in terms of adapted for this specific case accuracy, precision, recall, and F-score.

Firstly, it is needed to obtain the labels for the sentences included in the golden corpus by using the SBERT model. Further, these labels were compared with the ones assigned manually. The comparison was done automatically in Python. Based on the comparison of these labels, the elements of the confusion matrix were calculated. The confusion matrix is presented in table 6.

Table 6. Elements of the confusion matrix for the SBERT model

TP	**FP**	**TN**	**FN**
201	98	132	69

Based on the obtained values of the elements of the confusion matrix, there were calculated performance evaluation metrics – accuracy, precision, recall, and F-score, the results are presented in table 7.

Table 7. Performance evaluation metrics for the SBERT model

Accuracy	Precision	Recall	F-measure
0.666	0.672	0.744	0.706

As can be seen from the evaluation metrics, the results of model performance are satisfactory, especially taking into account that the task is multi-label classification – accuracy is almost 67% and F-score is equal to almost 71%. Based on most of the evaluations, anything higher than 70% can be named a realistically great model performance (Barkved, 2022). Also, comparing the performance of the model on the current dataset with the previous results presented in subchapter 3.7., it can be stated the SBERT model showed better results with accuracy and F-score being higher by more than 5 percentage points each.

What would be also interesting to look at is the distribution of errors by the sentences. It is important to check whether the model makes a lot of mistakes extracting the competencies from one sentence (is totally unable to catch the semantic meaning of a sentence), or whether these mistakes are distributed among all the sentences. The distribution of the errors (FP and FN elements) made by the model is presented in figure 18.

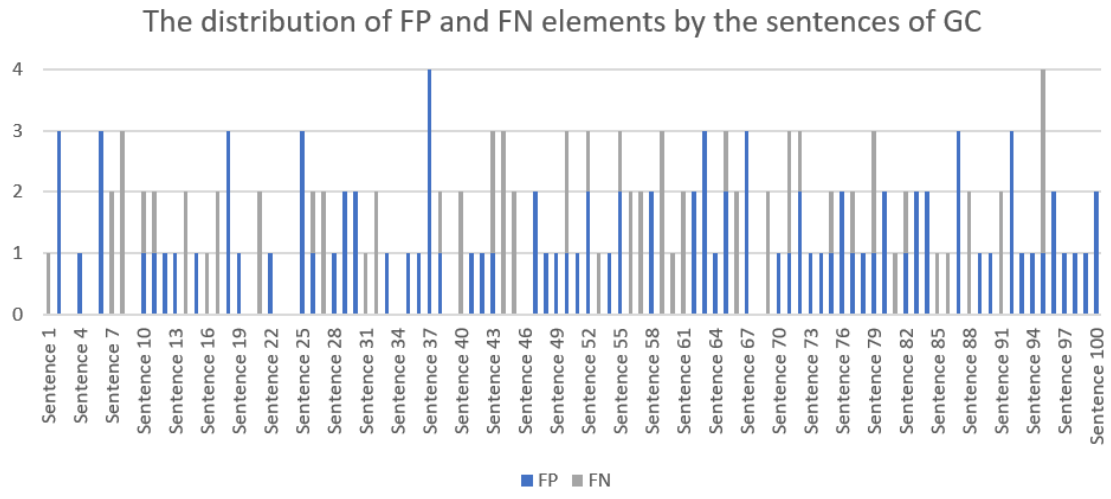

Figure 18. The distribution of FP and FN elements by the sentences of GC

According to the results obtained, for none of the sentences the model has identified all 5 competencies incorrectly, and only for two sentences within the Golden Corpus there were 4 errors made by the model – sentence № 37 (4 FP elements), sentence № 95 (1 FP and 3 FN elements). In 19 sentences within the Golden corpus, there were 3 errors per sentence made by

the model, in 33 sentences – 2 errors per sentence, in 36 sentences – 1 error per sentence, and in 10 sentences there were no errors made. This can be considered a satisfactory result because for almost 80% of sentences the model was able to identify correctly more than 50% of competencies and there was no sentence for which the model was absolutely incapable to understand the semantic meaning.

4.3. Vacancy – CV matching

Further, after the procedure of taxonomy enrichment has been completed, it becomes possible to use the updated taxonomy for matching task. The general algorithm for matching CVs to JDs is the following:

- Apply the NER component to extract competencies related to using specific tools ("Knowledge and experience with technologies and IT tools" section in the taxonomy) from both JDs and CVs
- Apply the Sentence Transformer model to extract competencies related to other sections of taxonomy than the "Knowledge and experience with technologies and IT tools" section
- Aggregate the competencies extracted by using both methods within each JD and CV to the corresponding lists
- For each JD rank the CVs based on the similarity measure described previously between the set of competencies extracted from the CV and the set of competencies extracted from the considered JD and obtain top-10 CVs

As for the validation part, there are two main steps that should be taken:

- Create a validation set consisting of 5 random JDs for each job position and CVs dataset containing CVs of candidates relevant and irrelevant for analyzed job positions
- Assess the quality of the JD – CV matching algorithm using a validation set based expert's evaluation

The application of two different approaches – NER and Sentence Transformer model is explained by the fact that competencies, that represent experience with specific tools or software, can be more accurately detected with NER rather than Sentence Transformer. These competencies, which are mentioned "Knowledge and experience with technologies and IT tools" section of the taxonomy, are one- or two-word skills, that represent the names of programs or tools, that is why it is needed to search for exact matches with these skills within the text and there is no need to apply Sentence Transformer model and generate word embeddings. Applying NER on such type of competencies, we can be sure that all the matches with them in the texts of JDs and CVs will be found.

4.3.1. Application of NER to JDs and CVs data

The competencies that are included in the ("Knowledge and experience with technologies and IT tools" section of the taxonomy represent different tools (Python), frameworks (ETL), and products (MS SQL). These kinds of competencies can be detected within the text by using the NER model because the dictionary-based NER model allows finding exact matches between the entities mentioned in the dictionary and these entities within the text. In accordance with the described methodology, there was custom NER from the SpaCy package used for this task.

Firstly, the SpaCy package was installed to be able to have access to the needed language model "en_core_web_sm". This is a trained pipeline for the English language, that has such components as tok2vec, tagger, parser, senter, ner, attribute_ruler, and lemmatizer (SpaCy, n.d.). A custom entity ruler was added to the pipeline .

After adding the custom NER pipe, it is needed to specify the list of named entities that should be found and labeled within the text of JDs and CVs. Dictionary-based NER is sensitive to case, so before applying it to the texts of JDs and CVs, it was important to transform them and also the competencies within the list of tools to the lowercase.

As it was stated previously, the list of competencies that should be labeled within the text consist of tools, products, and frameworks mentioned in the taxonomy. Thus, using the function "`add_patterns ()`", the list of competencies was added to the dictionary with

the label “tools”. The pipeline was run on both JDs and CVs and extracted named entities were stored for each JD and CV (example in fig. 19).

