NTNU
Norwegian University of
Science and Technology

Lazy Learned Screening for Efficient Recruitment

A Candidate Ranking System using Case
Based Reasoning and Semantic Data Models

Erik Espenakk
Magnus Johan Knalstad

Master of Science in Computer Science
Submission date: June 2018
Supervisor: Anders Kofod-Petersen, IDI

Norwegian University of Science and Technology
Department of Computer Science

Acknowledgements

The authors would like to thank all the recruiters and participants that helped us enable the foundation for this research. We would also like to thank our supervisor Anders Kofod-Pedersen with assistance and for being positive towards this project. We would also like to thank Gustaf Solerd Christian, COO at Highered for his good humour and allowing us to use his network in order to gather participants. Big thanks also goes to Ingraphic for facilitating a real world scenario allowing us to evaluate the system. Last but not least, we would like to thank our families and better half's for being patient and supportive during this research.

Summary

The transition from traditional paper based systems for recruitment over to the internet has resulted companies in getting a lot more applications. A majority of these applications are often unstructured documents sent over mail. This results in a lot of work sorting through the applicants. Due to this, a number of systems have been implemented in an effort to make the screening phase more efficient. The main problems consisting of extracting information from resumes and ranking the candidates the candidates for positions based on their relevance.

In this research we want to develop a system that can learn how to rank candidates for a position based on knowledge obtained from earlier screening phases. To this end we developed and integrated a Candidate Ranking System based on a lazy learning technique, namely Case Based Reasoning, combined with semantic data models. The systems performance was evaluated in conjunction with having Okapi BM25 as a baseline due to its widespread usage in comparing ranking system and other related work.

Summary

Overgangen fra tradisjonelle papirbaserte systemer i rekruttering over internett har resultert i at flere selskaper fr en stor andel flere sknader enn fr. Et flertall av disse sknadene er ofte ustrukturerte dokumenter som er sendt via epost. Dette resulterer i mye tid som gr bort til sortering og utplukking. Flere studier og systemer har derfor gtt ut p gjre denne prosessen mer effektiv. Der i blant finner vi systemer for kandidatrangering.

I denne forskningen s nsket vi automatisere deler av prosessen med plukke ut relevante kandidater. Dette har vi tenkt til gjre ved implementere ett system som kan gjennbruke erfaringer fra tidligere rekrutteringer. Vi har dermed utviklet og integrert et kandidatrangeringssystem som baserer seg p Case Based Reasoning og tar i bruk semantiske modeller.

Det implementerte systemet er blitt sammenlignet med Okapi BM25 som referansepunkt i en evaluering. Valget av denne falt p bakgrunnen av at denne var utbredt brukt i forbindelse med sammenligne rangeringssystemer, og annen relatert forskning. Evalueringen ble gjort i forbindelse med en ekte rekruttering. Det ble ogs gjennomfrt ett eksperiment inspirert av testen til Alan Touring.

Preface

This Master's Thesis is the final deliverable of the Computer Science Program at the Department of Computer Science and Faculty of Information Technology and Electrical Engineering. The Master's research in this program encompasses two parts. A pre-study conducted in the due to December 2017 followed by a main thesis, due to June 2018.

In the pre-study, the main goal was to gain a conceptual framework over the different approaches used within recruitment and with especially respect to AI. Based on this conceptual framework, another main goal was to propose a solution to candidate ranking by leverage an undiscovered technique to the problem.

This thesis builds on the pre-study, by having a main goal of integrating, developing and testing the proposed solution in a real world recruitment scenario aligned with and simulated experiments. In addition a goal of this thesis is to introduce new knowledge of using Case Based Reasoning in the field of recruitment.

Table of Contents

List of Tables

List of Figures

Abbreviations

Attribute = An atomic part of the case representation.
CBR = Case Based Reasoning
CV = Curriculum Vitae
CCBR = Conversational Case Based Reasoning
TCP = Transmission Control Protocol CRS = Candidate Ranking System
AI = Artificial Intelligence
BLL = Business Logic Layer
DAL = Data Access Layer
HR = Human Recruiter

Chapter 1

Introduction

This chapter address the background and motivation for this research project. Research goals will be presented as well as the research questions used to explore the area of research. It then outlines a plan to answer these questions, both the methodology and strategy used to answer as well as the structure of the thesis itself. We will also reserve a small part of this chapter to present the contributions made by this project.

1.1 Background and Motivation

Moving from the traditional paper based systems for recruitment and over to the internet, has resulted in companies getting a lot more applications. These application are often unstructured documents sent over email. The increase in the number of applications have led to a desire to automate parts of the recruitment processLee (2011).

> From an organizations viewpoint, a typical recruitment process can be divided into four main phases: describing the requirements of the job posting, publishing the job posting, receiving of applications, and final decision making
>
> (Maniu et al., 2009)

The use of technology to aid in this last step has exploded since 2008. Computerized assessments have become mainstream for both big and small organizations (Ryan and Ployhart, 2014). Several systems and solutions have been proposed to aid recruiters during this screening process, by automatically ranking the applicants based on their relevance.

In our preliminary study in the fall of 2017, we conducted a twofold literature review to gain a conceptual framework over the different approaches used and especially the AI techniques employed. Based on this, we proposed a novel approach for solving the ”candidate ranking problem” in recruitment by using Case Based Reasoning and semantic data models. The information retrieval ranking algorithm Okapi BestMatch25 was chosen as a baseline for evaluating the system. Okapi BM25 was chosen partially due to its widespread use in older information retrieval systems (Robertson, 1997) and because it has been used as a baseline in the related work by Gil et al. (2016).

Case based reasoning is based on lazy learning, meaning it will store previous knowledge without attempting to generalize it first. The generalization is attempted first when the knowledge is required. This is also the main reason behind our choice of using case based reasoning. The lazy learning aspect makes the system approximate target function locally. This will allow the system to continuously learn, without having to retrain a model.

In this paper we intend to develop, deploy and evaluate the proposed candidate ranking system. The system will be deployed into production in two companies and evaluated in a real world scenario. Due to this the system needs to be assessed for risks and the risk of breaching laws and regulations such as the General Data Protection Regulations(Council of European Union, 2016) needs to be low.

1.2 Goals and Research Questions

This section will introduce the research goals and questions that were created with a basis in our motivations.

1.2.1 Research Goals

Based on our motivations, our main goal is to make the recruiting process more cost efficient by automating part of the screening. Since research is about laying stone by stone in order achieve the main goal, this paper focuses on two sub goals that hopefully will enable some foundations for laying the first stone. Below we introduce the two goals for this paper:

G1 Develop and integrate a candidate ranking system to improve the efficiency of the screening phase in a recruitment scenario by reusing knowledge from earlier screening phases.

G2 Introduce new knowledge that may support the viability of using Case Based Reasoning in the field of Recruitment.

1.2.2 Research Questions

RQ:1 How does the lists produced by the Candidate Ranking System compare to those made by Okapi BM25 and human recruiters?
RQ:1.1 How does the CRS list correlate with human recruiters against the baseline Okapi BM25 in a top-25 list for a certain position in real world scenario?
RQ:1.2 when evaluated by other experts, how does the CRS top-5 list compare to The top-5 lists made/generated by human recruiters and Okapi BM25 given four job position scenarios?
RQ:1.3 When wish to juxtapose Okapi and CRS to the list produced by a list considered to be the most precise, namely those that are made by human experts
RQ2: How does system ensure compatibility in order to be deployed in the real world with respect to the new General Data Protection Regulations imposed by the European Union and ensuring that users find necessary attributes to reflect their competency?

1.3 Research Methods

We have applied the scientific method, that is a positivist approach to the research. The scientific method seeks to find all the regular laws or patterns in our universe, treating the world as ordered and regular (Oates, 2006). Our research questions are constructed to seek answers and provide conclusions in an objective and ordered manner. The Positivism-paradigm is considered suitable for the purpose.

1.3.1 Research Strategy

Design and creation has been used as the main research strategy. This strategy is also called a problem-solving approach Oates (2006). The main steps involved when using this strategy includes, creating a prototype software system, often called an artifact and the next step is evaluation. In order to determine the viability of an artifact it needs a way to be tested and have its effects validated. To this end a couple of options were considered and testing the artifact in a real world scenario was determined to be the most relevant way.

The real world scenario will be facilitated by comparing the ranked list of applicants created by our system, and one from the baseline BM25, to the ranked list created by a human recruiter for a job.

1.3.2 Data Generation Method

We have used questionnaires for generating data, we have also used questionnaires in order facilitate much of the data generation needed for the evaluation 4.

1.3.3 Scope of the Research

This research has had a span from being a preliminary project conducted in the fall of 2017 to evolve into this thesis in the upcoming spring of 2018. As explained in section 1.1, the starting point for the research had its origin in the curiosity of utilize AI in the domain of job recruitment and especially candidate ranking. This motivated in the establishment of finding the state of the art and derive an applicable design that could contribute to new research.

This thesis presents the second part which is about testing and evaluate the system in a real world scenario and conduct several experiments to match our system with the baseline and recruiters.

1.3.4 Related Work

The related work is described in more detail in chapter 2. The studies that we found the most relevant use some form of machine learning technique for ranking applicants in a recruitment scenario.

Faliagka et al. (2012b) compares a set of learning to rank methods in their ability to predict relevance for candidates compared to a human recruiter. They use spearman's correlation coefficient to compare the resulting lists. the study (Gil et al., 2016) presents a novel approach of applying machine learning to learn the relevance of applicants for a job.

Their solution is able to approximate a function for determining the distance between two resumes or a resume and a job description. They define the distance as a set of replacement, deletion and insertion costs on the attributes in a resume. The attributes are identified using the ISCO taxonomy. Machine learning is used to train a model of these (replacement, deletion and insertion costs), by comparing a resulting list with a list created by a human recruiter.

These studies all explore different approaches for solving the problem using an eager learning technique. This means that a set of data will have to be collected in advance, ”the company needs lists of CV’s sorted according to relevance, for each of the jobs they wish to train for”. The data models will have to be retrained if big changes happen, if for instance jobs in the future require higher education. Gil et al. (2016) also talks about the downside of using a list compiled by a human recruiter to learn from. The recruiters have a tendency to put a lot of effort into correctly ordering the first few candidates, and after a point the order starts to break down (Gil et al., 2016).

1.3.5 Evaluation

This research is evaluated using the most promising techniques within information retrieval in order to measure the system’s performance against the baseline. Some of these are the E-Measure, Spearman and Discounted Cumulative Gain that we explain in section 2.5.2. Due to research question 2, we chose to perform a risk management iteration using the ISO/IEC 27005 framework from International Organization for Standardization and International Electrotechnical Commission (2011) as a guide. An informal experiment has also been designed, the results obtained will be discussed in relation to research question (1,1.2,1.3).

1.4 Contributions

The first deliverable is the presentation of a candidate ranking system based on CBR with the objective to aid human resources in their candidate selection process. Implementing and evaluating this system was necessary to answer the research questions presented in 1.2.2. In addition to the CRS, this report contributes with new knowledge by presenting and discussing the empirical results obtained. A small contribution to the field of CBR can also be attributed, applying the CBR paradigm in this specific way to this problem had not been attempted before. This was determined in a structured literature review disseminated in our preliminary study.

The system’s source code has not been included for this deliverable due the protection of intellectual property. However, architectural blueprints are presented and modelled with UML[1]. These are made available to the reader in the architecture chapter 3. In addition the report contributes with pseudo code to help understand the more advanced processes. This pseudo code including the aforementioned diagrams should provide a foundation for reproducing the CRS.

[1] Abbreviation for Unified Modelling Language

1.5 Thesis Structure

This section will quickly summarize the structure of the thesis and give some insight into the content of the different chapters. The thesis is structured into six different main chapters, each chapter with its own introduction section and a summary at the end.

Chapter 2 - Background theory Chapter 2 gives the reader the proper theoretical basis on which to understand and critique the thesis. The thesis itself assumes a degree of familiarity with basic computer science, but the more specific topics are covered in chapter 2. Among these; Case based reasoning and semantic web.

Chapter 3 - Architecture Chapter 3 introduces the design and architecture used in the creation of the candidate ranking system. Details and design decisions are summarized and discussed for all parts of the system. Models and design details are presented using diagrams created with the Unified Modeling Language 2.0 convention.

Chapter 4 - Integration and deployment Chapter 4 contains a risk management procedure and discusses the steps required to get the system production ready and integrated into an existing platform with respect to information security.

Chapter 5 - Methods and Experiment Chapter 5 discusses the results obtained from the experiment, how this might relate to the system and the design decisions made.

Chapter 6 - Results This chapter is a formal chapter that presents the results from the different evaluation strategies applied.

Chapter 7 - Discussion This chapter discusses the results obtained in chapter 6 with respect to the goals and research questions. **Chapter 8 - Conclusion and future work** Chapter 8 concludes and summarizes the thesis and present the possible improvements and interesting directions for future work.

