DEGREE PROJECT IN INFORMATION AND COMMUNICATION TECHNOLOGY,
FIRST CYCLE
STOCKHOLM, SWEDEN 2017

Matching Performance Metrics with Potential Candidates

A computer automated solution to recruiting

OSCAR MELIN

KTH ROYAL INSTITUTE OF TECHNOLOGY
INFORMATION AND COMMUNICATION TECHNOLOGY

Matching Performance Metrics with Potential Candidates

A computer automated solution to recruiting

Oscar Melin

2017-06-02

Bachelor's Thesis

Examiner
Gerald Q. Maguire Jr.

Academic adviser
Anders Västberg

KTH Royal Institute of Technology
School of Information and Communication Technology (ICT)
Department of Communication Systems
SE-100 44 Stockholm, Sweden

Abstract

Selecting the right candidate for a job can be a challenge. Moreover, there are significant costs associated with recruiting new talent. Thus there is a requirement for precision, accuracy, and neutrality from an organization when hiring a new employee. This thesis project focuses on the restaurant and hotel industry, an industrial sector that has traditionally used a haphazard set of recruiting methods. Unlike large corporations, restaurants cannot afford to hire dedicated recruiters. In addition, the primary medium used to find jobs and job seekers in this industry often obscure comparisons between relevant positions. The complex infrastructure of this industry requires a place where both recruiter and job seeker can access a standardized overview of the entire labor market.

Introducing automation in hiring aims to better address these complex demands and is becoming a common practice throughout other industries, especially with the help of internet based recruitment and pre-selection of candidates. These solutions also have the potential to minimize risks of human bias when screening candidates.

This thesis aims to minimize inefficiencies and errors associated with the existing manual recruitment screening process by addressing two main issues: the rate at which applicants can be screened and the quality of the resulting matches.

This thesis first discusses and analyzes related work in automated recruitment in order to propose a refined solution suitable for the target area. This solution – semantic matching of jobs and candidates - is subsequently evaluated and tested in partnership with Cheffle, a service industry networking company. The thesis concludes with suggestions for potential improvements to Cheffle´s current system and details the viability of recruiting with the assistance of an automated semantic matching application.

Keywords:

Automation, Recruitment, Semantic Matching, Service industry, Hotel, Restaurant

Sammanfattning

Att välja den rätta kandidaten för ett jobb kan vara en utmaning. Det finns dessutom betydliga kostnader i att rekrytera ny arbetskraft. På grund därav finns det ett behov för noggrannhet och neutralitet från en organisation vid rekrytering av ny personal. Detta examensprojekt fokuserar på restaurang och hotellbranschen. Denna branchsektor har traditionellt sett använt undermåliga rekryteringsmetoder. Till skillnad från stora företag så kan inte restauranger avvara resurser för egna rekryterare. Därtill så försvårar de primära medierna för rekrytering i sektorn jämförelser mellan relaterade lediga jobb. Denna komplexa infrastruktur skapar ett behov av en plats där både företag och arbetssökande har tillgång till en standardiserad översikt av hela arbetsmarknaden.

Introduktionen av automatisering har som syfte att bemöta dessa komplexa krav och blir alltmer vanligt inom andra branscher. Speciellt med hjälp av internetbaserad rekrytering och förval av jobbkandidater. Dessa lösningar har även potentialen att minimera risken för mänsklig subjektivitet och opartiskhet vid förval av jobbkandidater.

Detta examensprojekt har som syfte att minimera ineffektiviteter och fel samhörande med den nuvarande manuella rekryteringsmetoden genom att tackla två huvudproblem: takten i vilken förvalet av arbetssökande kan göras och kvaliteten av detta förval.

Detta examensprojekt inleder med en diskussion och analys av relaterade arbeten inom automatiserad rekrytering för att sedan presentera en möjlig lösning för det behandlade målområdet. Denna lösning – semantisk matchning av jobb och jobbsökande - är senare utvärderad och testad i samarbete med Cheffle, ett nätverksföretag inom serviceindustrin. Detta examensprojekt avslutar med lösningsförslag för potentiell förbättring till Cheffles nuvarande system och en slutsats om genomförbarheten av automatisering inom rekrytering.

Nyckelord:

Automatisering, Rekrytering, Semantisk matchning, Serviceindustri, Hotell, Restaurang

Acknowledgments

I would like to thank:

Professor Gerald Q. Maguire Jr. for providing valuable input and advice.

Emil Karlsson at Cheffle Handelsbolag for offering and supervising this Bachelor's thesis project.

A special thanks to Megan Henry for useful discussion and suggestions.

Stockholm, June 2017
Oscar Melin

Table of contents

List of Figures

List of Tables

List of acronyms and abbreviations

API	Application Programming Interface
BFS	Breadth First Search
CG	Conceptual Graph
DISCO	European Dictionary of Skills and Competences
GUI	Graphical User Interface
ICT	Information and Communication Technology
JSON	JavaScript Object Notation

1 Introduction

This Bachelor's thesis project was conducted during the spring of 2017 at Cheffle Handelsbolag.

This chapter gives an introduction to what the company's recruiting process usually looks like today and the problems within it. These problems will be addressed in this thesis. Additionally, the chapter includes a problem statement, the purpose of this thesis, and the methods that have been used. Chapter 2 gives more details concerning the specific sub area of matching jobs with candidates and vice versa.

1.1 Background

Selecting the right candidate for a job is a challenge. Selecting the right candidate from a massive pool of applicants can be an overwhelming task. Large corporations or recruiting agencies typically receive a very large number of applications [1] and even small businesses can receive hundreds of applications per job listing. In some countries such as Sweden, an unemployed person is required to apply for a certain number of jobs per month in order to be eligible for unemployment benefits [2]. For small businesses recruiting for low-skill positions, this may cause undue and heavy strain on their limited human resources, as sifting through so many applications is costly, error prone, tedious, and often simply overwhelming.

The screening of applications and resumes is followed by interviews, background checks, and offers of employment. However, a large fraction of applicants who pass all the aforementioned steps may nonetheless decline offers, further increasing recruitment costs [3]. Equally pressing, and significantly more financially burdensome; an employee may leave the company shortly after recruitment, necessitating a doubled investment in the hiring process. These potential issues create a demand for an effective screening process: one that selects only the most promising and suitable individuals for interviews and thus helps minimize employee churn rates.

Even organizations with sufficient human resources to do high capacity processing of candidates face setbacks due to psychological bias and general human error. Furthermore, discrimination stemming from inferred ethnicity and tendencies towards homogenous hiring can decrease the likelihood of an interview for qualified applicants [3, 4]. Narrow bracketing or the assessment of individual subsets in isolation can lead to results that differ from objective consideration of the whole set [5]. For instance, after giving five consecutive applicants high ratings, an interviewer might be reluctant to do the same for the sixth.

The significant costs associated with recruiting new talent require precision, accuracy, and neutrality from an organization's hiring sector. Automation in hiring intends to better address these demands and is becoming a common practice throughout larger companies with the help of internet based recruitment and

pre-selection of candidates [6]. In practice, this process entails matching available candidate information with an employer's requirements while weighing individual preferences from both sides in order to calculate a compatibility rating between both parties. Automated solutions certainly have the potential to minimize the risks of human biases when screening candidates.

The restaurant and hotel industry addressed in this thesis project, specifically this industry sector in Sweden, has a haphazard set of recruiting methods. Unlike large corporations, restaurants can not afford to hire dedicated recruiters. At some of the largest hotel chains in the country, recruiting is done by receptionists or department managers - employees whose primary focus has little to do with headhunting*. These employees lack fundamental knowledge about recruitment that is necessary for bringing on the best staff. Moreover, the primary medium used to find jobs and job seekers in the service industry is Facebook, with some additional recruiting done on popular job portals such as Monster†, Blocketjobb‡, and Platsbanken§. As discussed in [7, 8], the traditional recruitment process (discussed further in Section 2.1) has several shortcomings with regard to discovering and assessing candidates through social media, especially when it comes to candidates who lack formal education or experience. Furthermore, these information islands are isolated from one another, therefore obscuring comparisons between relevant positions. The complex infrastructure of this industry requires a place where both recruiter and job seeker can access a standardized overview of the entire labor market.

Cheffle seeks to provide this meeting point between job seekers and employers in the hotel industry. It functions as both job portal and hands on recruiting agency. The combination of these two functions means that by manually screening the applications and job ads via the job portal, suitable applicants can be recommended to the advertising employers. Access to an industry wide meeting point provides a broad spectrum of job seekers and the open positions represented on Cheffle offers ample opportunity for improved hiring practices through automation.

This thesis seeks to minimize inefficiencies and avoid the errors associated with the existing manual recruitment screening process by addressing two main issues: the rate at which applicants can be screened and the quality of the resulting matches.

---

* Claims about hiring practices within the restaurant industry were asserted in interview with Cheffle in March of 2017.

† https://www.monster.se/

‡ https://jobb.blocket.se/

§ https://www.arbetsformedlingen.se/For-arbetssokande/Lediga-jobb.html

1.2 Problem

In order to create an effective system that pairs eligible and interested employee candidates with open positions, a recruitment system must address the following three priorities: preprocessing, weighting, and candidate/position matching. Figure 1-1 shows the overall process of matching candidates with open vacancies (i.e., jobs).

