ulm university universität
uulm

Ulm University
Faculty of Mathematics and
Economics

Mapping companies' readiness for the future: A novel data-driven approach to extract companies' future skills using social media analytics

Bachelor Thesis

in Physics and Management

submitted by
Evertz, Lennard, 2000811, Physics and Management, 7th Semester
on June 22, 2020

Reviewer

Prof. Dr. Mathias Klier

Table of contents

List of figures

List of tables

List of abbreviations

Information and communication technology	ICT
Partnership for 21st-century skills	P21
Continuous Bag of Words	CBOW
Skip-Gram	SG
Part-Of-Speech	POS
Distributed Dictionary Representations	DDR

1 Introduction

1.1 Motivation of Research Question

Future workspace is changing. Current literature argues that progress in technologies like robotics and artificial intelligence will have a big impact on the way employees will carry out their tasks, if they are needed to do so at all (Balliester & Elsheikhi, 2018; Daheim & Wintermann, 2016; Kirchherr et al., 2018; OECD, 2018; Smit et al., 2020). While most predict an increase of unemployment in low- and middle-skilled jobs as well as those characterized by administrative and predictive activities (Balliester & Elsheikhi, 2018; Chui et al., 2016), a study by the Bertelsmann Foundation anticipates that unemployment both in advanced and emerging economies could rise as high as 21% in Europe and over 25% in North America by 2050 (Daheim & Wintermann, 2016). Smit et al. (2020) find that 22% of current work activities will be automated by 2030. In a timeframe of two decades, about 50% of all work activities found today will be automated and require no human action (Manyika et al., 2016). Indeed, some of these changes can already be observed: Not least because of the COVID-19 pandemic, companies operate in virtual teams to be more flexible and productive (Daheim & Wintermann, 2016; Smit et al., 2020; Townsend et al., 1998), distance learning courses get more popular (Statista, 2020), workplaces realize a digitalization (Benson et al., 2002; Smit et al., 2020), and consequently employees as well as future employees are required to adapt into unprecedented digital environments. Manyiaka et al. (2016) find that 60% of all occupations are faced with a potential automation of at least 30% of their activities with today's available technology. Yet, in the advent of technological progress, employers are already left behind (Daheim & Wintermann, 2016). As technological change is thought to continue rising in the future, it will get harder to cope with this challenge for both employers and employees (Daheim & Wintermann, 2016). New work environments and forms of social interaction confront people to adapt into a new set of digital and social skills (Kirchherr et al., 2018). About 94 million employees will need retraining to handle the changed workspace, while 21 million need to change their workspace by 2030 (Smit et al., 2020). To accomplish a better transition into automated job environments, it is a crucial step for companies to examine the skills found in their employees and support their workforce through up- and reskilling (Balliester & Elsheikhi, 2018; Kirchherr et al., 2018; Smit et al., 2020; World Economic Forum, 2018). For instance, until 2023, around 700.000 people will have to learn advanced digital skills, to satisfy the need for tech-specialists in Germany alone (Kirchherr et al., 2018). Consequently, it has become more important than ever, that leaders establish an process to assess and map current workforce skills to reveal competitive advantages or uncover potential future skill gaps (Smit et al., 2020). As a proposal for the examination process, this thesis presents a model to effectively map companies' readiness for the future. By extracting *future skills* from employees' social media

profiles with the help of distributed dictionary representations, companies can be compared to industry standards and direct competitors. This helps companies to identify advantageous developments, when *future skills* are represented above average, and skill gaps, when represented below average. The following subsections present the research question and the remainder of the thesis, before an overview about the theoretical background as well as the implementation of the proposed process is given.

1.2 Research Question

Against this background this thesis aims at answering the following research question: How can companies' readiness for the future automatically be extracted using *future skills* and social media analytics? The thesis demonstrates and evaluates a novel approach to automatically extract companies' readiness for the future with respect to *future skills*. The approach is inspired by prior work on dictionary-based approaches for text classification, distributed word or text representations and in particular text mining approaches.

1.3 The remainder of the thesis

The remainder of this thesis is structured as follows: In the next section, an overview about related work on future skills and traditional skill identification approaches is provided. Thereafter, a novel approach to automatically extract companies' readiness for the future with respect to *future skills* is proposed. Then, the approach is demonstrated based on a real-world dataset obtained from New Work SE (New Work SE, 2020). Afterwards, a critical reflection on limitations is presented. The thesis concludes with practical implications, directions for further research and a brief summary.

2 Theoretical Background

In the following, clear distinctions between skills, abilities and knowledge and related work on *future skills* are presented. The approach derived in this thesis is based on *future skills*, which are defined in the next subsection. Thereafter, the research gap is carved out. Lastly, literature on traditional approaches of skill identification and particularly text mining is discussed, representing methodological foundations for the novel approach.

2.1 Definition of skill, knowledge and ability

In general, having a skill is far more than just knowing something about a specific topic. A skill is something that is learned rather than a generic gift (Julita, 2011). Attewell (1990) explains that the concept of a skill is rather complex but sums up existing definitions of skill as the ability to do something well (Attewell, 1990; Cambridge Dictionary, 2020). Until today, there is no consensus about the concept of skill. Scientists form different research disciplines and countries often have different conceptions about skills, for instance due to the variation of languages

they use (Green, 2011). Additionally, *skill* is often used alongside, or even as a synonym, with words with similar, yet different meanings. To those words belong words like *knowledge* or *ability* (Green, 2011). Since skills are the foundation of productivity and innovation (Cainelli et al., 2004) and they in turn are not only important for organizations survival and prosperity (Baumol, 2002) but also for economic and social growth (Cainelli et al., 2004; Kavoo-Linge & Kiruri, 2013) it should be worth considering defining skills in a uniform system. For that, it is important to delineate common miss-interpreted synonyms.

First, *knowledge* is a multilayered construct. A widely adapted definition separates knowledge into four dimensions (Krathwohl, 2002). The first one is the dimension of *factual knowledge*. Here, one can find terminology and specific details about a topic (Krathwohl, 2002). The second knowledge dimension is called *conceptual knowledge*. This dimension handles the intercorrelation among basic elements by means of classification and categories. Also, knowledge of generalizations, structures, models and theories belongs to this dimension (Krathwohl, 2002). The third knowledge dimension is called *procedural knowledge*. This dimension moves the knowledge space into a more specific area. When looking at a particular problem, procedural knowledge helps one to know what skills, methods, algorithms or techniques are needed to solve that problem (Krathwohl, 2002). Therefore, not only the knowledge of certain methods or techniques is needed, but also the ability to adapt these to a given situation. The last dimension is called *metacognitive knowledge*. This dimension describes knowledge of cognition in general and strategic knowledge (Krathwohl, 2002). What concerns skills, it is not enough to simply understand (theoretically or mentally) a matter. Compared to knowledge, which you can obtain by reading a book or researching online, skills need to be developed through training or experience (Attewell, 1990; Julita, 2011).

Contrasting skills from *abilities* is not as easy as doing so from knowledge. Although skills can also be abilities, they are not the same (Julita, 2011). An ability is generic and can be inherited from one's parents (Julita, 2011). Note that abilities therefore make up the basis of developing skills of an individual. Surely, abilities can be considered as skills in some cases. For example, having the ability to run a certain distance in a given time is also considered a remarkable skill, given an exceptional ratio of time and distance (Julita, 2011).

To sum up, skills are closely related to other concepts, and yet derive at an alone standing description. This not only implies the definition of ability mentioned above, but also knowledge and understanding. In literature, one can find several skillsets that map important skill categories or groups for being successful in different areas of life. The next subsection is meant to describe such groups a little further.

2.1.1 Defining skill categories

Literature provides various concepts to categorize skills (Cimatti, 2016; Green, 2011). These skill categories determined by different works may overlap in content (Cimatti, 2016; Green, 2011; Steptoe & Wardle, 2017; van Laar et al., 2017). According to literature, skills can describe a wide variety of areas, resulting in the most prominent categories social skills, life skills, soft skills, as well as job related and professional skills and hard skills. In the following, these categories are described in more detail.

Social, life and soft skills in general are skills that are needed in order to communicate and interact with others (Indeed, 2020; SkillsYouNeed, 2020). They can be used to build, maintain and grow strategic relationships (Indeed, 2020) and can therefore be beneficial for one's career in accomplishing tasks by working with others, learning about new opportunities and ensuring a better environment at the workplace. The importance of these skills is highlighted by the increased growth of employment and wage in jobs that require a high level of social skills (Deming, 2017). Examples of social skills are empathy, relationship management, active listening and conflict resolution (Indeed, 2020).

Job related skills on the one side describe the skills needed in specific fields of work (University of Nebraska-Lincoln, 2020). They are important to complete tasks in the desired field. For example, a hair stylist needs a different job skill set than a banker or carpenter. On the other side, professional skills are helpful in multiple occupations. These form the base of a successful job execution, since they make a person behave in an exemplary manner (Reverso-Softissimo, 2020). Together, these skills are the driving force for companies and whole industries regarding (technological) change and implementation of new techniques (Barbosa & Faria, 2008). Example skills are leadership, mentoring, collaboration and conflict management (University of Nebraska-Lincoln, 2020).

Hard skills are often referred to as the opposite of soft skills. Thus, they are not the ones characteristic of a personality (like speaking a certain language) but required to perform a specific task (Cimatti, 2016).

As one can see, these skill categories and examples aim to describe capabilities with that one can achieve a higher standard in life, job and other perspectives. Recent studies however find that nowadays it is not enough to have competencies in the above-mentioned fields (Binkley et al., 2012). This is partly due to the increased use of digital technologies (Chui et al., 2016; Manyika et al., 2016), their development and the accelerating globalization (OECD, 2018). Furthermore, scientists introduce a new skill concept dimension that considers such skills, that are becoming more and more important in everyday life as well as in a professional context (Ahonen & Kinnunen, 2015; Balliester & Elsheikhi, 2018; Binkley et al., 2012; Kirchherr et al., 2018; OECD, 2018; Partnership for 21st Century learning, 2015; van Laar et al., 2017): *Future skills*.

2.1.2 Future skills

Within the concept of skills, *future skills* have attracted wide interest (Balliester & Elsheikhi, 2018; van Laar et al., 2017). Indeed, assuring *future skills* is seen as a survival factor from companies' perspective (Fareri et al., 2020; van Laar et al., 2017). Consequently, there exists rich literature defining and examining *future skills* which predominantly differ in timeframe and scope of application. In the following an overview about this literature is given.

Table 1: Future skills according to Kirchherr et al. (2018).

Included skills	Skill category	Future Skills definition
Complex data analysis	Technological (expert) Skills	Future Skills are defined as skills that become much more important across all sectors and industries as well as in social life in the course of the next five years.
Smart hardware and robotics		
Web-engineering		
User-centric design		
Conception and administration of networked IT systems		
Blockchain technology development		
Tech translation		
Digital literacy	Digital Basic Skills	
Digital interaction		
Digital collaboration		
Agile working		
Digital learning		
Digital ethics		
Problem solving skills	Classical Skills	
Creativity		
Entrepreneurial action & self-initiative		
Adaptability		
Stamina		

First, Kirchherr et al. (2018) define future skills as competencies, that, in a timeframe of five years after publication in 2018, gain significant amounts of importance in terms of social participation and professional contexts. Including all industries and branches, they deliver an approximate overview over the competencies that are missing in German companies. That being said, they do exclude very important skills that are of high importance already and focus on skills that will become important in the future exclusively. By conducting workshops with

participants from start-ups, known companies, educational institutions and politicians, and surveying over 600 companies from a broad range of industries, they built a framework that includes three categories of future skills covering seven, six and five skills respectively: *Technological (expert) Skills, Digital Basic Skills and Classical Skills* (Kirchherr et al., 2018). A more specific description of the included skills can be found in table 1. Though the focus lays on German companies, the scope of this framework can be widened and taken for estimating the overall future skill need for international companies.

Second, Davies et al. (2011), who looked at future skills as skills that are needed due to the changed landscape of work, chose another approach. Like in the previous framework, the skills addressed are thought to scope across different work settings and industries without pointing out specific jobs of the future. The basic methodology of the framework creation was similar to that of Kirchherr et al. (2018), though the extent was very different: An expert workshop was held, where people with diverse backgrounds came together to brainstorm about key drivers that will shape the landscape of work and ways, in that these drivers will change the work skill requirements.

Table 2: Future skills according to Davies et al. (2011). (1)-(6) behind the given skills denote the assignment to the key-drivers (1)-(6).

Given skills	Key driver	Future Skills definition
Sense-Making (1)	rise of smart machines and systems (1)	Future work skills are defined as skills, proficiencies and abilities that are required across different jobs and work settings with a timeframe of ten years.
Novel and Adaptive Thinking (1) (2)		
Social Intelligence (1) (2)	globally connected world (2)	
Cross Cultural Competency (2) (3)		
Virtual Collaboration (2) (3)	super structured organizations (3)	
Design Mindset (3) (4)		
Cognitive Load Management (3) (4) (5)	computational world (4)	
New Media Literacy (3) (5) (6)	new media ecology (5)	
Computational Thinking (4) (5)	extreme longevity (6)	
Transdisciplinary (4) (6)		

As a result, Davies et al. (2011) identified a total of six key drivers and ten skills that will be dominant in shaping the work of the future. Also, the timeframe chosen was twice the length, looking ten years into the future from 2011 on. A uniqueness of this framework compared to all the other ones that can be studied in this chapter is that the individual skills are not exclusively assigned to one of the six key drivers and rather are included in multiple key driver categories. For example, the skill Novel and Adaptive Thinking is assigned to the two key drivers rise of smart machines and systems and globally-connected world (Davies et al., 2011). Again, a complete overview of this framework can be found in table 2.

Third, van Laar et al. (2017) summarized articles that focused on 21st-century skills. Starting at 1592 articles, they introduced four criteria the articles had to fulfill in order to be considered in their review. These can be summarized as follows:

- Articles had to focus on the 21st-century skill dimension or related terms, including technical aspects
- Articles had to present a measurement or specific conceptualization for 21-century skills
- As the review aims to present a framework scoping the current workforce, articles had to mention skills in context with workforce preparation
- Articles had to be published in a peer-reviewed journal for quality reasons

Table 3: 21-st-century digital skills according to van Laar et al. (2017).

Given skills	Skill category	21st-century digital skills definition
Technical	Core 21st-century digital skills	21st-century digital skills are first of all defined as skills that support the mastery of information and communication technologies (ICT) for work, skills that are not directly associated with any particular software, skills that support higher order thinking and skills favoring employees' continuous learning.
Information management		
Communication		
Creativity		
Critical thinking		
Problem Solving		
Ethical awareness	Contextual 21st-century digital skills	
Cultural awareness		
Flexibility		
Self-direction		
Lifelong learning		

Through these criteria, the reviewed articles shrunk to 75. Even though the extracted skills, e.g. *flexibility*, can be seen as non-digital skills, van Laar et al. (2017) focused on the digital aspects and therefore one has to see the skills contained in the framework in a digital context as well. They found two skill categories (*Core 21st-century digital skills* and *Contextual 21st-century digital skills*), which address six and five skills respectively (van Laar et al., 2017). Due to the nature of this review, there is no precise timeframe, other than that of the 21st-century, presented. A more detailed view on the framework can be found in table 3.
Fourth, Binkley et al. (2012) look at future skills as skills that can be classified into four concepts of living and working. The concepts include *Ways of Thinking, Ways of Working, Tools for Working* and *Living in the World* (Binkley et al., 2012). Though the scope of application on peoples' everyday life is new, the included skills can be found in all the other frameworks as well. This is of no surprise considering they analyzed twelve frameworks from different countries to derive at the proposed framework pictured in table 4. Due to the horizon of their work, the included skills can be used for a framework with international validity.

Table 4: Future skills according to Binkley et al. (2012).

Given skills	Skill category	Future Skills definition
Creativity and innovation	Ways of Thinking	Future skills are skills that enable people to communicate, share and use information to solve problems, to innovate and adapt into new environments, use technology for their advantages and enhance productivity.
Critical thinking, problem solving and decision making		
Learning to learn, metacognition		
Communication	Ways of Working	
Collaboration (teamwork)		
Information literacy	Tools for Working	
ICT literacy		
Citizenship - local and global	Living in the World	
Life and career		
Personal and social responsibility - including cultural awareness and competence		

Fifth, Ahonen and Kinnunen (2015) followed another approach in finding important future skills. They expanded the framework from Binkley et al. (2012) by two skills, namely *independent initiative* and *ecological awareness* and let 718 students (361 females and 357 males) between classes 5 and 9 value the skills' relative importance on a scale from 0 to 10. They found that

collaboration (8.6) was valued the most important, followed by *independent initiative* (8.3) and *work skills* (8.3). As the least important skill students chose *cultural awareness* (6.5). The remaining ratings can be observed in table 5. With the lowest average rating of 6.5, students value future skills in the upper scale of importance and believe that it gets more and more important for them to learn such skills to be successful in their future workspace (Ahonen & Kinnunen, 2015). Consequently, the scope of this framework addresses current students' perceptions about their future workplace, setting the timeframe to a minimum of seven years, the time by that students from fifth grade could enter working world.

Table 5: Future skills importance as found by Ahonen and Kinnunen (2015).

Given skills	**Skill Importance**
Collaboration	8.6
Independent initiative	8.3
Work skills	8.3
Learning skills and lifelong learning	7.7
Technical proficiency	7.7
Social responsibility	7.7
Creativity	7.6
Problem solving	7.6
Communication	7.4
Ecological awareness	7.0
Critical thinking	7.0
Citizenship	6.8
Information literacy	6.6
Cultural awareness	6.5

Sixth, the Partnership for 21st Century Skills (P21) is a joint government-corporate organization from the United States, making its framework for future skills widely accepted (Ahonen & Kinnunen, 2015; Binkley et al., 2012; Partnership for 21st Century learning, 2015; van Laar et al., 2017). The P21 (2015) framework identifies four different skill, knowledge and literacy categories (*21st-century themes, learning and innovation skills, Information and Communication Technology (ICT)* and *Life Skills*), that students must master in order to be successful in future work and life. Compared to the other frameworks, this one does not have a distinct focus on digital skills and rather deals with life skills. Even though *ICT* is an alone standing category, one can only find the skill *ICT literacy* in this category. On the opposite side, *life skills* includes eight different individual skills. This framework is applicable for current students and therefore

has a timeframe of multiple years (Partnership for 21st Century learning, 2015). The full framework can be seen in table 6.