Chapter 2

Background Theory and Motivation

This chapter provides an overview over the background for this thesis. The first section will briefly explain some of the results and motivation obtained from the preliminary project. Afterwards, an explanation for the underlying technology relevant for this project are undergone and discussed.

2.1 The Job Recruitment Domain

Job Recruitment is a large field composed by many actors and aspects. In this section we will briefly touch the surface of some of them and introduce our approach within this domain.

2.1.1 Recruitment

Traditionally recruitment has been separated into to the following three main steps, Sourcing, Screening and Selection. The sourcing step is where the candidate list gets filled with potential candidates. The sourcing step has traditionally been solved in a passive way, by using ads in papers and more recently over the internet. However exceptions include the use of recruitment agencies as some of these tend to actively seek out candidates via their online persona.

Screening is the second step of the recruitment process. This is also the step that we want to focus on in this study. The screening step is where you have a bunch of candidates for a job and you want to sort them from most suitable to least suitable. In the area of screening there are several assessment suites and tests that can be performed, each of these claiming that they can tell something about how the candidate will perform. We will however mostly stick to a slightly earlier phase of the screening, before resources are spent requesting tests. In the job market today, many of the job-applications come in as emails containing CV, cover letter and an application text. Some companies offering attractive positions can get hundreds of applications, some of which are not even relevant. This is where we would like to put our focus, using machine learning to do some pre-screening in order to reduce the human resource usage.

Selection is the last step of the recruitment process. The candidates that were picked out and scored well on tests and initial interviews are chosen and can proceed to final interviews. From these a few are chosen to fill the open positions.

2.1.2 E-Recruitment

As the internet came along, so did the tendency to handle job applications over e-mail. This resulted in a lot of work maintaining some form of order and control over the list of candidates and their respective time lines including interviews and tests. Several E-Recruitment systems have since been developed to maintain a database of candidates and reduce the amount of time spent by human resources. Today there are two main forms of systems available, we have the more traditional "e-recruitment" aiding individual companies and then we have talent-pools and recruitment platforms like linkedin. Linkedin with its massive database of profiles will be a big player in sourcing talent for years to come.

2.2 Motivation

This section will take you through some of the motivations behind the project. Our motivation can be described as a combination between a application specific and a technique specific motivation. In the subsections below we will detail the main motivators, and describe in what degree they influence our work.

2.2.1 Preliminary Study

The main motivator behind our project can be said to be our preliminary project. In the fall of 2017 we conducted a project in which one of the main goals was finding ways to automate and improve work-flow when dealing with human resources. During our preliminary literature study we scoped the goal down to improving the process of recruiting human resources.

We established what was the state of the art within digital recruitment systems and platforms. We discovered one specific issue that we wanted to address, namely the issue of ranking and filtering out suitable candidates in a recruitment process. Here with a focus on the first filtering and ranking that takes place before candidates are called in for interviews. We established that this problem was mainly a ranking problem, given that all the data came in structured and that applicants and the job itself shared a common vocabulary. This however is not always the case, some studies used natural language processing to separate out the useful information from job-applications while others used taxonomies and word-bag approaches. We did however not find a study that had attempted to solve the problem using a Case-based reasoning approach. This coupled with the fact that NTNU has some of the leading scientists in the field of Case-based reasoning motivated us to design a CBR based solution for this ranking problem.

2.2.2 Related Work

Kessler et al. (2012) has over the course of several studies (Kessler et al., 2007, 2008, 2009) developed the ”E-Gen” system, a Natural Language Processing (NLP) and Information Retrieval (IR) system. The system uses (Support Vector Machines, SVM) to analyse the content of a candidate’s email. In (Kessler et al., 2012) they continue the work by combining the E-Gen system with the Cortex in order to select the most relevant relevant candidates for a job.

Faliagka et al. (2012c) uses analytical hierarchy process (AHP) to rank candidates based on information from linkedin and text mining on a personal blog. In (Faliagka et al., 2012b) they take this further, comparing a set of learning to rank methods in their ability to predict relevance for candidates compared to a human recruiter. In (Faliagka et al., 2014) they add a taxonomy to distinguish between certain ICT skills.

Gil et al. (2016) presented a solution based on machine learning. Their solution is able to approximate a function for determining the distance between two resumes or a resume and a job description. They define the distance as a set of replacement, deletion and insertion costs on the attributes in a resume. The attributes are identified using the ISCO taxonomy. Machine learning is used to train a model of these (replacement, deletion and insertion costs), by comparing a resulting list with a list created by a human recruiter.

Kmail et al. (2015) uses NLP and semantic resources to extract and relate candidate concepts from job descriptions and resumes. Another approach is introduced in (Menon and Rahulnath, 2016). It exploits what they call ”Emotional Aptitude Evaluation Module”. The idea is to measure the applicants’ emotional intelligence based on their twitter posts.

A CBR approach is presented in (Siraj et al., 2011). In their approach they represent jobs and job seekers as a set of attributes (Gender, Age, Race, State/geographic location”, Qualification, Grade point average, etc.). A feature vector representation approach with feature similarity is used. The system was tested by a selection of employers and they were asked to fill out a questionnaire. The results from the questionnaire showed that the employers think that they could be more effective using the proposed system.

In the study Salazar et al. (2015), a CBR system using ontology as representation is proposed. The study explores the use of the HR-XML[1] standard for representing CV’s. In their proposed system, CBR is implemented as an agent in a multiagent environment. They use a small part of HR-XML to represent a candidate’s CV and the job offers. This limits their systems ability to do detailed comparisons.

Siting et al. (2012) survey different recommendation-system approaches applied in the job domain. These range from systems using collaborative recommendation that recommends jobs that similar profiles have liked, to bilateral recommendation systems that are able to provide recommendation to both job seeker and employer. The systems covered in this study are mostly approaches used in the early job portals in combination with information retrieval techniques.

[1] http://schemas.liquid-technologies.com/hr-xml/2007-04-15/

2.2.3 HigherEd

During preliminary research we were introduced to a company called HigherEd [2], a part of the EFDM Global Services[3]. This company has built the first truly global platform in order to connect, market and recruit top talent, from the absolute top graduate schools all over the world. This company has secured the rights to co-operate with the most acknowledged business schools through the accrediting body EFDM. This means they are given the direct access to the top schools around the globe. With a reach of more than 500 top schools and over 3 million students, they are given unprecedented diversity, corporate innovation and employer branding possibilities. Some of their customers are companies like Pepsico, Nike, Yara, Statoil, Volkswagen Truck & Bus, Accenture, Norges Bank, to mentioned some of the ones that are already leveraging HigherEd's web platform.

HigherEd explained to us that they planned to include a recruiting module into their platform in the upcoming year. An agreed contract has been written allowing us to evaluate our proposed system using their unique structured dataset explained under chapter 3.

2.3 The Case Based Reasoning Paradigm

In this chapter we will start with introducing the case based paradigm and briefly its background. Afterwards we will explain the CBR Cycle more in depth before looking into the term Case Representation and Similarity Measures.

2.3.1 Introduction

Case based reasoning (CBR) in computer science is an approach to problem solving and learning. Case based reasoning in simple terms is a system for using and retaining knowledge from past experiences. It does this by maintaining a library of cases. In CBR a case is a problem with an accompanying solution, stored in a fashion that enables reuse. CBR systems are different from simple databases of cases, due to their ability to retrieve and modify partially matching cases and apply these to attempt to solve a new problem.

Aamodt and Plaza (1994) wrote that "Case-based reasoning is a problem solving paradigm that in many respects is fundamentally different from other major AI approaches". The case-based reasoning paradigm differs in that it promotes the use of specific knowledge from earlier problem-solving situations "cases". Case-based reasoning is also suited for problems where one does not start off with a huge amount of data. This is due to the incremental learning nature of the CBR paradigm (Aamodt and Plaza, 1994).

2.3.2 The CBR Cycle

The case-based reasoning cycle is often depicted as a composition of four main activities. From the figure 2.1 we see; retrieve, reuse, revise and retain. In figure 2.1 Aamodt and Plaza (1994) describes retrieve as the first action given a new case. A new model R5 was suggested in the paper Finnie and Sun (2003). The R5 model adds a repartition step and

[2] http://highered.efmdglobal.org/
[3] http://www.efmd.org/what-is-efmd

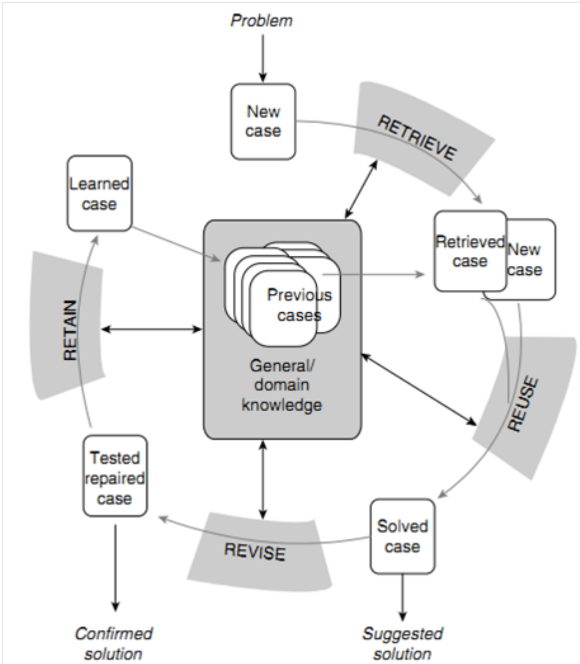


Figure 2.1: The CBR cycle as presented by Aamodt and Plaza (1994)

also a ”case base building” step as the first step. In this paper we will mainly refer to the R4 model suggested by Aamodt and Plaza (1994). To solve a problem using case-based reasoning we first define the problem as a case. We can then look for similar cases that the system has processed earlier. If a good match is found then this case is retrieved. The retrieved case is combined with the new case to find a solution. If the solution is accepted then the new case is retained. If not, then the case is revised. A functioning case-based reasoning system will have coherent solutions to all the mentioned activities. However there is one problem left, knowledge representation. How do we represent cases?

2.3.2.1 Retrieval

The first step of the cycle, Retrieval is about find which stored case is most useful to apply to a new problem. In order to retrieve cases the the retrieval step applies similarity measures. The step starts when a new problem is easily accessible and completes when a case is retrieved from the case base. In section 2.3.4 Retrieval and Similarity Measures, we undergo different retrieval techniques and similarity measures.

2.3.2.2 Reuse

Reuse is the step in the CBR Cycle when a case is selected for its solution to be reused. The idea is to reuse information and knowledge from the retrieved cases to solve a new problem. When a new problem turns out to be identical to the retrieved case problem, the

Reuse-step is quite simple, as the case will be reused. However, when they differ, Adaption is required. Section 2.3.5 Reuse and Adaption will go deeper into the possible adaption techniques.

2.3.2.3 Revise

The objective with the Revise cycle is to evaluate the applicability of the proposed solution. The revise step is concerned with a single problem and its solution. The evaluation step can either be evaluated in a simulation or in the real world. Since the observations come from the real world, it can be helpful recording the source of the observation. Therefore, it is recommended storing the case in a problem-solution fashion. This will be a benefit as experiences tell us something on how it was applied in the past.

2.3.2.4 Retain

If the revise step has generated a new case, the Retain step is responsible for updating the case base the new and learned case in order to help potential help solve other similar problems in the future.

2.3.3 Case representation

To represent knowledge in CBR systems one address some issues. Which attributes can represent a case, how can one measure the similarity between different cases, how should the cases be physically stored. Several solutions to these issues have been proposed. El-Sappagh and Elmogy (2015) surveyed and evaluated a range of different case representation methodologies used to address these issues. They group the methodologies into knowledge based case representation and traditional case representation, we will cover the knowledge based representation in a section below. The traditional representations will be Split into two subsections, one for basic representation and one for more advanced structures.

2.3.3.1 Basic representations

Richter and Weber (2013a) considers the following representations as part of the basic methods of representation.

- Text
- Images
- Speech
- Sensor data
- Conversational representation

2.3.3.2 Advanced case structures

Richter and Weber (2013a) considers the following representations as part of the basic methods of representation.

- Flat attribute-value representation
- Object-oriented representation
- Trees and graphs
- Hierarchies and taxonomies
- Generalized (set-oriented) cases

Flat attribute-value representation

Also called feature vector representation. This representation can be described as a set of attribute-value pairs of a specified common length. The main limitation of this representation is that cases cannot have unique structures, they must all have the same attributes and number of attributes. If this is not the case then the similarity measures will become much more complex.

The common way of computing similarity between cases using this representation is to use a global-local approach and either, directly compare each attribute value pair, or to include some weight for each attribute-value pair indicating their importance. Following the global-local principle we compute the similarity between each attribute-value pair and then combine these similarities into the global similarity. The local similarity function used to compare each attribute value pair depend on what kind of datatype has been used.