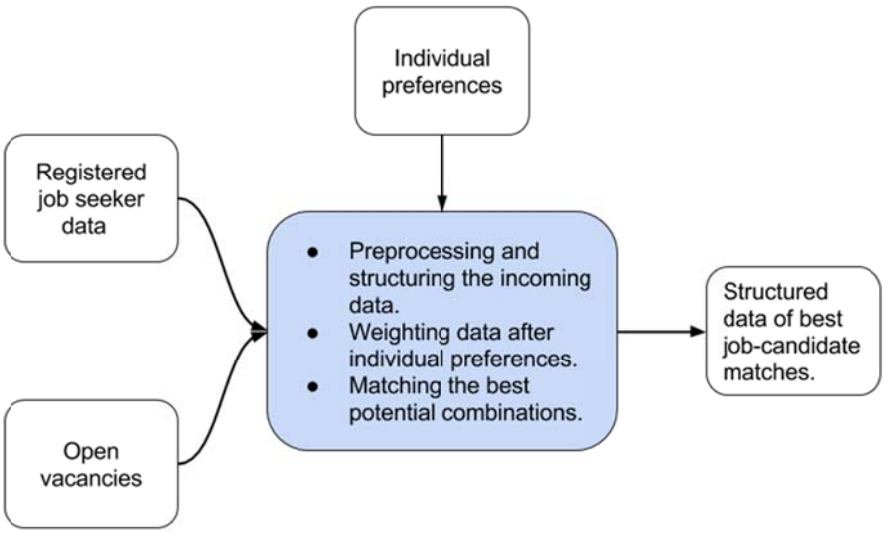

Figure 1-1: Overview of problem

1.2.1 **Preprocessing** data from proposed candidates and positions

The input data must be structured and represented according to a general set of rules. Dubious values, impossible data combinations and general issues of data quality must each be addressed before data can be subject to analysis [8]. As discussed in [9], users often leave some fields empty in an online form. However, missing values or small data sets impose significant constraints on the functionality of an automated system. Insufficient information on applicants or employment positions could potentially generate lower quality and/or unreliable matches.

1.2.2 **Weighting** the data with individual preferences

Employers will be able to clearly define their desirable candidate by providing the performance metrics they most value. These metrics could include differing values for preferences in education, skills, and experience. The algorithm applies the user's individual preferences to the available data and weights the data accordingly. Depending on the performance metric a recruiter is looking at, different variables may be more relevant; e.g. a recruiter might value years of experience above education or vice versa for a given position.

1.2.3 **Matching** the weighted and preprocessed data

With resources made available through the previous steps, it is possible to pair the best potential combinations of job seekers and open positions. Matching can be done either through manual analysis and reasoning or through a predefined automated algorithm.

1.2.4 Summary of problem

Letting the three aforementioned steps define the candidate/job matching process, this project attempts to address the following question: how can the automation process improve recruiting in terms of efficiency and quality?

1.3 Purpose

The purpose of this thesis and degree project is to research and develop a practical solution for job and candidate matching. This thesis seeks to provide job seekers with better opportunities for finding their most desired job, and to provide recruiters with a tool to assist them in finding better candidates for the positions they are aiming to fill.

This task cannot be accomplished without handling sensitive information, both personal and corporate. Dealing with confidential data always comes with a moral responsibility to neither purposefully nor mistakenly mishandle information.

Additionally, there is the issue of possible bad matches. A user will put a certain level of trust in the system will deliver on what it promises; hence, if the proposed solution produces incorrect job suggestions, it could lead to users missing out on a job they really want.

1.4 Goals

The goal of this project is to produce and analyze an automated job and candidate matching tool for Cheffle, in order to improve on their current manual job-candidate screening process. The following three sub-goals will define the direction, pace, and strategy for this project:

1. Understand the general practical needs of a job and candidate matching tool and the specific needs of Cheffle and its customers,
2. Translate these requirements into a working prototype using the available resources provided by Cheffle, and
3. Produce a result that satisfies the degree project requirements at KTH [10], Cheffle, and myself.

1.5 Research Methodology

The research methodologies used in this thesis project include:

- A literature study to gain the necessary fundamental understanding of the relevant topic areas and
- Iterative and continuous development, testing, and evaluation of the application.

1.6 Delimitations

This thesis is limited to matching sets of performance metrics in order to rank matches between jobs and candidates and does not include an implementation of data mining or parsing of resumes.

1.7 Structure of the thesis

Chapter 2 presents relevant background information about the context of which the solution is to be implemented as well as different theoretical matching concepts. Chapter 3 presents the methodology and method used to solve the problem. Chapter 4 presents the development, functionality, and implementation of the application. Chapter 5 presents results and analysis. Finally, chapter 6 presents conclusions and future work.

2 Background

This chapter provides basic background information about recruiting and methods that could be used for job-candidate matching. The main method described is *semantic matching*.

2.1 Recruitment

This section provides a general description of what a typical recruitment process might look like. Subsection 2.1.1 gives a overview of the process as a whole, Subsection 2.1.1.1 describes the pre-recruiting phase and planning, Subsection 2.1.1.2 describes the screening and selection process and finally Subsection 2.1.1.3 outlines how interviews and background checks of candidates are done.

2.1.1 What does a general recruitment process look like?

Recruiting a new employee is a multi-step process that can be summarized in three general phases. First, demand for a position is recognized and the job is published. Next, screening narrows the applicant pool to candidates that suit the position. Finally, interviews and background checks are conducted with suitable applicants [11]. Although the scope of this project is centered on the applicant screening process (step two), the first and third steps provide valuable context, hence they are briefly described.

A basic outline of the recruitment and selection hiring process consists of three initial steps. Each of these is described in a subsection below.

2.1.1.1 Vacancy identification and publishing

This first step lays foundation for the entire procedure. In this step (shown in Figure 2-1) the goals of hiring an employee and the competencies required of a candidate for this position are identified. A recruiting body develops a recruiting plan to hire the necessary employee(s).

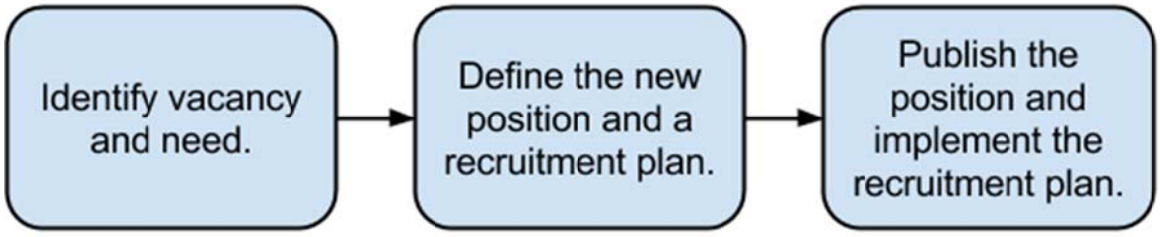

Figure 2-1: The vacancy identification and publishing step.

A well-constructed plan paired with accurately evaluated needs should provide the recruiting entity with the tools necessary to pinpoint the best possible candidate.

Additionally, in order to attract the most talented employee for the position and the most suitable employee for their team, it is crucial that the employer positively brands and effectively markets themselves to their potential candidates [12]. There will be nobody to hire if the company is not able to reach and attract qualified applicants.

2.1.1.2 Screening of applications

After the job listing has been published and possible candidates have submitted their applications, a screening entity must review them to produce a list of potential interviewees. This process is shown in Figure 2-2.

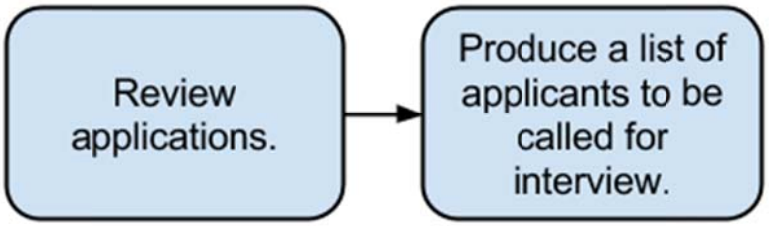

Figure 2-2: The screening of applications step.

In manual application reviews, the screening entity preferably consists of an assemblage of people that together possess the proficiency of a manager, a job specialist, and a potential future team member for the applicant [11]. The job of this entity is to evaluate candidates against a set of required skills and properties that were determined in the previous step. As these skills often overflow into related subjects, it is important to have a screening entity that is well informed about all relevant subjects. In addition to evaluating the applicant's technical skills, other crucial requirements such as team/company chemistry need to be factored in.

As applications tend to be very similar in their structure, the process of comparing one set of known structure (applications) with another (job positions) can productively lend itself to automation.

2.1.1.3 Interviews and background checks

The interview is arguably the most important part of the application process [11]. Meeting an applicant face to face is an indispensable component of the final recruitment decision [1]. This process is shown in Figure 2-3. The interview is at its heart a test of an applicant's technical skills. It requires an extensive investment of time and energy from the company to secure a close look at possible hires. To minimize excessive or wasted investment in unqualified applicants, pre-interview steps must be accurately and carefully conducted.

Figure 2-3: Describes the interviews and background checks step.