Table 6: Future skills according to the Partnership for 21st Century learning (2015) framework.

Given skills	Skill category	Future Skills definition
Global awareness	21st-century themes	Future Skills are skills that students need in order to succeed in work and life consisting of content knowledge, specific skills, expertise and literacies.
Financial, Economic, Business and Entrepreneurial Literacy		
Civic Literacy		
Health Literacy		
Environmental Literacy		
Creativity and innovation skills	Learning and innovation skills	
Collaboration and communication		
Critical thinking and problem solving		
Information literacy		
Media literacy		
ICT literacy	ICT	
Flexibility	Life skills	
Adaptability		
Independent (working)		
Personal productivity		
Ethics		
People skills		
Social direction		
Productivity		

Seventh, the OECD Learning Compass 2030 (2018) is another important orientation when analyzing *future skills*. The OECD (2018) defines skills as the ability to carry out a process and to be able to use one´s knowledge in a responsible way to achieve a goal (OECD, 2018). In the Future of Education and Skills 2030 initiative, the goal is to find skills, knowledge, attitudes and values that today`s students will need for shaping their world in 2030 (OECD, 2018). The organization and stakeholders around the world built, reviewed, tested and validated the first results of the framework, which consists of three main categories (*Cognitive- and metacognitive skills, Social and emotional skills* and *Practical and physical skills*) and found it to be relevant across the globe (OECD, 2018). The topically nature of this work and the global involvement make this framework a very important guideline when examining future skills. The

timeframe chosen is that of a complete school cycle of students who enter school in 2018 so that by the time they enter the working world, they have learned all the skills they need due to the changed workspaces. The complete framework can be seen in table 7.

Table 7: Future skills according to OECD (2018).

Given skills	Skill category	Future Skills definition
Critical thinking	Cognitive- and metacognitive skills	No definition of future skills given.
Creative thinking		
Learning-to-learn		
Self-regulation		
Empathy	Social and emotional skills	
Self-efficacy		
Responsibility		
Collaboration		
Use of new ICT (devices)	Practical and physical skills	

2.1.3 Extracting Future Skills

Considering the frameworks and definitions from above-mentioned publications, one has a better understanding of what *future skills* are. Kirchherr et al. (2018) define future skills as competencies that, during the next five years after publication and overlapping different industries, gain significance in both social and professional life. Similarly, Ahonen and Kinnunen (2015), OECD (2018) and P21 (2015) see *future skills* as skills current students must master in order to have a successful start into professional working life. According to discussed literature, future skills do not necessarily include completely novel skills and rather build a new scope on top of already existing skills.

The categories of *future skills* displayed in known frameworks have different terms describing the content but follow a recurring pattern. That way, the combination of selected digital, people and life skills together form a new concept called *future* or *21st-century skills*. Nevertheless, different studies employ different definitions of *future skills* with respect to for instance timeframe, i.e., the future time horizon for future skills being relevant, or scope of applicability, i.e., the target groups that are addressed with the proposed frameworks. Thus, in this thesis, an own framework, based on prior literature, is developed. In particular, this thesis focuses on digital skills, as automation and globalization take place in many industries: Companies operate in virtual teams to be more flexible and productive (Townsend et al., 1998), distance learning courses get more popular (Statista, 2020), workplaces realize a digitalization (Benson et

al., 2002), and consequently employees as well as future employees are required to adapt into unseen digital environments. Against this background, *future skills* are defined as competencies, that will foremost provide employees and future employees with a necessary skill set to cope with such workplace and life changes. Moreover, companies need the right distribution of such skills represented by their employees to realize a future-oriented and competitive company path (Kavoo-Linge & Kiruri, 2013; Kirchherr et al., 2018).

In summary of the depicted frameworks, the underlying future skills framework consists of four categories. First, *social, people and emotional skills* include skills and competencies that stand for a high affinity towards collaborative activity. In addition to *teamwork, empathy, cooperation* and other selected people skills one can find in the Binkley et al. (2012), Ahonen and Kinnunen (2015), P21 (2015) and OECD (2018) frameworks, this skill category incorporates the internationalization of companies as it integrates *intercultural* and *sociocultural* aspects. Second, inspired by the OECD Learning Compass 2030 (2018), the underlying future skills framework considers *cognitive- and metacognitive skills*. These are especially crucial for the strategic placement of employees in a company (Kavoo-Linge & Kiruri, 2013). Besides *critical thinking, creativity* and *problem solving,* this category focuses on the versatility of modern companies by including associated skills such as *innovative thinking* and *adaptability*. Third, *digital competencies* represent a future skill category as digitalization takes over a big part in today's frameworks (Binkley et al., 2012; Kirchherr et al., 2018). Two sub-categories are associated with *digital competencies*: On the one hand, *digital base skills* describe skills that are needed in everyday life and workplace in terms of participation in a digital world (Kirchherr et al., 2018). For instance, therein included are *digital learning, computer skills* and *digital interaction.* On the other hand, *expert digital skills*, covers competencies that deal with the handling and implementation of transformative technologies (Kirchherr et al., 2018). *Blockchain development*, *web engineering* and *software programming* are adopted skills from Kirchherr et al. (2018) and are supplemented by key drivers such as *data visualization, artificial intelligence, machine learning* and others for this skill category. Table 8 explains these categories in more detail.

Now, that a new framework dealing with future skills is introduced, the following subsection presents traditional approaches for skill identification, which are found to be inefficient and therefore are not optimal for the application of the framework. Thereafter, a more convenient method is carved out.

2.2 Traditional analysis of skills

The frameworks described either give an overview about competencies one has to learn in order to have a successful start into the working world (Ahonen & Kinnunen, 2015; Partnership for 21st Century learning, 2015) or are used to identify skill gaps in companies, industries, or countries (Kirchherr et al., 2018; OECD, 2018). The latter requires not only a suitable

framework, but also methods to determine the actual and target values of skill level. This subsection presents traditional approaches of skill identification. Companies map their employees' skills for various reasons. For instance, when new projects come up and the project manager has to do personnel planning, or, when paired with other characteristics like age and position, to realize a better future-oriented recruitment process (Kanning, 2019, pp. 17-19; Rosetti & Langhoff, 2015). Ultimately, the proposed method aims to identify companies' readiness for the future by extracting future skills from social media profiles of employees.

2.2.1 Overview

Employees' competencies can be mapped in different ways. First, though following no universal standards, a quality matrix is a famously used mapping method that displays different employees on one axis and workspaces or tasks on the other axis. The matrix can then be filled with values that represent a quality measure (Propp et al., 2003; Rosetti & Langhoff, 2015). Thus, one can easily read how good someone is at a certain job, or aggregate the values category-wise for a company value (Propp et al., 2003). As a consequence, process and organizational planning is more efficient (Rosetti & Langhoff, 2015). Additionally, the training need as well as future quality gaps can be highlighted by comparing actual and target values (Rosetti & Langhoff, 2015). Second, a competence pass is another in literature proposed way to organize peoples' skills (Rosetti & Langhoff, 2015). Competence passes list skills and abilities in which employees are sufficient in, focusing on soft skills (Rosetti & Langhoff, 2015). Due to the objectivity and comparability, these passes are especially helpful when companies seek to internally recruit for additional training or promotion (Rosetti & Langhoff, 2015).

No doubt, depictions like the quality matrix or competence pass are beneficial for companies. With them, employers know what qualities hide in their company. Better personnel planning, more targeted employee training and finally cost efficiency are just some reasons that speak for an implementation of such systems (Kanning, 2019, pp. 5-6; Rosetti & Langhoff, 2015). Therefore, the question arising is not whether or not such systems benefit companies, but how companies gather the information contained in systems like the quality matrix or competence pass. Most systems have a high complexity at first implementation in common (Kanning, 2019, p. 128; Rosetti & Langhoff, 2015). A prominent example of data collection is proposed by Rosetti and Langhoff (2015) with three steps for the implementation of a quality matrix.

First, a registration sheet has to be designed by supervisors and finalized by department-specific employees. This sheet later is used to determine the job-related actual values of competencies on a predefined scale. For that, employees have to fill out the sheet and discuss the result with a supervisor for objectivity reasons. Second, supervisors define a target number of employees that should have a certain skill-level for every skill contained in the registration sheet. This step is crucial for the comparability of actual and target values after the completed

registration sheets are entered into the matrix (Rosetti & Langhoff, 2015). Third, the emerging training needs are prioritized for later revision. The first step has to be completed in recurring sessions. What concerns competence passes, the overriding goal is to compare self- and external assessment with target values of transferrable skills (Rosetti & Langhoff, 2015). For that, these skills are categorized into subgroups first and further individualized into unique competencies after. Then, a statement catalogue is introduced. Statements are used to assess self- and external ranking of employees (Rosetti & Langhoff, 2015). The external ranking is done by a direct supervisor of each employee. Additionally, target values for each unique competence are set through an internal workshop, held by management and experienced employees. Rosetti and Langhoff (2015) propose a testing phase before adopting the system to company level as this reduces the risk of ambiguities in statement catalogue and target values. Companywide actual and target values can be observed by averaging the employees' values of each competence. Again, the degree of topicality maintains by assessing periodically only (Rosetti & Langhoff, 2015).

On a more general level, Kanning (2019) presents personnel diagnostics to consist of the three basic components *questioning, monitoring* and *testing*. First, *questioning* is used both in quality matrices and competence passes. Here, the focus lays on competencies. In general, questioning can be used for any information about not only competencies, but also attitudes, behavior and consequences from such behavior (Kanning, 2019, p. 117). Questioning can be done orally (e.g. job interview) or written (e.g. competence pass). The second aspect of personnel diagnostics deals with the behavior of people. *Monitoring* happens unintentionally in the usual workday by verbal and visual interaction (Kanning, 2019, p.132). Because the meaningfulness of such perceptions is questionable, one has to realize a more systematic way for transferrable input (Kanning, 2019, p.118). Literature shows that assessment centers achieve promising results in predicting managerial success (Klimoski & Brickner, 1987; Waldman & Korbar, 2004). Therefore, a good approach to systematically monitor a person is by establishing an environment similar to that of an assessment center (Kanning, 2019, p.118), where trained personnel can analyze ones' behavior. Third, *testing* focuses on intellectual competencies. Here, the test taker is asked to work through different tasks and questions, usually with a definite correct and multiple wrong answers (Kanning, 2019, p.118). The success and comparability can be measured by counting the correctly answered questions (Kanning, 2019, p.118). In any case, the integration of numerous actors, e.g. employees, supervisors and management, is needed in recurring sessions, in order to maintain a long-lasting and relevant skill overview.

Nowadays, it has become usual to handle the data obtained with any diagnostics system with a computer (Kanning, 2019, p.148; Obermann, 2018, pp. 418-419; Roussos et al., 2007). Tests or role play situations, which are often part of assessment centers (Obermann, 2018, pp. 136-

140), require a high degree of flexibility as emerging tasks or situations are more realistic and therefore more insightful when they carry on where previously solved issues have ended (Cudeck, 1985; Kanning, 2019, pp. 154-155; Obermann, 2018, pp. 418-421; Weiss, 2004). Consequently, not only the computer-based handling of data, but also computer-based adaptive tests, which take into account previously handled questions before choosing the next ones, are reasonable enhancements for modern questioning, testing and assessment (Cudeck, 1985; Kanning, 2019, pp. 154-155; Weiss, 2004).

2.2.2 Problems

Non-withstanding the benefits of these methods, several problems concerning implementing and maintaining these methods in organizations have to be considered.

First, a big problem of skill overviews like the ones mentioned is the topicality of the data collected. Due to the nature *of questioning, testing* and *monitoring*, the depicted competencies are only as new as their latest observations. Consequently, in order to form a good base for personnel planning at any given time, the procedures of adapted systems have to be repeated as often as possible (Rosetti & Langhoff, 2015). Questioning and tests usually consist of multiple questions, which, if paper-based documented, have to be transferred into a digital format for evaluation (Kanning, 2019, p. 153). Additionally to the employees filling out questionnaires, supervisors are an inherent part of the documentation process as this optimizes objectivity (Rosetti & Langhoff, 2015). Considering the workforce that is not available during the time of skill registration as well as supervisors' higher salary (Hermann & Zimmermann, 2020), this process gets more costly as more updates are being made.

Second, time intensity is a problem favoring the first problem. Besides the time consumed by maintenance and the resulting costs, the time consumed in the creational process takes up a large portion of the overall time consumption (Borg & Mastrangelo, 2009, pp. 153-155). Borg and Mastrangelo (2009) assert that the time consumed by the creational process of a questionnaire depends on the skill and experience of the creator and is dependent on whether or not the company has experience with similar surveys. Indeed, implementing and maintaining such tasks can emerge to be the main task of the human resource department (Borg & Mastrangelo, 2009, pp. 153-155), making it one of the main cost factor of the department. Literature suggests that computer-based questionnaires can counteract these circumstances by providing faster responses on a cheaper cost level with comparable or better data quality (Croteau et al., 2010; Kiesler & Sproull, 1986).

Third, another data quality problem arises from false or missing data. In fact, intraorganizational surveys can be seen as a control instrument and generate fear among employees (Croteau et al., 2010). One way to deal with this situation from examinee-side is to simply ignore questionnaires or to fake answers, which results in decreasing employee satisfaction

(Croteau et al., 2010). However, test creators can improve response rates, as the acceptance rate of employees regarding questionnaires and tests increases if they think job relevant characteristics are being measured (Truxillo et al., 2004). Croteau et al. (2010) also conduct that computer-based examinations are, compared to paper-based ones, similarly easy to use but benefit the enjoyment of participation as well as the involvement in recurring questionnaires. The difficulty, that examinees or observed people try to disguise themselves for a better workspace-related standing and therefore distort the measured data remains (Croteau et al., 2010; Kanning, 2019, p. 130).

To sum the points up, traditional approaches for skill identification are work capacity intensive and therefore costly in terms of time consumption during implementation and the process itself and computational expenditure when done by hand. One might therefore conclude that these methods might not help to instantly study companies' skills and competencies and automatically map them into the future skills framework for the identification of companies' readiness for the future effectively.

To address these issues, following approaches have been already pursued or might be an option. First, creating job profiles for every position occupied in a company partly eliminates the timewise effort for maintaining employees' skill overviews. Job profiles are descriptions of activities, skill level and training one needs in order to work in a job. Thus, a company only once has to create such profiles, which can be done by including currently in this particular job working people (Rosetti & Langhoff, 2015). The aggregation to company skill level can be computed by multiplying the skills included in a job profile with the number of people working in the job associated with the job profile. However, this process generalizes peoples' skills as it ignores peoples' individual skill sets. Second, skill identification based on applicants' CV's using text mining tools can be used for instant analyzation of individuals' competencies. For that, applicants' CV's are computer-read for skill tagging. Found skills can be compared with needed skills, i.e. predefined skills through open job descriptions containing such skills (Singh et al., 2010). The skills mentioned in the CV's of employees can then be used to map companies' skills. This method not only reduces the time needed both for maintenance and implementation but also fastens the recruiting process (Singh et al., 2010), the problem that the skills collected do not remain up to date for a longer period of time remains. Third, distributed representations of texts such as CV's, social media profiles or articles can be used with state-of-the-art accuracies for text classification (Garten et al., 2018). Here, the to be examined texts are represented as vectors. The classification works by calculation of similarities to the available category vectors. Garten et al. (2018) propose that this method can be used for dictionaries with a small number of categories, e.g. skill categories, and prove their results with a binary sentiment analysis, e.g. classification into either positive or negative sentiment, and multiclass morality detection of social media posts. The problem, especially with social media

posts, is that they often contain abbreviations or misspellings, that cannot be treated by the pretrained vector model Garten et al. (2018) use. Therefore, the method they propose helps to instantly classify texts but fails to incorporate unknown words, i.e. words with misspellings. The realization of skill detection through distributed representations has not yet found its way into companies' skill mapping process. Consequently, the problems with data topicality and cost and time inefficient identification processes remain reality. Against this background, this this thesis aims at closing the research gap by developing a novel method for extracting *future skills* for organizations. This method, similar to that of Garten et al. (2018), is based on text mining, distributed representations and concept dictionaries, for which prior literature is summarized in the following.

2.3 Text mining

Text mining describes the semi-automatic process of knowledge extraction, analyzation and structuration from text-based sources (Fareri et al., 2020; Heyer et al., 2006). It has shown promising results for a range of applications, for instance in summarization of customer reviews (Berezina et al., 2016; Jack & Tsai, 2015), social media study (Salloum et al., 2017; Stieglitz & Dang-Xuan, 2013) and finally for estimating the impact of industry 4.0 on job profiles (Fareri et al., 2020). Some works address the need of acceleration of recruitment processes with text mining approaches (Lumauag, 2019; Manad et al., 2018; Poonnawat et al., 2017) but only occasionally focus on the influence of employees' skills on firms performance (Caputo et al., 2019).

Depending on the purpose of the process, text mining algorithms follow different steps (Fareri et al., 2020; Hippner & Rentzmann, 2006). The now presented steps are common for the classification of text using text mining. First, preprocessing is the premier step for all purposes (Fareri et al., 2020). Here, the available texts written in natural language are processed into their most basic forms. For that, all unnecessary tags such as html tags or navigation bars are removed. Words are then handled as token that can be further processed and better interpreted (Fareri et al., 2020). Also, all punctuation marks are removed and every capital letter is modified into a lowercase letter during the tokenization process (Rajman & Vesely, 2004). It is then common to remove so called stop words (Hippner & Rentzmann, 2006; Kannan & Gurusamy, 2014). These are words that do little to benefit the understanding of the context (Kannan & Gurusamy, 2014). Since stop words like *I, and*, or *in* are usually found very frequently (Kannan & Gurusamy, 2014), the remaining tokens are now a lot less in number. The last step of preprocessing consists of lemmatization (Heyer et al., 2006; Kannan & Gurusamy, 2014). During this step, every word is brought into its very basic grammatical form (Fareri et al., 2020). For instance, the words *troubling, troubled* and *troubles* all can be found as *trouble* after lemmatization (Kannan & Gurusamy, 2014). Therefore, when using word count methods,

the word *trouble* now has a higher count than the unlemmatized word. Second, the next text mining step is the depiction of the text corpus (Hippner & Rentzmann, 2006). Literature often refers to vector models for this purpose (Rajman & Vesely, 2004). Suppose a training corpus has the vocabulary size n, the document vectors also have the dimensionality n, where the element wise values represent the occurrence of n different tokens within the text (Hippner & Rentzmann, 2006). The element wise values of the vector can either be of binary order (1 for existing, 0 for not existing) or have some other, weighted value (Hippner & Rentzmann, 2006). For example, word count could be a weight for individual words. In a three-dimensional vector space, a sentence consisting of the tokens *stock, rise* and *rise* again (originating from the sentence *The stocks rise and rise*) could have the form of the vector $v = (1,1,0)$ when dealing with binary representations, while word count methods could change the vector to $v = (1,2,0)$. The values represented in the vector stand for different word token of the corpus vocabulary, meaning that there is a word in the vocabulary other than *stock* and *rise*, for instance *fall*. Third, the last step of text mining is the original purpose of the process itself (Hippner & Rentzmann, 2006).