The two main characteristics of this representation is that it is flat and its content and length is not dynamic.

Object-oriented representation

The object-oriented representation tries to address some of the limitations of the flat attribute value representation. First it addresses the lack of structure by providing a way of defining structure in the case representation. This means that complex relationships and taxonomies can be expressed in the structure of the case representation.

In the object-oriented representation, an object describes an entity of the domain by a finite set of attributes. Each attribute can represent either a datatype or a link/relation (ex is-a, part-of) to another object. The objects are instances and their content, and internal structure can be described by their respective classes.

The main disadvantages are the additional costs related to a more complex similarity computation and a more complex retrieval procedure, also sequential orderings on objects cannot be formulated. The representation is recommended where complex properties occur, that is, if you have properties that cannot simply be described using a basic datatype.

Trees and graphs

Graphs and Trees are common data structures used in computer science they each come with certain advantages and disadvantages. Both graphs and trees consist of two main building blocks, nodes and edges. The edges represent relations between the nodes. The marks on the edges and nodes of an attributed graph are also called annotations. Edges and nodes and their annotations are the local elements of the representations. The annotations can, for instance, describe the costs of going from one node to another. Such costs can be summed along a path, which gives rise to the problem of finding optimal paths.

The properties of the objects are indicated in the node attributes while the relations between them are in the edge attributes.

A special attributed graph is a net of objects. Each relational attribute becomes an edge annotation. The set of all attribute-value pairs becomes an edge annotation. Other typical examples of such objects are:

- Road maps.
- Networks.
- Descriptions of machines

2.3.3.3 Knowledge based case representation

In some situations, domain knowledge might be required to get satisfactory reasoning capabilities. In (Bergmann et al., 1996), they device two rules to be used with the case base. A completion rule, this rule infers additional features in old cases using known features. The other rule is the adaption rule. The adaption rule covers how to adapt old cases to be used in solving the current case. These two rules are part of what they describe as general knowledge.

In Aamodt (2004) CREEK, the CBR system has a substantial amount of general knowledge and a smaller amount of cases compared to traditional CBR systems. Their cases are part of a general knowledge model, this model is semantic network. Both ontologies and semantic networks will be covered in a separate section. Knowledge based representations can benefit from using ontologies, as this will provide a formal language and in some cases already acquired knowledge.

2.3.4 Retrieval and Similarity Measures

In this section we will take a closer look regarding the first step of the CBR Cycle introduced in 2.3.2, retrieval. As mentioned Retrieval encompasses the process of finding the best case or cases based on a given query problem. This section will first explain sequential retrieval, before we detail the concept of Similarity and different measures.

2.3.4.1 Sequential Retrieval

Picking a sufficient retrieval technique depends on the context of how the problem is formulated. There exists a magnitude of different approaches in the domain of CBR Retrieval. One of them is sequential retrieval. According to Richter and Weber (2013a), Sequential

Retrieval is a retrieval technique that operates as a brute force approach to find the K-Nearest neighbours. It has a linear complexity. The time complexity is $O(n)$, meaning that the run time is linearly increasing with the input size. Because of this sequential retrieval is not very sufficient for larger case bases. However, it is very simple to implement and it's considered a good candidate when you deal with problems that requires several types of different similarity measures.

2.3.4.2 Similarity Measurement as a Concept

Richter and Weber (2013a) defines Similarity Measures as "the retrieval of similar cases based upon the use of similarity functions to compute the distance or similarity of two cases." This entails that to match different cases, one needs to find the similarity between the queries and cases. To apply similarity measures in a CBR System, you first need to define the conceptional meaning of the term "similarity". Since similarity is a relative concept, meaning it depends strongly on the domain of the objects to be compared, similarity does not have an absolute meaning Richter and Weber (2013a).

Having that in mind we can also observe that similarity is neither transitive. To state an example, lets consider finding the best case in a car purchase scenario. Your query wants to retrieve a case based on the similarity in color and price. Your constraints opted to be a blue color car that should be somewhere within the 500 000 NOK,- range. Finding out that Car A and Car B has the similarity in color, and Car B and Car C being similar as both are in the same price category, does not necessarily say that Car A and Car C are similar, since both Car A and Car C can have completely different color and price.

When talking about the concept of similarity, it is important to distinguish utility from similarity. Utility is considered a-posteriori criterion. This means that the utility usually is evaluated after solving the problem. Similarity however, is considered as an a-priori criterion Richter and Weber (2013a). This means that the similarity has to be computed before solving the problem. In short terms we can say that similarity approximates the utility of a case, given a specific requirement. Similarity is therefore considered as an heuristic to estimate the unknown utility function.

Similarity Measure is tightly coupled to the first step of the CBR-cycle, the retrieval step introduced in 2.3.4. Richter and Weber (2013a) describes that applying a similarity function should be considered a trade-off between:

- Modelling Effort: Amount of domain knowledge to be encoded and the procedure for encoding the knowledge

- Quality of the Similarity Measure

- Computational Complexity, which strongly influences the retrieval-efficiency.

2.3.4.3 Local-Global Principle

When we discuss similarity measures, it is important to distinguish between local and global measures. Similarity between the given query and a case, can be referred to as the global similarity; a result of the attributes combination of specific local similarities

Richter and Weber (2013a), discussed in 2.3.4.4. By achieving the global similarity, one has to apply a suited amalgamation function. This is introduced in section 2.3.4.5

In general terms, the local global principle says that each object A is constructed from atomic parts A_i that are attributes by some construction process

$$A = C(A_i | i \in I) \tag{2.1}$$

In simple terms the idea is to compare the first atomic parts before comparing the more complex constructs.

2.3.4.4 Local Similarity

This section start explaining the concept of local similarity and introduces some of the different approaches within the domain of local similarity that depends on the use case. We will introduce and explain textual, binary, numerical, taxonomic and object oriented similarity measures and provide some formulas where applicable.

The concept

Local measures has the objective to represent domain properties. Given attributes A, similarity is usually based on the difference of attribute-value and not on the attribute value themselves. Each local similarity has its own similarity function. However, the model type and function applied for the local measure depends on the attribute type Richter and Weber (2013a).

$$sim(q,c) = \{^{1\, if\, d(q,c) < S}_{0\, else}\; with\, d(q,c) \;=\; |q-c| \tag{2.2}$$

The local measures can either be symmetric or asymmetric. The latter means it is important that the attribute-value is either in the query or the answer. For instance consider a price tag where it is not necessarily important that a price matches exactly, say 1000 NOK, but that the case has a price lower than 1000. The Step Function 2.2 illustrates.

Textual Similarity

Richter and Weber (2013a) says that Textual similarity is considered hard. This is because you have to deal with syntactic texts that can have different meanings. To simplify this process one can consider utilize the ”Bag of Words” approach. This means that you tokenize each word that have great discrimination degree. In general terms, we first preprocess the text, removing stop words or using a stemmer. Consider the sentence ”I have been working as a Software Engineer for 5 years.” This Bag of Words after preprocessed will provide the word bag below:

$$T = \{Engineer, Software, Work, 5, years\} \tag{2.3}$$

As one can see stop words are removed, and the word ”Working” has been stemmed to Work, removing the ”ing” part. It is widely used as part of document preprocesing in information retrieval Oates (2006).

AS we can see from the word bag approach, there might be a drawback when applying this method. The textual meaning of a sentence can become unidentifiable. Therefore, it may not be very applicable in cases where you want to illuminate the textual meaning of sentences.

$$\begin{aligned} &similarity = cos(\Theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}, \\ &where\, A_i\, and\, B_i\, are\, components\, of\, Vector\, A\, and\, B\, respectively. \end{aligned} \tag{2.4}$$

Another standard approach adopted from Information Retrieval is the use of a Vector Representation. The concept here is to consider all texts as documents and create a vector representation for each document. The weights in each vector can be calculated using the TF-IDF weights, where TF stands for Term Frequency and IDF stands for Inverse Document Frequency. Term Frequency tells us how often a term or word occurs in a document, and Document Frequency says how frequent a term appears in the documents from a given corpus. The Inverse Document Frequency then punishes terms that occurs often in documents. When a query vector $\overrightarrow{q}$ and a document vector $\overrightarrow{d}$ have been calculated, we can use the Cosine Similarity formula to calculate the angel between the vectors displayed in formula 2.4.

Besides the latter two approaches, an approach based on transformational similarity can be used. This approach is called Levensthein distance, introduced in (Levenshtein, 1966). In short terms, the concept is that you count the number of changes needed to transform one string into another. These changes has to be applied with either insertion, deletion or a modification cost. The more similar, the cheaper the cost will get. It is specially a useful method for typo correction (Richter and Weber, 2013a).

Binary Similarity

$$H(x, y) = n - \sum_{i=1}^{n} x_i \cdot y_i - \sum_{i=1}^{n} (1 - x_i) \cdot (1 - y_i) = |\{i | x_i \neq y_i\}| \tag{2.5}$$

A good technique for calculating the similarity between binary attributes is the Hamming Distance, introduced in (Hamming, 1950). Briefly explained, the hamming distance in formula 2.5 is the total number of bits you have to change to make one bit-string equal to another. Consider for instance the two bit-strings 1100 and 0100. To make the second equal the first, we have to change the most significant bit, the third significant bit and the least significant.

Symbolic Similarity

Symbolic similarity can be either Ordered or unordered. In unordered symbolic, one set up a similarity table an explicitly enumrate all the possible value combinations. In ordered symbols the idea is that the ordered relation between symbols can bse used to derive the similarities. if we say that $small < medium < big$. This will imply that $sim(small, medium) > sim(small, big)$. In order to derive an adequate symbol ordering, one should start with assigning an Integer value to each symbol. The local similarity is the computed like for numerical values. For instance in the example above, we could assign very small(1), small(2), medium(3), big(4) $\rightarrow sim(verysmall, small) > sim(small, medium)$

Numerical Similarity

There are several common approaches in the field of Case Based Reasoning to calculate the similarity between numerical attributes. First we have the City block Metric, see formula 2.6

$$dist(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{2.6}$$

This is a simple numeric similarity that is applicable for k-dimensions. It is always greater or equal to zero. It constitutes that if the distance between x and y equals zero, then the values are the same.

Another common approach is to measure the euclidean distance between the numerical values, see formula 2.7

$$dist_{Euklid}(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.7}$$

The main difference between the euclidean distance vs the city block is that the euclidean calculates the similarity by taking a straight line between the points, as opposed to the city block that calculates one dimension at the time, hence the name city block as you have to go one road at the time around the blocks.

Taxonomic Similarity

The idea with taxonomic similarity is to have a taxonomic tree structure that represents some hierarchical general concept. The intention is that when you traverse the tree following a branch, it will point to more specific objects within the parent. This leads to having larger similarity values for child nodes. The overall similarity between two leaf nodes is given through the similarity value for the closest common parent node.

The simplest distance measure is called the graph distance, where you look at the deepest common predecessor for a given node u and v. This predecessor is called Deepest Common Predecessor.

$$d_{gr}(u,v) = length\,of\,shortest\,path\,from\,u\,to\,v. \tag{2.8}$$

Another similarity measure within the taxonomic similarities domain is the Wu-Palmer metric introduced in (Wu and Palmer, 1994). Formula 2.9 measures the depth for two given concepts c1 and c2. This is a path-oriented similarity, meaning that it relies on paths of a structure. In this case would be a taxonomy. The similarity between two leaf nodes is computed by the similarity value of the deepest common predecessor.

$$simWuPal(c1,c2) = \frac{2 \cdot depth(DCP(c1,c2))}{depth(c1) + depth(c2)} \tag{2.9}$$

Object Oriented Similarity

In order to derive the similarity of complex objects, there are two aspects to consider. The first has its origin in defining the attributes that defines the object, and the second is based on the classes that the objects can be located.

Introduced in in section 2.3.3.1, Object-Oriented representation may have relational attributes, for instance the "part-of" and/or attributes that has more than one value.

When dealing with similarity measure for object oriented similarity, we can consider it as an two step process. First we have what is called an Intra-class similarity. This similarity looks at the common properties between two objects. In Intra-class similarity the similarity is computed by taking the most specific common class between the objects. The similarity is isolated based on the attributes in the class. Since the two objects have its origin from the same class, local or object similarities can be computed for all the attributes. Each of the local values are then aggregated to compute the intra-class similarity. This can for instance been done by a weighted sum. The resulting values are aggregated to compute the Intra-class similarity, see section 2.3.4.3 local-global-principle. It is not sufficient describing the similarity between two objects alone. This is because not all the relevant properties can described properly this way. Therfore, we introduce the second step, the Inter-class similarity. The idea of Inter-class similarity is to measure the similarity between objects in the hierarchy without measuring the similarity for their respective attribute-values. The concept is quite frankly straight forward; It's about measuring how many values the two objects have in common. Therefore it is analogous to the similarity in taxonomies. The deeper the objects are located in the hierarchy, the more values they have in common.