2.2 Matching two sets

Matching two sets or determining an arbitrary distance between them can be done in several ways. Most simply, the elements in each set may be compared and then scored for similarity by calculating the number of skills required for the job that the applicant possesses [13]. Formally expressed as:

$$similarity\ (Job, Applicant) = |Job \cap Applicant| \ / \ |Job|$$

The following example computes the similarities between a job and applicants in order to find the best match. Consider that we have one job with four applicants:

Job = {Woking, Japanese Food, Desserts}
Applicant1 = {Pizza, Italian Food, Desserts}
Applicant2 = {Woking, Chinese Food, Desserts}
Applicant3 = {Wine, Japanese Food, Chinese Food}
Applicant4 = {Sushi, Japanese Food, Woking}

The similarity values between the job and the applicants are then:

similarity(Job, Applicant1) = {Desserts} / {Woking, Japanese Food, Desserts} = 1/3
similarity(Job, Applicant2) = {Woking, Desserts} / {Woking, Japanese Food, Desserts} = 2/3
similarity(Job, Applicant3) = {Japanese Food} / {Woking, Japanese Food, Desserts} = 1/3
similarity(Job, Applicant4) = {Japanese Food, Desserts} / {Woking, Japanese Food, Woking} = 2/3

The result is that several applicants have the same similarity value, meaning that this method of matching is *insufficient* to distinguishing between the applicants. As the goal is to find the best possible match, more information is needed in order to avoid the problem of two or more applicants scoring the same value.

Other possible methods include representing sets in some hyperdimensional space where a Euclidean distance [14] could be derived (as shown in Figure 2-4). In ecological applications, statistical coefficients used to compare similarities and diversities between two sets, namely Sørensen-Dice or Jaccard indices, have proven useful [15]. However, people's skills and abilities can be more complex than statistical data sets and the gradation of these skills can make comparisons more difficult. The set elements that define skills and abilities might imply information

crucial to the success of a matching system, but could be lost if not predefined by the data configuration (i.e., left out of the data). Therefore processes must be carefully calibrated to address all possible semantic nuances within a given data set. For instance, if a reviewing body were to analyze the resume of someone experienced in only theater, they could rather safely assume that this applicant is probably better suited for the role of "public speaker" than someone with experience in only C++. This is not to say that someone well versed in C++ cannot also be a skilled public speaker, it is an assumption based on predefined relationships between skills stemmed from the fact that public speaking is a component of theater. These grey area relations between domains of properties must be carefully considered when matching skills and people to avoid ignoring a potentially great match.

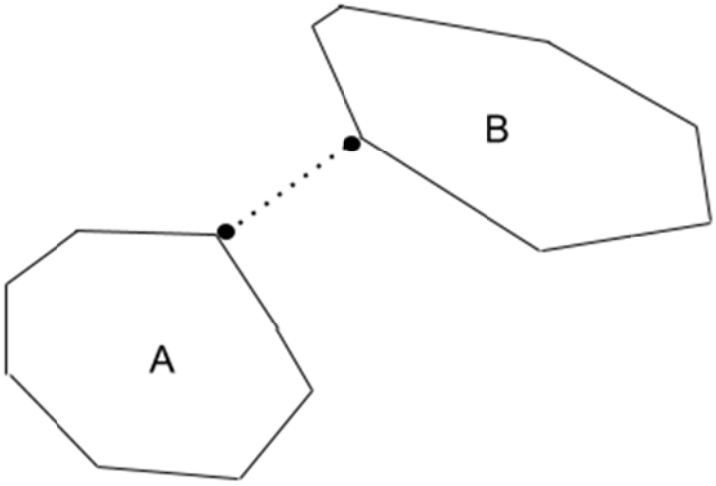

Figure 2-4: Euclidian distance between polyhedral A and B

2.2.1 Matching skills and properties of people

The extrapolation and fuzzy meaning of terms discussed above asserts that there is a need for clear and defined relationships between comparable elements. Resources to help define schemes of classifications already exist; among them: The European Dictionary of Skills and Competences (DISCO) [16], Standard Occupational Classification System [17], and International Standard Classification of Occupation [18]. These taxonomies classify occupational categories and identify their relations to one another. Relationships between these terms are then used to determine how precisely they correlate. For the purposes of this thesis, Figure 2-5 illustrates an example of the application of the Swedish version of DISCO* to the terms "fransk matlagning" (English: French cuisine) and "grillning" (English: grilling).

* http://disco-tools.eu/disco2_portal/terms.php

Conclusions are drawn from the relevancy between terms to determine an applicant's probability of successfully matching a certain job.

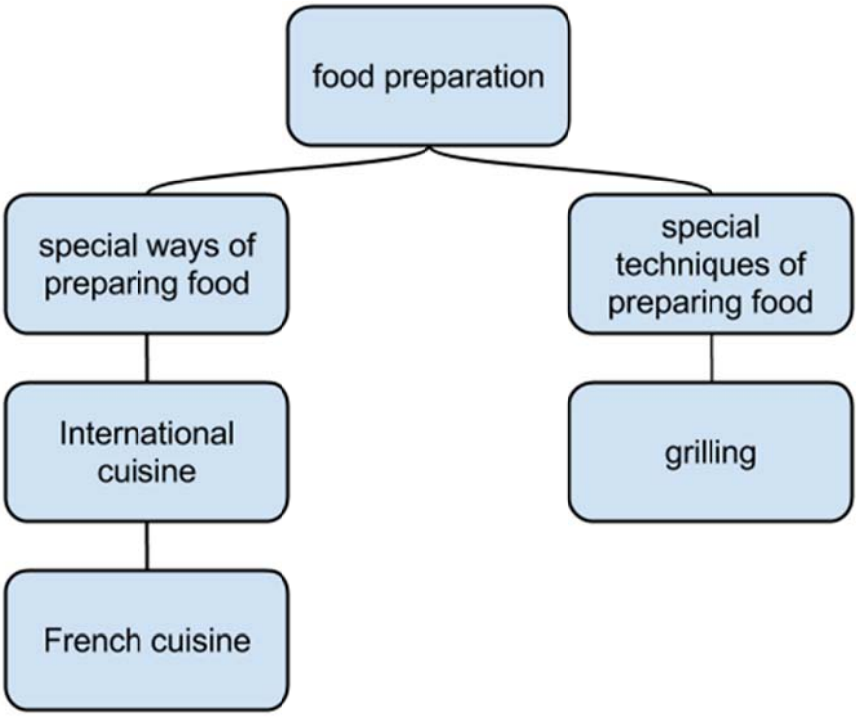

Figure 2-5: The connection between "French cuisine" and "grilling" in DISCO*

2.3 Semantic Matching

Semantic matching is a technique used to identify relationships between data sets using predefined and controlled vocabularies, assuming all data can generally be represented as a graph [19]. To better derive information not explicitly stated in the elements, but rather in related and/or implied data, information is semantically matched to yield a more accurate depiction of an applicant's qualifications. Suppose two partially ordered sets or graph like data structures are given; semantic matching compares and identifies nodes in both graphs that are semantically similar to each other [20]. When applied to matching candidates and jobs, two semantically equivalent terms might be "maître d" and "waiter" - if they are synonymous in the context of the application. In accordance with applied taxonomies and ontologies, semantically similar terms will be evaluated as good matches, whereas queries such as "Why is flying expensive" and "Why is gold expensive" from Table 2-1 will not, although three out of four words are identical, the major semantic difference between "flying" and "gold" significantly negates any match between the two queries.

* http://disco-tools.eu/disco2_portal/terms.php

Table 2-1: Examples of terminology match versus semantic match

		Term match	Semantic match
Chef Stockholm	Chefs Stockholm	partial	yes
pool	billiards	no	yes
Why is flying expensive	Why is gold expensive	partial	no
Java	C#	no	partial

2.3.1 Structural Overview of this project

The basic outline for this project is illustrated in Figure 2-6. The blue shaded boxes describe the semantic matching process, the white boxes describe pre-existing data, and the green boxes represent the anticipated product.

As depicted in Figure 2-6, the semantic matching process utilized for this project will incorporate both concept taxonomy and relational hierarchy in order to better analyze and address the lexicon variability of users. The concept taxonomy acts as a thesaurus and provides the possibility to match words with similar meanings. The purpose of a relational hierarchy is to provide a structure describing the similarity between terms. When both are combined in a conceptual graph (CG), distance between two concepts in the CG is a measure of the similarity between them.