	Candidate №	Occupation	Skills from CVs
0	Candidate 1	data analyst	[etl, matlab, google sheets, python, tableau]
1	Candidate 2	data analyst	[php, artificial intelligence, matlab, sql, ja...
2	Candidate 3	senior data analyst	[etl, python, matlab]
3	Candidate 4	marketing data analyst	[sql, python, tableau]
4	Candidate 5	financial data analyst	[sql, tableau]
5	Candidate 6	data analyst	[bi, sql, python, oracle, excel, tableau]
6	Candidate 7	junior data analyst	[sql, python, oracle, excel, tableau]
7	Candidate 8	healthcare data analyst	[javascript, php, sql, html/css]
8	Candidate 9	business data analyst	[etl, tableau, bi, spss, sql, erp]
9	Candidate 10	power bi analyst	[sas, bi, sql, in-memory, tableau]
10	Candidate 11	intern data analyst	[etl, sql, olap, excel, tableau]

Figure 19. Extract from the table of competencies extracted by NER from CVs

4.3.2. Application of Sentence Transformer to JDs and CVs data

The procedure of applying the Sentence Transformer model to extract competencies from JDs and CVs for further matching is the same as it was for the taxonomy enrichment task – firstly, there were extracted key phrases using the RAKE algorithm and later Sentence Transformer (SBERT) model was applied to the extracted key phrases and competencies from the taxonomy and obtain cosine similarity scores for each pair of a key phrase (within JD or CV) and competency.

However, in this case from the list of competencies, there were excluded the competencies mentioned in the section “Knowledge and experience with technologies and IT tools” as they were already processed by the NER tool.

As well as for the taxonomy enrichment task, the competencies extracted for phrases were ranked based on the cosine similarity score (with a threshold value equal to 0.55 as in the taxonomy enrichment task) and aggregated within sentences in order to choose the top 5

competencies for each sentence. Further, the competencies were aggregated within each JD or CV to obtain the list of competencies mentioned in them. So, for each JD and CV there were obtained 2 lists of competencies extracted using NER and Sentence Transformer, these lists were aggregated within each JD and CV (fig. 20).

Candidate №	Occupation	Skills from CVs
Candidate 1	data analyst	[tableau, business intelligence management, de...
Candidate 2	data analyst	[tableau, developing a data model, programm ma...
Candidate 3	senior data analyst	[business scenario, strategic planning, vision...
Candidate 4	marketing data analyst	[business intelligence management, tableau, da...
Candidate 5	financial data analyst	[business scenario, strategic planning, vision...
Candidate 6	data analyst	[metric system design, development of the syst...
Candidate 7	junior data analyst	[development of the system concept, tableau, b...
Candidate 8	healthcare data analyst	[sql, it trends, data analysis, learning abili...
Candidate 9	business data analyst	[tableau, business intelligence management, de...
Candidate 10	power bi analyst	[methods and standards of business analysis, t...

Figure 20. Extract from the table of competencies extracted from CVs

4.3.3. Calculating similarity between sets of competencies and obtaining top-10 CVs for each JD

In order to rank the CVs based on their suitability for each of the job positions, it is needed to compare the lists of competencies, stated in the JDs and the lists of competencies, that were mentioned in the CVs. Consistent with the methodology, there was used the similarity measure, which calculates the number of shared competencies in the sets from JD and CV, divided by the number of competencies from the JD. This measure was calculated for each of the JDs' sets and each of the CVs' sets, and then for each JD CVs were ranked based on the similarity score, and the top-10 CV was stored. This was done using the following Python code:

```
"candidate_num=list(range(1,len(skills_CV)))

result=[]

for i in range (0,len(skills_vacancies)):
```

```
        res_sub=[]
        for j in range (0, len(skills_CV)):
            res=len(set(skills_vacancies.iloc[i,1]) &
set(skills_CV.iloc[j,2])) /
float(len(set(skills_vacancies.iloc[i,1])))
            res_sub.append(res)
        result.append(tuple(zip(candidate_num,res_sub)))
    result_sort=[]
    for i in range (0, len(result)):
        result_sort.append(sorted(result[i], key = lambda x:
x[1], reverse=True))
    top_10=[]
    for i in range (0, len(result_sort)):
        top_10.append(result_sort[i][0:10])".
```

This code allows obtaining the nested list, that contains the index numbers of top-10 suitable candidates with the respective similarity score. Further, if needed, it is possible to present the result in the form of a table to make it look more visually understandable.

4.3.4. Creation of validation set and evaluation of the quality of JD – CV matching algorithm using based expert's evaluation

In order to evaluate the quality of the algorithm it is needed to form the validation set. The validation set consists of 5 job descriptions for each of the 4 job positions analyzed (20 JDs in total), and of the dataset of CVs consisting of 54 CVs. The JDs for the validation set were randomly chosen from the Pandas DataFrame of JDs using the "`sample()`" method and stating the number of rows that should be selected (fig. 21).

	Job position	Vacancy
0	Enterprise Data Architect	(architectural, expertise, with, big, data, di...
1	Sr Enterprise Data Architect	(leads, data, analysis, ,, database, design, ,...
2	Data Architect	(designing, large, complex, table, structures,...
3	Data Architect	(collaboratively, work, with, functional, spec...
4	Health System Data Architect	(implement, as, required, ,, data, integration...
5	Senior Data Analyst	(bachelor, 's, degree, in, a, quantitative, fi...
6	Data Analyst	(serve, as, the, data, and, business, analytic...
7	Data Analyst	(5, +, years, of, data, management, ,, busines...
8	Senior Data Analyst	(minimum, of, 5, years, of, healthcare, experi...
9	Data Analyst	(1, +, years, of, experience, using, tableau, ...
10	Business Analyst	(collaborate, with, multi, -, functional, team...

Figure 21. The extract from the validation set of vacancies

As for the CVs dataset, the set of relevant CVs for the analyzed job positions consists of the CVs taken from the Resume Worded website, as these CVs are well-organized and designed specifically for these job positions. Other CVs were chosen randomly from the original set of CVs. The dataset consists of:

- 11 CVs for the position of "Data Analyst"
- 12 CVs for the position of "Business Analyst"
- 10 CVs for the position of "Enterprise Architect"
- 10 CVs for the position of "Data Architect"
- 6 CVs for the position of "Data scientist"
- 2 CVs for the position of "HR specialist"
- 3 CVs for the position of "Sales Manager"

As it can be seen, the dataset of resumes consists of three types of CVs – CVs that belong to the job positions analyzed in this thesis ("data analyst", "business analyst", "data architect" and "enterprise architect"), CVs for the job position of "data scientist", which belongs to the same IT field as analyzed job positions, and absolutely irrelevant CVs for the positions of "HR specialist" and "Sales manager", which are needed to assess the quality of the

algorithm and to see whether they do not appear in the list of top-10 candidates for the analyzed job positions.

The respective positional numbers for job positions within the dataset of CV are the following:

- Data analyst – №1 - №11
- Business analyst – №12 - №21, №42 - №43
- Data architect – №22 - №31
- Enterprise Architect – №32 - №41
- Data scientist – №44 - №49
- HR specialist – №50 - №51
- Sales manager – №52 - №54

Figure 23 demonstrates the results of the ranking obtained for the validation set. At the first glance, it can be seen that none of the irrelevant CVs ("HR specialist" and "Sales Manager") appeared in the list of the top 10. This can be named a good desirable result, that the algorithm does not suggest the irrelevant candidates from completely different fields. Also, for the position of "data architect" 82% of CVs in the top 10 belong to this job position, for "data analyst" – 66%, for "business analyst" – 76%, and for "enterprise architect" – 64%. As for the "data scientist" resumes, it can be seen, that one of the candidates with CV № 47, has appeared in the top – 10 for all job descriptions for "data architect" and in one job description for "data analyst". This can be explained by the fact that this resume contains a lot of competencies related to using specific tools and some competencies related to data architecture design, developing data modeling standards, etc. As will be shown later, Candidate №47 was considered be the expert a good candidate for data analyst positions, but a non-suitable candidate for the remaining three positions.