2.3.4.5 Global Amalgamation Measure

Amalgamation Functions are global measures that aggregates the local measures. There are several approaches that can be used. An approach is the weighted average:

$$F(S_1, ..., S_n) = \sum_{i=1}^{n} W_i \cdot S_i \;\; with \;\; \sum_{i=1}^{n} W_i = 1 \;\; and \;\; S_i = sim_i(q_i, d_i) \tag{2.10}$$

This formula includes weights to distinguish the importance for each local similarity. A generalization of this equation is shown below

$$F(S_1, ..., S_n) = \sqrt[\alpha]{\sum_{i=1}^{n} W_i \cdot (S_i)^{\alpha}} \quad with \quad \alpha \in R^{+}, \sum_{i=1}^{n} W_i = 1 \tag{2.11}$$

Another popular measure is the generalization of the weighted hamming measure. This generalization is possible when you allow an arbitrary range of attributes and local similarity measures. When we define $sim = (sim_i, ..., sim_n)$ as a measuring the vector, we can then define the generalized Hamming measure as follow

$$H_{\omega,sim}((a_1, ..., a_n), (b_1, ..., b_n)) = \sum (\omega_\iota \cdot sim_i(a_i, b_i) | 1 \leq i \leq n). \tag{2.12}$$

Richter and Weber (2013b) explains that these measures assumes that attributes are independent. It is said to be very efficient when it comes to computing and easy to learn.

2.3.5 Reuse and Adaption

The process of reusing solutions generally requires a change of previous solutions, which in turn requires actions. This is called change adaption. Richter and Weber (2013b) explains that Actions that causes changes are usually described by rules that explains which certain conditions that adaptions are possible or recommended. Adaptions are in general special kinds of actions that are formally described by rules. These descriptions encompasses types of input and output, side effects when performing the action and preconditions that have to be met to pursue the action. There are two ways to adapt solutions. Either by transforming the previous solution, or use the strategy from the previous solution. Adaption is performed by actions that change a given model.

To use adaption systematically, a rigorous formalism is required. A rule has the form

$$\phi_1 \wedge \phi_2 \wedge ...\phi_n \Rightarrow Action.$$

Rules are distinguished in two different kind of forms. The first are completion rules. These type of rules adapt case description, most particulary in the query. Second are adaption rules that adapt the solution.

The adaption problem can be eased by reducing the need for adaptation. This can be done by improving the representation and/or retrieval of more relevant/similar cases where less adaptation is required.

2.3.6 Conversational Case Based Reasoning

Described in (Aha and Breslow, 1997), Conversational Case Based Systems is a powerful approach that do iterative interaction with a user in order to solve a query. This query is composed by a set of questions that are selected and answered by a user during this conversation. These systems are abbreviated further as CCBR. Aha and Breslow (1997) defines a good CCBR library as a system that compromises high precision and efficiency. More detailed we can say that if the retrieved case's actions solves the user query, the precision is considered high. Efficiency on the other hand, is defined as the magnitude of questions required before a case is considered solved.

The following guidelines are proposed in (Aha and Breslow, 1997) to produce a CCBR Library:

1. Reuse questions when possible
2. Order context questions before detail questions
3. Eliminate questions that do not distinguish cases
4. Ask for only one thing in a question
5. Use a similar, short number of questions per case.

However, Aha and Breslow (1997) points out that some of the bullet points may conflict. Cases should be be described by a large set of questions even though the reuse of questions is considered important. The outcome may therefore yield in inefficient retrieval. This is especially important when dealing with a CCBR that handles a vast amount of questions and cases.

Richter and Weber (2013a) states that when human experts try to solve different problems, they are not asking a large set of questions to elicit the problem. Instead human experts ask a few questions at a time. Figure 2.2 illustrates a flowchart that shows a general interaction between a user and the conversational CBR system. The user starts entering a brief description of the problem. Based on this brief description, the system captures the values and tries to populate a new problem and present this to the user. From there, the user is prompted to answer one or more questions. If the user is satisfied, the case is being reused, if not, the user will be prompted to answer a new question until satisfaction.

2.3.7 CBR Frameworks

CBR frameworks such as MyCBR and JColibri have been developed for the purpose of simplifying the the process of creating CBR systems. Creating a CBR system from scratch can be a daunting task and a lot of the industrial CBR systems running today use and/or extend some CBR framework. In this paper we also opt to use a framework in order to limit time spent developing. The two main contenders are myCBR and jColibri. While both are offering some measure of similarity and support for different case representation, they do have their pro's and con's.

2.3.7.1 MyCBR

MyCBR is one of the more popular CBR frameworks. One of the most popular versions of myCBR is the old plugin for the protg ontology editor. The new version of myCBR is standalone, open-source and mostly written in java. The main motivation behind myCBR was the need for a tool for building prototype CBR applications in research and teaching, as well as some smaller industrial projects. The myCBR package consists of a SDK and a set of tools that simplify the development of CBR applications this is called the myCBR Workbench.

MyCBR Workbench

MyCBR workbench allows for both easy modelling of case representations and similarity measures. The case representations that are supported out of the box include an object oriented representation that can use both is-a and part-of-a relationships, this can of course be extended if the need arises. Furthermore the workbench provides a easy to use gui for setting up several types of similarity measures, including both symbolic, taxonomy based and string similarity. MyCBR Workbench also include tools that can be used to generate case representations from raw data.

MyCBR SDK

The myCBR SDK is written in java and allows for integration into other software and applications. The SDK provides interfaces for extending functionality beyond what is originally offered, such as adding new similarity measures.

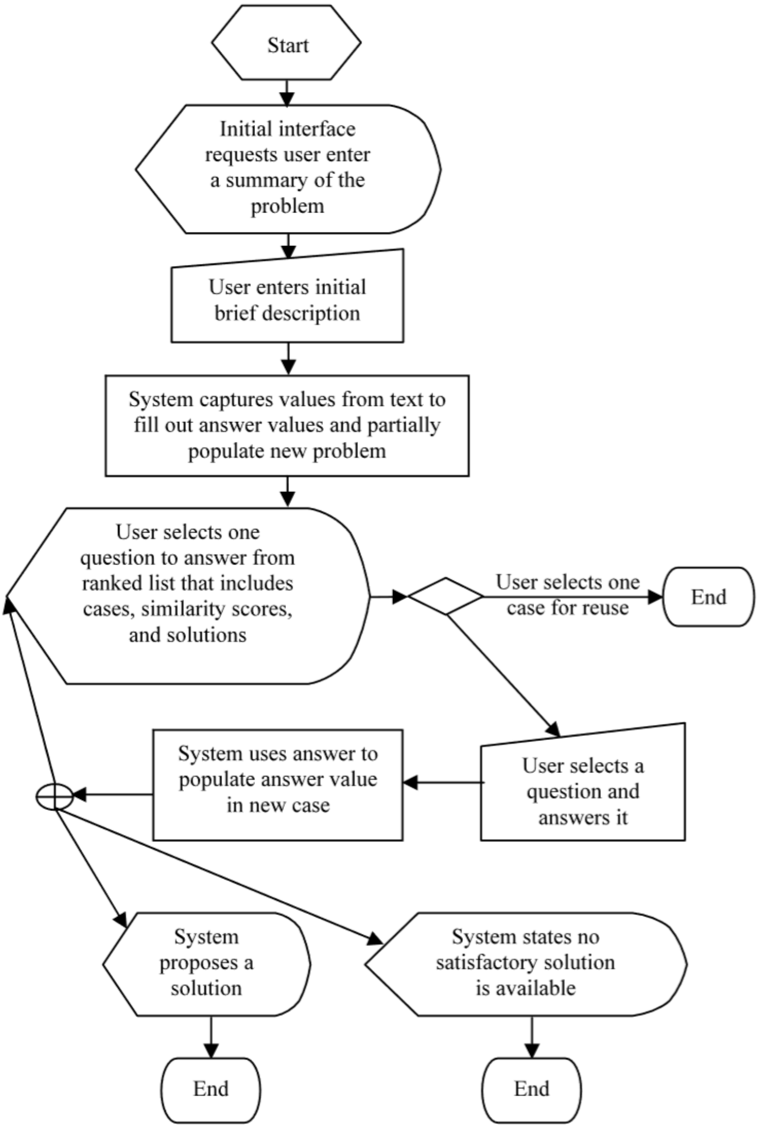


Figure 2.2: Flowchart showing simple interaction over CCBR Richter and Weber (2013a)

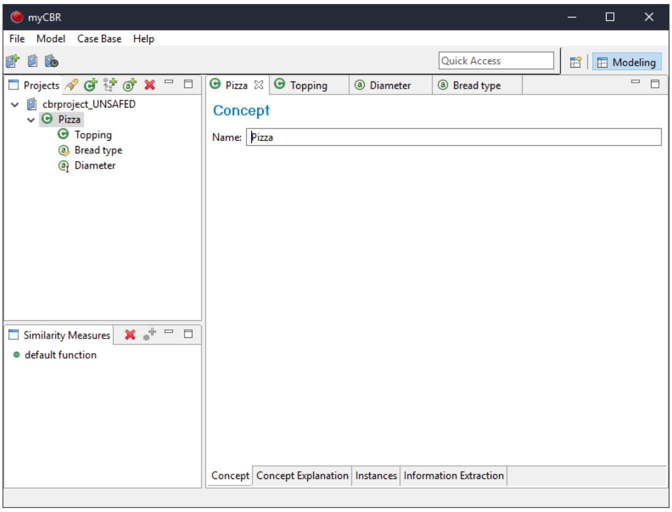


Figure 2.3: MyCBR Workbench, editing a pizza case structure

Using myCBR and myCBR Workbench

The myCBR Workbench supports creation and maintenance of knowledge models that can be used in CBR systems. There are two main parts of the myCBR Workbench, a view for modelling the knowledge base and a view for manipulating the case-base.

The workflow in myCBR Workbench can be described the following way. First you create your knowledge-model/case structure, then you define a vocabulary. The last step is creating local similarity measures for each attribute in a concept and a global similarity measure for each concept description. The case structure itself is described by these concepts, these concepts support both is-a and part-of relationships, this means that a concept can contain another concept as an attribute as well as normal datatypes. An attribute can have the following datatypes: Integer, Double, Boolean, Interval, String and Symbol. MyCBR provides one or more similarity measure for all of these different datatypes.

Similarity is set up locally for each attribute of a concept and a global amalgamation is done for each concept, gathering the similarity data of all its attributes using a function like the weighted sum for instance.

The myCBR Workbench also supports adding instances from a CSV file. These instances can then be reviewed and added to a Case-base.

2.3.7.2 COLIBRI

COLIBRI provides an infrastructure for developing CBR systems. The main goal of COLIBRI is to function as an environment where users can share and contribute their efforts creating CBR applications so that this can be used towards developing the field of CBR itself. Users can contribute with code and designs that can then be reused by others.

COLIBRI Architecture

The two main layers of the COLIBRI architecture consists of a set of tools that are designed to assist users in implementing and sharing new CBR systems and components, these tools have recently been integrated into its own COLIBRI Studio development environment. The reference implementation of COLIBRI is called jCOLIBRI framework. The jCOLIBRI framework is implemented in Java and is the collimation of several years of experience in CBR systems development shared by a multitude of users.

jCOLIBRI Framework

The jCOLIBRI framework itself does not include any visual tools but contains most of the code needed to create a wide range of CBR systems: Traditional CBR systems, Textual CBR, Knowledge-intensive CBR, Data-intensive, Recommender systems and distributed CBR systems. It does how even include a small set of tools that can be used to evaluate and maintain a case-base, as well as a tool for visualizing the case-base. A large part of the components in jCOLIBRI have been contributed through research projects.

After jCOLIBRI was sufficiently mature, a set of graphical development tools were developed. These tools are now enclosed in a IDE package that is called the COLIBRI Studio IDE GAIA (2012). The COLIBRI Studio IDE is built as a extension to the popular Eclipse IDE for Java.

COLIBRI Studio

COLIBRI Studio is the implementation of the top layer in the COLIBRI platform GAIA (2012). It is built on top of the jCOLIBRI Framework and enables users to generate CBR systems without having to deal with source code. It does this by providing a set of visual builder tools. COLIBRI Studio comes as an extension to the popular Eclipse IDE, this means that all the project and java source code will be handled by Eclipse. The COLIBRI Studio tools include a Case Designer, Case-Base Selector, Plaintext and database connector and a Similarity tool.

- **The Case Designer** is used to visually set up a case representation. The standard case structure in jCOLIBRI include a description, solution, result and a justification. Each of these can consist of one or more attributes, and these attributes can also contain a value or more attributes.