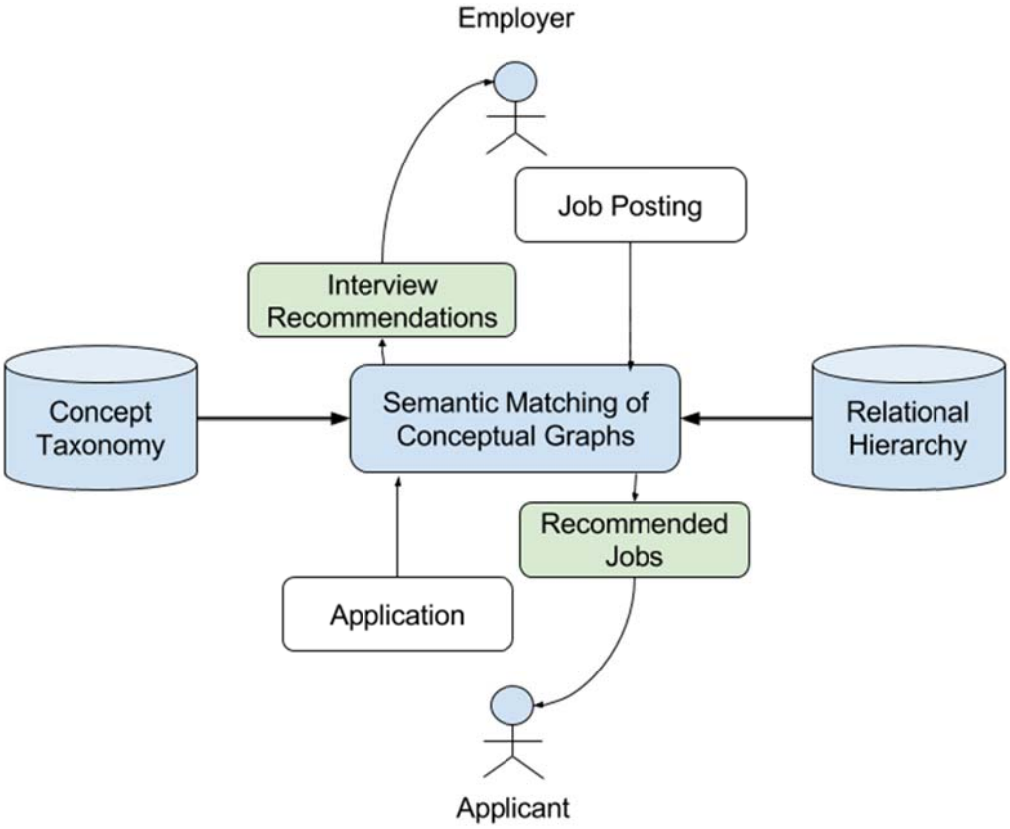

Figure 2-6: Overview of the project structure

2.3.2 Semantic Matching of Conceptual Graphs

The approach used in this project is based on ideas from [1, 21, 22]. The similarity between two concepts, c_1 and c_2, in a conceptual graph [23] is derived from the distance between them (denoted as $d_c(c_1, c_2)$). Using this distance, the similarity between c_1 and c_2 is defined as:

$$sim_c(c_1, c_2) = 1 - d_c(c_1, c_2) \quad [1]$$

Additionally, every node in a CG is assigned a milestone value:

$$milestone(n) = (1 / 2) / k^{l(n)}$$

Where k is a predefined factor (such as $k>1$) that indicates the rate at which the value decreases along the hierarchy and $l(n)$ is the depth of node n in the graph where $l(root) = 0$. The numerator is set to ½ so that $d_c(c_1, c_2) = 1$, if c_1 and c_2 are nodes at the deepest level with the root as their closest common parent (c_{cp}). The milestone value indicates degree of differentiation as one proceeds down the graph.

Due to the fact that the shortest path between two nodes in a hierarchical graph will go through their common parent, c_{cp}, the distance between the two nodes will be calculated by their milestones and their c_{cp} as follows:

$$d_c (c_1, c_2) = d_c(c_1, c_{cp}) + d_c(c_2, c_{cp})$$
$$d_c(c, c_{cp}) = milestone(c_{cp}) - milestone(c).$$

This model is designed under the assumption that the semantic difference between higher level concepts is greater than the difference between lower level concepts. That is, generalized concepts will differ more from each other than specialized ones. This model also implies that the semantic difference between "siblings" is larger than the difference between "parent" and "child" concepts, as the distance is calculated through their closest common parent [21].

For example, to find the distance between the two skills "Pastry Baking" and "Cold Smoking", begin with the ontology shown in Figure 2-7.

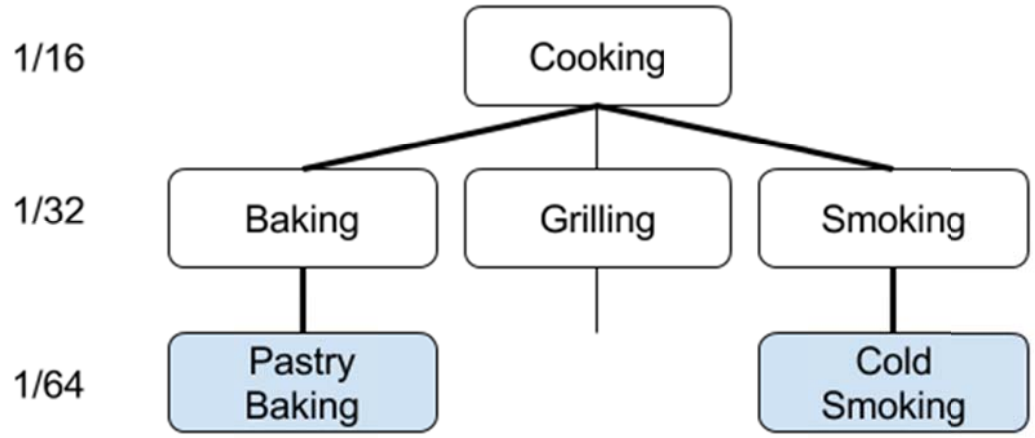

Figure 2-7: Segment of a possible CG with corresponding milestone values

By first identifying their closest common parent, "Cooking", and using $k = 2$, the distance between the two concepts can be calculated as follows:

$$c_1 = \text{Pastry Baking}$$
$$c_2 = \text{Cold Smoking}$$
$$d_c(c_1, c_2) = d_c(c_1, Cooking) + d_c(c_2, Cooking)$$
$$= (1/16 - 1/32) + (1/16 - 1/32)$$
$$= 0.0625$$

The similarity between the two concepts is then:

$$sim_c(c_1, c_2) = 1 - 0.0625 = 0.9375$$

This similarity can be compared to a reference similarity threshold defined by the user, then only results with a $sim_c(c_1, c_2)$ value above the user's defined threshold will appear. If too many or too few results are presented (i.e.,), then the user may dynamically adjust the value of their threshold. For example, increasing $sim_c(c_1, c_2)$ to increase match precision or lowering it.

2.4 Implementation

This subsection describes the methods and technologies needed to implement the previously described concepts.

2.4.1 Finding shortest path in a graph.

In order to find the distance between two concepts according to the formula given in Section 2.3.2, the shortest path between them must first be established.

Dijkstra's algorithm [24] is probably the first solution that comes to mind when trying to find the shortest path between nodes in a graph. It finds the shortest path in a weighted graph (containing only positive edges) between a given node and every other node by extending the best path found so far. Dijkstra's algorithm works even for unweighted graphs, such as the one used in this thesis project, however it will not be the most efficient solution. Even when all edge weights are identical, the algorithm will spend time unnecessary looking for alternative paths throughout the entire graph.

To take advantage of the fact that the graph is unweighted, we use the Breadth-First Search (BFS) method* as it produces a more efficient solution. BFS traverses the graph breadth wise from the source node and when first coming to any node *v*, it will have done so by the guaranteed shortest path, i.e., the path with the lowest number of edges between the source node and *v*. As Dijkstra's algorithm has the time complexity of $O(V^2)$ and BFS has $O(E + V \log V)$ [25], hence BFS is theoretically the best solution for this application.

* This method is well described in [25].

3 Methodology

The purpose of this chapter is to provide an overview of the research method used in this thesis. Section 3.1 describes the research process. Section 3.2 focuses on the data collection techniques used. Section 3.3 describes the experimental design. Section 3.4 explains the techniques used to evaluate the reliability and validity of the data collected.

3.1 Research Process

The research process was divided into three phases. Phase 1 represents the pre-study phase, whereas Phases 2-3 concerned developing and evaluating the application. Figure 3-1 shows the general timeline of this thesis project.

3.1.1 Phase 1: Information gathering and Literature study phase

The first two weeks were to be spent preparing for the practical and analytical parts of the project. This included a literature review of theoretical and practical contributions within those topics closely related to this project in order to discover where there was room for improvement and to develop a theoretical foundation for the following phases.

3.1.2 Phase 2: Developing the application

This phase is divided into three sub phases:

1. Plan the development process and research technologies to be used when developing the application.
2. Build the application piece by piece in conjunction with weekly meetings with Cheffle to ensure that development proceeds in a direction that satisfies all parties.
3. Test, analyze, and refactor each piece of code to ensure functionality and scalability.

3.1.3 Phase 3: Evaluation and Analysis

In this phase, the entire application is evaluated and analyzed to determine how well it performs and to find any missing functionality. A performance evaluation was conducted regularly together with Cheffle to determine if any additions or improvements needed to be made.

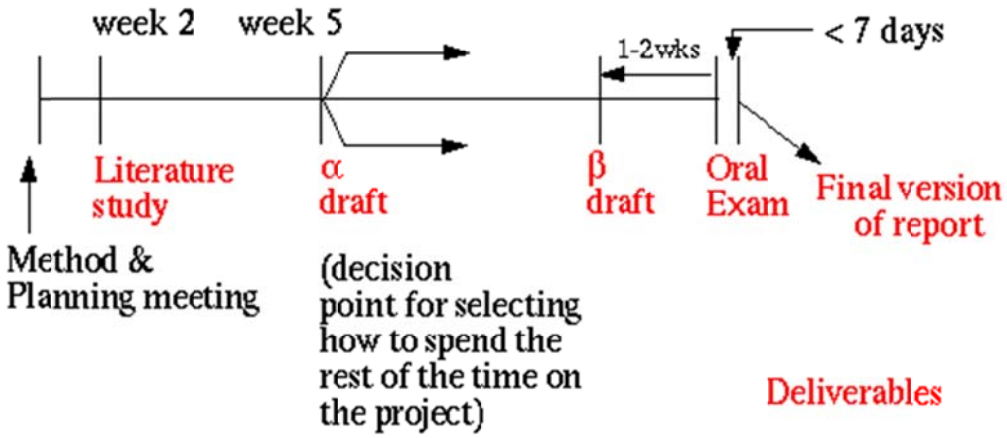

Figure 3-1: Rough timeline of the thesis project*. (Figure appears here courtesy of G. Q. Maguire Jr.)