JD №	JD title	Candidate № 1	Candidate № 2	Candidate № 3	Candidate № 4	Candidate № 5	Candidate № 6	Candidate № 7	Candidate № 8	Candidate № 9	Candidate № 10
1	Analytics Data Architect	23	24	27	25	31	26	30	29	47	22
2	Data Architect	30	34	22	31	23	26	27	25	47	29
3	Sr Enterprise Data Architect	30	24	25	23	26	36	31	27	47	29
4	Enterprise Data Architect	29	22	24	31	36	27	33	23	26	47
5	Data Architect	24	26	30	23	29	31	27	22	25	47
6	Senior Data Analyst	6	47	5	1	2	4	3	10	11	14
7	Data Analyst	2	24	9	10	8	7	5	11	17	20
8	Senior Data Analyst	3	5	20	4	11	17	6	1	14	30
9	Senior Data Analyst	14	2	7	13	9	5	18	20	30	39
10	Data Analyst	1	14	10	11	5	31	3	6	2	4
11	Business Analyst	18	27	20	21	14	31	17	13	30	12
12	Business Analyst	13	14	43	12	24	30	27	15	19	18
13	Business Analyst	14	13	21	17	12	18	30	15	11	20
14	Business Analyst	19	14	17	12	27	15	31	13	20	22
15	Business Analyst	14	30	13	12	24	17	43	20	27	22
16	Enterprise Architect	35	27	34	41	30	33	31	32	36	39
17	Enterprise Architect	36	39	33	29	30	38	34	32	27	31
18	Enterprise Architect	34	41	22	30	36	39	26	33	37	32
19	Enterprise Architect	30	36	39	43	41	32	34	31	27	38
20	Enterprise Architect	34	39	30	33	36	43	29	37	32	24

Figure 22. Top-10 candidates ranked based on the used similarity measure

Consistent with the literature review, there was used expert's evaluation to assess the performance of the algorithm used for vacancy-CV matching. As an expert there acts the director of the LLC company "The MR. Wolfe Group". "The MR. Wolfe Group" is a privately held company operating in Australia with international subsidiaries in Armenia and Russia. The spheres of the company's activities include business development for digital companies, SEO optimization, email marketing for eCommerce companies, etc. The expert is independently engaged in the selection of both full-time staff and employees for specific projects. The expert was asked to assess each of the 54 CVs within the dataset by their suitability for "data analyst", "data architect", "business analyst" and "enterprise architect" using three degrees of suitability "good", "average" and "not suitable". The table that represents the expert's evaluation is presented in Appendix 2 of this thesis.

For the “business analyst” position there were identified 7 good candidates, 9 average candidates, and 38 not suitable ones, for the “data analyst” position there were identified 9 good candidates, 8 average candidates, and 37 not suitable ones, for the position of “data architect” – 10 good candidates, 1 average candidate, and 43 not suitable ones, and for the “enterprise architect” – 7 good candidates, 8 average candidates, and 39 not suitable ones. Given the fact that for 3 out of 4 job positions there were less than 10 good candidates identified (based on the expert's assessment), the most desirable outcome of the JD-CV matching algorithm is that all these good ones being suggested and then the rest being suggested from the average candidate category for that position. Suggestions of "not suitable" candidates are considered undesirable, as are omissions of good candidates. On the other hand, the expert did not assess the suitability of the candidates for the specific JDs, but for an overall concept of the specific position. Some suggestions of average or non-suitable candidates can therefore also be a result of a strange (unusual) JD for the given position. That is of one that contains non-standard or less standard requirements.

Based on the expert’s evaluation it can be stated that:

- Out of all the top 10 candidates suggested by the algorithm for the position of Data Architect, 86% were considered good by the expert for this position. There was only one candidate marked as “average” by the expert, so it is of no surprise that this candidate did not appear in the top and the remaining14% of candidates in the top 10 were candidates labeled as “Not suitable”

- Out of all the top 10 candidates suggested by the algorithm for the position of Data Analyst, 54% were considered by the expert as “Good” for this position, 22% as “Average”, and 24% as “Not suitable”

- Out of all the top 10 candidates suggested by the algorithm for the position of Business Analyst, 54% were considered as “Good” by the expert, 16% as “Average” and 30% as “Not suitable”

- Out of all the top 10 candidates suggested by the algorithm for the position of Enterprise Architect, 52% were considered as “Good” by the expert, 40% as “Average” and 8% as “Not suitable”

Besides, it is important to check that the algorithm does not miss good candidates within the list and does not include unsuitable ones in the top 10 (fig.23). In figure 23, candidates who

were marked as “Good” are presented in green color, candidates who were marked as “Average” – in yellow color, and unsuitable candidates – in red color.

For the position of “Data architect” there were 10 candidates, who were marked as “good” by the expert; in the list of the top 10 for this position for three out of five job descriptions there were 9 out of 10 candidates identified, and for two of them – there were 8 good candidates identified (on average 86% of all good candidates), that could be named an accurate result. Apart from that, all the candidates denoted as “Good” by the expert appeared among the suggested top 10 candidates. However, for all job descriptions for this position, there were no average candidates identified within the list of the top 10, for two job descriptions 2 “Not suitable” candidates appeared, and for three job descriptions – 3 “Not suitable” candidates. However, there was one CV № 47, marked as “Not suitable”, that appeared in all 5 job descriptions due to the fact that the candidate has experience in using a lot of tools and dealing with data models. Nevertheless, this candidate was ranked low – either 9^{th} or 10^{th} position. CV № 34 possesses a bigger problem, as it appears in the 2^{nd} position; this CV belongs to the “Enterprise Architect” occupation. In general, there are 3 unique not suitable candidates out of 43 possible ones who appeared in the top 10 for this position.

For the position of “Data Analyst” there were 9 candidates marked as “Good”; for one JD there were 7 out of 9 good candidates identified, for two JDs – 6 out of 9, for one JD – 5 out of 9, and for one JD – 3 out of 9 (on average 60% of all good candidates). Only one of the good candidates – Candidate № 44 did not appear in the top 10 for this position. As for the JD, which has the biggest number of good candidates within the top 10, there was just one unsuitable candidate identified – Candidate № 14 (Business analyst), the same candidate was identified as the best for another data analyst position. This can be explained by several reasons – both JD and CV have many soft skills mentioned (written and verbal communication skills, organizational skills, presenting information), all MS Office programs are written separately, and as well communicating to stakeholders related competency. In fact, the JD № 9 in general is composed in a non-standard way for this occupation, with an emphasis on soft skills and such cognitive competencies as analytical skills or logical thinking, and also on communication with stakeholders and technical and non-technical audiences. This can be an explanation, of why so many mistakes were made for this exact non-standard JD (because the expert marked candidates with their suitability level for the occupation in general, but not for each exact JD). In general, there were 7 unique CVs out of 37 CVs marked as “Not suitable” by the expert, which appeared in the top 10.

For the position of “Business Analyst” there were 7 candidates marked as “Good” by the expert; in two JDs there were 6 out of 7 good candidates appeared in the top 10, and in three JDs – 5 out of 7, so on average 77% of good candidates were identified by the algorithm. All the candidates who were marked as good by the expert have appeared in the suggested lists of candidates. Candidate № 18 (Entry Level Business Analyst) who was marked as “Average” by the expert, has appeared in the 1st position for JD № 11; that can be explained by the difference in the approaches used by the algorithm and by the expert. While marking the candidates, the expert was motivated not only by the presence of competencies but also by the expected proficiency level, which is why the candidate was ranked as “average”, whereas the algorithm considers only the presence of the competencies. Candidate № 21 (Agile Business Analyst), marked as “Average” also appears in the 3rd and 4th positions in the top 10, which could also be explained by the subjective opinion of the expert regarding this candidate. In general, there were 6 unique CVs out of 38 CVs marked as “Not suitable” by the expert, which appeared in the top 10 for this position.