- **The Case-Base Selector** is used to show and manipulate the case bases that are in memory.

- **The Plaintext and database connector** is used to load cases from textfiles or databases. When loading from a text file it is important that the text file contains attributes for the cases and that these are separated by comma, semi-colon or a custom separator.

- **The Similarity tool** can be used to configure similarity functions for each of the attributes in a case. This tool and these similarity measures are mostly used alongside a k-Nearest neighbour retrieval method, if another retrieval method is used then this tool might not be relevant.

Template-based design

COLIBRI Studio supports something that they have called template-based design. The template-based design mechanism allows you to reuse CBR designs that are tried and true as well as generating a CBR system based on a template that you can design yourself. The template-based design paradigm is similar to case based design itself in that expert users design cases "templates". These cases "templates" can be retrieved and adapted by normal users into CBR systems, if these normal users create new methods then these can be offered as options for the template. This allows for collaboration between multiple users and also makes CBR available to non academic users.

When creating a new system from an existing template you first get to pick the template family, here you can pick between Recommender system, Conversational, Knowledge-intensive and many more. After you have selected a family you get to pick the type of CBR within the family, you are also given the option to view flow graphs of the different systems.

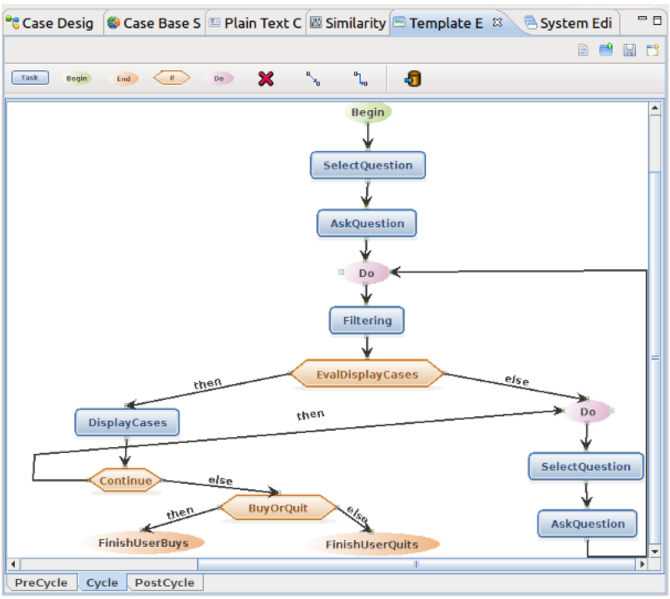


Figure 2.4: COLIBRI Studio from GAIA (2012)

Each template is composed of several task that define the workflow of the system. The functionality defined by each task has to be implemented by using one method of the jCOLIBRI framework. In case that jCOLIBRI will not provide the desired method, users can program it directly in Java (using Eclipses facilities) and integrate it in any template. Roth-Berghofer et al. (2013)

2.3.7.3 Using myCBR knowledge models with COLIBRI Studio

In their paper, Roth-Berghofer et al. (2013) look at using myCBR and COLIBRI Studio in combination. Specifically to use myCBR Workbench to model the case structure and to then use this case structure in a jCOLIBRI based CBR System. This provides the best of both worlds as we get the flexible knowledge modeling of myCBR Workbench combined with the template engine and code generation of COLIBRI Studio.

The workflow when using both myCBR and jCOLIBRI can be described as follows:

- 1. Knowledge modeling in myCBR Workbench(Case structure and similarity)
- 2. The working knowledge model is then exported to COLIBRI Studio.
- 3. COLIBRI Studio generates a CBR system, manually or using a template. The knowledge model from myCBR is integrated into the system.

2.4 Semantic Web

This section encompasses the concept of Semantic Web. The concept section of semantic web, RDF, OWL, Ontology and Knowledge Graph have been included from our previous preliminary work as it is a necessity to understand our use of case representation, and the use of external semantic resources. The new sub sections that will be examined is the online semantic resource section that discusses the ESCO ontology.

2.4.1 The Concept

Berners-Lee and Hendler (2001) describes that ”The Semantic Web is not a separate web, but can be viewed as an extension of the current Web in which information has a well-defined meaning. This section describes the Resource Description Framework(RDF), OWL, SPARQL, the meaning of an ontology, and knowledge graphs. This section will explain their key concept and terms.

2.4.2 RDF

Classical semantic network approaches lacks formal syntax and semantics. Therefore, RDF(Resource Description Framework)[4] was proposed and has become a standard data model on the web. RDF allows to use URIs[5] to name the relationship between concept of ”things” and the two ends of the link. This is referred to as a ”triple”Consortium (2004). Triples consist of a subject, predicate and an object, see 2.5.

In (Pan et al., 2017), a ”is-a” relationship can be represented by a subClassOf property. This provides clearly defined semantics for relational presentation. However, it does not allow a user to define concepts, therefore it does not address all limitation in semantic networks. This is being addressed by the Web Ontology Language(OWL), explained in the section below.

[4]https://www.w3.org/RDF

[5]https://www.w3.org/Addressing/URL/uri-spec.html

Figure 2.5: A triple consisting of a subject, predicate and object

2.4.3 OWL

The standard ontology language is called OWL[6] (Web Ontology Language). This is a Semantic Web Language designed to represent rich and complex knowledge about things, relation between things and grouping of things. It can define classes, object properties, data properties and much more. This is the W3C standard for defining vocabularies for RDF graphs Pan et al. (2017). OWL is a logic-based language that is computational. It allows you to express knowledge in a form that can be understood and exploited by computer programs. It is based on formal knowledge representation called Description LogicBaader et al. (2003), a powerful language used in artificial intelligence. It is used to both describe and reason with and around relevant concepts.

2.4.4 SPARQL Query Language

In order to extract results from the knowledge graph and query the ontologies you need a query language. SPARQL[7] is a query language designed to retrieve these results. The advantage of using the SPARQL query language is that you can express queries across a great variety of data sources. It also has capability for querying both required and optional graph patterns, in addition to conjunction and disjunction. The results obtained for the SPARQL query can both be a result set in a structured table or be presented as composed RDF graphs.

2.4.5 Ontology

An ontology defines a common vocabulary for researchers who need to share information in a domain. One of the main purposes by creating an ontology is being able to answer different sorts of competency questions. This are questions you want the ontology to be able to answer. Example of such questions can be ”Who has experience with object oriented programming?” or ”Which candidates has C# as their favourite programming language?”. This could be examples in a software developer ontology. An ontology usually has one or more class taxonomies. These taxonomies defines different concepts in the ontology. An ontology can be modelled with the Web Ontology Language ”OWL”, introduced in 2.4.3.

[6]https://www.w3.org/OWL/
[7]https://www.w3.org/TR/rdf-sparql-query/

2.4.6 Knowledge Graph

According to Pan et al. (2017), the term ”Knowledge Graph” became a well known term back in 2012 when Google implemented the use of knowledge graph in their search engine. This allowed people to search for places, people and things as opposed to just matching strings in the search queries. A knowledge graph consists of a set of interconnected typed entities and their attributes. The difference between a knowledge graph and linked data is that a knowledge graph is a local graph, and linked data is a network of connected graphs.

2.4.7 Online semantic resources

Following the invention of the world wide web there has been a lot of focus on not only linking data but to add meaning to these links. Semantic web can be seen as the web 3.0 where data is not only linked, but the links themselves describe a relation. In the recruitment domain there are some existing ontologies available to us. Among these we have the ESCO ontology, developed and maintained by a European commission. We also have a skill ontology developed and maintained by DICE. The biggest ontology resource available is dbpedia. Dbpedia is comprised of mostly general knowledge and linked data derived from Wikipedia. We will further explain the ESCO Ontology as it is an important factor in our architecture, explained in 3.

2.4.7.1 ESCO

The ESCO ontology has been specifically developed to aid in developing suggestion systems, job search algorithms and job matching algorithms. The ESCO ontology includes modules that contain elements such as occupations, knowledge, skills, competencies and qualifications. This is then combined with the hierarchy specified in the International Standard of Occupations (ISCO) to then form a useful ontology which can be used for classification purposes. The ontology supports all 24 official EU languages in addition to Icelandic and Norwegian.

The ESCO data model

The data model of ESCO is structured into three main pillars.

- The occupations pillar;
- The knowledge, skills and competencies pillar;
- The qualifications pillar.

As you can see from the figure 2.6, the pillars are interlinked. Skills can be both attributed to an occupation as required skills, and to a qualification. This makes it easy to query for skills required for a certain occupation, and also makes it easy to query for occupations that require a certain skill.

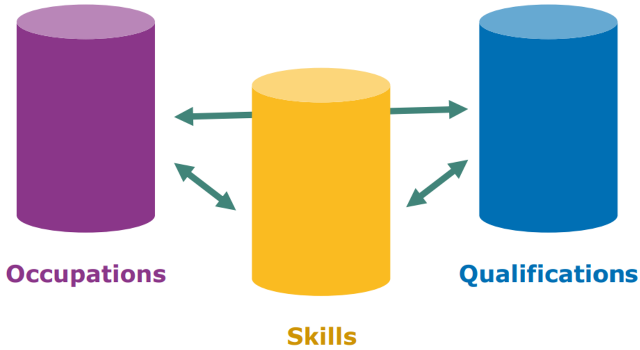


Figure 2.6: The three pillar structure of the ESCO data model EU-Commission (2017b)

The occupations pillar

The occupations pillar should not be confused with jobs, jobs are not covered by ESCO. An occupation is a grouping of jobs that require the same type of skills and involve similar tasks, while a job is a set of tasks and duties meant to be excecuted by one person ILO (2008). The first version of the ESCO classification features 2950 different occupations. Each occupation is linked to their own set of metadata as well as as an ISCO-08 code. The ISCO-08 code can be used as a hierarchical structure for the occupations pillar. The full view of the metadata of an occupation in ESCO can be viewed in the following list.

- Description
- Alternative label
- Regulatory aspect
- ISCO-08 code
- Definition
- Scope notes
- Hierarchy
- Narrower occupations
- Essential skills and competences
- Essential Knowledge
- Optional skills and competences
- Optional Knowledge
- Status

- Replaces
- Replaced by
- Concept URI

The ISCO-08 standard

The ISCO-08 standard by ILO (2008), divides all occupations into ten major groups. In the current version of the ESCO classification, ISCO-08 provides the top four levels for the occupations pillar. ESCO occupations are located at level 5 and lower.

- 1 Managers
- 2 Professionals
- 3 Technicians and Associate Professionals
- 4 Clerical Support Workers
- 5 Services and Sales Workers
- 6 Skilled Agricultural, Forestry and Fishery Workers
- 7 Craft and Related Trades Workers
- 8 Plant and Machine Operators and Assemblers
- 9 Elementary Occupations
- 10 Armed Forces Occupations

The skills pillar

The skills pillar consists of both knowledge, skills and competencies. There are in total about 13492 skill concepts in ECSO v1. The ESCO classification uses the following definitions for skill, knowledge and competence, taken from European-Parliament (2008):

- **Knowledge:** The body of facts, principles, theories and practices that is related to a field of work or study. Knowledge is described as theoretical and/or factual, and is the outcome of the assimilation of information through learning.
- **Skill:** The ability to apply knowledge and use know-how to complete tasks and solve problems. Skills are described as cognitive (involving the use of logical, intuitive and creative thinking) or practical (involving manual dexterity and the use of methods, materials, tools and instruments).
- **Competence:** The proven ability to use knowledge, skills and personal, social and/or methodological abilities in work or study situations, and professional and personal development.

Like for the competence pillar, each skill concept in the skill pillar contains a set of useful metadata that both describe the skill and its context. The data includes the type of skill and if there exists, a relation to a broader skill. The following is a list of the possible metadata that ESCO v1 currently supports.

- Description
- Definition
- Scope notes
- Skill type
- Broader skills/competences
- Narrower skills/competences
- Essential skills and competences
- Optional skills and competences
- Essential skill/competence of
- Regulatory aspect
- Replaces
- Replaced by
- Concept URI

As you can see from the list of skill meta-data 2.4.7.1, the relationship between knowledge, skills and competences has been captured to a certain degree. An example mentioned in EU-Commission (2017b) ”Procedural law is essential for pleading a case in court; labour law is optional for pleading a case in court”, illustrates how this meta-data structure can be used to define these relationships.

The in addition to the meta-data mentioned above, the skills are assigned a reusability level. The reusability level can be either ”transversal”, ”cross-sector” or ”sector-specific”. Transversal skills are relevant to a broad range of occupations and are often referred to as ”core skills”, ”memorise information” is an example of such a skill. ”Cross-sector” skills are relevant across several economic sectors, but not as general as the transveral ones, an example of a cross-sector skill is ”drive vehicles”, which is a skill used in both law enforcement and public transport for example. A ”sector-specific” skill is used in a specific sector, but can be used in a multitude of occupations whitin that sector. Programming in Java is seen by ESCO as a sector specific skill, specific to the ICT sector and used in some of the occupations related to this sector.