Depending on what one wants to achieve with text mining, one can use vector representations of texts to automatically classify the texts into predefined categories (Garten et al., 2018), group them into clusters in terms of similar contents (Hippner & Rentzmann, 2006; Rajman & Vesely, 2004) or, for instance, analyze the frequency of words in a given timeframe for trend analysis. The latter most recently has been applied on posts on Twitter for trend analysis during the COVID-19 pandemic (Rajput et al., 2020). As this thesis aims at analyzing skills from existing textual data and later classify them into skill categories (e.g. *future skills*), further explanations will focus on the classification process in text mining.

Before one can classify a given text into a category effectively, a classification model has to be trained. Otherwise, the used model does not know what characteristics should be used for classification. On the one hand, training of a model can be done with labeled data, split into training and testing datasets (Caruana & Niculescu-Mizil, 2006; X. Zhang et al., 2015). As the names suggest, training data is used for model tuning and testing data is used for testing the validity of the model (Caruana & Niculescu-Mizil, 2006; X. Zhang et al., 2015). This kind of training is referred to as supervised learning, as the correctly classified categories of the training and testing datasets are available (Caruana & Niculescu-Mizil, 2006; X. Zhang et al., 2015). On the other hand, unsupervised learning deals with unlabeled data. Here, the goal is to find similarities, regularities or other rules within the data to, for instance, form clusters (Caron et al., 2018). In recent years, neural networks, as classification models, gained significant popularity due to their surprisingly good results (Bojanowski et al., 2017; Caruana & Niculescu-Mizil, 2006; Collobert & Weston, 2008). Even though neural networks are not necessarily needed for the classification process introduced later, they are at the center of vector representation

models like Word2Vec (Mikolov, Chen, et al., 2013) or FastText (Bojanowski et al., 2017). These work with unsupervised neural networks for the creation of vector representations and are used in the proposed classification method of this thesis. Therefore, the following introduction into neural networks presents how they work at the example of a simple classification problem.

2.4 Neural networks

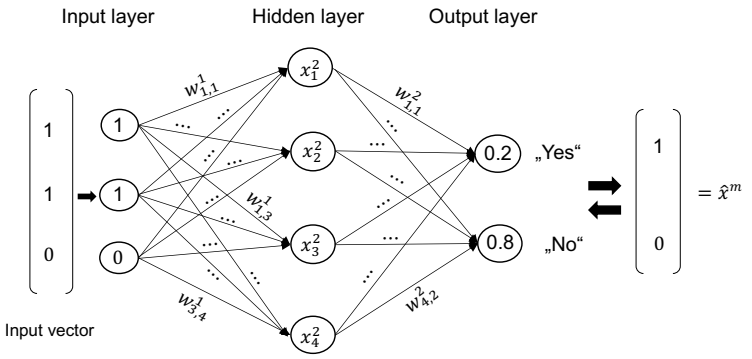

Figure 1: Simplified neural network architecture with one hidden layer. Numbers taken from example above.

The name and functionality of artificial neural networks have their origin at real neural networks. The human brain contains around 100 billion ($\sim 10^{11}$) neurons (Herculano-Houzel, 2009) that spread information in the form of electrical impulses to other neurons. These are connected by synapses that transport information between neurons and form a large natural neural network (Herculano-Houzel, 2009). As displayed in figure 1, the mathematical modeling of neural networks is as follows: The network consists of three types of layers, namely the input layer, hidden layer and output layer. In the input layer, a vector of dimensionality n is split into n individual neurons. For instance, this can be the three-dimensional vector described in the previous subsection. The values of the different neurons are better defined as x_i^j, with $i \in \{1, \dots, n_j\}$ denoting the i-th neuron in the j-th layer. The values of the neurons of the next layer are calculated as the sum over all neuron values of the previous layer multiplied with the weights $w_{k,i}^{j-1}$ associated with the links between the neurons k from layer $j-1$ and i from layer j. This value then is put into an activation function ϕ, which has the ability to map a real number onto another real number in a fixed interval. Therefore, the i-th neuron of the j-th layer has the value $x_i^j = \phi(s_i^j)$, with $s_i^j = \sum_{k=1}^{n_j} x_k^{j-1} w_{k,i}^{j-1}$. One can choose between different activation functions ϕ, but an often-used function is the sigmoid function $\phi(s) \in (0,1)$, $\phi(s) = \frac{1}{1+e^{-s}}$, which norms any real number on the interval (0,1). One can have one or multiple hidden layers, depending on the complexity of the classification problem (Huang, 2003). After the calculation of the last hidden layer $m-1$, the output layer m can be calculated. For that, the resulting layer is put into a softmax function $\sigma(a)_j = \frac{\exp(a_j)}{\sum_{k=1}^{V} \exp(a_k)}$, where a is a vector and V and j denote the

vocabulary size and the unit of the resulting vector respectively, which norms the units in the resulting vector so that the sum of them is 1. This step is done for interpretability reasons, as the result can be seen as a probability. Thus, the neuron (category) with the highest probability is chosen for classification. Say the output layer has dimensionality $n_m = 2$, describing the answers *yes* in the first neuron and *no* in the second neuron of a binary decision (e.g. *Does the stock rise?*), the value of the neuron with the correct answer should be somewhat close to 1 and the other one close to 0 (Rong, 2014).

Now, the network begins with randomly assigned weights and therefore the initial values of the output vector are not significant in meaning. Without training, the classification based on the neural network has no value. The parameter that predominantly influence the result are the weights as multiplicators for the neurons' values (Rong, 2014). For parameter tuning in a supervised model, the model is fed with training data. In this example case, we know that the correct answer to the question *Does the stock rise?* is *yes* when dealing with the sentence *The stocks rise and rise*, so the desired output could be described as the vector $\hat{x}^m = (1,0)$. Thus, if the resulting vector of the first epoch is $x^m = (0.2, 0.8)$, meaning the category *no* would be chosen as the probability is higher, the weights have to be modified in a way that in the next epoch, the first value would be higher than 0.2 und the second value would be lower than 0.8. These adjustments of values can be done by a process called backpropagation (Collobert & Weston, 2008; Mikolov et al., 2010; Mikolov, Chen, et al., 2013; Rong, 2014; X. Zhang et al., 2015). The goal of backpropagation is to modify the weights beginning from the last weights, meaning the ones between the layer m and $m-1$ and then iteratively continue to the beginning, meaning the input layer (Rong, 2014). First, the cost function $c = \frac{1}{2}\sum_{k=1}^{n_m}(x_k^m - \hat{x}_k^m)^2$ is used to determine the deviation of the output vector and desired output vector by squaring the elementwise difference. The factor of $\frac{1}{2}$ is useful for the minimization process that follows. Hence, the resulting cost of this simplified example is $c = \frac{1}{2}(0.2-1)^2 + \frac{1}{2}(0.8-0)^2 = 0.64$. Then the goal is to minimize the cost by determining the change of cost when changing the weights $\frac{\partial C}{\partial w_{k,i}^{m-1}}$. Note that because $x_k^m = \phi(s_k^m) = \phi(\sum_{k=1}^{n_j} x_k^{m-1} w_{k,i}^{m-1})$, the factor $\frac{1}{2}$ vanishes after the first derivative is calculated. This process can be repeated until a predefined quality, measured by values like precision, recall or accuracy, is reached. In the following, two algorithms that create word embeddings with the help of neural networks are introduced.

2.4.1 Word2Vec

Word embeddings created by Word2Vec enjoy high popularity in today's literature (Choi & Lee, 2020; Garten et al., 2018; Grave et al., 2018). Embeddings are numerical vector representations of textual content that can be used for further handling by computers. Mikolov, Chen, et

al. (2013) introduced Word2Vec as an algorithm that creates vectors based on words' semantic meaning. Following the Distributional Hypothesis (Harris, 1954), words that often appear close to each other in the learning corpus should have similar vector representations. Word2Vec is a semi-supervised neural network, e.g. it has some information about the surrounding words of a target word, with one hidden layer that takes a text corpus as its input and creates a list of vectors corresponding to the words of the text corpus as its output. In the following, methods used by the algorithm for vector creation are presented.

2.4.1.1 Methods

The methods used by the Word2Vec algorithm are the famously used continuous bag of words (CBOW) model and the skip-gram (SG) model. With CBOW, the training objective is to predict the target word by looking at a predefined number of context words around the target word. Contrasting CBOW, SG unintuitively tries to predict surrounding context words given the target word.

In the following a few variables are used that are introduced as follows:

- Vocabulary size V
- Input vector $x, dim(x) = V, x_i$ denoting the i-th value of vector x
- Hidden layer vector $h, dim(h) = N$
- Output vector $y, dim(y) = V$
- Weights matrix W from the input layer to hidden layer, being a $V \times N$ matrix (containing the N dimensional weight vectors associated with word i in row i for every word in vocabulary V).
- Weights matrix $\widehat{W}$ from the hidden layer to output layer, being a $N \times V$ matrix

2.4.1.1.1 CBOW

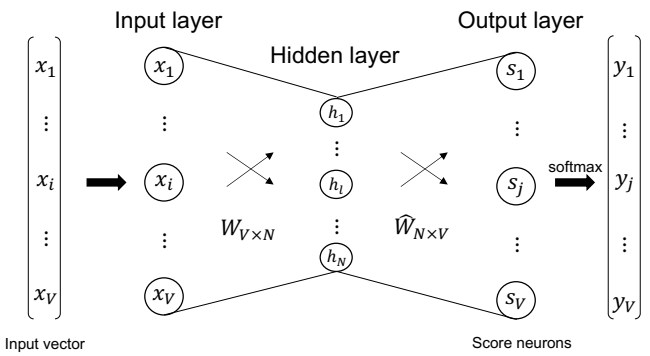

Figure 2: CBOW model with only one input word, represented by the one-hot input vector.

The CBOW model disregards grammar and word order by viewing a predefined context window around the target word as a list, e.g. bag, of randomly shuffled words (Mikolov, Chen, et al., 2013; Rong, 2014). For better understanding of the CBOW model, it is easiest to start with a one-word-context (Rong, 2014). Thus, the model will predict the target word given only one context word. The input vector x used in the algorithm is a one-hot vector, meaning it has the value 0 in every row except the one associated with the context word w_c,

where it is 1. Suppose this row is the i-th row of x. Now, the hidden layer vector is calculated by multiplying the weight matrix so that $h = W^{\mathrm{T}}x$ essentially is a copy of the i-th row of matrix W (Rong, 2014). h can now be seen as the N dimensional vector representation of the input word. Finally, the vector produced in the output layer is calculated in two steps. First, multiplying $\widehat{W}$ with h so that $s = \widehat{W}^{\mathrm{T}}h$ creates a score vector with units corresponding to each word in vocabulary V. Then, Mikolov, Chen, et al. (2013) propose to then use softmax, e.g. a log-linear classification model, to obtain the words' distribution y, hence $y_i = p(w_i|w_c) = \frac{\exp(s_i)}{\sum_{k=1}^{V}\exp(s_k)}$ is the i-th unit of the output vector (Rong, 2014). As explained in subsection (2.4), the objective of the following model training is to adjust the weights, e.g. minimize the cost function $C = -\log\big(p(w_t|w_c)\big)$, where $p(w_t|w_c) = y_t$ is the probability to find the actual target word w_t (Rong, 2014). The logarithmic function comes into play because of the minimization problem regarding C. It is a strictly increasing function and therefore does not change the location of the initial function's extrema when applied on this function, because the derivative is strictly positive (especially never 0). Also, the logarithmic function has practical mathematical implications when dealing with exponential functions and products.

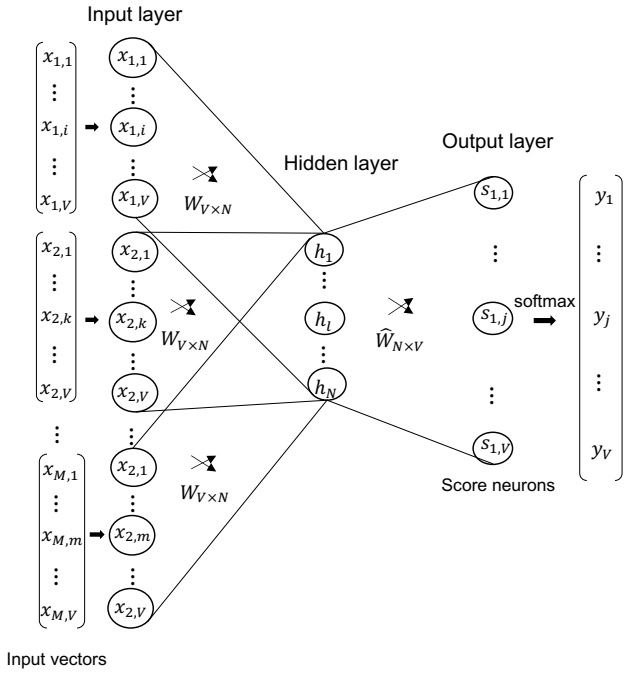

Figure 3: CBOW model with M context words as input.

For instance, in this example $\log\big(p(w_i|w_c)\big) = \log\left(\frac{\exp(s_i)}{\sum_{k=1}^{V}\exp(s_k)}\right) = s_i - \log\big(\sum_{k=1}^{V}\exp(s_k)\big) = -\mathrm{C}$. For a multi word context with M words, CBOW computes h as the averaged sum over the product of W and one-hot vectors of context words so that $h = \frac{1}{M}W^{T}\big(x^{(1)} + \cdots + x^{(M)}\big)$. Note that now, the cost function changes to $C = -\log\left(p\left(w_t\middle|w_c^{(1)}, \ldots, w_c^{(M)}\right)\right)$. The rest of the computation remains the same (Rong, 2014).

2.4.1.1.2 Skip-gram

The SG model was introduced by Mikolov, Chen, et al. (2013). It is often referred to as the opposite of the CBOW model as its architecture is reversed (Rong, 2014). Here, the input (target) word is used to find the surrounding context words. The matrices multiplied have the same characteristics as in the CBOW model. Therefore, h is the same as in the one-word-context of CBOW pictured in figure 2 (Rong, 2014). The difference lays in the output layer,

which now consists of M instead of one V dimensional vectors, calculated by the multiplication of h with $\widehat{W}$. This operation yields $y_{m,i} = p(w_{m,i} = w_{c,m}|w_t) = \frac{\exp(s_{m,i})}{\sum_{k=1}^{V}\exp(s_k)}$ as the i-th unit of the m-th (context) output vector. Note that here the role of context and target words are changed: The input vector corresponds to the target word and the output vectors correspond to the M context words. The idea behind the remaining calculations is the same (Rong, 2014).

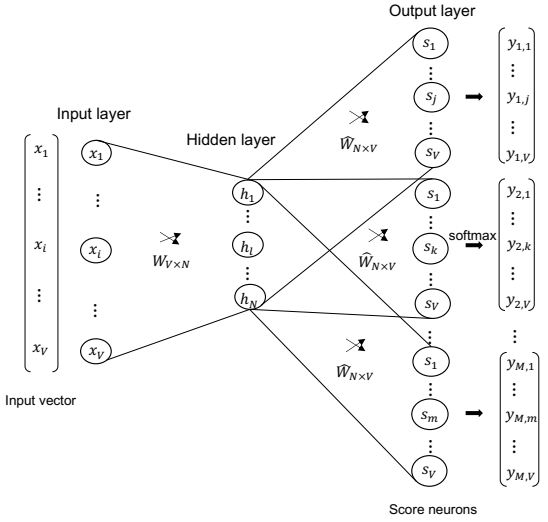

Figure 4: CB model.

2.4.1.2 Possibilities

With the help of distributed representations of words, Mikolov, Chen, et al. (2013) created an algorithm that is able to capture surprising semantic similarities (Mikolov, Chen, et al., 2013; Mikolov, Yih, et al., 2013; Rong, 2014). As vectors are mathematical constructs, one can use mathematical operations to find dependencies. Most famously the finding, that the vector of the word *king* minus the vector of the word *man* plus the vector of the word *woman* is closest to the vector of the word *queen* (Mikolov, Yih, et al., 2013), resulted in wide interest of the Word2Vec algorithm. Indeed, interconnections such as “A is to B as C is to __” can be found with 40% accuracy, proving that vector representations outperform any of the other previously tested systems (Mikolov, Yih, et al., 2013). This includes not only semantic analogies like “Berlin is to Germany as Paris is to France” or “clothing is to shirt as dish is to bowl” but also grammatical properties like “car is to cars as apple is to apples” or “family is to families as car is to cars”. Mathematically spoken, if x_i is the vector representation of word i, then $x_{apple} - x_{apples} \approx x_{familiy} - x_{families} \approx x_{car} - x_{cars}$ (Mikolov, Yih, et al., 2013). Through this relation, one can see how Mikolov, Yih, et al. (2013) arrived at the above mention relation with *king* and *queen*: $x_{king} - x_{man} \approx x_{queen} - x_{woman} \Leftrightarrow x_{king} - x_{man} + x_{woman} \approx x_{queen}$. Notice that these are no equations as the calculations result in multidimensional points which do not necessarily

equal the vector representations of the searched words. The algorithm however choses the most similar words as the output of such requests. In the past, word embeddings like those created by Word2Vec have been successfully used for several natural language processing applications. First, Part-of-Speech tagging used for identifying grammatical properties has seen improvements adding vector representations to traditional methods by achieving more than 97% accuracy (Collobert et al., 2011). Clearly, Word2Vec alone is not considered to outperform Part-of-Speech tagging models as it disregards word order in both of the provided models CBOW and SG (Ling et al., 2015). However, due to its simplicity it is still a convenient choice for distributed representations (Ling et al., 2015). Second, Word2Vec has been investigated on how it can complement traditional methods for text classification and is found to effectively outperform methods if used in combination with other methods by considering semantic similarities (Lilleberg et al., 2015). Finally, Garten et al. (2018) introduced distributed dictionary representations based on Word2Vec and found state-of-the-art results in sentiment analysis and morality detection in twitter posts, pushing the study of small dictionaries with few core words for text classification applied on small pieces of text forward.