For the position of “Enterprise Architect” there were 7 candidates marked as “Good” by the expert; in one JD there were all 7 good candidates identified correctly in the top 10, for three JDs there were 5 out of 7 good candidates identified correctly, and for one JD only 4 out of 7 (on average for the position – 74% of all good candidates). All the candidates labeled as “Good” were suggested by the algorithm in the top 10 for this position. The JD № 16 can be a representation of how the algorithm should perform – in addition to correctly identifying all 7 good candidates, it suggested 3 average candidates and none of the unsuitable ones. The presence of that many “average” candidates could be explained by the severity of the expert’s evaluation (not all the candidates with the occupation stated as “Enterprise Architect” were marked as “good” by the expert). However, as for the unsuitable candidates, only 3 unique ones out of 39 possible appeared in the top 10.

JD №	JD title	Candidate № 1	Candidate № 2	Candidate № 3	Candidate № 4	Candidate № 5	Candidate № 6	Candidate № 7	Candidate № 8	Candidate № 9	Candidate № 10
1	Analytics Data Architect	23	24	27	25	31	26	30	29	47	22
2	Data Architect	30	34	22	31	23	26	27	25	47	29
3	Sr Enterprise Data Architect	30	24	25	23	26	36	31	27	47	29
4	Enterprise Data Architect	29	22	24	31	36	27	33	23	26	47
5	Data Architect	24	26	30	23	29	31	27	22	25	47
6	Senior Data Analyst	6	47	5	1	2	4	3	10	11	14
7	Data Analyst	2	24	9	10	8	7	5	11	17	20
8	Senior Data Analyst	3	5	20	4	11	17	6	1	14	30
9	Senior Data Analyst	14	2	7	13	9	5	18	20	30	39
10	Data Analyst	1	14	10	11	5	31	3	6	2	4
11	Business Analyst	18	27	20	21	14	31	17	13	30	12
12	Business Analyst	13	14	43	12	24	30	27	15	19	18
13	Business Analyst	14	13	21	17	12	18	30	15	11	20
14	Business Analyst	19	14	17	12	27	15	31	13	20	22
15	Business Analyst	14	30	13	12	24	17	43	20	27	22
16	Enterprise Architect	35	27	34	41	30	33	31	32	36	39
17	Enterprise Architect	36	39	33	29	30	38	34	32	27	31
18	Enterprise Architect	34	41	22	30	36	39	26	33	37	32
19	Enterprise Architect	30	36	39	43	41	32	34	31	27	38
20	Enterprise Architect	34	39	30	33	36	43	29	37	32	24

Figure 23. . Expert's evaluation of the top 10 candidates for each JD

Figure 24 presents a distribution of positions of unsuitable CVs in the top10 rankings. As it can be seen from the graph, the majority of unsuitable candidates were not ranked within top 5 for each JD. In order to avoid suggesting only unsuitable candidates, the aim is to suggest more than one candidate and then leave the selection of the most appropriate candidate on the HR manager. For all the job descriptions within the top 10 there were at least 3 good candidates suggested.

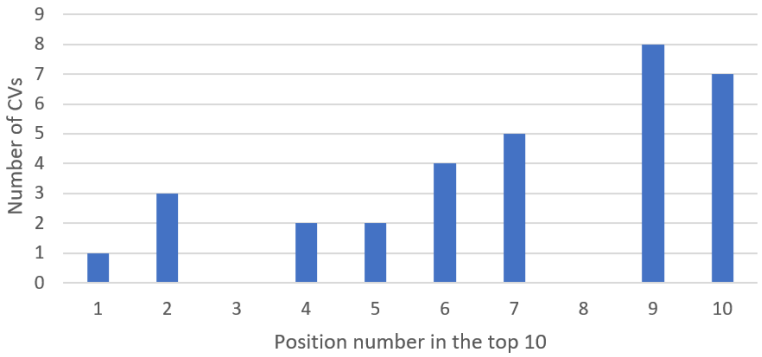

Figure 24. The distribution of positions of unsuitable CVs in the top 10 rankings presented in Figure 23

As it was previously mentioned, one of the drawbacks of the algorithm proposed is that there is no possibility to mark some of the competencies as absolutely necessary ones and to filter out candidates who do not have these competencies, even though their similarity with a JD is overall high. One of the solutions for that is demonstrated in figure 25. This solution aims to present the top10 lists the way it will make the decisions as simple as possible for the HR specialists. The presented table shows the competencies, that were extracted from the JD № 18 for the position of Enterprise Architect and the top 10 candidates for this position with the respective skillset. So, if a candidate has a skill, that was mentioned in the JD, then the skill is marked as "+", if a candidate does not have that skill, then the skill is marked as "-". Thus, if, for example, an HR specialist considers the skill "Solution design" to be a must for the candidate, then, candidate № 32 will be excluded from the list, and the next candidate in the list who has this skill will be added to the top 10. The table presented in figure 25 allows HR specialists to see which matches the candidates have with the competencies from the JD and to filter out the candidates who do not have the specific necessary skill. This table can serve as a substitute for the filtering algorithm.

Candidate №	Candidate occupation	document and archive management systems	it architecture design	systems engineering	strategic thinking	architecture design	solution design	developing a logical data model	logical analysis	building relationships	evaluate new system software	it architecture modeling	development of technical specifications for the system	strategic planning	logical data model	technology architecture design
22	data architect	-	-	-	-	-	+	+	-	+	-	+	-	+	+	-
26	data architect	-	-	-	-	-	+	-	+	+	+	-	-	-	-	+
30	asst. data archietct	-	-	-	-	-	+	+	+	-	-	+	-	-	+	+
32	enterprise architect	-	+	-	-	+	-	-	-	-	-	+	-	-	-	-
33	sr. enterprise architect	-	+	-	-	-	+	+	-	-	-	+	-	-	-	-
34	jr. enterprise architect	-	+	-	+	+	+	+	+	+	-	-	-	+	+	-
36	enterprise architect	-	+	-	+	-	+	+	-	-	-	-	+	-	+	-
37	asst. enterprise archietct	-	-	-	+	-	+	-	-	+	-	-	-	+	-	-
39	enterprise architect	+	+	-	-	+	+	+	-	-	-	-	-	-	+	-
41	enterprise architect	-	+	-	+	-	+	+	+	-	+	-	+	-	-	-

Figure 25. Example of present and missing competencies for the top-10 applicants for the position of Enterprise Architect

In general, it can be stated the algorithm can be useful in terms of applying it to the automated screening of resumes and filtering out irrelevant candidates. All the candidates labeled as “Good” by the expert except for 1 candidate who belongs to “Data Analyst” occupation, were suggested by the algorithm within the top 10 for respective job positions. This can be named a quite accurate result in terms of the selection of the suitable candidates and suggesting them to the HR managers in the first place. Also, for each of the job positions out of all the top 10 candidates suggested by the algorithm there were more than 50% of good candidates. As for the irrelevant CVs (for the positions of “Sales manager” and “HR specialist”) present in the dataset, none of them was suggested within the top 10 by the matching algorithm, from which we can conclude that the algorithm performs its function of filtering out the candidate from other non-related fields. Apart from that, the candidates who were labeled as “Not suitable” for analyzed job positions have appeared mostly in the last positions in the top 10. Thus, the performance of the vacancy-CV matching algorithm can be named as acceptable and satisfactory since it performs primary functions which were expected and desired. However, it is obvious that the lists of suggested candidates should be further analyzed by the HR specialists.