3.2 Data Collection

Data collection for this project was done during two different periods. Initially, all of the available relevant data from registered Cheffle users was collected and analyzed for the purpose of adjusting the taxonomies mentioned in Chapter 2. The ontological elements from the user data as well as relevant elements in the taxonomies make up the structure of the main ontology. Secondly, feedback from industry insiders on the final application will provide data on which the final application's performance will be evaluated.

3.2.1 Sampling

Two different kinds of sampling will be done for this project. Subsection 3.2.1.1 describes gathering data for the relational hierarchy and Subsection 3.2.1.2 describes gathering feedback to aide in the final evaluation.

3.2.1.1 Gathering data for the relational hierarchy of terms

The relational hierarchy/ontology is the foundation on which all matching between performance metrics are calculated. Consequently, it is crucial that the ontology is created as accurately as possible. To achieve this, data was collected and analyzed in order to create a set of elements that populate the ontology.

* https://people.kth.se/~maguire/b-exjobb-time-line-20070906a.gif

Three methods were to be used to identify which terms the ontology would require:

1. Frequency and relevancy of the term's usage in job ads and job profiles
2. Frequency and relevancy of the term's usage in international skills and competences classifications.
3. Analysis of user needs

Frequency and relevancy of a term's usage in job ads and job profiles will be determined through sampling of Cheffle user profile data. The user profile database contains 4 columns of data relevant to this task. They are:

Education	A list of schools the candidate has attended and a list of any specific qualifications.
Work experience	
Title	The candidate's work title, e.g. "Chef" or "Waitress"
Skills	Any combination of 74 predefined skills selected when creating a profile, e.g. "Sushi making" and "Pastry baking"

Due to the fact that only one of the 4 columns in each user row is formatted according to a controlled vocabulary, while the rest are free text, problems will occur when parsing and analyzing this data. Lexical differences between "Bartender/Waitress", "bartender and waitress", and "Hi, I am someone who bartends" need to be taken into account for a practical solution. As the scale of this project does not encompass dealing with the full range of possible nuances, within the project I have only used those entries containing clear and readable data; this is believed to be sufficient to demonstrate the relevant concepts.

3.2.1.2 Gathering feedback from industry insiders

The performance of the finished application will be evaluated based upon feedback from Cheffle employees.

3.2.2 Sample Size

The sample size for the data described in Section 3.2.1.1 is 400 users with registered job profiles. As for the evaluation of the project (Section 3.2.1.2), a minimal number of people will be ask to provide feedback – sufficient to provide some feedback for this proof of concept prototype.

3.2.3 Target population

The target population is Cheffle's registered users. Due to ethical and legal responsibilities and restrictions, such as PUL [26] and the Cheffle terms of service*, all of the user data is stripped of personal information and does *not* contain the applicant's name, sex, or age.

*https://cheffle.se/integritetspolicy/

3.3 Experimental design/Planned Measurements

This thesis project is evaluated through a comparison with other matching methods and tested with real world data in conjunction with Cheffle. Section 3.3.1 describes the test environment, Section 3.3.2 describes the software and data structures to be used. Finally, Section 3.3.3 describes the way in which python and JSON are used in the implementation.

3.3.1 Test Environment

All software and data models used for this thesis project are cross platform compatible, hence they should work on Microsoft's Windows, Apple's OSX, and GNU/Linux platforms. Nevertheless, using Ubuntu 16.04 Linux is recommended as it is the only platform on which this project has been extensively tested.

3.3.2 Software and data structures to be used

This thesis project will use Python 2.7 [27] for the application and JSON [28] for representing any data. The main parts of the application are shown in Figure 3-2.

The concept taxonomy is the main structure describing how different skills and performance metrics relate to each other and is represented in JSON format in a separate file. Each element in the ontology will contain a title, a link to the term in the relational hierarchy, and a list of JSON objects containing elements further down the ontology. The relational hierarchy acts as a thesaurus with the purpose of limiting the need for redundant elements with the same meaning in the ontology.

The advantages of using JSON for representing the relationships between skills are:

- Visual representation. The structure of JSON makes it easy to view and intuitive to edit, even for very large files, especially when assisted by visual aids such as jsonviewer [29].
- There is a well-documented built-in library [30] in Python for supporting encoding and decoding of JSON.

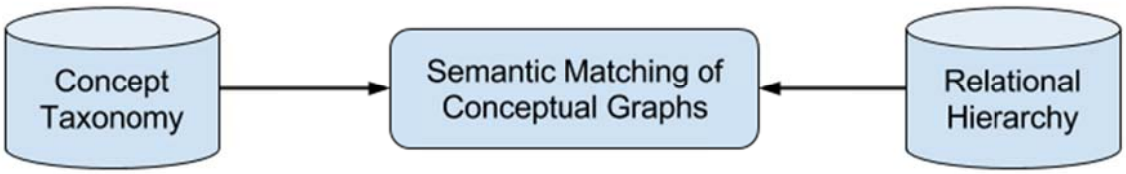

Figure 3-2: Main parts of the application

3.4 Assessing reliability and validity of the data collected

This section explains the reliability and validity of the collected data. Section 3.4.1 explains reliability and Section 3.4.2 explains validity.

3.4.1 Reliability

The information gathered from Cheffle's employees is mostly based on their professional experience and understanding of skills and competences relevant to the hotel and restaurant industry. However, only a small number of people participated in the evaluation of the final product. This means that the feedback given is dependent upon the personal experience of each individual and that the reliability of this feedback is based on qualitative sources and does not in and of itself provide quantitative results.

3.4.2 Validity

The information gathered from Cheffle's employees is presumed valid as it is based on their personal feedback regarding the potential impact of the prototype produced in this thesis project on their work, in which they possess extensive experience and knowledge.

4 The Application

The purpose of this chapter is to describe the application's development process and the final prototype as well as its functionality. Section 4.1 describes the design and development of the application. Section 4.2 describes the process of creating the ontology. Finally, Section 4.3 describes possible functionality and discusses different implementations of the application.

4.1 Design

This section describes the design process and design decisions made during the development of the prototype.

4.1.1 Python

Python was chosen as the programming language for several reasons. First and foremost, Python is platform independent; hence it can run on a wide variety of platforms [31] including Microsoft's Windows, Apple's Mac OS X, and Linux operating systems, such as the Debian based Ubuntu. Secondly, this project was written to be implemented behind a web interface which runs a Python-Flask [32] backend. These reasons, especially the ability of simple integration with a Python written web framework, motivated the choice of Python as the programming language for the entire project.

4.1.2 Description of the application

This subsection describes the different modules that make up the application and what they do. The application consists of three modules: graph.py, node.py, and main.py.

4.1.2.1 Node module

The Node module represents an element in the ontology and contains 4 different attributes as depicted in Figure 4-1.

The "title" attribute describes the skill or performance metric that each individual node represents and additionally works as a unique identifier for the object. The purpose of the "meta" field is to hold metadata used when pairing a node with a search term. As there will not be enough nodes to cover every possible search term, information in the metadata field describes each element in more depth than is possible using only a single "title" field.

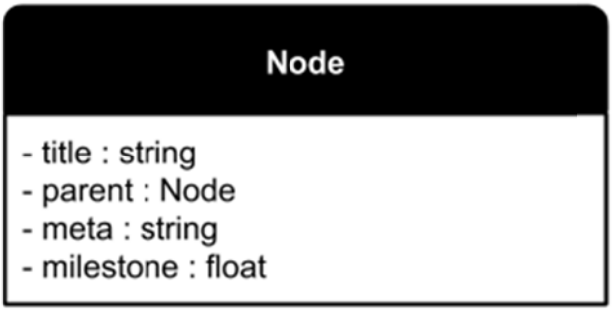

Figure 4-1: Node class and its attributes

4.1.2.2 Graph module and calculating distance between terms

The purpose of the "Graph" module is to realize a custom graph data structure in the form of a Python dictionary of sets, i.e., an associative array made up of *key : value* pairs, where all keys must be unique. *Keys* represent each node in the graph and the *values* correspond to each node's children. Safeguards against adding nodes that lead to cycles or duplicates exist to ensure that the graph is structured correctly. In addition to realizing a graph data structure, the "Graph" module also supplies several class methods for maintaining and analyzing the graph as outlined in Figure 4-2.