2.4.1.3 Problems

The text corpus on which Mikolov, Chen, et al. (2013) trained their model contains around 320 million words, resulting in a vocabulary of 82 thousand words (Mikolov, Chen, et al., 2013). Literature often uses the Wikipedia corpus consisting of articles of the desired language (Garten et al., 2018) or Google News articles of similar word occurrences (Garten et al., 2018; Mikolov, Chen, et al., 2013) for model tuning. As semantic similarities rely on the Distributed Hypothesis, e.g. words' meaning can be found in the words surrounding them (Harris, 1954), models have to be specifically trained for each individual application because of the topic relation. For instance, in a technical environment the vector of the word *python* should be significantly closer to that of the word *java* than that of the word *snake.* Unfortunately, this difficulty is hard to overcome due to missing training data, considering that Mikolov, Chen, et al. (2018) propose training on datasets with multiple hundreds of millions of words. Even though this is a lot, the algorithm struggles to deal with new, e.g. unknown words. In other words, new or unknown words cannot be associated with a vector and therefore have to be eliminated from the to be classified text (Garten et al., 2018). Bojanowski et al. (2017) found a way around this by introducing an algorithm for word vectors with subword information, called FastText (Bojanowski et al., 2017).

2.4.2 FastText

FastText (Bojanowski et al., 2017) eliminates the difficulty to deal with new or generally unknown words by introducing word vectors with subword information. The algorithm that

produces these vector representations fundamentally equals that of Word2Vec. As with Word2Vec, the embeddings created by FastText can be produced through SG, with the difference, that FastText creates embeddings for character n-grams and sums them up to arrive at a word vector (Bojanowski et al., 2017). For instance, in a 3-gram model, the word *house* would have the vector representation $x_{house} = x_{<ho} + x_{hou} + x_{ous} + x_{use} + x_{se>}$, with "<" and ">" denoting pre- and suffixes.

With that change, unknown and infrequent words now have a vector representation as the n-grams contained in such words are most likely to exist in other words as well (Bojanowski et al., 2017; Choi & Lee, 2020; Grave et al., 2018). In other words, it is highly unlikely that a character n-gram is unknown. Bojanowski et al. (2017) also found that FastText, compared to Word2Vec's CBOW model, is better suited when training with little data. This, in fact, enables the implementation of word embedding models for smaller fields of study. Furthermore, misspelled words, which are often found in social media posts, are no longer treated as unknown words. Twisted characters for example only partially change the vector representation of a word, as it does not affect every n-gram of the word. Grave et al. (2018) underpinned the efficiency of these embeddings by creating embeddings for 157 languages. Though the results show the vectors to be of high quality, datasets with more training data yielded the best results (Grave et al., 2018). In the following, dictionary-based approaches for text classification are described before an application of such combined with Word2Vec word embeddings is presented.

2.5 Dictionary-based approach

Dictionary-based approaches for information retrieval are commonly used concepts (Hjørland, 2016; McMath et al., 1989). In dictionaries (often called concept-dictionaries), one can find a number of categories, matched with their unique description. When looking at a text, paragraph, or even a single word (or symbol), one can classify it as belonging to a certain category by comparing and counting the words contained in the concept descriptions and the to be classified text (Park & Kim, 2016). With respect to creating a dictionary, it is very important to find the correct concept descriptions. Only small changes can drastically impact the performance of a dictionary (Park & Kim, 2016). For example, if too few words are contained in a dictionary, it gets very hard to classify especially short text corpora like social media posts. When no matching words can be found, no classification can be accomplished. A walk-around for this can be found with so-called seed words. Seed words are words that form the base of a concept in a dictionary. Concept descriptions in a dictionary get build around those seed words by collecting synonyms and antonyms from existing (online) dictionaries. As one can imagine, the selection of the correct seed words plays a crucial role (Park & Kim, 2016).

2.5.1 Existing dictionaries

In the past, researchers used dictionaries to identify and classify things in many different areas of study, e.g. machine translation (Tripathi & Sarkhel, 2010), protein identification (Egorov et al., 2004) and hate speech detection (Gitari et al., 2015). One very extensively studied field in that dictionary-based approaches show promising results is sentiment analysis (Hardeniya & Borikar, 2016). Here, the goal is to extract expressive information or people's opinion from text, social media posts, reviews or other text-based sources (Park & Kim, 2016). As mentioned before, concept descriptions, the dictionaries' cores, get build around seed words by collecting synonyms and antonyms from existing (online) dictionaries. There are a few commonly used, online available and free dictionaries especially for sentiment analysis, most famously WordNet (Miller et al., 1990) and SentiWordNet (Esuli & Sebastiani, 2006). Both are publicly available online dictionaries that held information about 54.000 lexical entries in their first released versions and get updated regularly. While WordNet is a collection of English nouns, verbs and adjectives organized into synonym sets (synsets) (Miller et al., 1990), SentiWordNet tags each synset of WordNet with three numerical values ($Obj(w), Pos(w)$ and $Neg(w)$) between 0.0 and 1.0 (that sum up to 1.0), describing how objective, positive or negative the terms w in the synset are (Esuli & Sebastiani, 2006). Some works went even further and built dictionaries to cover the semantic relation between word types like adjectives or verbs only (Hatzivassiloglou & McKeown, 1997; Taboada et al., 2006). Of course, those existing dictionaries can only be used for specific use cases. Other existing dictionaries were created to classify the grammatical property of a word or sentence (e.g. give them a part-of-speech (POS) tag). Existing dictionaries based on word count methods are suited better whenever the object of inquiry is a closed set of words (Garten et al., 2018). Linguistic categories, for example pronouns, articles and conjunctions, are considered to be composed of a relatively fixed set of terms (Garten et al., 2018). The study of these classes has proven to be very insightful in terms of linguistic and psychological research (Pennebaker, 2011). On the opposite side, open class terms make things more complicated.

2.5.2 Self-written dictionaries and methodology

Whenever one deals with a relatively closed set of words, creating a dictionary is a good choice (suppose there is no suitable publicly available dictionary yet) (Park & Kim, 2016). Dictionaries can help to create an overview about the data one has to classify in a very understandable and reproducible way. By comparing the content of the data and the dictionary, the classification process can be monitored at all times. Because dictionaries have a high topic relation, one cannot simply take any concept dictionary and expect promising results in terms of classification (Were et al., 2007). This chapter identifies key points of the creation of a suitable dictionary.

Park and Kim (2016) propose an approach to classify sentences in terms of sentiment by word matching (Park & Kim, 2016). As this method could be used for the creation of dictionaries for other use cases and, furthermore, is easy to understand, this paragraph is meant to lay out the basics of one way to create a concept dictionary. The classification method used in this case requires a well-developed dictionary, because it directly affects the performance of the classifier. The clearer the seed words used to build the dictionaries are, the better the classifier is (Park & Kim, 2016).

Park and Kim (2016) propose a seed word selection based on the word's frequency in the so-called sentiment lexicon. This is a set of words that have either positive or negative sentiment and can be obtained through training with labeled data. For that, each word is classified with an existing POS tagging model (Manning, 2011) and counted, to find the frequency of a word. Thus, the same word can have different POS-tags and therefore is considered a different feature. For example, in the sentences *I like to eat ice cream* and *It looks like rain* the word *like* has a different grammatical position (different POS tag) and consequently a different overall word frequency. After tagging, the seed words are selected by $\mathrm{Difference}$ and $\mathrm{Polarity}$, where

$$Difference = |Frequency\ of\ Positive - Frequency\ of\ Negative| \text{ , and}$$

$$Polarity = \frac{Difference}{Frequency\ of\ Positive + Frequency\ of\ Negative}.$$

That being said, seed words have a strong sentiment (high polarity) and a large gap between the frequency of positive and the frequency of negative tagged words (high difference) (Park & Kim, 2016). In other words, seed words have an unambiguous nature. Translated into a skill dictionary, the seed words $\{s_1, s_2, \dots, s_n\}$ could take the position of the n concept (skill) names. These should also have a definite meaning. Each concept can then be described as a set of synonyms or closely related terms $\{t_1, t_2, \dots, t_m\}$, so that the $i-th$ concept consists of the words $\{t_{1,i}, t_{2,i}, \dots, t_{m,i}\}$, $i \in \{1, \dots, n\}$ and $m \in \mathbb{N}$ but small. After collecting words and terms to build a concept description of satisfactory length m, one can run small classification tests to examine the performance. This process can be repeated multiple times until the results are on the required level.

2.5.3 Weaknesses

Until today, there is no suitable dictionary for the classification of skills from freely written text on social media platforms. One possibility to explain this circumstance is the difficulty to extract information from short texts like social media posts. For instance, texts on Twitter average at 34 character per post (Ihara, 2017), have a maximum length of 140 (Kwak et al., 2010) and hold misspellings or abbreviations, not always follow grammatical conventions and use a range of descriptions for the same thing. Another possibility is that dictionaries often capture a wide range of descriptive words because without them, something containing words that are not

part of the dictionary could not be classified. Unfortunately, it is nearly impossible for a person (e.g. dictionary creator) to be familiar with all possible synonyms of a word (Louwerse, 2004). Thus, there will always be words that cannot be captured. This is especially crucial for small dictionaries that are used to classify into a small number of categories. Following, not only the absence of good existing dictionaries for this use case, but also the disability to use the same dictionary in different fields of study effectively and the difficulties with short text passages mentioned above bring the need for another method of creation and usability of dictionaries forward. Garten et al. (2018) propose a promising new approach to build dictionaries suited for the classification of short text corpora (Garten et al., 2018). The biggest difference is the usage of semantic similarity instead of wordcount methods by introducing Distributed Dictionary Representations (DDR).

2.6 Distributed Dictionary Representations

Garten et al. (2018) argue that word count methods can safely be used whenever one deals with closed-class terms but struggle to achieve good results in classifying open-class terms. They examined a novel approach to create a dictionary by considering the semantic similarity of words rather than the morphological one. Therefore, one does not encounter difficulties like capturing morphological similarities that do not make much semantic sense, for instance *adore, adoration* and *adornment* when capturing the pattern *ador**, but include closely related terms with different word stems, which one would expect when looking at the words *father, mother* and *son*. DDR uses vector representations of words to not only build distributed dictionaries, but also uses them to classify terms into the concept categories contained in the dictionary. This enables the applicability of dictionaries for short texts like social media posts. They no longer have to be aggregated into big documents for classification and can be compared on a semantic level of similarity with the concepts, that themselves have a defined vector representation (Garten et al., 2018). The big advantage that follows this method is that only the core of a concept rather than every somehow associated word has to be found for the purpose of classification. Hence, the creation of a dictionary can be focused on the most important descriptive words of a concept (Garten et al., 2018).

DDR consists of four main steps, which are as follows: First, a list of words characteristic of a category is being created. This list can be obtained by finding synonyms or closely related terms describing a concept (e.g. finding words with high semantic similarity and adding them to the list) or by choosing words from existing dictionaries. Second, they use pre-trained distributed representations of words trained with the Word2Vec algorithm (Mikolov et al., 2010) to create concept representations by averaging the word vectors. Formally, for every word $w \in \{w_1, \dots, w_n\}$ in the dictionary D, the distributed representation can be displayed as a m-dimensional vector $R(w) = [d_1, \dots, d_m]$ where $d_i \in \mathbb{R}, i \in \{1, \dots, m\}$. Since Word2Vec works

with training through whole words only, one first has to take the intersection of the words in the dictionary D and the vocabulary V of the training corpus $D_I = D \cap V$ to be able to create such representations. Third, the average of the vector representations yields into the concept representation $C_R = \frac{\sum_{\forall w \in D_I} R(w)}{||\sum_{\forall w \in D_I} R(w)||}$. This, as well, is a vector of dimensionality m. Finally, the similarity of the to be classified text, e.g. movie reviews or social media posts, can be computed by calculating the cosine similarity between the concept vector C_R and the text vector T, which is obtained in the same manner as the concept vector C_R. The cosine similarity is a continuous measurement ranging between -1 and 1, and is defined as the scalar product of two vectors of same dimensionality, divided by its absolute value: $cosinesimilarity(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1 \cdot v_2|}$. Therefore, when two vectors show into the same direction, their standardized value equals 1 (they are perfectly matching). On the opposite side, two vectors that are showing in the most opposite direction, share the product value of -1. One can compare this with the words *positive* and *negative* or *yes* and *no*. They could both be the answer for the same question but could not be any more different. Additionally, two vectors that form a 90-degree angle have the cosine similarity of 0. These vectors have nothing in common. Any vector having the slightest contribution into the same direction of the other vector, has a cosine similarity greater than 0. Garten et al. have proven distributed dictionaries to be a powerful tool for dictionary creation especially for short text classification and small sized dictionaries (Garten et al., 2018). They found the perfect dictionary size to be 30 for a concept, but left room for deviations in different kinds of study. DDR combines the theory driven structure of dictionaries with the strength of distributed representations. Therefore, a continuous measure instead of word count, the power of modern technology through high dimensional neural machine learning and the applicability of short texts (like social media posts that often exist of very few or single words) raise attention for future work. The following section presents a novel method with consideration of the discussed options for information retrieval and word embeddings.

3 Methodology

Predominantly inspired by Garten et al. (2018), this thesis presents a method that uses a distributed dictionary with vector representations created by the FastText algorithm for skill identification from social media profiles. In fact, with the help of concept vectors, this model classifies certain characteristics, competences or expertises (*haves*) represented by short text components from social media profiles into predefined *future skill* concepts. It does so by calculating the cosine similarity from the vector of a person's description to given concept vectors and ascribing it to be included into that given future skill category when the similarity is above a certain value. The benefit of this model to that of Garten et al (2018) is that it uses the FastText algorithm as it has a better management of rare and misspelled words. It further analyzes the

claim of Garten et al. (2018) that their method of dictionary-based text classification can be applied to short texts like social media posts and with small and open-class dictionaries. As explained in subsection 2.1.3, the dictionary used in this model is derived by other future skills frameworks (Ahonen & Kinnunen, 2015; Davies et al., 2011; Kirchherr et al., 2018; OECD, 2018; Partnership for 21st Century learning, 2015) and completed with online databases (ESCO, 2020; OnetOnline, 2020) that hold descriptions of jobs with their corresponding skills, abilities and knowledge needed for this job.

The concepts then get finalized by examination of closest neighbors of the concept vectors.

This method suggests different figures for the evaluation of companies' readiness for the future that all can be used for direct comparison with competitors or sector wide averages. For vector creation, a pretrained FastText model trained on the English Wikipedia text corpus with 300-dimensional vector representations (Grave et al., 2018) is used.

In the following, the process of dictionary creation is fully described. Furthermore, the proposed method is explained in more mathematical detail. At the end of this section, three depictions of the framework are offered.

3.1 Finding descriptions of included skills

This subsection aims at addressing the creational process of the future skills dictionary. Initially, skill categories are extracted from existing future skills frameworks (Ahonen & Kinnunen, 2015; Davies et al., 2011; Kirchherr et al., 2018; OECD, 2018; Partnership for 21st Century learning, 2015). In summary, four skill categories have been found. First, *social, people and emotional skills* include skills and competencies that stand for a high affinity towards collaborative activity. Second, *cognitive- and metacognitive skills* are considered as they are important for future employee orientation (Kavoo-Linge & Kiruri, 2013). Third, *digital base skills* cover digital skills that are needed for future every day professional and private life. Fourth, expert digital skills include competencies dealing with transformative technologies, as they shape our future (Kirchherr et al., 2018; Smit et al., 2020). Before creating concept vectors, the description words for each of the concept have to be found. For that, seed words for each category are chosen by summarizing and comparing existing future skills frameworks. The number of seed words is not fixed, but oriented at the extend of category appearances in other frameworks. What concerns the total number of description words, Garten et al. (2018) propose a range of about 30 words per category, which can vary depending on the topic. The number of words contained in this models' categories is determined in the course of the model optimization process explained later. After seed word collection, this method follows a procedure used by Fareri et al. (2020) by incorporating closely related skills, competencies and job information found in online available databases.

Table 8: Future skill categories and therein included description words. The superscripts 1, 2, 3 and 4 denote whether a term or word is a seed word, included after the online dictionary phase, included as a closest neighbor and/or is part of the final word list of a category respectively.

Future skills category	Included skills			
Social, people and emotional skills	social[1]	cooperation[1]	intercultural[2,4]	accountability[2]
	culture[1]	customer[3,4]	enthusiasm[2,4]	communication[2,4]
	listening[2,4]	people[1,4]	acceptance[2]	thoughtfulness[3,4]
	clarity[2]	empathic[2]	coordination[2,4]	sociocultural[3]
	advice[3]	employee[2]	personnel[3,4]	collaboration[2,4]
	empathy[1]	teamwork[3,4]	admiration[2,4]	responsibility[3,4]
	ethical[2,4]	speaker[2]	recruiting[3,4]	
	team[1,4]	emotion[2]	participation[2]	
Cognitive- and metacognitive skills	creativity[1]	curiosity[2,4]	adaptability[1,4]	analytical thinking[2]
	innovative[2]	mutability[3,4]	engagement[2,4]	phenomenological[3]
	endurance[2]	judgment[3]	commitment[3,4]	time management[2]
	solving[3,4]	flexibility[2,4]	accountability[2,4]	innovative thinking1
	sociability[3]	analytical[3]	critical thinking[1]	problem solving[1]
	passion[3,4]	strategic[2]	sophisticated[3,4]	decision making[2,4]
	awareness[2]	intellectual[2,4]	responsibility[2,4]	
Digital base skills	excel[2,4]	skype[2,4]	digital learning[1]	microsoft office[2,4]
	software[2,4]	electronic[3]	web conference[2]	virtual collaboration[1,4]
	digital[2,4]	internet[3,4]	mathematics[2,4]	digital interaction[1]
	information technology[1]	system[3]	digital competence[1]	communication technology[1]
	computer[1,4]	technology[2,4]	international[3]	virtual[2]
Expert digital skills	blockchain[1,4]	data base[2]	data analysis[2]	business intelligence[2,4]
	syntax[3,4]	database[3]	data mining[2]	artificial intelligence[2]
	algorithm[2]	automation[2]	data science[2]	web engineering[1]
	data[2]	visualization[2]	programming[3,4]	mathematics[2]
	robotic[1]	analytics[2]	industry 4.0[2,4]	software programming[1]
	code[2]	text mining[2]	machine learning[1,4]	

In this case, European Skill/Competencies Qualification and Occupation (ESCO, 2020), which classifies jobs, competencies and qualifications in Europe and O*net (OnetOnline, 2020), developed for the US department of Labor, are the available databases used for leveraging the concept descriptions. For instance, when analyzing the seed word *teamwork* from category *social, people and emotional skills*, one encounters related terms like *interpersonal* or *communication*.