As for the average and unsuitable candidates, their appearance in the top 10 can be explained by several reasons:

- for 3 out of 4 job positions there were fewer than 10 candidates marked as “Good” by the expert, which is why “Average” candidates could appear in the top 10

- some of the JDs can be compiled in the non-standard for that job position way (this is the case with the vacancy № 9), which is why unsuitable candidates could be suggested
- the suitability of the CV for the job position was defined by the expert based on the average or prototyped job description, without taking into consideration the specificity of each job description

5. Potential usage of the research

As was mentioned in the introduction part, the recruitment process becomes more and more time-consuming and difficult, and the number of CVs that HR specialists need to consider is one of the reasons for that. The role of automated screening systems is to provide convenience and savings for recruiters by sorting out irrelevant resumes (Daryani et al., 2020).

In terms of taxonomy enrichment, the results presented in the thesis can be used by the companies being in need for using the enriched taxonomy for competencies analysis or related purposes. The taxonomy was updated with the new competencies, which are now required from successful candidates. The updated taxonomy can be used to create employees' profiles and assess the sufficiency of competencies within a company. Also, the algorithm proposed, and the Sentence Transformer model applied can be used for similar tasks of taxonomies enrichment.

Besides, there was developed a modification of the Jaccard similarity measure, that suits this specific task of ranking the resumes based on their suitability for the analyzed job descriptions. The modified version allows calculating the proportion of competencies from a CV, which are also mentioned in the JD. In contrast to the original Jaccard similarity, the new similarity measure takes into account only competencies from the JD as a basis for calculating the proportion.

Moreover, in order to deal with one of the limitations of the proposed algorithm – the inability to mark absolutely necessary competencies, there was suggested a table, which demonstrates the present and missing competencies for each candidate from the top 10 within each job position. Using this table, HR specialists can sort out the candidates, who do not have absolutely required competencies.

The proposed algorithm allows to extract the competencies from JDs and CVs and present a JD or CV as a set of competencies, that could make them more visually understandable. Also, HR specialists could easily and rapidly evaluate the candidate in terms of their skills and knowledge. Besides, the algorithm filters out inappropriate candidates for specific job positions, which could potentially reduce the time to analyze potential candidates and also the time per hire. The implementation of the matching procedure that aims to provide an automated pre-screening of CVs can make the recruitment process more efficient and less time-consuming.

Besides, the same matching algorithm can be applied to match the vacancies to the CVs, which can be used by job seekers to explore the possible job positions they can apply to based on the skillset they have. Also, applicants can in advance assess their chances while applying for a certain job position, taking into account the competencies required and the competencies they have. This could as well be beneficial, because the comparison of these two sets of skills required and skills present can provide candidates with insights about their future professional development and give ideas for the competencies they should develop to become in-demand professionals.

Regarding the validation procedure, there was suggested a case-specific approach for the expert's evaluation. The approach does not require obtaining the rankings from an expert for each CV and JD, which would make the evaluation more subjective and definitely more time-consuming. On the contrary, the task of the expert was to assess in general whether each CV can be suitable for specific job positions using just three possible markers "good", "average" and "not suitable". This procedure allowed to assume, based on the expert's assessment whether each of the candidates could appear in the list of top 10 candidates for analyzed job positions.

6. Limitations and ideas for further research

The approach to the task of vacancy–CV matching that was developed has shown acceptable results, however, the work performed shows that there is much room for improvement. The first direction of changes is related to the model used to capture the competencies within JDs and CVs. The model used is a pre-trained one, and even though it was trained on a tremendous amount of data, the type of this data is far from what was analyzed in this thesis. The vocabulary of JDs and CVs is very specific due to the business and IT-related

terminology used to write them, so in case of having a large corpus of labeled field-specific data, it should be more effective to train the model on this data rather than to use pre-trained pipelines and models.

As for the information, extracted from the JDs and CVs and serving as a basis for the matching algorithm, it could be beneficial to include in the analysis not only the competencies. When HR specialists analyze CVs, they pay attention also to the location of the applicant, because if it does not match the location of the intended place of work, then it can become an additional problem. Also, it is advantageous to know the educational information, the university the candidate graduated from, the obtained degree and major, and to have this information separately from the information about the competencies. Information about the professional experience could also go separately. In most of the JDs HR specialists specify the required work experience in years and job positions held, so adding the filter, which would reject the candidates with insufficient work experience could also be beneficial. While analyzing only the competencies mentioned in the CV, it is impossible to say which level of proficiency was reached in, for example, using the specific tool, so that intern candidates could be assigned to senior positions. Another piece of information that could be analyzed is related to the languages the candidate speaks. This especially is relevant for the candidates whose mother tongue is different from the language that is spoken in the country of the company they apply to. So, if the company requires knowledge of any other language except for English, this should be considered.

Besides, the algorithm used does not account for the negations that could sometimes be met in the texts of JDs. For example, if in the JD it is written “Knowledge of any programming language other than Matlab”, the algorithm will extract “Matlab” competency based on its match with the taxonomy of competencies, without accounting for negation. This type of statement can be rarely found in JD; however, this could still be a problem.

Other limitations are associated with the similarity measure used to assess the suitability of CVs for job positions. As it was mentioned previously, the similarity measure used treats all the competencies, mentioned in the set from JD equally, however, in reality, it is not true. It can possess a problem in two ways, the first one is that for IT-related field, which includes analyzed job positions, hard skills, and proficiency in using various tools and programs is more important than some of the soft skills. Another way it could bias the algorithm is that usually, IT specialists do not mention or mention not that many soft skills in their CVs. So, if there is

one candidate who mentioned a lot of soft skills that were matched with JD's soft skills, they will appear in the list of top candidates, even though there is the possibility that in terms of hard skills this candidate does not suit. Additionally, the algorithm does not allow to choose the core competencies from the JD, the absence of which would not allow the candidate to get the job position. The implementation of the opportunity to specify these core requirements can make the algorithm more useful for HR specialists.

As for the potential improvement, apart from what was mentioned previously, there could be several things done to make the algorithm easy and convenient to use for HR specialists:

- Automatic parsing of resumes from the websites on job search
- Tool for converting the resumes from such formats as PDF, JPG, etc. to text format, which will be then readable for the models
- Additional tools for filtering the candidates to make the ranking procedure more customized

As for the validation algorithm, even though there was suggested a case-specific working approach for evaluating the quality of the algorithm, in order to make the validation more objective, there could be asked several experts to perform the evaluation. This could allow for avoiding subjectivity while marking the candidates as good or average.

7. Conclusion

The goal of this research paper was to present a taxonomy-based approach to performing the task of vacancy – CV matching for the job positions of "business analyst", "business architect", "data architect" and "enterprise architect".

In order to do this, there were JDs parsed from the website Monster.com, and also CVs collected from datasets of resumes available on Kaggle.com and on Resume Worded website. To make the taxonomy of competencies more appropriate and updated, there was performed the enrichment of the current taxonomy. This was done using the pre-trained Sentence Transformer model, which allowed to extract the competencies from the JDs and to analyze missing ones. After the enrichment of the taxonomy, there was performed the vacancy-CV

matching task. In order to extract competencies from the JDs and CVs there were use two approaches – NER to extract the competencies related to the knowledge and experience with specific tools and software, and the Sentence Transformer model to extract other competencies. The comparison of skillsets extracted from JDs and CVs was performed by using the similarity measure – the modification of the Jaccard similarity measure. The assessment of the quality of the matching algorithm was done on the validation set by the expert's evaluation.