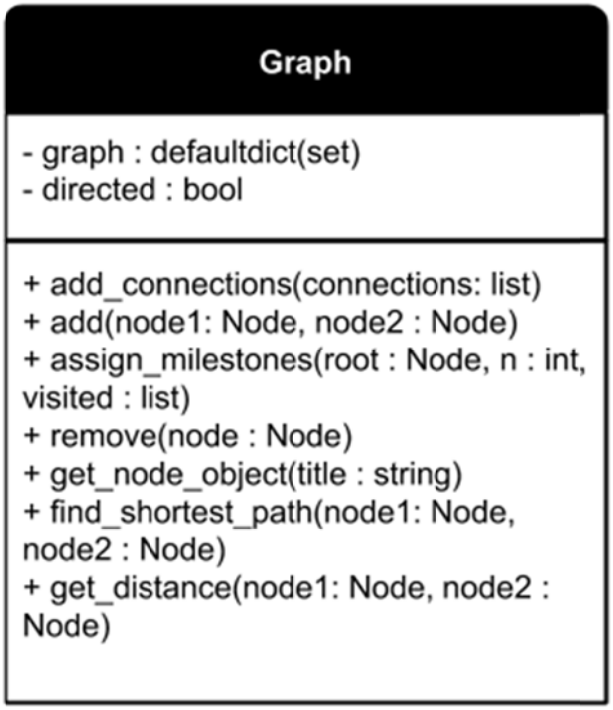

Figure 4-2: Graph class and its attributes

The most important method of this application resides in the Graph module under the name get_distance(node1 : Node, node2 : Node). The purpose of this method is to calculate the distance(c1, c2) between two concept nodes, c1 and c2, in the graph according to the formula given in Section 2.3.2. Two "Node" objects are required as input to produce as output a number between 0 and 1. An output of 0 is returned if the two "Node" objects are identical, while 1 indicates that the inputs are nodes in the deepest level of the tree with the root node as their closest common parent.

To calculate the shortest path, get_distance works in two steps: first, a call to find_shortest_path finds the shortest path between node1 and node2 using BFS. Second, the milestone values of the nodes along the path are used to calculate the distance.

4.1.2.3 Main module and building the graph

The Main module, defined as shown in Figure 4-3, ties everything together. All job and candidate data is provided as a database dump into a text file. Direct access to this database is restricted for privacy reasons. The file was filtered to remove any personally identifiable information (as described in Section 3.2.3). When initializing the application, *Main* parses the filtered database into candidate objects and the ontology is loaded from JSON into a Graph-object. The Main module uses this data and provides an API with methods to match skills as well as maintain and analyze user and ontology information.

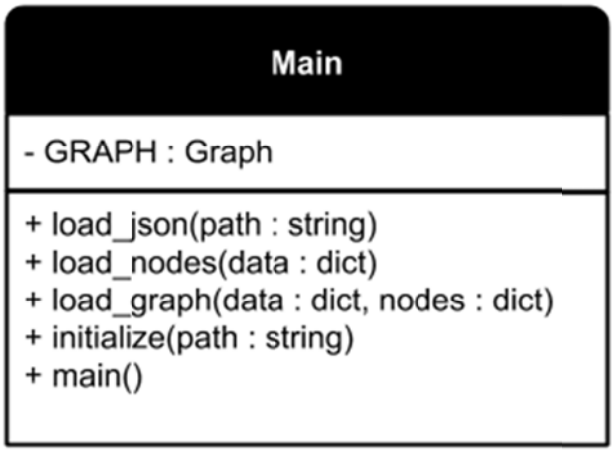

Figure 4-3: Main class and its attributes

4.2 Building the ontology

The ontology defines how different performance metrics relate to each other and is the base on which this application stands. This section describes how the ontology was built and what problems occurred along the way.

4.2.1 Fetching ontological elements

The purpose of the ontology is to provide relationships between concepts used for describing candidates and job postings. The best place to look for concepts to populate the ontology is in real world user data and comprehensive international skills repositories, such as DISCO [16].

In order to apply for a job on the Cheffle platform, a user must first register a profile. To do so, the user provides information about themselves including: title, skills, experience, and education. The elements used to populate the ontology were collected from these fields for a total of 476 registered user profiles. Sections 4.2.1.1-4.2.1.3 gives a short rundown of these 4 different fields.

4.2.1.1 Title

The title element is perhaps the most crucial when defining a candidate. It is important that any potentially occurring titles are represented in the ontology. Table 4-1 shows the most frequently used titles in the job profiles. Note that the titles are given in Swedish as this is the language that most users used to enter the data.

Table 4-1: Most frequent skills

Title	Frequency
Kock	64
Servitris	36
Bartender	21
Student	18
Servitör	17
Hovmästare	7
Kökschef	7
Restaurangchef	6

4.2.1.2 Skills

The skills field is the only field in which the entries come from a predefined vocabulary consisting of 74 different terms of which zero or more can be chosen. Additionally, when submitting a job posting, these terms can be used to describe the skillset required for the position. Figure 4-4 shows how skills can be added to either a job profile or a job posting.

This thesis project initially used only *predefined* skills for ontology building and searching. However, later when doing user data analysis, it was discovered that few

users actually entered anything in the skills field. This is far less data than necessary for usability as the sole resource when searching. Thus, a decision was made to expand the scope of the search to find performance metrics in other fields (specifically, Title, Education, and Experience).

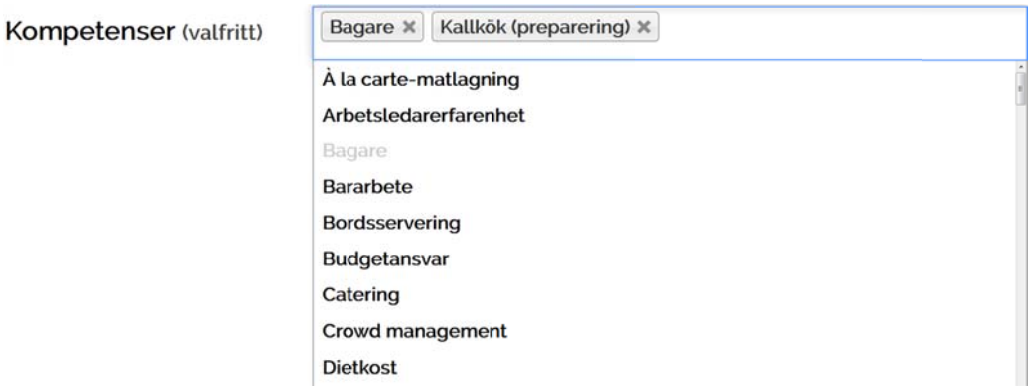

Figure 4-4: Adding skills to a job profile or job ad on Cheffle.

4.2.1.3 Education and Experience

The education and experience sections both follow the same format. They can contain zero or more entries with each entry having 4 sections: Employer/School, Title/Qualifications, Date, and Notes. See Table 4-2 for an example. For the sake of finding performance metrics to populate the ontology only the Title/Qualifications field was considered. This field proved to be the most useful and was easily parsed. All 4 sections are free text. Parsing information in the "Notes" section (which contained long and sometimes unclear text) proved too cumbersome and time consuming to be included within the scope of this thesis project.

Table 4-2: Example of an Education or Experience entry in a job profile

Employer/School	EBS (European Bartender School)
Title/Qualifications	Bartender
Date	Augusti 2013
Notes	Högklassig grundutbildning i allt som ingår i baryrket. Teorikunskaper om sprit och dess historia, drinkrecept, barträning, preppning, bar vett och etikett, flairing, free pouring etc.

4.2.2 Structuring the retrieved data

Collecting all of the terms gathered from the user database and filtering out duplicates and irrelevant or meaningless data, resulted in a list of 109 unique

terms. To build the ontology, these terms needed to be structured into a relational hierarchy that modeled the relationships between terms after their use within the industry. To accomplish this, two different tools were utilized: DISCO [16] and input from Cheffle employees. The purpose of this combined method was to utilize the DISCO definitions and to create an ontology based on international labor market standards which could be refined to suit Cheffle's specific needs. The main structure of the resulting ontology is laid out as shown in Figure 4-5.

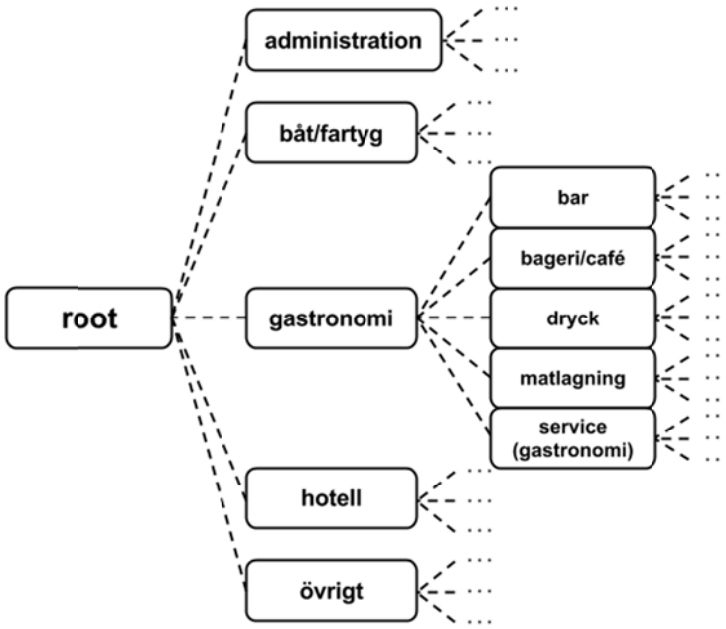

Figure 4-5: Overview of the ontology used in the application, showing 10 out 109 total elements

4.3 Testing

A short experiment was conducted to investigate the success of the matching algorithms performance in comparison with a human. Three items (one question item and two response items) were randomly selected from the ontology. Each human participant was asked to identify the most closely related response (the second or third item) to the addressed question (the first item). Each of the three participants completed 50 unique comparisons. Figure 4-6 shows the GUI of the program that was created to conduct the testing.