The skills hierarchy is divided into five major categories, thinking; language; application of knowledge; social interaction; attitudes and values. Dividing the skills into this hierarchy makes it easier to use esco in certain situations, such as in a CV creation situation. An example could be if someone was filling out a skill section for language and searched for Chinese, restricting to language type skills would limit the suggestions to ”read, write Chinese” instead of getting suggestions like ”traditional Chinese medicine”.

The qualifications pillar

The qualifications pillar is intended to become a comprehensive listing of all the qualifications that are relevant for the European labor market. ESCO includes qualifications both directly and indirectly. The indirect-inclusion is based on data gathered from EU countries' national qualification databases. The direct-inclusion will include qualifications that are not part of these databases but may still be relevant. ESCO v1 contains only a small sample of international qualifications.

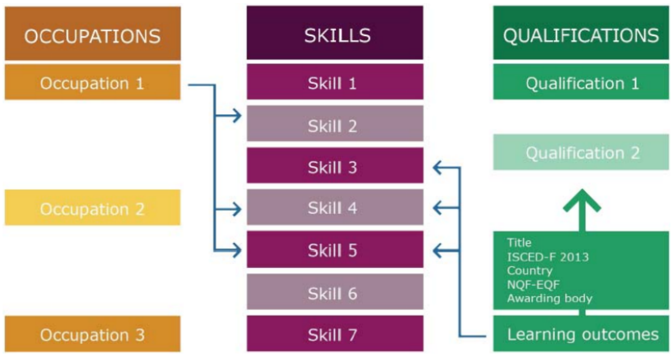


Figure 2.7: The interconnection between skills, occupations and qualification in ESCO EU-Commission (2017b)

The figure 2.7 illustrates how the skills interconnect the qualifications and occupations. Some occupations can have mandatory skills on a national level, and/or skills that are optional. As an example, the "driving vehicle" skill is an essential skill of a delivery driver but an optional skill of a vehicle technician.

2.5 Software Architecture

In this section we will explain some details regarding describe some design patterns, tactics and architectural patterns that is needed to understand the essence of the architecture introduced in 3.

2.5.1 N-Tier Architecture

An N-Tier Architecture of consists of minimum three tiers or layers. These three typical tiers are:

- Presentation Tier
- Middle Tier
- Data Tier

2.5.1.1 Presentation Tier

The presentation tier is the layer where users interacts with an application. This may involve application logic as well. The presentation layer communicates with the Middle Tier and does not access the data tier directly.

2.5.1.2 Middle Tier

The middle tier often involves a Business logic Layer with the responsibility for handling data communication data between the presentation layer and the Data Access Layer. In addition, the Business Logic Layer, abbreviated as BLL, can be considered a guard that handles validation and rules.

2.5.1.3 Data Tier

The Data Tier can be considered the server that stores an application's data. Typically this tier consists of the layer that handles the interaction with the server to both store and request data and the database layer, that may be an SQL Server containing different data storage endpoints.

Microsoft (2015) explains that when we separate application components into separate tiers, you increase the applications maintainability and scalability. This is because when seperating out, one enables new technology to be adopted easier since each tier is unaffected bu each other. In addition with regard

Separating application components into separate tiers increases the maintainability and scalability of the application (Sheriff, 2002).

2.5.2 Evaluation Formulas

Baeza-Yates and Ribeiro-Neto (2008) describes these measures as the gold standard when evaluating ranking systems. Below we describe the different formulas and their respective definitions. First we explain the Okapi BM25 which we will use as a baseline in our evaluation explained in 5. Afterwards we describe Precision and Recall that are used to measure the system for top-k results in the ranking. After that we will explain the F-Measure that is a harmonic mean based on the latter two and E-Measure that allows to weight either one more than the other. At the end we examine the Spearman's rank correlation coefficient that we use in the real world scenario experiment, explained in 5.4.

2.5.2.1 Okapi BM25 Ranking Formula

$$B_{i,j} = \frac{(K_1 + 1)f_{i,j}}{K_1[(1-b) + b \times \frac{len(d_j)}{avg_doclen}] + f_{i,j}} \tag{2.13}$$

$$sim_{BM25}(d_j, q) \sim \sum_{ki[q,dj]} B_{i,j} \times log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right) \tag{2.14}$$

From Information Retrieval, Okapi BM25 is a ranking formula that positions as an extension of the probabilistic model. This formula has been derived by conducting different experiments through the Okapi System Robertson et al. (1993). The key by using the BM25 formula is that it is fully automated. This is because it can be computed without any relevant information provided to the user in advance. Baeza-Yates and Ribeiro-Neto (2008) tells us that this method has been used as the baseline for evaluating new ranking methods. Equation 2.5.2.1 presents the equation.

2.5.2.2 Precision and Recall

In information retrieval explained in Baeza-Yates and Ribeiro-Neto (2008) Precision and Recall are presented as retrieval metrics.

Precision is defined as the fraction of the retrieved documents, referred to as set A, that are relevant.

$$Precision = p = \frac{|R \cap A|}{|A|} = \frac{RelevantAndRetrieved}{Retrieved} \tag{2.15}$$

Recall is the fraction of the relevant documents, referred to as set R which has been retrieved.

$$Recall = r = \frac{|R \cap A|}{|R|} = \frac{RelevantAndRetrieved}{Relevant} \tag{2.16}$$

2.5.2.3 Mean Average Precision(MAP)

$$MAP = MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{2.17}$$

The mean average precision for a set of queries is the mean of the average precision score for each of the queries.

2.5.2.4 F-Measure and E-Measure

$$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}} \tag{2.18}$$

In information retrieval, F-Measure is described as a harmonic mean. It is an approach to combine both precision and recall into a single number. This metric assumes a great value if and only if both recall and precision are great. In equation 2.18, r(j) is the recall at the j-th position in the ranking. P(j) is the precision at the j-th position in the ranking, giving F(j) the harmonic mean at the j-th position Baeza-Yates and Ribeiro-Neto (2008).

The E-Measure is based on the F-Measure, but it allows the user to weight either precision or recall more, depending on the ranking system.

$$E = \frac{(\beta +^2)PR}{\beta^2 P + R} = \frac{(1 + \beta^2)}{\frac{\beta^2}{R} + \frac{1}{P}} \tag{2.19}$$

if $\beta = 1$, it equals to F-Measure, $\beta > 1$ weight precision more, $\beta < 1$ weight recall more.

2.5.2.5 Spearman's Rank Correlation Coefficient

Introduced by Charles Spearman Spearman (1906), the Spearman's Rank Correlation tries to find out whether it is a correlation between two sets of variables. The Spearman's Rank Coefficent is useful when you have data that has been scored. The equation is presented below

$$r_s = 1 - \frac{6\sum d^2}{n(n^2-1)} \tag{2.20}$$

The values for r_s are listed below:

- **Approximating 1:** If you end up with an Spearman's rank coefficient that approximates 1, it means that for each value of x it is an increase in y. This means that it has a strong positive relationship. When the correlation equals 1, it means that the agreement is perfect.
- **Approximating -1:** If the value however approximates -1, it means that for each value x there is a decrease in y. When the correlation equals -1, it means that the disagreement between the two rankings is perfect.
- **Zero Value:** If the value is 0 it means that the rankings are completely independent.

$$D = \sum d^2 \tag{2.21}$$

$$E(D) = \frac{n(n-1)(n+1)}{6} \tag{2.22}$$

$$V(D) = \frac{n^2(n+1)^2(n-1)}{36} \tag{2.23}$$

$$Z = \frac{D - E(D))}{\sqrt{V(D))}} \tag{2.24}$$

We can find the Z-score for a normal distribution of the spearman coefficient by taking the sum of the squared distance D 2.21, the expected value of D 2.22, the variance of D 2.23 and then apply these in 2.24. This gives a Z score which approximately follows the standard normal distribution under the null hypothesis that there is statistical independance (rho=0).

2.5.2.6 Discounted Cumulated Gain

Discounted Cumulated Gain aims to take into consideration multiple level relevance instead of binary relevance such as the precision and recall metrics. The whole idea and advantage with DCG is that highly relevant documents that appears in the lower search result of the list can be penalized. This is because the graded relevance value is reduced logarithmically in proportion to the ranking position.

Oates (2006) tells us that in situations where highly accurate evaluation is required in the comparison of retrieval systems that have similar retrieval quality, the DCG metric is an interesting alternative to the more established precision and recall.

The formulala for the DCG is given as:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2(i+1)} \tag{2.25}$$

Computing the DCG alone does not necessarily say much. Oates (2006) says that since the normal DCG is not computed relatively to any baseline, it may be confusing to use it directly in the comparison between two distinct retrieval algorithms. It is therefore important to normalize the DCG in order to achieve a percentage of how well the gain is. We therefore normalize the DCG. The normalization formula is given as:

$$nDCG_p = fracDCG_pIDCG_2 \tag{2.26}$$

where IDCG is the ideal of the discounted cumulative gain which means the perfect rank based on the relevance. The ICDG formula is given as:

$$IDCG_p = \sum_{i=1}^{|REL|} \frac{2^r el_i - 1}{log_2(i+1)} \tag{2.27}$$

Chapter 3

Architecture

In this chapter we present our architecture. Designing a good architecture is an art of picking the right compromises. The following sections will give some insight into these compromises and why they were made.

3.1 Introduction

In chapter 1 we explained that we wanted to create an artifact based on the design and creation research strategy in order to contribute with new knowledge within the domain of candidate ranking in recruitment. This chapter will detail the architecture used when for this artifact. UML diagrams will be presented and used to describe the implementation of the system. Different architectural blueprints will be used to present the system from different angles. We recommend readers to use chapter 2 as a reference when reading this chapter.

A software architecture provides a high-level structure over the software that can present different viewpoints of the system. Using a model that are composed of multiple views or perspectives have shown to provide stakeholders and readers the insight needed Booch (1994). Therefore, we have created architectural blueprints inspired by the ”4+1” View Model introduced in Kruchten (1995). The reader can expect some of these viewpoints modelled with UML:

- A use case diagram presents the scenarios in terms of user functionality that the system must account for.
- Both class diagrams and a sequence diagram presents logical viewpoints.
- Activity diagrams and state machines describes the system processes.
- Component diagrams and package diagrams are included in order to show a development view of the system.
- A deployment diagram presents a physical view over system.

Figure 3.1 shows the deployment diagram over the CRS, that presents the physical view over the CRS. This view is concerned with presenting the software components and the physical nodes they are deployed to. Therefore, this view displays the system from a high-level perspective. Some of the design aspects have been influenced by the possibility of this system going into production in the future. Certain steps are therefore taken to insure good maintainability and the possibility of future upgrades. Measurements have been taken to account for this. Section 3.3 presents logical viewpoints over each of the tiers. The multi-tiered architecture that consists of tiered interconnected modules. Logical viewpoints are included for each layer in this section, except for the CBR-System which is dedicated to section 3.5 CBR Architecture. Class diagrams share the same color as its respective tier presented in figure 3.1.

Storing and interaction between data is an important part of the CRS. We have therefore dedicated an own section that explains the database architecture and the storage container. The database architecture can be examined from the Entity relationship model presented in section 3.4 Database Architecture and Storage Container.

Section 3.5 examines all the details regarding CBR that are utilized in the CRS. Each cycle presented in the 2.1 are explained. This includes retrieval techniques and case representations to mentioned some.

Section 3.6 Application Modules is dedicated to present our system modules. From 3.19 the user scenarios to be solved are presented. This section examines each of these modules, a development view for each module by using UML Component Diagrams, the motive for having them, and screen-dumps from the modules to assist the reader seeing the system from a user perspective.

3.2 Requirements

This section describes a set of requirements that we derived in advance for the system. These requirements contains both functional requirements, and non-functional requirements.

3.2.1 Functional Requirements

Functional requirements describes the systems embodiment and can be defined as the system requirements. We present a Use Case diagram to have an overview over the stakeholders interaction. Further, we explain user stories for the different actors involved. These user stories constitute our functional requirements.

3.2.2 Non Functional Requirements

The non-functional requirements describes how the system is supposed to implement the functional requirements. We slice these further down to two types forrequirements, organizational requirements and external requirements

3.2.2.1 Quality Attributes

Since we were dealing with a creation of a system that both could be deployed and consolidated for new changes, we wanted to focus on that the system was modifiable and deployable. This motivated in using the Multi-Tiered Architecture pattern described in 3.3.

3.2.3 Organizational Requirements

Organization Requirements describes the requirements that are a consequence of policies and procedures in an organization. This means process standards and implementation requirements. The two given are explained below.