As for the research questions stated in this thesis, there could be given the following answers:

- Given the fact that the task of extracting the competencies from JDs and CVs is basically multilabel classification, the Sentence Transformer model performance can be named satisfactory. The F-score calculated for the Golden Corpus consisting of 100 sentences is equal to 70,6%.
- The rankings of the CVs obtained by the algorithm can be described as quite accurate considering the usage of only pre-trained model. None of the irrelevant CVs ("HR specialist" and "Sales Manager") appeared in the list of the top 10. Also, for the position of "data architect" 82% of CVs in the top 10 belong to this job position, for "data analyst" – 66%, for "business analyst" – 76%, and for "enterprise architect" – 64%. As for the expert's evaluation, for the position of "data architect" on average for each JD 86% of all good candidates were identified within the top 10, for the position of "data analyst" – on average 60%, for the position of "business analyst" – on average 77%, for the position of "enterprise architect" – on average 74%.

References:

Agarwal, K. (2022, September 24). Rake: Rapid automatic keyword extraction algorithm. Medium. Retrieved October 11, 2022, from https://medium.datadriveninvestor.com/rake-rapid-automatic-keyword-extraction-algorithm-f4ec17b2886c

Ai recruiting software: Online recruitment software. CVViZ. (n.d.). Retrieved November 18, 2022, from https://cvviz.com/

Arslan, M., & Cruz, C. (2022). Semantic enrichment of taxonomy for BI applications using multifaceted data sources through NLP techniques. Procedia Computer Science, 207, 2424–2433. https://doi.org/10.1016/j.procs.2022.09.533

Arslan, M., Cruz, C. (2022). Semantic taxonomy enrichment to improve business text classification for dynamic environments.

Barkved, K. (2022, March 9). How to know if your machine learning model has good performance: Obviously ai. Data Science without Code. Retrieved October 24, 2022, from https://www.obviously.ai/post/machine-learning-model-performance

Bhatia, V., Rawat, P., Kumar, A. & Shah, R. (2019). End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT.

Bhawal, S. (2021, August 8). Resume dataset. Kaggle. Retrieved November 23, 2022, from https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset

Biswas, M. (2020). Taxonomy enrichment.

Bogush, P. (2022, April 9). How has technology changed the hiring process. Businesstechweekly.com. Retrieved November 6, 2022, from https://www.businesstechweekly.com/hr-and-recruitment/technology-hiring-process/

Brin, D.W. (2021). Employers Embrace Artificial Intelligence for HR. Retrieved from SHRM

Chang, J. (2022, January 14). 101 hiring statistics you must read: 2021/2022 Data Analysis & Market Share. Financesonline.com. Retrieved September 2, 2022, from https://financesonline.com/hiring-statistics/

Charatan, Q., Kans, A. (2022). Object-Oriented Python: Part 1. In: Programming in Two Semesters. Texts in Computer Science. Springer, Cham. https://doi.org/10.1007/978-3-031-01326-3_8

Charles, L., P. Coutts, A., Xia, S. (2022, August 26). Digitalization and employment. Report: Digitalization and Employment, A Review. Retrieved November 6, 2022, from https://www.ilo.org/employment/Whatwedo/Publications/WCMS_854353/lang--en/index.htm

Chiarello, F., Fantoni, G., Hogarth, T., Giordano, V., Baltina, L., & Spada, I. (2021, September 9). Towards Esco 4.0 – is the European classification of skills in line with industry 4.0? A text mining approach. Technological Forecasting and Social Change. Retrieved September 27, 2022, from https://www.sciencedirect.com/science/article/pii/S0040162521006107

Dadzie, A.-S., Sibarani, E., Novalija, I., & Scerri, S. (2018). Structuring visual exploratory analysis of Skill Demand. SSRN Electronic Journal.

D'Agostino, A. (2022, August 20). Keyword extraction - a benchmark of 7 algorithms in Python. Medium. Retrieved October 11, 2022, from https://towardsdatascience.com/keyword-extraction-a-benchmark-of-7-algorithms-in-python-8a905326d93f

Dai, H., Wu, C., Tzong, R., Tzong-Han Tsai, R., Hsu, W. (2012). From Entity Recognition to Entity Linking: A Survey of Advanced Entity Linking Techniques.

Daryani, C., Chhabra, G. S., Patel, H., Chhabra, I. K., & Patel, R. (2020). An automated resume screening system using natural language processing and similarity. ETHICS AND INFORMATION TECHNOLOGY. https://doi.org/10.26480/etit.02.2020.99.103

Deng, L., & Liu, Y. (Eds.). (2018). A Joint Introduction to Natural Language Processing and to Deep Learning. In Deep learning in natural language processing. Springer.

Devika, R., Vairavasundaram, S., Mahenthar, C. S., Varadarajan, V., & Kotecha, K. (2021). A deep learning model based on Bert and sentence transformer for semantic keyphrase extraction on Big Social Data. IEEE Access, 9. https://doi.org/10.1109/access.2021.3133651

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL.

Dutta, M. (2022, August 2). Word2vec for word embeddings -A beginner's guide. Analytics Vidhya. Retrieved November 10, 2022, from https://www.analyticsvidhya.com/blog/2021/07/word2vec-for-word-embeddings-a-beginners-guide/

Eftimov, T., Koroušić Seljak, B., & Korošec, P. (2017). A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. PLOS ONE, 12(6). https://doi.org/10.1371/journal.pone.0179488

EntityRuler · Spacy API Documentation. EntityRuler. (n.d.). Retrieved November 24, 2022, from https://spacy.io/api/entityruler

ESCO database and network design and Administration. Esco. (n.d.). Retrieved September 27, 2022, from https://esco.ec.europa.eu/en/classification/skills?uri=http%3A%2F%2Fdata.europa.eu%2Fesco%2Fisced-f%2F0612

Farrugia, A. (2022, September 6). ATS explained: What is it and what does it mean for my cv? LinkedIn. Retrieved September 29, 2022, from https://www.linkedin.com/pulse/ats-explained-what-does-mean-my-cv-adam-farrugia

Fernández-Reyes, F. C., & Shinde, S. (2019). CV retrieval system based on job description matching using hybrid word embeddings. *Computer Speech & Language*, *56*, 73–79. https://doi.org/10.1016/j.csl.2019.01.003

Fletcher, S., & Islam, M. Z. (2018). Comparing sets of patterns with the jaccard index. Australasian Journal of Information Systems, 22. https://doi.org/10.3127/ajis.v22i0.1538

Goyal, C. (2021, June 23). Semantic Analysis: Guide to master natural language processing (part 9). Analytics Vidhya. Retrieved November 23, 2022, from https://www.analyticsvidhya.com/blog/2021/06/part-9-step-by-step-guide-to-master-nlp-semantic-analysis/

Haldar, Rishin & Mukhopadhyay, Debajyoti. (2011). Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach. Computing Research Repository - CORR.

Harrison, O. (2019). Machine learning basics with the K-nearest neighbors algorithm. Medium. Retrieved September 23, 2022, from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Harsha, T. M., Moukthika, G. S., Sai, D. S., Pravallika, M. N. R., Anamalamudi, S. & Enduri, M. (2022). Automated Resume Screener using Natural Language Processing(NLP)

Hoff, M. (2022, February 27). Businesses are still struggling to attract workers, but it isn't because employers and job seekers have mismatched goals. Business Insider. Retrieved November 6, 2022, from https://www.businessinsider.com/job-mismatch-impacting-hiring-challenges-during-pandemic-indeed-2022-2

Horev, R. (2018, November 17). Bert explained: State of the art language model for NLP. Medium. Retrieved September 11, 2022, from https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

HR software for growing businesses: Freshteam. Freshworks. (n.d.). Retrieved November 18, 2022, from https://www.freshworks.com/hrms/

Huang J., Ren Z., Xin Zhao W., He G., Wen J., Dong D. 2019. Taxonomy aware multi-hop reasoning networks for sequential recommendation. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining,

Human Resources Director (2021). Revealed: Five recruitment priorities for 2021. Retrieved from Human Resources Director

Hunkenschroer, A. L., Luetge, C. (2022). Ethics of ai-enabled recruiting and selection: A review and research agenda. Journal of Business Ethics, 178(4), 977–1007. https://doi.org/10.1007/s10551-022-05049-6

Jabeen, H. (2018, October 23). Stemming and lemmatization in python. DataCamp. Retrieved September 11, 2022, from https://www.datacamp.com/tutorial/stemming-lemmatization-python

Jayanthi, S., Embar, V. & Raghunathan, K. (2021). Evaluating Pretrained Transformer Models for Entity Linking in Task-Oriented Dialog.