Figure 4-6: Test program graphical interface

The results of this experiment are summarized in Table 4-3. Two Cheffle employees and one independent recruiter participated in the test. "Mean of correct/incorrect similarities" refers to the mean of all answers chosen either in or not in accordance with the underlying matching algorithm.

Table 4-3: Comparison of the different resulting values between matching methods. Simple matching refers to the example given in section 2.2

Test Person	# Correct	# Incorrect	Mean of correct similarities.	Mean of incorrect similarities.
Cheffle 1	35	15	0.72834	0.41041
Cheffle 2	33	17	0.71993	0.47380
Recruiter	35	15	0.69531	0.30833
Average	**34.33**	**15.67**	**0.71453**	**0.39751**

4.4 Functionality and Implementation

This section describes the prototype's functionality and discusses different potential implementations of the application.

4.4.1 Functionality

The main function of the application is to rank match quality between two sets of performance metrics based on an underlying relational hierarchy, commonly referred to in this report as "ontology". Similarity between two single metrics is calculated with the formula described in Section 2.3.2 and between two sets of metrics by averaging the similarity of each term in the first set to the term in the second set with the highest respective similarity. Table 4-4 shows a comparison

between the example matching method from Section 2.2 and the new semantic method using the same parameters.

Table 4-4: Comparison of the different resulting values between matching methods. Simple matching refers to the example given in Section 2.2

Applicant	Simple matching		Semantic method	
	Similarity	Position	Similarity	Position
Applicant1	0.33	3, 4	0.9375	3
Applicant2	0.66	1, 2	0.9896	1
Applicant3	0.33	3, 4	0.9323	4
Applicant4	0.66	1, 2	0.9427	2

As shown in Table 4-4, the issue of simple matching being unable to sufficiently distinguish between applications does not exist for the same example job and applicants when using the semantic method. One more feature of the matching function is the possibility of prioritizing certain metrics above others. Following the same example, we could choose to value “Japanese Food” higher by adding it a second time in the set of requested performance metrics. I.e. Job = {Woking, Japanese Food, Desserts, Japanese Food}. Table 4-5 shows the new matching values after weighting.

Table 4-5: Comparison between unweighted and weighted for 2x “Japanese Food” in the Job set of performance metrics from the previous example.

Applicant	Unweighted		2x Japanese Food	
	Similarity	Position	Similarity	Position
Applicant1	0.9375	3	0.9296	4
Applicant2	0.9896	1	0.9843	1
Applicant3	0.9323	4	0.9492	3
Applicant4	0.9427	2	0.9570	2

Additionally, the matching method is a non-commutative operation, meaning that similarity (Set 1, Set 2) is not the same as similarity(Set 2, Set 1). Table 4-6

shows an example of the difference in matching score when swapping the order of a match. This occurs because Set 1 is intended to describe the set of required metrics while Set 2 represents the set to be evaluated according to the requirements.

Table 4-6: Shows an example of the non-commutativity in matching

Set 1	Set 2	Matching score
{Bartender}	{Bartender, Server}	1
{Bartender, Server}	{Bartender}	0.796875

4.4.2 Implementation

This subsection describes potential implementations of the prototype. Section 4.3.3.1 describes the planned implementation at Cheffle. Section 4.3.3.2 describes possible implementations for application in other areas.

This application is intended to serve two purposes at Cheffle. Namely:

1. The ability for companies buying job ads to instantly receive a list of candidates with high matching skillsets.
2. The ability to provide a better service for job seekers by providing a more accurate search function as well as automated suggestions/notifications of open positions that fit both the candidates' preferences and qualifications.

For example, if a posted job ad required the skillset {"restaurangchef", "hovmästare", "bordsservering", "vinkunskap"}, the best matching candidates from the database of registered users would be as shown in Table 4-7.

Table 4-7: Result of requesting the top 5 matches to {"restaurangchef", "hovmästare", "bordsservering", "vinkunskap"} of all users in the Cheffle database.

	Title	Summary of skills, education and experience.	Matching score
1	Restaurangchef	{”hovmästare”, ”restaurangchef”, ”kock”, ”sommelier”, ”servitör”}	0.9609375
2	Restaurangchef / hovmästare / barmästare	{”bar”, ”runner”, ”hovmästare”, ”restaurangchef”, ”bartender”, 'servitör}	0.9140625
3	Restaurangchef	{”administration”, ”restaurangchef”, ”restaurang”, ”hovmästare”, ”kock”, 'kökschef}	0.875
4	Servitör/Hovmästare	{”bartender”, ”restaurangchef”, ”servitör”}	0.859375
5	Caféansvarig, Servitris och Bartender	{”bartender”, ”marknadsföring”, ”administration”, ”receptionist”, ”servitris”}	0.828125

4.4.2.1 Other uses

As the scope of matching skills is only limited by the scope of the ontology, this application could easily serve other industries if provided with the corresponding industry specific ontology or if the of current ontology were expanded. However, in the case of expansion, possible application wide scaling performance issues due to using a much larger ontology might need to be addressed.

5 Results and Analysis

In this chapter, the results are presented and discussed.

5.1 Major results

The major results of this thesis project are divided into two parts. Subsection 5.1.1 describes the applications ability to derive additional knowledge from provided concepts and Subsection 5.1.2 discusses results from the user test.

5.1.1 Ability to access additional knowledge

One of the main issues this thesis project sought to address was the insufficient capability of distinguishing between two sets of performance metrics, as described in Section 2.2. This insufficiency resulted in different applications that could too easily score the same similarity value when compared to a set of required metrics. As the goal was to find the best possible match, more information was needed in order to avoid such occurrences.

To solve this, Chapter 4 introduced specialized relations among performance metrics, thus forming a hierarchy between the different concepts. The application could then use the additional knowledge derived from these relations in order to find the most suitable match between two sets of performance metrics with higher precision than the simple matching method described in Section 2.2. As shown in Table 4-4, the issue of insufficient differentiation between applications due to simple matching does not exist for the same example job and applicants when using the new proposed semantic method.

5.1.2 User tests

In order to create a system able to pair jobs with potential candidates, it is of paramount importance that the matching algorithm is able to accurately match the performance metrics required for a job with those representing each candidate. In order to measure this, a test was created to compare matching done by the application against matching done by qualified recruiters.

At first glance, the test results shown in Section 4.3 seem poor in comparison with the matching ability of the application. Only 69% of the best matches chosen by the test participants aligned with those of the application. However, all performance metrics used in the tests were chosen at random, which produced some difficult questions. For example, one question might be: Which of the following is closest related to “Indian food (cooking)”: “Receptionist” or “Cleaner (Hotel)”? Questions like this one are a dice throw for the human participant and would seldom be relevant in a real world scenario, as the matching algorithm would rank both the options so low that they would be considered irrelevant.

Table 5-1: Shows the relationship of similarity span (similarity of correct answers with posed performance metrics) to frequency of agreement (between test participant and matching method).

Similarity span	# Correct	# Incorrect	% Correct
0.9 - 1	10	11	91 %
0.8 - 0.9	37	40	92.5 %
0.7 - 0.8	12	14	86 %
0.6 - 0.7	21	28	75 %
0.5 - 0.6	14	21	67 %
0.4 - 0.5	0	0	-
0.3 - 0.4	1	3	33 %
0.2 - 0.3	4	6	67 %
0.1 - 0.2	4	25	16 %
0.0 - 0.1	0	2	0 %
0.0 - 1	103	150	69 %

To evaluate the three unrelated questions as possible cause for the user tests' low scoring percentage, consider Table 5-1. When the correct answer and question had a similarity value larger than 0.6, test participants chose the closest matching option 86% of the time. Accuracy fell to only 40% if the similarity value was below 0.6. This supports the notion that the application matches with accuracy comparable to a qualified person, if it is assumed that questions containing three much unrelated concepts are considered less relevant due to the difficulty of comparing them. However, more extensive research should be done to reach conclusive results.

5.2 Reliability Analysis

The information gathered from Cheffle employees is mostly based on their professional experience and understanding of skills and competences relevant to the hotel and restaurant industry. Additionally, only a small number of people participated in the evaluation of the final product. This means the feedback varies dependent on each individual's personal experiences and that the reliability of the feedback is based on qualitative sources and not quantitative results.

5.3 Validity Analysis

The data gathered is considered valid to a high degree, as previously discussed in Section 3.4.2; it is based on the knowledge of people in the industry this thesis project addresses. However, since all ontology elements are represented using solely a short title, there were a few instances during user tests where a given object was perceived as ambiguous. This ambiguity is never a problem for the matching algorithm since it has the entire relational hierarchy to fetch an element's context, but it could cause some invalid results due to insufficient context provided for test participants. A revised and better designed test would be necessary for future research.

5.4 Discussion

After two weeks of testing the application's matching method on real user data, two points surfaced: well-structured user information is crucial to accurate matching and scaling could be a future concern. Due to the structure of Cheffle user data at the time of this thesis project, the sheer preprocessing of data to extract matchable terms caused limitations in both quality and speed. Speed was not then a paramount issue; however concern was raised that it could cause complications at a later point. Variables like total size of user and job data and extended ontology could cause scalability problems, were Cheffle to expand their reach beyond the service industry. Structuring new data for better indexing and less preprocessing, or dividing the ontology into smaller more specific segments might prove helpful.