3.2.3.1 Development

- Use Angular and typescript to control the dynamic for the application.
- Material Design Framework[1], CSS and HTML5 for designing the web UI.
- Visual studio Code and Visual Studio 2017 as code developing environments.
- Databases and Storage Containers shall be developed on the Azure Platform.
- MyCBR SDK

3.2.3.2 Code implementation

In order to achieve a readable and fine structured code, that can be reusable. The proposed system should follow a code convention. We propose the use of the following [2]

3.2.4 External Ethical Requirements

3.2.4.1 Ethical - Privacy and compliance with the GDPR regulations

HigherED is currently working on ways to integrate the GDPR regulations into the recruitment module that they are building. We will get personal access to much of this data during the spring semester and our proposed system will be integrated into their module. The integration into HigherEd and the potential of the system going into production imposes a stricter set of laws and regulations. The considerations and actions required in order to successfully deploy the system is discussed in chapter 4.

[1] Design Framework by Google, see https://material.angular.io

[2] https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions

3.3 Multi Tiered Architecture

From the elicitation of requirements discussed in the previous section, it motivated in using a multi tiered architecture as an architectural pattern for developing the system. This section therefore presents the Multi Tiered Architecture that the CRS is built upon. We will explain each tier and their qualities. Building the system by following an Multi Layered Convention means that the system will consist of different interconnected modules, as explained further in section 2.5.1. The CRS encompasses four tiers, whereas six layers are incorporated. Figure 3.1 presents a deployment diagram for the CRS. We present each tier and their corresponding layers. Class diagrams are included for each layer.

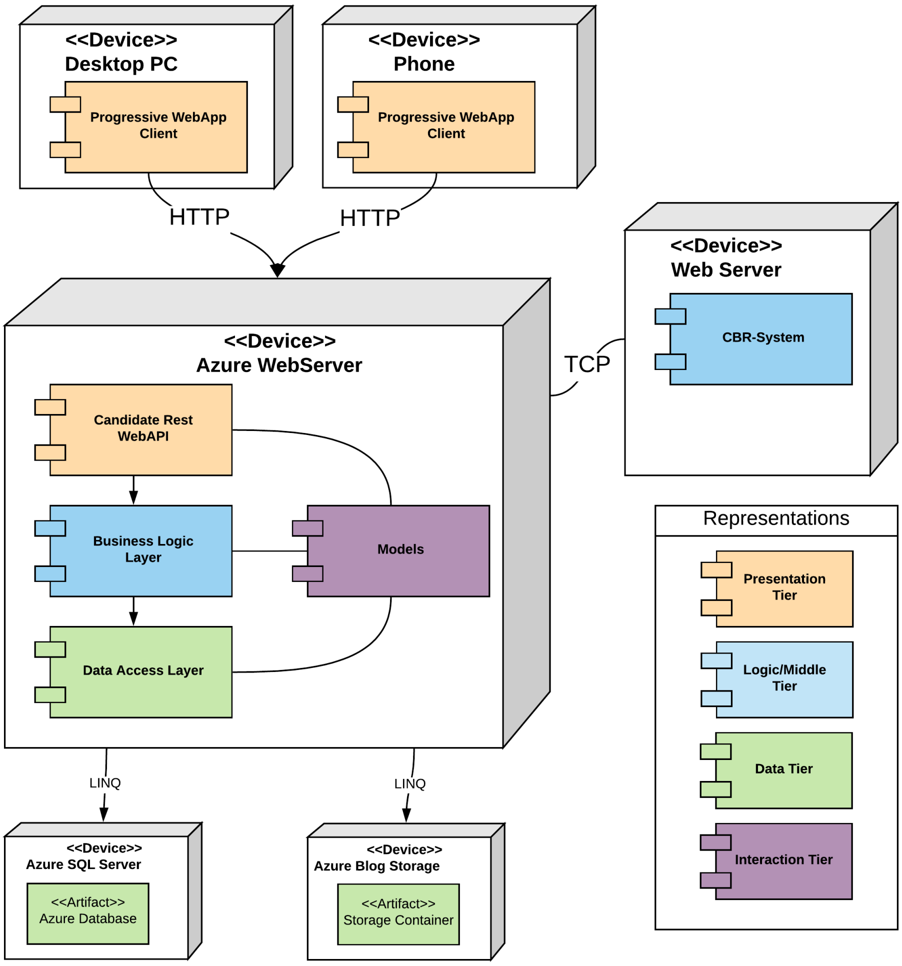


Figure 3.1: Deployment Diagram CRS System

3.3.1 Presentation Tier

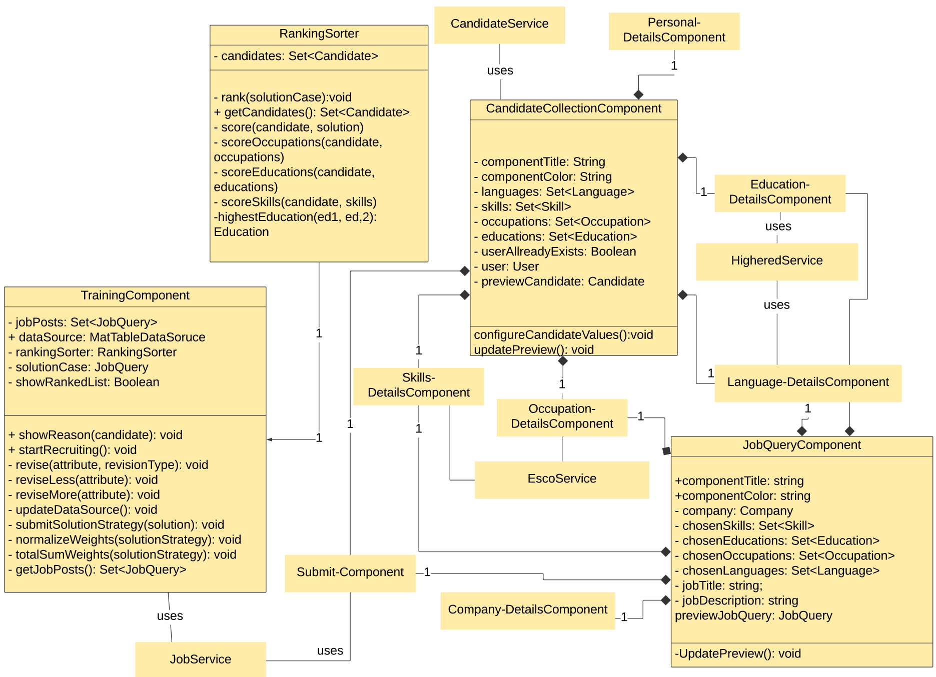


Figure 3.2: Class Diagram showing Services and Components in the Frontend layer

The presentation tier conveys the layers whom the users interacts with. From figure 3.1 wee see that we have implemented two layers into this presentation tier colored in beige. First, there's a standalone progressive web application developed. Since we utilized the Angular-5 Javascript, we took advantage of its component driven development approach. This means that certain parts can be reused and therefore it may reduce the bundle size of the application. this was important for us as we wanted to provide the users with a welcoming and responsive user interface. Figure 3.2 presents a class diagram over the parts that encompasses the components and services in the frontend layer. For better readability, the front end model classes are separated out and presented in figure 3.4. Services and detail components that are used by several of the main components are detailed in in figure3.3.

The main components are the TrainingComponent, CandidateCollectionComponent and the JobQueryComponent. Their properties are presented in figure 3.2. The TrainingComponent uses the RankingSorter class in order to provide runtime scoring of the candidates. The Rankingsorter includes methods to provide these scorings. A more rigorous explanation is presented in section 3.6.2. The TrainingComponent contains methods to update the candidate ranking list and revise methods to weight certain attributes more. It also includes method to normalize the weights, so the total sum weight sums up to a total of 100 %.

The CandidateCollection-Component is the main component for the DCM Module that we introduce in 3.6.1. It is responsible for handling the attributes received from the details components, such as skill and education details, presented in 3.3. The submit-Component is a shared component between the JobQueryComponent and CandidateCollectionComponent, responsible for calling the services to store data.The JobQuery-Component is responsible from handling new job advertisements. It behaves similar to CandidateCollectionComponent except it utilizes the Company-DetailsComponent as well and some differences in the design.

From 3.3 one can see several service-classes. These classes works as a data sharing point between components and they are responsible for handling secured http connection with the backend web api. For instance, the JobService has API rules for handling the connection with the JobQueryController presented in 3.5.

This is developed using a state of the art framework within the domain of JavaScript technology. It is also developed with Typescript, CSS and HTML. The second layer is the Candidate Rest Web API. This is a Rest API consisting of different controller classes with the objective to handle incoming HTTP requests from the web applications different modules.

The Progressive Web App consists of four modules, that we discuss in their respective sections below. These modules are the Data Collection Module, Training Module, Job Query Module and the Experiment Module. The latter is a module utilized in order to conduct the experiment regarding this research. The training module derives the Revision part of the CBR System. This is discussed further in 2.3.6.

3.3.2 Logical Tier

This Tier conveys the brains of our implemented system. These are colored in blue from figure 3.1. This tier consists of two layers. The first is the Business Logic Layer that is developed in C#, presented in figure 3.6 and the CBR System coded in Java. The latter has an own dedicated section presenting the architecutre, see section 3.5.

The business logic layer has the responsibility to validate incoming data and work as a general checkpoint between the Rest-API and the Data Access Layer. It is also responsible for establish a TCP-connection that enables secure exchange of data between the Web Servers. From figure 3.6 there are three classes and two interfaces. The CandidateBLL is responsible for storing and retrieving candidate data. It communicates with the respective attribute access layers for storing new attributes that the new candidate may have registered. The JobQueryBLL has the responsibility handling the new job advertisement that are being registered through the Job Query Module explained in section 3.6.3. The SocketCommunicator handles all data exchange between the Java Web Server that runs the CBR part of the system and the C# that runs the backend for the application.

3.3.3 Data Tier

The Data Tier consists of three layers, colored in green from 3.1. The Data Access Layer is responsible for handling the communication directly with the data endpoints. This is presented in figure 3.7.

From the figure there are a total of 7 DAL-classes and a Context Class. The reason for the CandidateRankingContext class is because we are utilizing a Code-First approach in