Jugran, S., Kumar, A., Tyagi, P. S., & Anand, V. (2021). Extractive Automatic Text Summarization using SpaCy in Python & NLP.

Jung, A. (2022). Machine Learning: The Basics. Springer.

Keyser, P. (2012). Taxonomies and Ontologies. Indexing, 121–142. https://doi.org/10.1016/b978-1-84334-292-2.50007-6

Knight, M. (2020, December 16). Taxonomy vs ontology: Machine learning breakthroughs. DATAVERSITY. Retrieved November 6, 2022, from https://www.dataversity.net/taxonomy-vs-ontology-machine-learning-breakthroughs/

Konys, A. (2015). An approach for ontology-based information extraction system selection and evaluation

Kulkarni, A., Chong, D., & Batarseh, F. A. (2020). Foundations of data imbalance and solutions for a data democracy. Data Democracy, 83–106.

Lad, A., Ghosalkar, S., Bane, B., Pagade, K., & Chaurasia, A. (2022). Machine Learning Based Resume Recommendation System. International Journal of Modern Developments in Engineering and Science.

Lazzareschi, I. (2022, May 23). Skill Data Dictionary, part 2: Taxonomies, Ontologies, and more: ATD. Main. Retrieved September 25, 2022, from https://www.td.org/atd-blog/skill-data-dictionary-part-2-taxonomies-ontologies-and-more

Li, J., Sun, A., Han, J., & Li, C. (2020). A Survey on Deep Learning for Named Entity Recognition

Lutkevich, B. (2020, January 27). What is Bert (language model) and how does it work? SearchEnterpriseAI. Retrieved October 7, 2022, from https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model

Mansour, A. (2021, December 31). Fast and effective ways to extract keyphrases using TFIDF with python. Analytics Vidhya. Retrieved November 7, 2022, from https://www.analyticsvidhya.com/blog/2021/12/how-to-extract-key-phrases-using-tfidf-with-python/

Mao, Y., Zhao, T., Kann, A., Zhang, C., Dong, X.L., Faloutsos, C., Han, J. (2020). Octet: Online Catalog Taxonomy Enrichment with Self-Supervision.

Marrero, M., Urbano, J., Sánchez-Cuadrado, S., Morato, J., & Gómez-Berbís, J. M. (2013). Named entity recognition: Fallacies, challenges and opportunities. Computer Standards & Interfaces, 35(5), 482–489. https://doi.org/10.1016/j.csi.2012.09.004

Meyer, D. (2016). How exactly does word2vec work?

Mihalcea, R. (2004). TextRank: Bringing order into texts. Proceedings of EMNLP.

Mohammad, A. (2022). Basic Implementation of sentiment analysis using BERT.

Mokhtari, N. I. (2022, February 15). What are Siamese neural networks in deep learning? Medium. Retrieved October 11, 2022, from https://towardsdatascience.com/what-are-siamese-neural-networks-in-deep-learning-bb092f749dcb

Monster. (n.d.). About. http://about.monster.com/about/

Monster.com website traffic, ranking, analytics [October 2022]. Semrush. (n.d.). Retrieved November 23, 2022, from https://www.semrush.com/website/monster.com/overview/

Morrison, L. (2021, August 14). Privacy: The use of Artificial Intelligence in recruitment. Gerrish Legal. Retrieved November 18, 2022, from https://www.gerrishlegal.com/blog/2020/07/07/2020-3-2-privacy-artificial-intelligence-in-recruitment

Moses, K. (2021, July 18). Encoder-decoder Seq2Seq models. Medium. Retrieved November 10, 2022, from https://medium.com/analytics-vidhya/encoder-decoder-seq2seq-models-clearly-explained-c34186fbf49b

Naseer, S., Ghafoor, M., Khalid Alvi, S., Kiran, A., Rehman, S.- U., Murtaza, G., Campus, J., Jehlum, P. (2022). Named Entity Recognition (NER) in NLP Techniques, Tools Accuracy and Performance.

Neha, V. (2021, March 9). How to find element by xpath in selenium. BrowserStack. Retrieved October 13, 2022, from https://www.browserstack.com/guide/find-element-by-xpath-in-selenium

Nikishina, I., Logacheva, V., Panchenko, A., & Loukachevitch, N. (2020). Studying taxonomy enrichment on diachronic WordNet versions. Proceedings of the 28th International Conference on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.276

Nikishina, I., Tikhomirov, M., Logacheva, V., Nazarov, Y., Panchenko, A., & Loukachevitch, N. (2022). Taxonomy enrichment with text and graph vector representations. Semantic Web, 13(3), 441–475. https://doi.org/10.3233/sw-212955

Pai, A. (2022, June 21). What is tokenization: Tokenization in NLP. Analytics Vidhya. Retrieved November 10, 2022, from https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/#:~:text=Tokens%20are%20the%20building%20blocks,n%2Dgram%20characters)%20tokenization.

Phan, T. T., Pham, V. Q., Nguyen, H. D., Huynh, A. T., Tran, D. A. & Pham, V. T. (2021). Ontology-based resume searching system for job applicants in Information Technology. Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices, 261–273.

Polner, E. (2022, September 20). Best job search websites. The Balance. Retrieved November 23, 2022, from https://www.thebalancemoney.com/top-best-job-websites-2064080

Powers, D., & Ailab. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. J. Mach. Learn. Technol.

Pretrained models¶. Pretrained Models - Sentence-Transformers documentation. (n.d.). Retrieved April 15, 2022, from https://www.sbert.net/docs/pretrained_models.html

Rasool, A., Tiwari, A., Singla, G. & Khare, N. (2012). String Matching Methodologies: A Comparative Analysis.

Reimers, N., Gurevych, I. (2019). Sentence-bert: Sentence embeddings using Siamese Bert-Networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). https://doi.org/10.18653/v1/d19-1410

Reman, P. (2022). Developments in HR: The future of recruitment, current trends, and challenges.

Resume worded. Resume Examples for 2022 [Handpicked by Recruiters]. (2022). Retrieved November 23, 2022, from https://resumeworded.com/resume-examples

Riabchenko, A. & Zheleiko, I. (2022). Competency demand analysis based on job advertisement data.

Rogushina, J.V., Gladun, A.Y., Pryima, S.M., & Strokan, O.V. (2019). Ontology-Based Approach to Validation of Learning Outcomes for Information Security Domain. ITS.

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. Text Mining, 1–20. https://doi.org/10.1002/9780470689646.ch1

Rule-based matching · Spacy Usage Documentation. Rule-based matching. (n.d.). Retrieved December 1, 2022, from https://spacy.io/usage/rule-based-matching

Russell, S., & Norvig, P. (2003) Artificial Intelligence: A Modern Approach. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition, 848-850

Sabharwal, N., Agrawal, A. (2021). BERT Algorithms Explained. In Hands-on question answering systems with bert: Applications in neural networks and natural language processing. essay, Apress.