6 Conclusions and Future work

This final chapter summarizes and concludes the report. Conclusions and limitations are discussed, as well as potential future work and reflections.

6.1 Conclusions

The goal of this thesis project as stated in Section 1.4 is “To produce and analyze an automated job and candidate matching tool for Cheffle, in order to improve on their current manual job-candidate screening process.” The following three sub-goals defined the direction, pace and strategy for this project:

- Understand the general practical needs of a job and candidate matching tool and the specific needs of Cheffle and its customers.
- Translate these needs into a working solution using the available resources provided by Cheffle.
- Reach a result that satisfies KTH [10], Cheffle and myself.

The first sub-goal was reached in the pre-study phase by combining a literature review with company and industry specific insight from Cheffle. The second and third sub-goals were partially reached by producing a prototype and proof of concept, which improved upon the current manual job-candidate screening process by using a proposed matching method. It became apparent that no feasible working solution could be produced and deployed within the given time constraints of this thesis. Instead, focus was shifted to creating and proving the functionality of the proposed matching method by conducting both theoretical tests with Cheffle employees (Section 5.1.2) and practical tests using Cheffle data (Section 4.4).

I have gained multiple insights during this thesis project that have further increased my knowledge. These insights include:

- Gained knowledge of different matching techniques to identify information.
- Concepts of recruitment and human resource processes.
- Improved scientific writing and research.

An important suggestion for others working in this area and something I would have done if I had the thesis project to do all over again is to include data analysis and preprocessing in the literature study, in order to better analyze and test where results are not affected by the limits of elementary text parsing.

6.2 Limitations

One of the main limitations of this project was the difficulty of testing the matching method on poorly structured real user data. This limited some potential for evaluating the application’s performance, which would have been more clearly evaluated if user profiles with more restrictions were in place.

Other limitations included:

- Time. The aim of the project shifted from creating a full working solution to creating and proving the functionality of such a solution.
- Language. The application only supports matching of Swedish terms and concepts, limiting testing on users with English profiles.

6.3 Future work

The natural next step of this application is for it to reach preparedness for deployment to a production environment. To reach this point, compatibility of input data and further insight into scalability and performance with larger data sets must be considered. Though the matching method of this application has been shown to work, it is limited by user information lost to misspelling and poor formatting. My suggested solution to this problem is to let a controlled vocabulary of ontological elements aide in the creation of user profiles.

Currently, the application only supports use within the hotel and restaurant industry as the scope of matchable skills is limited by the scope of the ontology. A possible future effort would be to extend this reach and serve other industries; either by creating industry specific ontologies or expanding the one developed in this project.

Additional future work to consider is as follows:

- Set a noise floor by comparing the matching algorithm's performance with the developed ontology to random ontologies. This would measure the success of my applications' performance against random selections.
- Further research into the matching applications performance compared to real people. Would a test that filters out questions containing three unrelated terms yield more conclusive results? Can the test results be used to improve the ontology structure?
- DISCO exists in 15 parallel versions in different languages. The ontology could be expanded to cover additional languages by applying the same methods as those used to develop this Swedish version.

6.4 Reflections

If the working prototype presented in this report were to be further developed and integrated into Cheffle's platform, it could provide economic benefits to Cheffle and its customers, as well as social benefits to Cheffle users. By automating the screening process, employers will save time and effort while job applicants will be able to apply to a position with confidence that their selection is free from human error or bias.

The ethical aspects regarding this thesis should be noted. Aside from ethical concerns and responsibilities implicit in handling personal information [26], the potential for the application to produce suboptimal matches is concerning. Users put a certain level of trust in the system to deliver on its promises. For example, if the proposed solution produced incorrect job suggestions, job applicants might unknowingly accept positions when more fitting or desirable opportunities were available.

References

[1] C. Bizer, R. Heese, M. Mochol, R. Oldakowski, R. Tolksdorf, and R. Eckstein, "The Impact of Semantic Web Technologies on Job Recruitment Processes," in *Wirtschaftsinformatik 2005*, 2005, pp. 1367–1381.

[2] "Ersättning från a-kassa." [Online]. Available: http://www.arbetsformedlingen.se/For-arbetssokande/Stod-och-service/Ersattning-fran-a-kassa.html. [Accessed: 11-May-2017].

[3] M. Lavergne and S. Mullainathan, "Are Emily and Greg More Employable Than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination," *The American Economic Review*, vol. 94, no. 4, pp. 991–1013, Sep. 2004.

[4] A. Stafsudd, "People are strange when you're a stranger: senior executives select similar successors," *European Management Review*, vol. 3, no. 3, pp. 177–189, Jan. 2006.

[5] U. Simonsohn and F. Gino, "Daily Horizons: Evidence of Narrow Bracketing in Judgments from 9,000 MBA Admission Interviews," 2013.

[6] F. Lievens, K. van Dam, and N. Anderson, "Recent trends and challenges in personnel selection," *Personnel Review*, vol. 31, no. 5, pp. 580–601, Oct. 2002.

[7] A. Capiluppi, A. Serebrenik, and L. Singer, "Assessing Technical Candidates on the Social Web," *IEEE Software*, vol. 30, no. 1, pp. 45–51, Jan. 2013.

[8] D. Pyle, *Data Preparation for Data Mining*. Morgan Kaufmann, 1999.

[9] X. Yi, J. Allan, and W. B. Croft, "Matching Resumes and Jobs Based on Relevance Models," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2007, pp. 809–810.

[10] "KTH, II143X Degree Project in Information and Communication Technology, First Cycle 15.0 credits." [Online]. Available: https://www.kth.se/student/kurser/kurs/II143X?l=en. [Accessed: 11-May-2017].

[11] "Human Resources: Recruitment & Selection Hiring Process." [Online]. Available: https://hr.ucr.edu/recruitment/guidelines/process.html. [Accessed: 22-Mar-2017].

[12] E. Ärnström and J. Bergman, *Attracting the right employees : A study of successful employer branding*. 2011.

[13] G. Rácz, A. Sali, and K.-D. Schewe, "Semantic Matching Strategies for Job Recruitment: A Comparison of New and Known Approaches," in *Foundations of Information and Knowledge Systems*, 2016, pp. 149–168.

[14] M. M. Deza and E. Deza, *Encyclopedia of distances*. Springer Berlin Heidelberg, 2009.

[15] G. Gilbert, "Distance between Sets," *Nature*, vol. 239, no. 5368, pp. 174–174, Sep. 1972.

[16] European Dictionary of Skills and Competences - Swedish version, "DISCO II Portal." [Online]. Available:http:/disco-tools.eu/disco2_portal/terms.php. [Accessed: 08-Apr-2017].

[17] "Standard Occupational Classification (SOC) System." [Online]. Available: https://www.bls.gov/soc/. [Accessed: 11-May-2017].

[18] “ISCO - International Standard Classification of Occupations.” [Online]. Available: http://www.ilo.org/public/english/bureau/stat/isco/. [Accessed: 11-May-2017].
[19] F. Giunchiglia and P. Shvaiko, “Semantic matching,” *The Knowledge Engineering Review Journal*, vol. 18, no. 3, pp. 265–280, 2003.
[20] F. Giunchiglia, M. Yatskevich, and P. Shvaiko, “Semantic Matching: Algorithms and Implementation,” in *Journal on Data Semantics IX*, 2007, pp. 1–38.
[21] J. Zhong, H. Zhu, J. Li, and Y. Yu, “Conceptual Graph Matching for Semantic Search,” in *Conceptual Structures: Integration and Interfaces*, 2002, pp. 92–106.
[22] J. Poole and J. A. Campbell, “A novel algorithm for matching conceptual and related graphs,” in *Conceptual Structures: Applications, Implementation and Theory*, 1995, pp. 293–307.
[23] J. F. Sowa, *Conceptual Structures: Information Processing in Mind and Machine*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1984.
[24] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.
[25] J. Kleinberg and E. Tardos, *Algorithm Design*, 1st ed. Pearson, ISBN-13: 978-0321295354 .
[26] Datainspektionen, “Personuppgiftslagen - Datainspektionen.” [Online]. Available: http://www.datainspektionen.se/lagar-och-regler/personuppgiftslagen/. [Accessed: 11-May-2017].
[27] “Python.org,” *Python.org*. [Online]. Available: https://www.python.org/download/releases/2.7/. [Accessed: 11-May-2017].
[28] T. Bray, “The JavaScript Object Notation (JSON) Data Interchange Format.” [Online]. Available: https://tools.ietf.org/html/rfc7159.html. [Accessed: 11-May-2017].
[29] “Online JSON Viewer.” [Online]. Available: http://jsonviewer.stack.hu/. [Accessed: 11-May-2017].
[30] “JSON encoder and decoder — Python 2.7.13 documentation.” [Online]. Available: https://docs.python.org/2/library/json.html. [Accessed: 11-May-2017].
[31] “PythonImplementations - Python Wiki.” [Online]. Available: https://wiki.python.org/moin/PythonImplementations. [Accessed: 11-May-2017].
[32] “Flask (A Python Microframework).” [Online]. Available: http://flask.pocoo.org/. [Accessed: 11-May-2017].

TRITA-ICT-EX-2017:47