After the set of words is a reasonably closed set of competencies, e.g. it describes the skill category to a satisfactory level, temporary distributed concept representations are created. This is done by averaging the sum of every words' vector contained in category i. Mathematically, the temporary vector is computed as $c^{(t)} = \frac{1}{|C|}\sum_{k=1}^{|C|} x_k$, where the capital $C = \{X_1, \dots, X_{|C|}\}$ is the set of description words and $vector(X_i) = x_i$ the vector representation of the i-th word in C. Note that this averaging method is different to the method of Garten et al. (2018) as it does not exclude out-of-vocabulary words. The next step makes use of a predefined function of the FastText model called model.*get_nearest_neighbors(*$c^{(t)}, k = n$*),* which outputs the n closest word embeddings and their corresponding words based on cosine similarity. Through this process of closest neighbor analyzation, some useful additions can be found. Following, the process of temporary vector creation and closest neighbor analyzation can be repeated until the desired number of words is reached. A complete overview of the resulted skill categories and therein included description words can be studied in table 8.

3.2 Application

Before the dictionary can be applied on the dataset, the present data has to be preprocessed. First, the data is split into a two-dimensional list, representing different users in each row and their respective *haves* in each column. Therefore, not every row has the same length, e.g. the length depends on the number of *haves* given by a user. Second, punctuation marks are removed as they are not part of the training vocabulary. The removed marks include ";", ":", "/", "(", ")", "&", "%", ",", "_" and "-". Third, every character is rewritten into lower case format. Following Garten et al. (2018), this method does not remove stop words as they can transport important psychological meanings, including social hierarchies (Kacewicz et al., 2014). Also, lemmatization is no part of this method because it does not rely on word count methods and rather builds semantic unities in vector space. Before the classification process begins, another two-dimensional list is created. Considering the model is applied on n different individuals (users), the second list has n rows and four columns (corresponding to four different skill categories). More precisely, the first, second, third and fourth unit $(u_{i,c1} - u_{i,c4})$ of every row $i \in \{1,2, \dots, n\}$ correspond to the category *social, people and emotional skills, cognitive- and*

metacognitive skills, digital base skills and *expert digital skills,* respectively. As it will later be used to save the extent to which a user has what skill, this list will be called *skList*, with $skList = \left[\left[u_{1,c1}, u_{1,c2}, u_{1,c3}, u_{1,c4}\right], \ldots, \left[u_{i,c1}, u_{i,c2}, u_{i,c3}, u_{i,c4}\right], \ldots, \left[u_{n,c1}, u_{n,c2}, u_{n,c3}, u_{n,c4}\right]\right]^T$, from now on. At the beginning, this list is empty, e.g. it contains only zeros as countable variables, so that $skList = \left[[0,0,0,0], \ldots, [0,0,0,0]\right]^T$. This implementation, and not one that simply counts the skill occurrences on company level, is chosen for the sake of comparability on user-level. For example, if a company lacks employees in a specific skill category, it can choose to analyze a single person's skills on whether or not he or she contributes to close that skill gap. Eventually, the analysis and classification of future skills can begin.

3.3 Distribution analysis

This subsection explains how the classification process works. First, four concept vectors describing each skill category in vector space are created and saved. These are later used to calculate the similarity of haves to future skill categories. For that, the sum of word vectors contained in the concept is divided by the number of terms in the concept, so that the concept vector is $c = \frac{1}{|C|}\sum_{k=1}^{|C|} x_k$. If a term in the concept consist of multiple words, for instance *business intelligence* or *machine learning,* again the average word vector is taken so that it only has a simple weight contributing to the overall concept vector. Second, in an iterative process, every to be classified text is attributed to a vector representation by the same averaging method. After vector creation, the model calculates the similarity in terms of the widely used cosine-similarity (Mikolov, Yih, et al., 2013) to each concept vector. If the similarity to any skill category is higher than the predefined threshold t, it gets attributed to that category by activating the counter variable in the earlier defined $skList$. Thus, a single skill can also be attributed to multiple skill categories, considering the threshold is surpassed in multiple categories, or to neither of them, considering the threshold is nowhere crossed. For clarification, when looking at the *have* "development of machine learning algorithm", one would likely classify it as an *expert digital skill*. Additionally, one could also infer that an individual having that competence has at least some *digital base skills*. The question remaining is how to set the threshold t. For that, the model goes through a supervised learning process. In this case, t is chosen through solving a maximization problem of similarity measure with regards to the F-measure of the model. The F-measure calculates as

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

with $precision = \frac{TP}{TP+FP}$ and $recall = \frac{TP}{TP+FN}$ (Lumauag, 2019; J. Zhang & Mani, 2003). TP denote the number of data points that are correctly classified into their specific category. FP and FN describe the number of wrongly into a category classified datapoints and the number of

wrongly not classified datapoints, respectively (Lumauag, 2019; J. Zhang & Mani, 2003). This measure is taken, as it is the harmonic mean of the fractions of $precision$ and $recall$, which represent the percentages of correctly classified datapoints from all classified ones (precision) and of the correctly classified datapoints from all datapoints truly belonging to the specific category (recall) (Lumauag, 2019; J. Zhang & Mani, 2003). As this model can classify data into four categories, each possible classification is taken as a new datapoint. The benefit of this measure is that it can be taken for unbalanced data (J. Zhang & Mani, 2003). Accuracy, which is the most intuitive goodness measure as it is defined as the number of correctly classified points divided by the total number of points, in turn is not useful when dealing with unbalanced data. For instance, when trying to identify spam mails and having a dataset with 100 mails, five of which are spam, a model classifying every mail as *no spam* would have an accuracy of 95%. However, no actual spam mail was found (J. Zhang & Mani, 2003). $precision$ and $recall$ are goodness measures that accommodate such classification problems. Of a random classification, considering a balanced dataset, one would expect a F-measure of 0.5. Consequently, a higher F-measure than 0.5 would stand for a better classification model than a random classification. Now, let C be the set of description words for a category that are found by the process described in subsection 3.1. During the process of F-measure optimization, the number of concept description words is iteratively increased from 1 until the maximum number of words in C is reached. In the i-th iteration, i distinct words from C are chosen at random, forming a temporary set of words. To ensure that different combinations are considered, this process is repeated multiple times. These temporary word sets are then used as temporary concept descriptions for the supervised learning process. Here, the F-measure is calculated for different thresholds t, ranging from 0 to 1 and increased in steps of 0.01. The highest F-measure and associated threshold t is saved until every iteration is done calculating. Finally, the temporary word set resulting in the highest F-measure is taken as the actual concept description word set.

The final step of the classification process is the depiction of the results for later interpretation. The following subsection addresses three methods that arrive at a similar scope of interpretation.

3.4 Depiction

This final step takes $skList$ as its input and results in vivid depictions. In the following, three results with different possibilities of interpretation are presented. The first version is based on a binary skill behavior, e.g. a user either has a skill or not. Thus, if a user describes his competence as *Microsoft Office Skills*, he or she should get a 1 in the third unit, e.g. the one belonging to *digital base skills*, of his or her respective $skList$ row. However, if the second description the same user gives is *Microsoft Excel Skills*, this user still has a 1 (and not a 2)

attributed to *digital base skills*. The original $skList$ containing counting variables can easily be adjusted by applying $\text{binary}(skillcount) = \begin{cases} 0, & skillcount = 0 \\ 1, & skillcount \geq 1 \end{cases}$ on every unit in $skList$ whilst copying them into a new list $skListBinary$. For a company skill view, the sum over all n rows in $skListBinary$ is taken to arrive at a four-units vector with units $u_{cj} \in \{0, \dots, n\}, j \in \{1,2,3,4\}$, corresponding to the skill categories. For the sake of comparability with other companies of different size, the vector gets divided by n, so that the final company skill vector of version one $s_1 = [u_{1,c1}, u_{1,c2}, u_{1,c3}, u_{1,c4}]$ has only normalized values in the interval [0, 1]. Consequently, the units $u_{1,ci}$ can be interpreted as percentages that represent the proportion of employees in a company that hold skills associated with category i.

The second version takes into consideration the number of haves an employee provides. For this method, let a be the number of provided *haves* by all employees of company A. The temporary company skill vector $s_2^{(t)}$ is created by taking the sum over all rows in $skList$. This time, the normalization for comparability reasons is done by dividing by a, so that

$$s_2 = \frac{s_2^{(t)}}{a} = [u_{2,c1}, u_{2,c2}, u_{2,c3}, u_{2,c4}].$$

Note that $s_1 \neq s_2$, as users might have more than one *have* associated with a skill category. Here, the units $u_{2,ci}$ represent a percentage value of a, e.g. $u_{2,ci} \cdot 100\%$ of all given *haves* can be associated with future skill category i.

The third version is a mixture of the above-mentioned ones. It seeks to represent the average skill distribution per employee in a company A. Let $a_i, i \in \{1, \dots, a\}$ be the number of *haves* given by employee i. Then the i-th row of $skList$ is divided by a_i for every $i \in \{1, \dots, n\}$ to yield the percentage of skills in each category per person. s_3 is now calculated by summing the rows of $skList$ and dividing by the number of employees n. The resulting third company skill vector $s_3 = [u_{3,c1}, u_{3,c2}, u_{3,c3}, u_{3,c4}]$ represents the average percentage $u_{3,ci}$ of *haves* an employee of company A has, that can be associated with future skill category i. Regardless of the version, a horizontal bar plot is used for the depiction of the results.

In the following, the pros and cons of the presented versions and the method as a whole are examined.

3.5 Pros and Cons

In direct comparison with DDR introduced by Garten et al. (2018), the main difference is the change of the vector algorithm from Word2Vec to FastText. By choosing FastText, unknown words can be handled better because of the subword information contained in each word (Bojanowski et al., 2017). This is especially true for misspelled words, which one can often find in social media posts and profiles. As these display this models' focus point of application, this is an important addition for the models' capabilities. Another big adjustment is the dictionary

creation process. Here, an optimization process yields the best suitable concept descriptions with regards to description words chosen from a larger pool of selected words. Indeed, this method will later prove to outperform dictionaries incorporating whole word lists created by the process introduced by Garten et al. (2018). Yet, on the skill assessing perspective, the biggest advantage of this model is the time saved during the collection and classification process. Employees do not have to be interviewed or tested for the data collection process as the information needed can be taken from their respective social media profiles. The assessing time can alternatively be put into other value-adding processes. This method also benefits data-topicality, because people seek to keep a most current status of themselves on social media platforms (Jackson, 2011). Another positive aspect is the available perspective on both individual employee-level and company-level for higher comparability company intern among the employees and outer company between the company itself and industry standards. For instance, company intern personnel planning for a new technology-based project can be accomplished through the model by asking what employee has above average *expert digital skills?* and comparison with industry standards could highlight areas in which a company is ahead of its competitors and can therefore take advantage of these skills. In the end, all values resulting from the proposed analysis represent an easily understandable statement.

On the other hand, this method requires access to the social media data of a companies' employees. Not all social media platforms are well suited for this model as it aims at extracting mostly skills in and for a professional context. For instance, Twitter or Facebook are heavily used in a private context (Jackson, 2011) whereas LinkedIn and Xing are social media platforms that users are part of to connect with working partners (LinkedIn, 2019; XING, 2020). Another negative aspect is that the company-level skill view is estimated by only a fraction of the overall workforce since not necessarily every employee is part of the same platform. Hence, the resulting skill distribution may be biased.

A possible downside of the first version of the model is that it may overestimate the skill diversity. For example, if a person has ten different skills attributed to category *social, people and emotional skills* only, and accidentally the model makes a false classification by also classifying one of the skills to category *cognitive- and metacognitive skills*, there can be made no distinction between the two skill categories. This vulnerability is decreased by versions two and three of the model as one false classification would not have the same impact on the skill distribution as ten correctly classified skills. However, very versatile employees can be undervalued. For example, if an employee has ten skills in each category, no category would be considered outstanding even though represented by multiple skills. Another employee with only three skills, two of them in the first category and the remaining one in the second category, would seem more skilled in both of the categories. Therefore, the second and third version of the

model are better suited for a company-level overview, whereas the first version is better suited for personnel planning.

The following section examines the applicability of the model by demonstrating it on real world data from Xing featuring all three versions presented.

4 Demonstration with data from XING

4.1 Description of the data

The present dataset that is used for demonstration of the model is provided by New Work SE (New Work SE, 2020). It shows information from a social media network, namely XING (XING, 2020), from three different companies, which are not shown in the individual datapoint because of anonymity reasons. At XING, users predominantly connect with business contacts for strategic reasons (XING, 2020). On their profiles, they can upload personal information such as the age, working position, interests, hobbies, competencies and other. The dataset consists of 4532 anonym users and has the information “WANTS”, explaining the interests of a user, “HAVES”, explaining competencies, abilities and knowledge of users, “HOBBIES”, “JOB DESCRIPTION” and “AGE”. This model predominantly focuses on the column “HAVES”, as it holds the information needed for current *future skills* identification. The different “HAVES” (*haves*) a user communicates on his profile are separated by comma. In the following, the characteristics of users in the dataset are further described. Thereafter, the results of the presented versions are presented and interpreted. The thesis then concludes with a detailed discussion on limitations, future research and a brief summary.

Users in this dataset present between zero and 111 *haves*, with a median of nine, resulting in a total of 50184 *haves*. As XING is heavily used in German-speaking countries, the majority of descriptions is in German language. Because the model is trained on English words, the first preprocessing step was to translate the data into English language. This has been done by using the python implementation of the Microsoft translate API (*Translator*, 2020; Yin, 2017). During the data collection process, the encoding switched from UTF-8 (Unicode) to ISO 8859-1 (Latin-1). This resulted into the depiction of German umlauts as different characters, for instance “ä” as “Ã¤”, which had to be adjusted before the translation process. The results were saved in a two-dimensional list, with each row representing different user and each column representing a *have*. Second, the preprocessing steps explained in 3.2 were applied. In addition to punctuation marks, the model removed the words *skill, skills, ability, competence, competences* and *knowledge*, as *haves* containing those words proved to be likely to be classified into wrong, e.g. all, categories. This might be because the mentioned words all describe every of the four skill categories. To further improve the model, a dictionary containing abbreviations focusing on (expert) digital contexts, such as *business intelligence (BI)* or *user interface* (*UI)*,

and programming languages, such as *Python* or *R*, was created. As the model showed poor capabilities in handling such terms, an additional query that finds terms from this dictionary in given *haves* was added to the algorithm. This query automatically classifies a *have* into both *digital base skills* and *expert digital skills,* after a match was found.

For the evaluation of the model, 700 randomly chosen datapoints were labeled by hand, resulting into 2800 different classification decisions. Through the F-measure optimization process, the four category descriptions were shortened to 16, 13, 10 and 7 words, respectively. The finalized concept description words can be found in table 8 marked with superscript 4. For comparability reasons, the average F-measure of a random classifier was calculated by randomly classifying datapoints to categories and yielded $F_{random} = 0{,}267$ after 100 iterations. While the method without the dictionary of abbreviations and programming languages achieved a maximum of $F_{simple} = 0.507$ with a threshold of $t = 0.41$, the final version arrived at a significant higher $F_{final} = 0.564$. F_{final} was found at a similarity threshold of $t = 0.45$. This threshold is further used as the minimum similarity for classification. An additional validation test was run to test the performance of the shortened dictionary descriptions and showed that the model with uncut word sets achieved $F_{uncut} = 0.464$ at $t = 0.48$. This proves that the optimization process of this model in fact reaches better dictionary descriptions for classification than the dictionary creation process presented by Garten et al. (2018).

The results and a complete depiction of this model's results including all three presented versions can be observed in the next subsection.

4.2 Results of skill identification through the presented model

In the following graphs, a new variable called *Maximum Value* is introduced. This variable is normed to one and has the function of stretching the color scale. The color scale is usually set by the industry average so that the greener a category value, the better off is the company compared to the industry average. The three versions of depiction are now closer discussed. Furthermore, interpretations of the derived values are proposed.

First, the binary skill version was studied. From all 4532 users, 4099 had a minimum of one future skill. While 42.6% of users have at least one *digital base skill*, only 25.4% possess *expert digital skills*. As shown in figure 5, with 72.2% and 81.6% of examinees the majority of users share to have *cognitive- and metacognitive skills* as well as *social, people and emotional skills.*

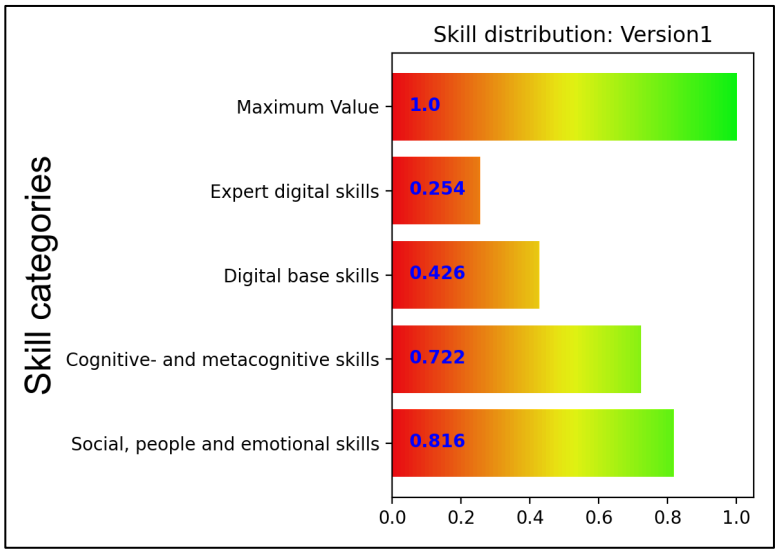

Figure 5: Results of distributed future skill extraction version 1.