Sajwani, M. (2022, January 5). 5 Recruitment Challenges & Solutions for 2022. LinkedIn. Retrieved November 8, 2022, from https://www.linkedin.com/pulse/5-recruitment-challenges-solutions-2022-muhammad-sajwani

Sakamoto, K., & Honiden, S. (2018, January 18). Information extraction apparatus, information extraction method, and information extraction program.

Sanyal, S. (2021, October 26). Rake algorithm in Natural Language Processing: What is rake? Analytics Vidhya. Retrieved October 11, 2022, from https://www.analyticsvidhya.com/blog/2021/10/rapid-keyword-extraction-rake-algorithm-in-natural-language-processing/

Schuster, M., & Nakajima, K. (2012). Japanese and Korean Voice Search. 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). https://doi.org/10.1109/icassp.2012.6289079

Shakya, A., Paudel, S. (2019). Job-candidate matching using ESCO Ontology. Journal of the Institute of Engineering, 15(1), 1–13.

SHRM. (2022, April 13). Fresh SHRM research explores use of automation and AI in HR. SHRM. Retrieved November 6, 2022, from https://www.shrm.org/about-shrm/press-room/press-releases/pages/fresh-shrm-research-explores-use-of-automation-and-ai-in-hr.aspx

Sibarani, E. M., Scerri, S., & Dadzie, A.-S. (2020). Skills and Recruitment Ontology. Skills and Recruitment Ontology (SARO). Retrieved November 22, 2022, from https://elisasibarani.github.io/SARO/

Song, H.-J., Jo, B.-C., Park, C.-Y., Kim, J.-D., & Kim, Y.-S. (2018). Comparison of named entity recognition methodologies in biomedical documents. BioMedical Engineering OnLine, 17(S2). https://doi.org/10.1186/s12938-018-0573-6

Spacy · Industrial-strength natural language processing in Python. · Industrial-strength Natural Language Processing in Python. (n.d.). Retrieved November 24, 2022, from https://spacy.io/

Spacy models documentation. English. (n.d.). Retrieved November 24, 2022, from https://spacy.io/models/en

Stone, D. L., & Deadrick, D. L. (2015). Challenges and opportunities affecting the future of Human Resource Management. Human Resource Management Review, 25(2), 139–145. https://doi.org/10.1016/j.hrmr.2015.01.003

Tainter, M., Davis, D., Turkewitz, N. (2022, January 6). What is the difference between a taxonomy and an ontology? Copyright Clearance Center. Retrieved November 6, 2022, from https://www.copyright.com/blog/taxonomy-vs-ontology/

Takeoka, K., Akimoto, K., & Oyamada, M. (2021). Low-resource taxonomy enrichment with pretrained language models. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. https://doi.org/10.18653/v1/2021.emnlp-main.217

Tarpey, M. (2022, October 13). 72% of employers expect talent acquisition roles will be automated by 2027. CareerBuilder's Employer Resource Center. Retrieved November 8, 2022, from https://resources.careerbuilder.com/recruiting-solutions/talent-acquisition-automated-by-2027

Terry-Jack, M. (2019, May 3). NLP: Pretrained named entity recognition (NER). Medium. Retrieved November 24, 2022, from https://medium.com/@b.terryjack/nlp-pretrained-named-entity-recognition-7caa5cd28d7b

The Esco Classification. ESCO. (n.d.). Retrieved November 7, 2022, from https://esco.ec.europa.eu/en/classification

The Occupational Information Network (O*NET) official website. O*NET Resource Center. (n.d.). Retrieved September 27, 2022, from https://www.onetcenter.org/

Tikhomirov, M., Loukachetitch, N. (2021). Domain-specific Taxonomy Enrichment based on Meta-Embeddings.

Todorov, G. (2022, November 1). Best recruitment stats and trends 2022. Learn Digital Marketing. Retrieved November 8, 2022, from https://thrivemyway.com/recruitment-stats/

Updated resume dataset. Kaggle. (2022, February 1). Retrieved November 29, 2022, from https://www.kaggle.com/datasets/jillanisofttech/updated-resume-dataset

Vickery, S. (2022, July 11). Automated hiring systems could be making the worker shortage worse. UWM REPORT. Retrieved November 6, 2022, from https://uwm.edu/news/automated-hiring-systems-could-be-making-the-worker-shortage-worse/

Vijaymeena, M. K., & Kavitha, K. (2016). A survey on similarity measures in text mining. Machine Learning and Applications: An International Journal, 3(1), 19–28. https://doi.org/10.5121/mlaij.2016.3103

Wang, Y., Allouache, Y., Joubert, C. (2021). Analysing CV Corpus for Finding Suitable Candidates using Knowledge Graph and BERT.

What is Esco? ESCO. (n.d.). Retrieved November 7, 2022, from https://esco.ec.europa.eu/en/about-esco/what-esco

Wimalasuriya, D. C., & Dejing, D. (2010). Ontology-based information extraction: An introduction and a survey of current approaches. Journal of Information Science, 36(3), 306–323.

Wissler, L., Almashraee, M., Monett, D., & Paschke, A. (2014). The Gold Standard in Corpus Annotation.

Yang, L. & Tao, Y. (2017). Word Embedding for Understanding Natural Language: A Survey.

Zong, C., Xia, R., & Zhang, J. (2021). Text Data Mining. Springer.

Appendix 1. Extract from the Golden Corpus (sentences from the JDs from Monster.com website)

Sentence from the JD	**Skills 1**	**Skill 2**	**Skill 3**	**Skill 4**	**Skill 5**
`Must possess effective oral and written skills and strong analytical and problem-solving capabilities.	Written Communications	Verbal Communications	Analytical mindset	Troubleshoot problems	New
Participate in an agile scrum team writing user stories, error mapping, and service mapping and working with the lead product manager.	Scrum	User stories	New	New	New
Design queries to perform data analytics, and data extractions across business databases using various data mining tools (such as SQL).	Data analysis	Extracting information and knowledge	SQL	MS SQL	New

Sentence from the JD	Skills 1	Skill 2	Skill 3	Skill 4	Skill 5
Perform quantitative and qualitative analysis of data used to prepare and report various metrics of the company to both internal and external parties using PowerPoint, Visio, and Excel.	Data analysis	MS Powerpoint	MS Excel	MS Visio	New
Responsible for supporting planning activities and supporting the on-schedule delivery of milestones and deliverables.	Planning, accounting, control, and adjustment of work deadlines	Planning and organization of work	New	New	New
Must be a team player able to work in a dynamic environment and have a working knowledge of the SDLC and the associated processes and documentation.	Teamwork	New	New	New	New

Appendix 2. Suitability of CVs obtained by expert's evaluation

	Business Analyst	**Data Analyst**	**Data Architect**	**Enterprise Architect**
"Good"	Candidate № 12 - 14, 17, 19, 20, 43	Candidate № 1 - 3, 5 - 8, 11, 44, 47	Candidate № 22 – 27, 29 – 31, 36	Candidate № 31 – 35, 39, 41
"Average"	Candidate № 3, 4, 6, 9, 15, 16, 18, 21, 42	Candidate № 4, 9, 10, 19, 20, 22, 24, 49	Candidate № 28	Candidate № 22, 27, 29, 30, 36 – 38, 40
"Not suitable"	Candidate № 1, 2, 5, 7, 8, 10, 11, 22 – 41, 44 - 54	Candidate № 12 - 18, 21, 23, 25 – 43, 45, 46, 48, 50-54	Candidate № 1 – 21, 32 – 35, 37 - 54	Candidate № 1 – 21, 23 – 26, 28, 42 - 54