However, the second version of depiction clearly shows that only a small number of *haves* given by a user can be categorized into a future skill category. For instance, only 5.3% of all given *haves* can be associated with category *expert digital skills*. A little more, 10.7% of *haves* are categorized into category *digital base skills*, 21.6% into *cognitive- and metacognitive* and 28.4% into *social, people and emotional skills*. Further investigation showed that only 20017 of all *haves* are part of at least one of the four *future skills* categories, meaning that the remaining 30167 or 60.1% of all *haves* cannot be classified as a *future skill*. What concerns the positively classified *haves*, investigation showed that 71.3% of them can be found in category *social, people and emotional skills,* 54.1% in category *cognitive- and metacognitive skills,* and 26.8% as well as 13.4% in categories *digital base* and *expert digital skills,* respectively.

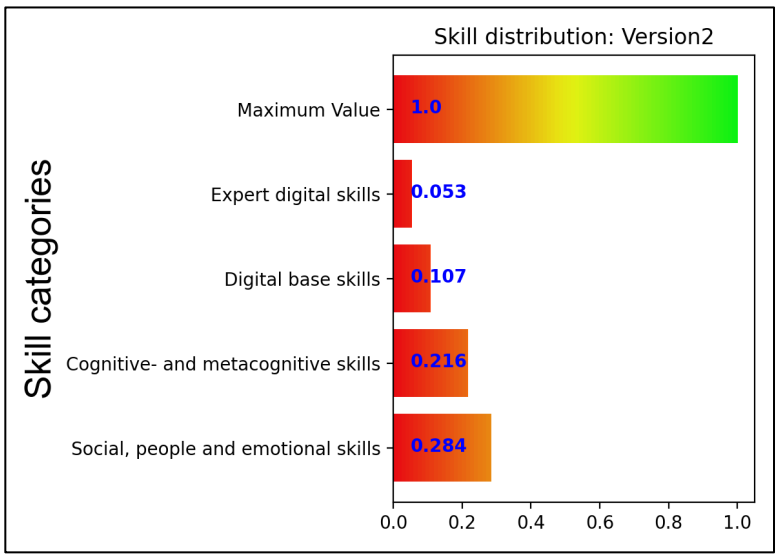

Figure 6: Results of distributed future skill extraction version 2.

Version 3 highlights the average distribution of *future skills* among the examinees a little further. Values in this version unsurprisingly show a strong correlation to those in version 2.

Results of skill classification display the average user to possess 5% of given *haves* associated with category *expert digital skills*, surpassed by all other categories *digital base skills* (9.3%), *cognitive- and metacognitive skills* (22.8%) and finally *social, people and emotional skills* (29.3%).

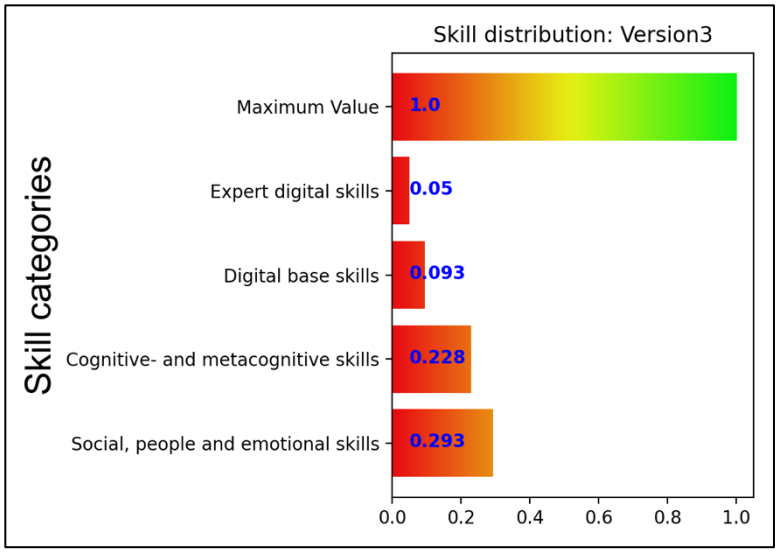

Figure 7: Results of distributed future skill extraction version 3.

5 Discussion

The following section first of all summarizes the execution of the model. Thereafter, limitations of the proposed approach of skill identification are critically presented. Finally, this thesis ends with proposals for further investigations on this topic and suggests options that could eliminate some of the limitations of the current model.

Nowadays, companies face the challenge to effectively identify hidden talents (Rosetti & Langhoff, 2015) and to translate into a world that experiences a long-lasting technological shift (Kirchherr et al., 2018; Smit et al., 2020). Since future skills are seen as a survival factor (Fareri et al., 2020; van Laar et al., 2017), rich literature already discussed potential future skills frameworks (Balliester & Elsheikhi, 2018; Davies et al., 2011; Kirchherr et al., 2018; OECD, 2018; Partnership for 21st Century learning, 2015), aimed at different scopes of application. Nevertheless, they do not suggest an effective way of application of such frameworks. Other works discuss retrieval systems for skill identification (Kanning, 2019, pp. 120-160; Rosetti & Langhoff, 2015) that struggle with high time exposure. Only little interest is shown in automated skill or technology identification (Fareri et al., 2020; Kanning, 2019, pp. 153-154). Therefore, this thesis presents its own distributed future skill extraction method applied on a newly summarized framework.

The presented method uses distributed dictionaries (Garten et al., 2018) and vector representations of social media profiles to instantly identify future skills from employees of the desired company.

First, an algorithm creates the best suited dictionary representations of future skill categories that later are used to calculate the similarity to user profile information. Second, XING-user profile information is prepared for skill extraction. Only *haves* of users are considered as they depict the competencies, abilities and skills a user currently has. Third, every distributed representation of a user's *haves* gets compared with every concept in terms of similarity and classified above a certain threshold. As a result, three versions of depictions are presented including interpretations. With an F-measure of $F_{final} = 0.564$, the model that considers abbreviations and programming languages with an exclusive dictionary for word tagging outperforms the simple model ($F_{simple} = 0.51$) that works with vector representations only. A comparison with a random classifier ($F_{random} = 0.27$) highlights the performance of this model. Garten et al. (2018) conducted a similar study in which they classified Twitter posts into ten different morality categories and one neutral category and achieved $F_{moral} = 0.496$ with their best classifier. This, combined with the short amount of time needed for implementation, proves the applicability of this model. Overall, the model was easy to execute with the given data. After writing the code that carries out the preprocessing steps, an instant analyzation of the *haves* was automatically computed, incorporating no further human action. Indeed, this model shows an uncomplicated behavior that saves the time usually consumed by traditional skill assessment methods.

5.1 Limitations

Besides the positive aspect of the model, predominantly meaning the time savings, the model suffers from limitations that hold the performance of the classifier down.

First, this model uses pretrained, 300-dimensional vector representations from FastText (Bojanowski et al., 2017). These are trained on the Wikipedia corpus and therefore not optimal for domain specific tasks such as specific future skill identification. This also includes that the vector of the programming language *python* is closer in terms of cosine similarity to the representation of the word *snake* ($sim\big(vec(\text{python}), vec(\text{snake})\big) = 0.556$) than that of the word *javascript,* another programming language ($sim\big(vec(\text{python}), vec(\text{javascript})\big) = 0.363$). Second, the datapoints used for model validation are too few for a significant evaluation of the models' validity. Besides only having 2800 labeled decisions for training, the real benefit of this model, the companywide comparison with industry standards, could not be studied with only 4532 examinees without employment information. For now, all examinees are treated as belonging to one company. Moreover, because of the lack of sufficient data, dictionary representations of future skills may have experienced an overfitting during the optimization process and have to be optimized with a larger data pool in future examinations. Third, the question whether the three proposed versions depict the actual skill distribution in companies remains. As stated before, the depictions are believed to be biased since not every employee has the same affinity

to social media accounts. Fourth, the additional dictionary aiming at tagging abbreviations and names from a specific field (in this case programming languages), is rather incomplete as new fields can emerge, e.g. by looking at companies from other sectors and other tools. The danger of missing out on notations is hard to eliminate and yields the necessity to input research in the desired field before an application of the model can be successful. Nevertheless, the model can be improved for removing some of the limitations to successfully assist modern diagnostics systems for skill identification and analyzation of a companies' readiness for the future. The following subsection clarifies the possibilities and suggests future research to assist such enhancements.

5.2 Practical implications

The proposed model is the first known approach to automatically identify future skills as a measurement for a companies' readiness for the future. However, for now it is just a construct that yet has to be unfolded. To arrive at the full potential, it is crucial for future researchers to collect data for different purposes. First, a promising approach would be to create a text corpus from job descriptions, skill definitions and CV's as the basis for FastText vector model training. For this, existing collections can be combined. Second, further data from social networks like XING should be crawled to create industrywide averages. Then, the readiness for the future of companies can be visualized by comparison with industry standards of *future skills* distributions. Additionally, more high-quality labeled data has to be created.

Third, future study should put supplementary focus on how other information from employees' social media pages can benefit the estimation of a companies' readiness for the future. For instance, the column "WANTS" in the XING dataset could potentially be used to find internal interest in fields that can benefit the closure of skill gaps or even enhance advantages over competitors. Another opportunity would be to analyze the value in different styles of *haves* given from employees. While most users list their respective qualities as key points, some write in whole sentences or even paragraphs. As one might infer that longer answers correlate with more creative character traits, it would be interesting to see what other properties can be used for (hidden) skill identification.

Fourth, especially the category *expert digital skills*, due to its relatively vague definition that includes transformative technologies, is able to be transformed in a way, that more specific statements can be retrieved. For instance, one could separate the category into subcategories devoted to only one area of transformative technologies. That way, companies could see how well they are prepared for specific trends like *blockchain technology* or *big data* by analyzing skills needed specifically for that trend.

Fifth, it remains to be seen how the model translates into other social networks similar to XING. Even though other social networks might be alike, take LinkedIn, they have different handling,

e.g. free text vs. pre-selected skills in the “HAVES” column. The model therefore has to be adjusted depending on the desired source of data.

Summing up, this model can help to effectively map in a company’s included skills and suggests comparison with competitors or industry standards for a clearer vision on how *future skills* can benefit, when above average, or harm, when forming a skill gap, future company development. Yet, it is still at the beginning of a whole new challenge: the quantization of a companie's readiness for the future.

Appendix

Appendix 1: Translate.py

```
# -*- coding: utf-8 -*-
import pandas as pd
from translate import Translator

# convertion dictionary because of faulty export ä, ö, ü, ß in original data
characterConv = {
    'Ã¤': 'ä',
    'Ä': 'Ä',   # no example found
    'Ã¶': 'ö',
    'Ã–': 'Ö',
    'Ã¼': 'ü',
    'Ãœ': 'Ü',
    'ÃŸ': 'ß'
}

#import data
df = pd.read_csv("/Users/lennard/Desktop/BADataRaw.csv", sep=";")

haves = df['HAVES'].tolist()

newData = []
charcount = 0
for row in haves:
    # if no entry, NaN is saved (which is not a string)
    if isinstance(row, str):
        # check if wrongly exported encoding exist
        for key in characterConv.keys():
            # replace with utf-8 character
            row = row.replace(key, characterConv[key])
        # create list for every user, listing all the "haves"
        newData.append(row.split(','))
    # still save an empty list for NaN cases for later normalization
    else:
        newData.append('')

count = 0
# microsoft api key
secret = 'SECRETKEYSTRING'
# empty list in which the result gets saved in
translated = []
# look at every user
for user in newData:
    tempuser=[]
    # iterate through all the haves of user
    for i, s in enumerate(user):
        # initialize translator every time to avoid high frequency ban error
        translator = Translator(provider = "microsoft", secret_access_key= secret,
from_lang ='de', to_lang='en')
        # encountered problems with lengthy descriptions
        if len(user[i])>500:
            print(len(user[i]), user[i])
        else:
            # actual translation
            tempuser.append(translator.translate(user[i]))
            # keep track during process bc. it takes a while
            print("I just did the ", count, "-th translation")
        count += 1
    # add translated haves of user to result list
    translated.append(tempuser)
# create data frame bc. its easier to write df to excel
```

```
df = pd.DataFrame(translated)
df.to_excel("translated.xlsx", sheet_name='batch1')
```

Appendix 2: fasttext.py

```
import fastText
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt; plt.rcdefaults()
import matplotlib as mpl
import matplotlib.pyplot as plt

#load pre trained 300-dim model wikipedia based englisch version
model = fastText.load_model("result/pretrained.bin")

# model options: model.get_nearest_neighbors("word", k=x), k stands for the amount
of neighbors displayed
#                model.get_word_vector("word"), displays the vector as an array
#                model.save_model("path"), saves model when trained
#                [model.get_word_vector(x) for x in ["asparagus", "pidgey", "yel-
low"]], list of vectors
#                model.get_analogies("a", "b", "c"), what a is to b, c is to OUTPUT
#                more to be added..

#calculate the concept vector of a skill-category
def conceptVector(ConceptWordList):
    #create empty numpy vector of dimension 300
    returnVec = np.zeros(len(model.get_word_vector("word")))
    #run through whole list of descriptive words of concept
    for word in ConceptWordList:
        splitwordret = np.zeros(len(model.get_word_vector("word")))
        for splitword in word.split(" "):
            splitwordret += np.array(model.get_word_vector(splitword))
        splitwordret = splitwordret/len(word.split(" "))
        returnVec += splitwordret
    returnVec = returnVec/len(ConceptWordList)  #average, like in ddr
    return returnVec

#calculate the cosine similarity
def similarity(v1, v2):
    if np.linalg.norm(v1) == 0 or np.linalg.norm(v2) == 0:
        return 0
    #normalization simplifies the cosine to a simple dot-product
    norm1 = v1/np.linalg.norm(v1)
    norm2 = v2/np.linalg.norm(v2)
    return np.dot(norm1, norm2)

# dictionary containing all the skills and their descriptions, now filled with im-
aginary names
dict = {
    'Social, people and emotional skills' : ['people', 'responsibility', 'collabo-
ration', 'personnel', 'coordination', 'recruiting', 'enthusiasm', 'communication',
'intercultural', 'team', 'admiration', 'ethical', 'thoughtfulness', 'listening',
'customer', 'teamwork'],
    'Cognitive- and metacognitive skills' : ['passion', 'curiosity', 'responsibil-
ity', 'mutability', 'accountability', 'intellectual', 'commitment', 'decision
making', 'sophisticated', 'flexibility', 'solving', 'engagement', 'adaptability'],
    'Digital base skills' : ['internet', 'virtual collaboration', 'microsoft',
'digital', 'technology', 'excel', 'computer', 'software', 'skype', "data"],
    'Expert digital skills' : ['machine learning', 'syntax', 'business intelli-
gence', 'programming', 'industry 4 0', 'blockchain', 'data'],
    'Maximum Value' : ["max"]
}
exDiSkAbr = {
    "python" : "python",
```

```
    "java" : "java",
    "c" : "c",
    "c#" : "c#",
    "c##" : "c##",
    "c++" : "c++",
    "x++" : "x++",
    "do178b" : "do178b",
    "html" : "html",
    "sql" : "sql",
    "mysql" : "mysql",
    "mdx" : "mdx",
    "spss" : "spss",
    "sap" : "sap",
    "eclipse" : "eclipse",
    "scrum" : "scrum",
    "oracle" : "oracle",
    "css" : "css",
    "srummaster" : "scrummaster",
    "c #" : "c #",
    "xml" : "xml",
    "javascript" : "javascript",
    "r" : "r",
    "matlab" : "matlab",
    "do 254": "do 254",
    "php" : "php",
    "ssh" : "ssh",
    "typo3" : "typo3",
    "nosql" : "nosql",
    "mssql" : "mssql",
    "nlp" : "nlp",
    "ml" : "ml",
    "vb" : "vb",
    "vba" : "vba",
    "bi" : "bi",
    "ai" : "ai",
    "ui" : "ui",
    "ux" : "ux",
    "shell" : "shell",
    "vbscript" : "vbscript",
}
changeDict = {
    ";" : " ",
    ":" : " ",
    "/" : " ",
    "(" : " ",
    ")" : " ",
    "&" : " ",
    "%" : " ",
    "," : " ",
    "-" : " ",
    "_" : " ",
    "skill" : " ",
    "skills" : " ",
    "competence" : " ",
    "competences": " ",
    "knowledge" : " ",
    "ability" : " "
}
ConceptVecs = []
skList = []
# load translated haves
list = pd.read_excel("/Users/lennard/Desktop/Ich/BA/results/translated.xlsx",
header = None).values.tolist()
# preprocessing data
for user in list:
    for i in range(len(user)):
```

```
        if pd.notnull(user[i]):
            user[i] = user[i].lower()
            for key in changeDict.keys():
                user[i] = user[i].replace(key, changeDict[key])
                user[i] = ' '.join(user[i].split())
# calculate concept vecs
for concept in dict:
    ConceptVecs.append(conceptVector(dict[concept]))
# set threshold for  classification
threshold = 0.45
for user in list:
    # initialize vector of concept names, has values 0 or 1
    counts = [0]*len(dict)
    # look at every "have" a user has
    for have in user:
        # check if there is a have
        if pd.notnull(have):
            for category in range(0, len(counts)):
                # for max value depiction of colour scale
                if category == len(counts)-1:
                    counts[category] += 1
                else:
                    # category similarity
                    catSim = similarity(conceptVector(have.split(" ")),Con-
ceptVecs[category])
                    # check if have has needed similarity
                    if catSim > threshold:
                        counts[category] += 1
                    # else check for abbreviations or programming languages for
digital categories
                    elif category > 1:
                        for key in exDiSkAbr.keys():
                            for word in have.split(" "):
                                if key == word:
                                    counts[category] += 1

    # 2D-list of skills of every user
    skList.append(counts)
# set desired  version
version = 3

if version == 1:
    # calculate version 1
    for user in skList:
        for j in range(len(user)):
            if user[j] >0:
                user[j] = 1
            else:
                pass
    # create a numpy array to apply simple LA
    s = np.array([0] * len(dict))
    for user in skList:
        # instead of having counter for all categories, just add the skill vectors
        s = s + np.array(user)
    s = s/len(skList)

elif version == 2:
#version 2
    a = 0
    for user in list:
        for have in user:
            if pd.notnull(have):
                a+=1
    s = np.array([0]*len(dict))
    for user in skList:
        s = s + np.array(user)
```

```
    s = s/a
elif version == 3:

#version3
    for user in range(len(list)):
        ai = 0
        for have in list[user]:
            if pd.notnull(have):
                ai += 1
        for value in range(len(skList[user])):
            if ai != 0:
                skList[user][value] = skList[user][value]/ai
    s = np.array([0]*len(dict))
    for user in skList:
        s = s + np.array(user)
    s = s/len(skList)


#prepare the plot
plt.rcParams.update({'figure.autolayout': True})
fig, ax = plt.subplots()
objects = tuple(dict)
y_pos = np.arange(len(objects))
data = s
plt.barh(y_pos, data, align='center', alpha=0.1)
plt.yticks(y_pos, objects, rotation = 0)
plt.ylabel('Skill categories', fontname = "arial", fontsize = 20)
plt.title('Skill distribution: Version'+str(version))
bar = ax.barh(range(len(data)), data)
for i, v in enumerate(data):
    plt.text(0.05, i - 0.05, str(round(v, 3)), color='blue', fontweight='bold')
# for colour scale and optics
def gradientbars(bars):
      ax = bars[0].axes
      lim = ax.get_xlim()+ax.get_ylim()
      for bar in bars:
          bar.set_zorder(1)
          bar.set_facecolor("none")
          x,y = bar.get_xy()
          w, h = bar.get_width(), bar.get_height()
          grad = np.atleast_2d(np.linspace(0,1*w/max(data),256))
          ax.imshow(grad, extent=[x,x+w,y,y+h], aspect="auto", zorder=0,
norm=mpl.colors.Normalize(vmin=0,vmax=3), cmap = plt.get_cmap('hsv'), alpha = 1)
      ax.axis(lim)
gradientbars(bar)
plt.show()
```

Appendix 3: Validation.py

```
import fastText
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt; plt.rcdefaults()

#load pre trained 300-dim model wikipedia based englisch version
model = fastText.load_model("result/pretrained.bin")
# model options: model.get_nearest_neighbors("word", k=x), k stands for the amount
of neighbors displayed
#                model.get_word_vector("word"), displays the vector as an array
#                model.save_model("path"), saves model when trained
#                [model.get_word_vector(x) for x in ["asparagus", "pidgey", "yel-
low"]], list of vectors
#                model.get_analogies("a", "b", "c"), what a is to b, c is to OUTPUT
#                more to be added..

#calculate the concept vector of a skill-category
```

```
def averageVector(wordList):
    #create empty numpy vector of dimension 300
    returnVec = np.zeros(len(model.get_word_vector("word")))
    for word in wordList:  #run through whole list of descriptive words of concept
        splitwordret = np.zeros(len(model.get_word_vector("word")))
        for splitword in word.split(" "):
            splitwordret += np.array(model.get_word_vector(splitword))
        splitwordret = splitwordret/len(word.split(" "))
        returnVec += splitwordret
    returnVec = returnVec/len(wordList)     #average, like in ddr
    return returnVec

#calculate the cosine similarity between vector v1 and v2
def similarity(v1, v2):
    if np.linalg.norm(v1) == 0 or np.linalg.norm(v2) == 0:
        return 0
    #normalization simplifies the cosine to a simple dot-product
    norm1 = v1/np.linalg.norm(v1)
    norm2 = v2/np.linalg.norm(v2)
    return np.dot(norm1, norm2)

def calcF(threshold):
    skList = []
    for have in list:
        # initialize vector of concept names, has values 0 or 1
        counts = [0]*len(dict)
        # check if there is a have
        if pd.notnull(have):
            for category in range(0, len(counts)):
                #calc category similarity
                catSim = similarity(averageVector(have.split(" ")), Con-
ceptVecs[category])
                # check if have has needed similarity
                if catSim > threshold:
                    counts[category] = 1
                elif category >1:
                    for key in exDiSkAbr.keys():
                        for word in have.split(" "):
                            if key == word:
                                counts[category] = 1
        # 2D-list of skills of every user
        skList.append(counts)
    # initialize counter variables for True/False classification
    TP = 0
    TN = 0
    FP = 0
    FN = 0
    # count all TP/TN/FP/FN
    for i in range(len(label)):
        for j in range(len(label[i])):
            if label[i][j] == skList[i][j] and skList[i][j] == 0:
                TN += 1
            elif label[i][j] == skList[i][j] and skList[i][j] == 1:
                TP += 1
            elif label[i][j] != skList[i][j] and skList[i][j] == 0:
                FN += 1
            elif label[i][j] != skList[i][j] and skList[i][j] == 1:
                FP += 1
            else:
                pass
    # catch zero cases
    if TP+FP == 0:
        precision = 0
    else:
        precision = TP/(TP+FP)
    if TP+FP == 0:
```

```
        recall =0
    else:
        recall = TP/(TP+FN)
    if precision+recall == 0:
        f = 0
    else:
        f = (2*precision*recall)/(precision+recall)
    return f

# dictionary containing all the skills and their descriptions, now filled with im-
aginary names
dict = {
    'Social, people and emotional skills' : ['social', 'culture', 'collaboration',
'communication', 'ethical', 'intercultural', 'team', 'enthusiasm', 'empathy', 're-
sponsibility', 'listening','participation', 'people', 'cooperation', 'coordina-
tion', 'acceptance', 'speaker', 'clarity', 'accountability', 'thoughtfulness', 'em-
pathic', 'emotion', 'admiration', 'advice', 'customer', 'employee', 'sociocultur-
al', 'teamwork', 'communication', 'personnel', 'recruiting'],
    'Cognitive and meta-cognitive skills' : ["creativity", "solving", "innovative",
"awareness", "mutability", "adaptability", "responsibility", "analytical thinking",
"curiosity", "endurance", "passion", "judgement", "decision making", "flexibility",
"strategic", "time management", "engagement", "commitment", "accountability", "so-
phisticated", "intellectual"],
    'Digital base skills' : ["microsoft office", "excel", "digital", "digital in-
teraction", "digital learning", "web conference", "technology", "mathematics",
"computer", "software", "virtual", "virtual collaboration", "international", "in-
ternet", "electronic", "information technology", "communication technology", "digi-
tal competence", "digital interaction", "skype", "system"],
    'Expert digital skills' : ["blockchain", "robotic", "data analysis", "analyt-
ics", "visualization", "artificial intelligence", "data science", "machine learn-
ing", "data mining", "text mining", "algorithm", "business intelligence", "automa-
tion", "industry 4 0", "database", "data base", "web engeneering", "programming",
"code", "syntax", "mathematics"]
}
exDiSkAbr = {
    "python" : "python",
    "java" : "java",
    "c" : "c",
    "c#" : "c#",
    "c##" : "c##",
    "c++" : "c++",
    "x++" : "x++",
    "do178b" : "do178b",
    "html" : "html",
    "sql" : "sql",
    "mysql" : "mysql",
    "mdx" : "mdx",
    "spss" : "spss",
    "sap" : "sap",
    "eclipse" : "eclipse",
    "scrum" : "scrum",
    "oracle" : "oracle",
    "css" : "css",
    "srummaster" : "scrummaster",
    "c #" : "c #",
    "xml" : "xml",
    "javascript" : "javascript",
    "r" : "r",
    "matlab" : "matlab",
    "do 254": "do 254",
    "php" : "php",
    "ssh" : "ssh",
    "typo3" : "typo3",
    "nosql" : "nosql",
    "mssql" : "mssql",
    "nlp" : "nlp",
```

```
    "ml" : "ml",
    "vb" : "vb",
    "vba" : "vba",
    "bi" : "bi",
    "ai" : "ai",
    "ui" : "ui",
    "ux" : "ux",
    "shell" : "shell",
    "vbscript" : "vbscript",
}
changeDict = {
    ";" : " ",
    ":" : " ",
    "/" : " ",
    "(" : " ",
    ")" : " ",
    "&" : " ",
    "%" : " ",
    "," : " ",
    "-" : " ",
    "_" : " ",
    "skill" : " ",
    "skills" : " ",
    "competence" : " ",
    "competences": " ",
    "knowledge" : " ",
    "ability" : " "
}
fDict = {
}
ConceptVecs = []
labelList = pd.read_excel("label.xlsx")
# load translated haves
list = labelList["Beschreibung"].values.tolist()
for i in range(len(list)):
    list[i] = list[i].lower()
    for key in changeDict.keys():
        list[i] = list[i].replace(key, changeDict[key])
        list[i] = ' '.join(list[i].split())
label = labelList[["social, people and emotional skills", "cognitive- and metacog-
nitive skills", "digital base skills", "expert digital skills"]]
label = label.fillna(0).values.tolist()
# calculate concept vecs
for concept in dict:
    ConceptVecs.append(averageVector(dict[concept]))
t = 0
learnRate = 100
while t < 1:
    fDict[str(t)] = calcF(t)
    t += 1/learnRate

maximum = max(fDict, key=fDict.get)
print(maximum, fDict[maximum])
```

Appendix 4: WordListFinder.py

```
import fastText
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt; plt.rcdefaults()
import random

def averageVector(wordList):
```

```
    # create empty numpy vector of dimension 300
    returnVec = np.zeros(len(model.get_word_vector("word")))
    # run through whole list of descriptive words of concept
    for word in wordList:
        splitwordret = np.zeros(len(model.get_word_vector("word")))
        for splitword in word.split(" "):
            splitwordret += np.array(model.get_word_vector(splitword))
        splitwordret = splitwordret/len(word.split(" "))
        returnVec += splitwordret
    returnVec = returnVec/len(wordList)
    return returnVec

# calculate the cosine similarity between vector v1 and v2
def similarity(v1, v2):
    if np.linalg.norm(v1) == 0 or np.linalg.norm(v2) == 0:
        return 0
    # normalization simplifies the cosine to a simple dot-product
    norm1 = v1/np.linalg.norm(v1)
    norm2 = v2/np.linalg.norm(v2)
    return np.dot(norm1, norm2)

def getWordList(threshold, category):
    skList = []
    cat = category
    for have in list:
        # initialize vector of concept names, has values 0 or 1
        counts = [0]*4
        # check if there is a have
        if pd.notnull(have):
            # category similarity
            catSim = similarity(averageVector(have.split(" ")), averageVector(tem-
pList))
            # check if have has needed similarity
            if catSim > threshold:
                counts[cat] = 1
            # look for abbreviations
            else:
                for key in exDiSkAbr.keys():
                    for word in have.split(" "):
                        # needed when looking at 3rd or 4th category
                        if key == word and cat > 1:
                            counts[cat] = 1
        # 2D-list of skills of every user
        skList.append(counts)
    # initialize classification counters
    TP = 0
    TN = 0
    FP = 0
    FN = 0
    # count them
    for i in range(len(label)):
        if label[i][cat] == skList[i][cat] and skList[i][cat] == 0:
            TN += 1
        elif label[i][cat] == skList[i][cat] and skList[i][cat] == 1:
            TP += 1
        elif label[i][cat] != skList[i][cat] and skList[i][cat] == 0:
            FN += 1
        elif label[i][cat] != skList[i][cat] and skList[i][cat] == 1:
            FP += 1
        else:
            pass

    if TP+FP == 0:
        precision = 0
    else:
        precision = TP/(TP+FP)
```

```
    if TP+FP == 0:
        recall =0
    else:
        recall = TP/(TP+FN)
    if precision+recall == 0:
        f = 0
    else:
        f = (2*precision*recall)/(precision+recall)

    return f

#load pre trained 300-dim model wikipedia based englisch version
model = fastText.load_model("result/pretrained.bin")

exDiSkAbr = {
    "python": "python",
    "java": "java",
    "c": "c",
    "c#": "c#",
    "c##": "c##",
    "html": "html",
    "sql": "sql",
    "scrum": "scrum",
    "srummaster": "scrummaster",
    "c #": "c #",
    "xml": "xml",
    "javascript": "javascript",
    "r": "r",
    "matlab": "matlab",
    "do 254": "do 254",
    "php": "php",
    "ssh": "ssh",
    "nosql": "nosql",
    "nlp": "nlp",
    "ml": "ml",
    "bi": "bi",
    "ai": "ai",
    "ui": "ui",
    "ux": "ux",
    "shell": "shell",
    "vbscript": "vbscript",
    "css":"css",
    "emc": "emc",
    "scripting" : "scripting",
    "typo3" : "typo3",
    "erm" : "erm",
    "powershell": "powershell",
    "dhcp" : "dhcp"
}
changeDict = {
    ";" : " ",
    ":" : " ",
    "/" : " ",
    "(" : " ",
    ")" : " ",
    "&" : " ",
    "%" : " ",
    "," : " ",
    "-" : " ",
    "_" : " ",
    "skill" : " ",
    "skills" : " ",
    "competence" : " ",
    "competences": " ",
    "knowledge" : " ",
    "ability" : " "
```

```
}

labelList = pd.read_excel("label.xlsx")
# load translated haves
list = labelList["Beschreibung"].values.tolist()
for i in range(len(list)):
    list[i] = list[i].lower()
    for key in changeDict.keys():
        list[i] = list[i].replace(key, changeDict[key])
        list[i] = ' '.join(list[i].split())
label = labelList[["social, people and emotional skills", "cognitive- and metacog-
nitive skills", "digital base skills", "expert digital skills"]]
label = label.fillna(0).values.tolist()

tempList = []
tempmax = 0
wordLists = []        # insert whole word list
categoryNum = 0       # insert correct category, count from 0
for numWords in range(1, len(wordLists)):
    for j in range(10):
        # keep track of process (computanionally lengthy)
        print(numWords, " from ", len(wordLists), "iteration: ", j)
        tempList = []
        # save all the f-measures
        fDict = {
        }
        # increase number of words
        for i in range(numWords):
            # take random word
            newWord = wordLists[random.randint(0, len(wordLists)-1)]
            while True:
                print("looking for word")
                # make sure to only catch unique words
                if newWord in tempList:
                    newWord = wordLists[random.randint(0, len(wordLists)-1)]
                else:
                    break
            tempList.append(newWord)
        # run validation loop with tempList
        t = 0
        learnRate = 100
        while t < 1:
            fDict[str(t)] = getWordList(t, categoryNum)
            t += 1 / learnRate
        maximum = max(fDict, key=fDict.get)
        # save current best list
        if fDict[maximum] > tempmax:
            tempmax = fDict[maximum]
            tempBestList = tempList
        else:
            pass
```

Appendix 5: RandomClassifier.py

```
import pandas as pd
import random

def calcRandomF(threshold):
    skList = []
    for have in list:
        # initialize vector of concept names, has values 0 or 1
        counts = [0]*4
        # check if there is a have
        if pd.notnull(have):
            for category in range(0, len(counts)):
                # randomly (not) classify
```

```
                counts[category] = random.randint(0,1)
        # 2D-list of skills of every user
        skList.append(counts)
    # initialize calssification counters
    TP = 0
    TN = 0
    FP = 0
    FN = 0
    # count classifications
    for i in range(len(label)):
        for j in range(len(label[i])):
            if label[i][j] == skList[i][j] and skList[i][j] == 0:
                TN += 1
            elif label[i][j] == skList[i][j] and skList[i][j] == 1:
                TP += 1
            elif label[i][j] != skList[i][j] and skList[i][j] == 0:
                FN += 1
            elif label[i][j] != skList[i][j] and skList[i][j] == 1:
                FP += 1
            else:
                pass
    # catch zero cases
    if TP+FP == 0:
        precision = 0
    else:
        precision = TP/(TP+FP)
    if TP+FP == 0:
        recall =0
    else:
        recall = TP/(TP+FN)
    if precision+recall == 0:
        f = 0
    else:
        f = (2*precision*recall)/(precision+recall)
    return f

randomF = []
ConceptVecs = []
labelList = pd.read_excel("/Users/lennard/Desktop/untitled1/label.xlsx")
# load translated haves (for upper for loop)
list = labelList["Beschreibung"].values.tolist()
label = labelList[["social, people and emotional skills", "cognitive- and metacog-
nitive skills", "digital base skills", "expert digital skills"]]
label = label.fillna(0).values.tolist()
t = 0
# run severral times for average random score
repeatRate = 100
while t < repeatRate:
    randomF.append(calcRandomF(t))
    t += 1
print("Average is: ", sum(randomF)/repeatRate)
```

Appendix 6: FurtherInvestigation.py

```
#how many "haves" are future skills?
count = 0
for user in list:
    for have in user:
        if pd.notnull(have):
            for category in range(4):
                catSim = similarity(conceptVector(have.split(" ")),ConceptVecs[cat-
egory])
                if catSim>0.45:
                    count += 1
                    break
```

```
# how many users have a minimum of one future skill?
count = 0
for user in skList:
    for num in range(len(user)):
        if user[num] == 1:
            if num != 4:
                count += 1
                break
```

References

Ahonen, A. K., & Kinnunen, P. (2015). How Do Students Value the Importance of Twenty-first Century Skills? *Scandinavian Journal of Educational Research*, *59*(4), 395–412. https://doi.org/10.1080/00313831.2014.904423

Attewell, P. (1990). What Is Skill? *Work and Occupations*, *17*(4), 422–448.

Balliester, T., & Elsheikhi, A. (2018). The Future of Work: A Literature Review. In *ILO Research Department Working Paper* (Issue 29).

Barbosa, N., & Faria, A. P. (2008). Technology adoption: Does labour skill matter? Evidence from Portuguese firm-level data. *Empirica*, *35*(2), 179–194. https://doi.org/10.1007/s10663-007-9056-x

Baumol, W. J. (2002). Towards microeconomics of innovation: Growth engine hallmark of market economics. *Atlantic Economic Journal*, *30*(1), 1–12. https://doi.org/10.1007/BF02299142

Benson, A. D., Johnson, S. D., & Kuchinke, K. P. (2002). The Use of Technology in the Digital Workplace: A Framework for Human Resource Development. *Advances in Developing Human Resources*, *4*(4), 392–404. https://doi.org/10.1177/152342202237518

Berezina, K., Bilgihan, A., Cobanoglu, C., & Okumus, F. (2016). Understanding Satisfied and Dissatisfied Hotel Customers: Text Mining of Online Hotel Reviews. *Journal of Hospitality Marketing and Management*, *25*(1), 1–24. https://doi.org/10.1080/19368623.2015.983631

Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Miller-Ricci, M., & Rumble, M. (2012). Defining twenty-first century skills. In *Assessment and teaching of 21st century skills*. Springer. https://doi.org/10.1007/978-94-007-2324-5

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051

Borg, I., & Mastrangelo, P. M. (2009). *Employee surveys in management: Theories, tools, and practical applications*. Hogrefe Publishing.

Cainelli, G., Evangelista, R., & Savona, M. (2004). The impact of innovation on economic performance in services. *The Service Industries Journal*, *24*(1), 116–130. https://doi.org/10.1080/0264206041233130116 2

Cambridge Dictionary. (2020). *Definition: Skill*. https://dictionary.cambridge.org/de/worterbuch/englisch/skill (visited on 04/20/2020).

Caputo, F., Cillo, V., Candelo, E., & Liu, Y. (2019). Innovating through digital revolution: The role of soft skills and Big Data in increasing firm performance. *Management Decision*, *57*(8), 2032–2051. https://doi.org/10.1108/MD-07-2018-0833

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. *Proceedings of the European Conference on Computer Vision (ECCV)*, 132–149. https://doi.org/10.1007/978-3-030-01264-9_9

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd International Conference on Machine Learning*, 161–168. https://doi.org/10.1145/1143844.1143865

Choi, J., & Lee, S. W. (2020). Improving FastText with inverse document frequency of subwords. *Pattern Recognition Letters*, *133*, 165–172. https://doi.org/10.1016/j.patrec.2020.03.003

Chui, M., Manyika, J., & Miremadi, M. (2016). Where machines could replace humans-and where they can't (yet). *McKinsey Quarterly*, *30*(2), 1–9.

Cimatti, B. (2016). Definition, development, assessment of soft skills and their role for the quality of organizations and enterprises. *International Journal for Quality Research*, *10*(1), 97–130. https://doi.org/10.18421/IJQR10.01-05

Collobert, R., & Weston, J. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. *Proceedings of the 25th International Confer- Ence on Machine Learning*. https://doi.org/10.1145/1390156.1390177

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*, *12*, 2493–2537.

Croteau, A. M., Dyer, L., & Miguel, M. (2010). Employee reactions to paper and electronic surveys: An experimental comparison. *IEEE Transactions on Professional Communication*, *53*(3), 249–259. https://doi.org/10.1109/TPC.2010.2052852

Cudeck, R. (1985). A Structural Comparison of Conventional and Adaptive Versions of the ASVAB. *Multivariate Behavioral Research*, *20*(3), 305-322.

Daheim, C., & Wintermann, O. (2016). *2050: Die Zukunft der Arbeit.*

Davies, A., Fidler, D., & Gorbis, M. (2011). Future Work Skills 2020. *Institute for the Future for University of Phoenix Research Institute*, 540.

Deming, D. J. (2017). The growing importance of social skills in the labor market. *Quarterly Journal of Economics*, *132*(4), 1593–1640. https://doi.org/10.1093/qje/qjx022

Egorov, S., Yuryev, A., & Daraselia, N. (2004). A Simple and Practical Dictionary-based Approach for Identification of Proteins in Medline Abstracts. *Journal of the American Medical Informatics Association*, *11*(3), 174–178. https://doi.org/10.1197/jamia.M1453

ESCO. (2020). *Europäische Klassifikation für Fähigkeiten, Kompetenzen, Qualifikationen und Berufe (ESCO)*. https://ec.europa.eu/esco/portal/skill (visited on 05/10/2020).

Esuli, A., & Sebastiani, F. (2006). SentiWordNet: A Publicly Available Lexical Resource for

Opinion Mining. *LREC*, *6*, 417–422.

Fareri, S., Fantoni, G., Chiarello, F., Coli, E., & Binda, A. (2020). Estimating Industry 4.0 impact on job profiles and skills using text mining. *Computers in Industry*, *118*, 103222. https://doi.org/10.1016/j.compind.2020.103222

Garten, J., Hoover, J., Johnson, K. M., Boghrati, R., Iskiwitch, C., & Dehghani, M. (2018). Dictionaries and distributions: Combining expert knowledge and large scale textual data content analysis: Distributed dictionary representation. *Behavior Research Methods*, *50*(1), 344–361. https://doi.org/10.3758/s13428-017-0875-9

Gitari, N. D., Zuping, Z., Damien, H., & Long, J. (2015). A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, *10*(4), 215–230. https://doi.org/10.14257/ijmue.2015.10.4.21

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning word vectors for 157 languages. *LREC 2018 - 11th International Conference on Language Resources and Evaluation*.

Green, F. (2011). *What is Skill? An Inter-Disciplinary Synthesis*. Published by the Centre for Learning and Life Chances in Knowledge Economies and Societies at: http://www.llakes.org.

Hardeniya, T., & Borikar, D. A. (2016). Dictionary based approach to sentiment analysis -A Review. *International Journal of Advanced Engineering, Management and Science (IJAEMS)*, *2*(5), 317–322.

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

Hatzivassiloglou, V., & McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, 174–181. https://doi.org/10.3115/979617.979640

Herculano-Houzel, S. (2009). The human brain in numbers: A linearly scaled-up primate brain. *Frontiers in Human Neuroscience*, *3*. https://doi.org/10.3389/neuro.09.031.2009

Hermann, D. A., & Zimmermann, D. T. (2020). *Stepstone Gehaltsreport 2020*.

Heyer, G., Quasthoff, U., & Wittig, T. (2006). Text mining: Wissensrohstoff text. *W3l, Herdecke*, *18*.

Hippner, H., & Rentzmann, R. (2006). Text mining. *Informatik-Spektrum*, *29*(4), 287–290. https://doi.org/10.1007/s00287-006-0091-y

Hjørland, B. (2016). Does the traditional thesaurus have a place in modern information retrieval? *Knowledge Organization*, *43*(3), 145–159. https://doi.org/10.5771/0943-7444-2016-3-145

Huang, G.-B. (2003). Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Transactions on Neural Networks*, *14*(2), 274–281. https://doi.org/10.1109/TNN.2003.809401

Ihara, I. (2017). *Our discovery of cramming*. Twitter, Inc. https://blog.twitter.com/engineering/en_us/topics/insights/2017/Our-Discovery-of-Cramming.html (visited on 05/17/2020).

Indeed. (2020). *Social Skills: Defintion and Examples*. Indeed.Com. https://www.indeed.com/career-advice/career-development/social-skills (visited on 04/06/2020).

Jack, L., & Tsai, Y. D. (2015). Using Text Mining of Amazon Reviews to Explore User-Defined Product Highlights and Issues. *Proceedings of the International Conference on Data Mining (DMIN)*, *July*, 92–98.

Jackson, N. (2011). *Infographic: Who Is Using Twitter, How Often, and Why?* The Atlantic. https://www.theatlantic.com/technology/archive/2011/07/infographic-who-is-using-twitter-how-often-and-why/241407/ (visited on 06/18/2020).

Julita. (2011). *Difference Between Ability and Skill*. DifferenceBetween.Net. http://www.differencebetween.net/language/difference-between-ability-and-skill/ (visited on 04/06/2020).

Kacewicz, E., Pennebaker, J. W., Davis, M., Jeon, M., & Graesser, A. C. (2014). Pronoun Use Reflects Standings in Social Hierarchies. *Journal of Language and Social Psychology*, *33*(2), 125–143. https://doi.org/10.1177/0261927X13502654

Kannan, D. S., & Gurusamy, V. (2014). Preprocessing Techniques for Text Mining Preprocessing Techniques for Text Mining. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16.

Kanning, U. P. (2019). *Standards der Personaldiagnostik* (2nd ed.). Hogrefe Verlag.

Kavoo-Linge, D. T., & Kiruri, J. K. (2013). The effect of placement practices on employee performance in small service firms in the information technology sector in kenya. *International Journal of Business and Social Science*, *4*(15), 213–219.

Kiesler, S., & Sproull, L. S. (1986). Response Effects in the Electronic Survey. *Public Opinion Quarterly*, *50*(3), 402–413. https://doi.org/https://doi.org/10.1086/268992

Kirchherr, J., Klier, J., Lehmann-Brauns, C., & Winde, M. (2018). Future skills: Welche Kompetenzen in Deutschland fehlen. In *Future Skills-Diskussionspapier 1*.

Klimoski, R., & Brickner, M. (1987). Why Do Assessment Centers Work? the Puzzle of Assessment Center Validity. *Personnel Psychology*, *40*(2), 243–260. https://doi.org/10.1111/j.1744-6570.1987.tb00603.x

Krathwohl, D. R. (2002). A Revision of Bloom's Taxonomy: An Overview, Theory into Practice.

American Journal of Psychology, *41*(4), 212–218. https://doi.org/10.1207/s15430421tip4104

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, s Social Network or a News Media? *Proceedings of the 19th International Conference on World Wide Web*, 591–600. https://doi.org/10.4321/S0004-05922011000200015

Lilleberg, J., Zhu, Y., & Zhang, Y. (2015). Support vector machines and Word2vec for text classification with semantic features. *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, 136–140. https://doi.org/10.1109/ICCI-CC.2015.7259377

Ling, W., Dyer, C., Black, A., & Trancoso, I. (2015). Two / Too Simple Adaptations of Word2Vec for Syntax Problems. *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1299–1304.

LinkedIn. (2019). *What is LinkedIn and How Can I Use It?* https://www.linkedin.com/help/linkedin/answer/111663/what-is-linkedin-and-how-can-i-use-it-?lang=en (visited on 06/18/2020).

Louwerse, M. M. (2004). Semantic Variation in Idiolect and Sociolect: Corpus Linguistic Evidence from Literary T exts. *Computers and the Humanities*, *38*(2), 207–221. https://doi.org/10.1023/B:CHUM.0000031185.88395.b1

Lumauag, R. G. (2019). Decision support system for personnel selection. *International Journal of Recent Technology and Engineering*, *8*(1), 177–179.

Manad, O., Bentounsi, M., & Darmon, P. (2018). Enhancing Talent Search by Integrating and Querying Big HR Data. *2018 IEEE International Conference on Big Data (Big Data)*, 4095–4100. https://doi.org/10.1109/BigData.2018.8622275

Manning, C. D. (2011). Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? *International Conference on Intelligent Text Processing and Computational Linguistics.* https://doi.org/10.1007/978-3-642-19400-9_14

Manyika, J., Chui, M., Madgavkar, A., & Lund, S. (2016). *Technology, jobs, and the Future of Work*. https://doi.org/10.5089/9781484374979.001

McMath, C. F., Tamaru, R. S., & Rada, R. (1989). A graphical thesaurus-based information retrieval system. *International Journal of Man-Machine Studies*, *31*(2), 121–147. https://doi.org/10.1016/0020-7373(89)90024-2

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*.

Mikolov, T., Karafiát, M., Burget, L., Cernocky, J., & Khudanpur, S. (2010). Recurrent neural

network based language model. *Eleventh Annual Conference of the International Speech Communication Association*.

Mikolov, T., Yih, W., & Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. *Proceedings of NAACL-HLT 2013*, 746–751.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, *3*(4), 235–244. https://doi.org/10.1093/ijl/3.4.235

New Work SE. (2020). *Das Unternehmen*. https://www.new-work.se/de/ (visited on 06/18/2020).

Obermann, C. (2018). *Assessment Center* (6th ed.). Springer. https://doi.org/10.1007/978-3-658-18716-3

OECD. (2018). The Future of Education and Skills: Education 2030. In *OECD Education Working Papers*. https://doi.org/10.1111/j.1440-1827.2012.02814.x

OnetOnline. (2020). *Onet Dictionary*. https://www.onetonline.org (visited on 05/10/2020).

Park, S., & Kim, Y. (2016). Building thesaurus lexicon using dictionary-based approach for sentiment classification. *2016 IEEE/ACIS 14th International Conference on Software Engineering Research, Management and Applications, SERA 2016*, 39–44. https://doi.org/10.1109/SERA.2016.7516126

Partnership for 21st Century learning. (2015). *P21 Framework Definitions*. http://www.p21.org/our-work/p21-framework (visited on 05/10/2020).

Pennebaker, J. (2011). *The secret life of pronouns* (211th ed.). New Scientist.

Poonnawat, W., Pacharawongsakda, E., & Henchareonlert, N. (2017). Jobs analysis for business intelligence skills requirements in the aseasn region: A text mining study. *The Joint International Symposium on Artificial Intelligence and Natural Language Processing*, 187–195. https://doi.org/10.1007/978-3-319-94703-7

Propp, D. A., Glickman, S., & Uehara, D. T. (2003). ED Leadership Competency Matrix: An Administrative Management Tool. *American Journal of Emergency Medicine*, *21*(6), 483–486. https://doi.org/10.1016/S0735-6757(03)00164-5

Rajman, M., & Vesely, M. (2004). *From Text to Knowledge: Document Processing and Visualization: a Text Mining Approach*. Springer. https://doi.org/10.1007/978-3-540-45219-5_2

Rajput, N. K., Grover, B. A., & Rathi, V. K. (2020). Word frequency and sentiment analysis of twitter messages during Coronavirus pandemic. *arXiv preprint arXiv:2004.03925*.

Reverso-Softissimo. (2020). *DefProfessionalSkills*. https://dictionary.reverso.net/english-definition/professional+skills (visited on 05/07/2020).

Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Rosetti, K., & Langhoff, P. D. T. (2015). *Interne Potenziale*.

Roussos, L. A., DiBello, L. V., Stout, W., Hartz, S. M., Henson, R. A., & Templin, J. H. (2007). The fusion model skills diagnosis system. *Cognitive Diagnostic Assessment for Education: Theory and Applications*, 275–318. https://doi.org/10.1017/CBO9780511611186.010

Salloum, S. A., Al-Emran, M., & Shaalan, K. (2017). Mining Social Media Text: Extracting Knowledge from Facebook. *International Journal of Computing and Digital Systemss*, *6*(2), 73–81. https://doi.org/10.12785/ijcds/060203

Singh, A., Rose, C., Visweswariah, K., Vijil, E., & Kambhatla, N. (2010). PROSPECT: A system for screening candidates for recruitment. In *International Conference on Information and Knowledge Management, Proceedings*. https://doi.org/10.1145/1871437.1871523

SkillsYouNeed. (2020). *What are social skills?* Skillsyouneed.Com. https://www.skillsyouneed.com/ips/social-skills.html (visited on 04/06/2020).

Smit, S., Tacke, T., Lund, S., Manyika, J., & Thiel, L. (2020). *The future of work in Europe: Automation, workforce transitions, and the shifting geography of employment* (Issue June). https://doi.org/10.4324/9781351146609

Statista. (2020). *Percentage of students in the United States taking distance learning courses from 2012 to 2018*. https://www.statista.com/statistics/944245/student-distance-learning-enrollment-usa/ (visited on 06/10/2020).

Steptoe, A., & Wardle, J. (2017). Life skills, wealth, health, and wellbeing in later life. *Proceedings of the National Academy of Sciences of the United States of America*, *114*(17), 4354–4359. https://doi.org/10.1073/pnas.1616011114

Stieglitz, S., & Dang-Xuan, L. (2013). Emotions and information diffusion in social media - Sentiment of microblogs and sharing behavior. *Journal of Management Information Systems*, *29*(4), 217–248. https://doi.org/10.2753/MIS0742-1222290408

Taboada, M., Anthony, C., & Voll, K. (2006). Methods for creating semantic orientation dictionaries. *Proceedings of the 5th International Conference on Language Resources and Evaluation*, 427–432.

Townsend, A. M., DeMarie, S. M., & Hendrickson, A. R. (1998). Virtual teams: Technology and the workplace of the future. *Academy of Management*, *12*(3), 17–29. https://doi.org/10.5465/ame.1998.1109047

Translator. (2020). Microsoft.Com. https://azure.microsoft.com/en-us/services/cognitive-services/translator/#pricing (visited on 05/14/2020).

Tripathi, S., & Sarkhel, J. K. (2010). Approaches to machine translation. *Annals of Library and Information Studies*, *57*(4), 388–393.

Truxillo, D. M., Steiner, D. D., & Gilliland, S. W. (2004). The importance of organizational justice

in personnel selection: Defining when selection fairness really matters. *International Journal of Selection and Assessment*, *12*(1–2), 39–53. https://doi.org/10.1111/j.0965-075X.2004.00262.x

University of Nebraska-Lincoln. (2020). *Professional Skills*. University of Nebraska-Lincoln; University of Nebraska-Lincoln. https://www.unl.edu/gradstudies/current/development-/professional-skills-building-your-vita-0 (visited on 04/06/2020).

van Laar, E., van Deursen, A. J. A. M., van Dijk, J. A. G. M., & de Haan, J. (2017). The relation between 21st-century skills and digital skills: A systematic literature review. *Computers in Human Behavior*, *72*, 577–588. https://doi.org/10.1016/j.chb.2017.03.010

Waldman, D. A., & Korbar, T. (2004). Student Assessment Center Performance in the Prediction of Early Career Success. *Academy of Management Learning & Education*, *3*(2), 151–167.

Weiss, D. J. (2004). Computerized adaptive testing for effective and efficient measurement in counseling and education. *Measurement and Evaluation in Counseling and Development*, *37*(2), 70–84. https://doi.org/10.1080/07481756.2004.11909751

Were, M. C., Mamlin, B. W., Tierney, W. M., Wolfe, B., & Biondich, P. G. (2007). Concept dictionary creation and maintenance under resource constraints: lessons from the AM-PATH Medical Record System. In *AMIA Annual Symposium Proceedings* (Vol. 2007, p. 791). American Medical Informatics Association.

World Economic Forum. (2018). *The future of jobs report 2018*. Geneva: World economiic Forum. https://doi.org/10.1177/0891242417690604.

XING. (2020). *Facts and figures*. https://advertising.xing.com/facts-and-figures/ (visited on 06/18/2020).

Yin, T. (2017). *translate 3.5.0*. Pypi. https://pypi.org/project/translate/ (visited on 05/14/2020).

Zhang, J., & Mani, I. (2003). kNN approach to unbalanced data distributions: a case study involving information extraction. *Proceedings of Workshop on Learning from Imbalanced Datasets*, *126*.

Zhang, X., LeCun, Y., & Zhao, J. (2015). Character-level Convolutional Networks for Text Classification. *Advances in Neural Information Processing Systems*, 649–657. http://arxiv.org/abs/1502.01710

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit mit dem Titel

Mapping companies' readiness for the future: A novel data-driven approach to extract companies' future skills using social media analytics

selbständig angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ich bin mir bewusst, dass eine unwahre Erklärung rechtliche Folgen haben wird.

Ulm, 22.06.2020	Lennard Evertz
Ort, Datum	Vorname Nachname

Unterschrift