Figure 3.3: Class Diagram Services and Details components

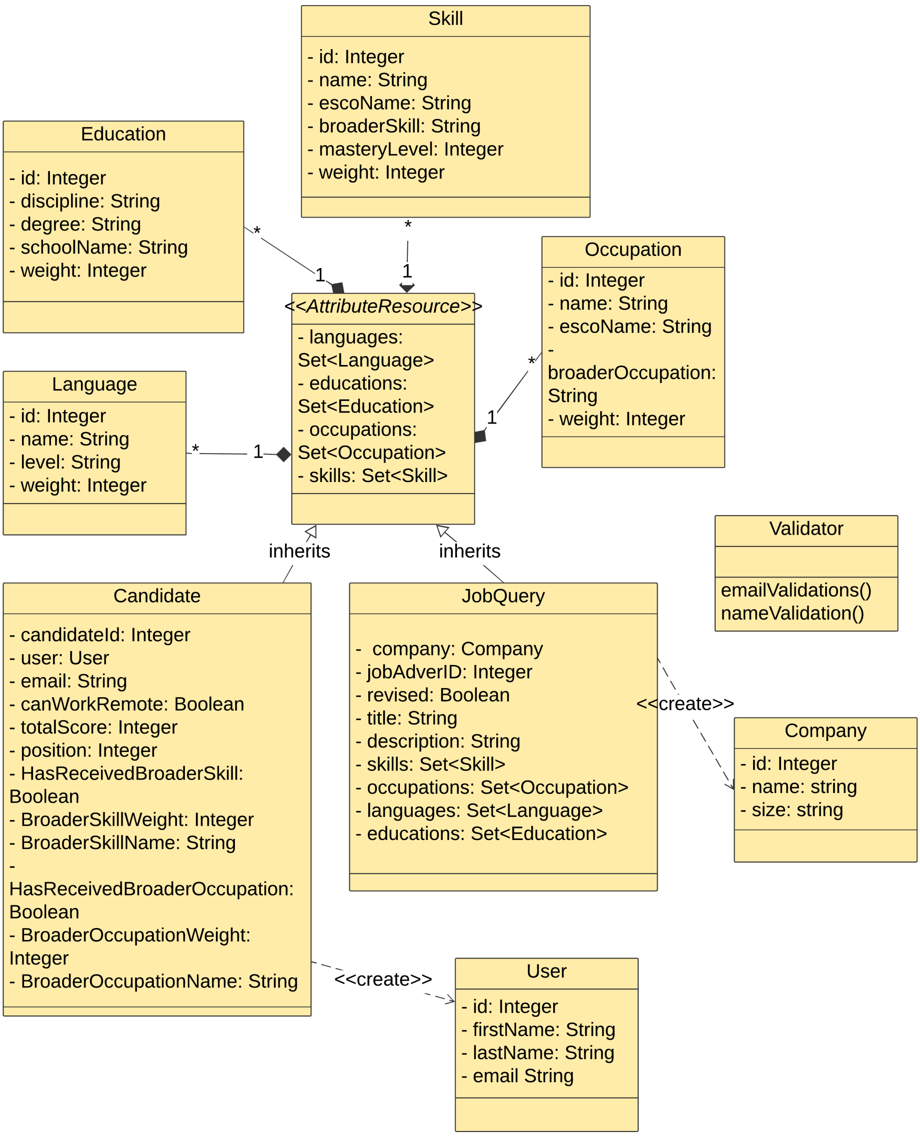


Figure 3.4: Class Diagram showing models in the Frontend layer

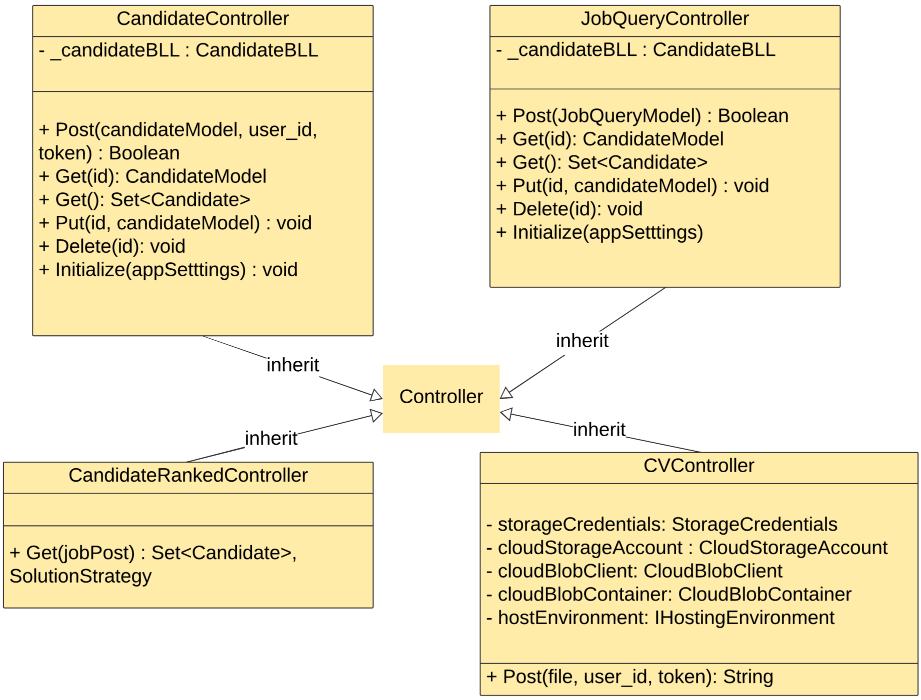


Figure 3.5: Class Diagram showing the Web API Controllers

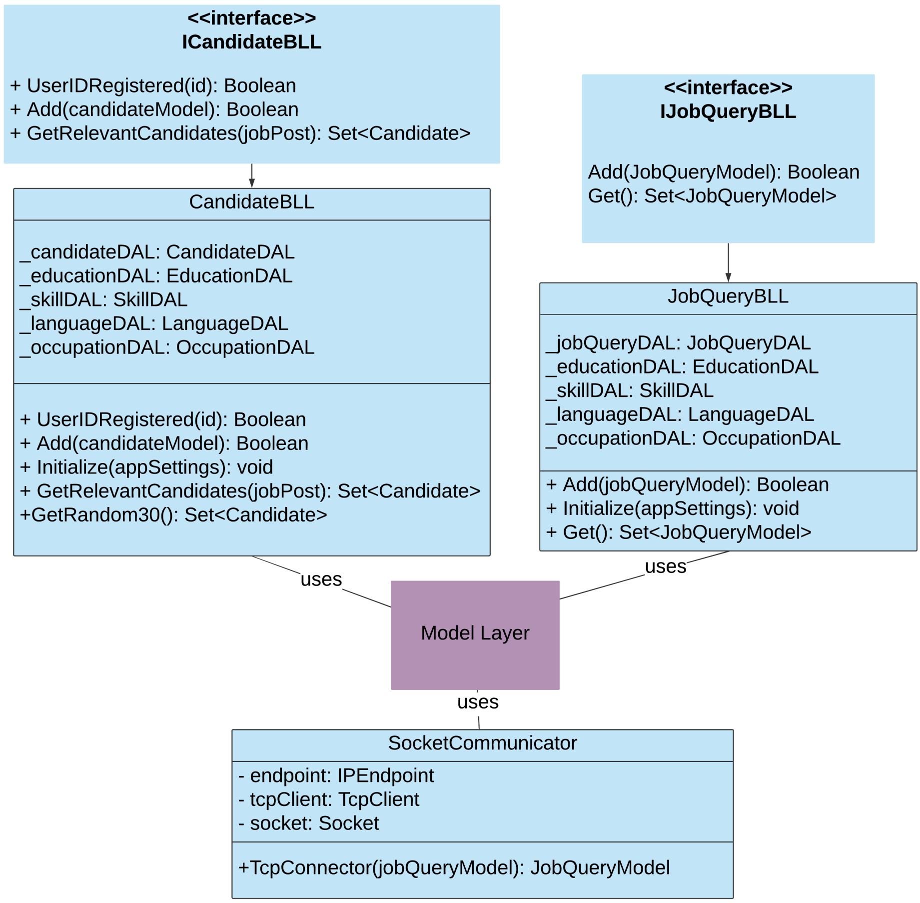


Figure 3.6: Class Diagram Business Logic Layer

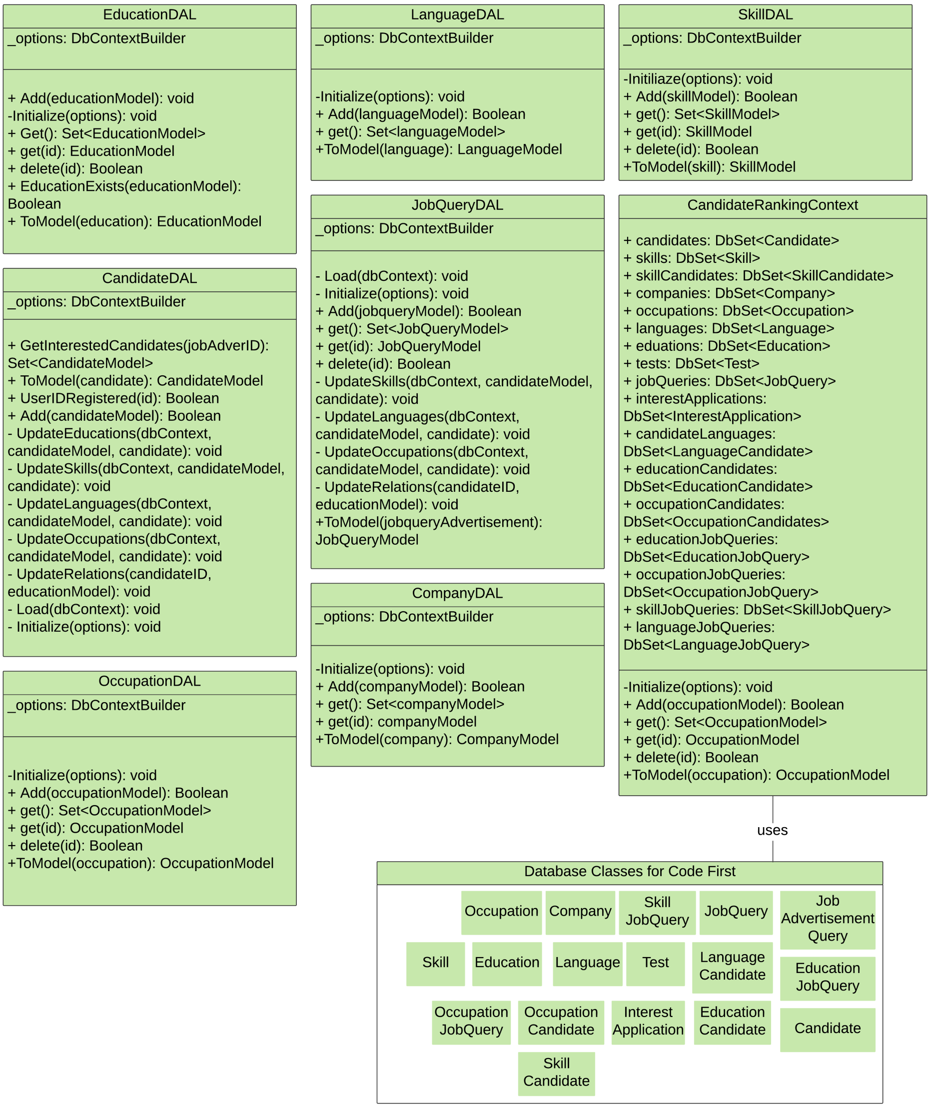


Figure 3.7: Class Diagram Data Access Layer

order to connect and create the database. When using a code-first approach one focus on the domain of the application and create classes for the domain entities rather than designing the database first and create classes to match the design of the database. Therefore, as presented in figure 3.7, there is outlined a pool of domain classes that the context class uses to create the database tables. As presented in figure 3.9, these 17 tables are presented by each class in the pool. For simplicity, we excluded the data properties and methods for this classes in the diagram since they are almost identical with the interaction models that we present from the figure in section 3.3.4

The Data tier also includes the Database and the Storage Container. The Data Access Layer, abbreviated as DAL uses LINQ as a query language. Under this tier we also find the SQL Server and the Blog Storage.

3.3.4 Interaction Tier

The interaction tier is a standalone tier that can reference to each tier independent of hierarchy. From figure 3.1 we have included a ”Models”-layer in the interaction tier. By having an independent layer, we can create and exchange classes or models that incorporates data fields through the layered stack. Figure 3.8 presents the models that are used in order to interchange data between the layers.

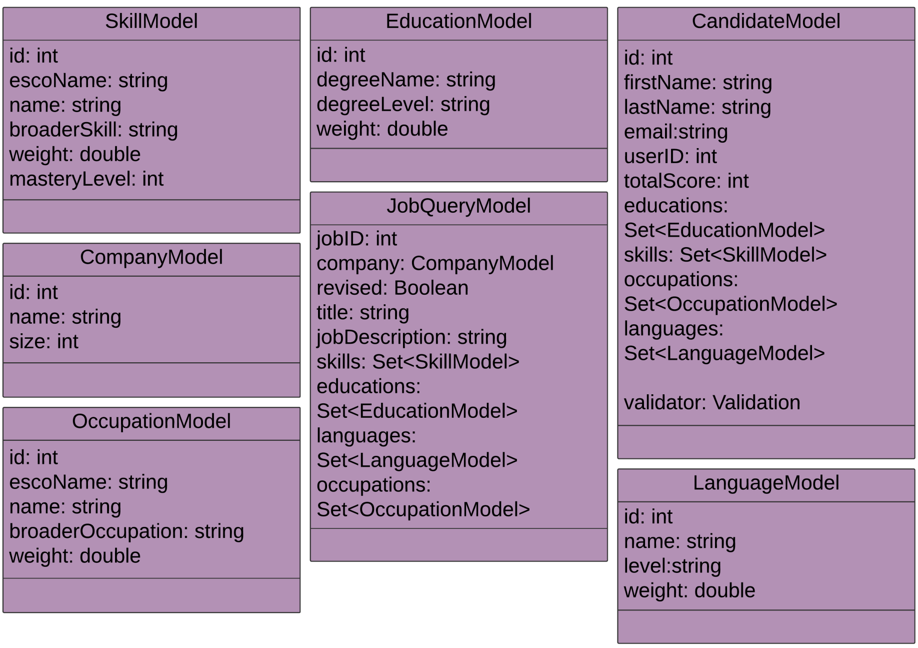


Figure 3.8: Class Diagram over Model Layer

3.4 Database Architecture and Storage Container

This section will justify the choice of implementation when it comes to storing data. We start by describe how the architecture encompasses third normal form, then present each table in the database, before reviewing the Blog Storage.

3.4.1 Third Normal Form

The database architecture includes all tables that is necessary to provide satisfactory nutrition for the CRS. An overview of this architecture is presented in figure 3.9. The blue painted tables, we call main tables with the responsibility for storing the facts. The green tables we call the relational tables used to connect two main tables that has the objective to store historical data between main tables if applicable. The reason for this coupling is with regards to having the database normalized to what is commonly known in the database world as "third normal form" or abbreviated 3NF, introduced in Codd (1971). When a database is normalized to 3NF, all column references that are not depending on the primary key are removed. Only a foreign key within that table is used to access another table. There are a numerous of advantages for having the database normalized to 3NF. From Codd (1971) we learned that the most important factor is that it removes the data redundancy since data is only present in one table. In addition normalizing our database to 3NF, gives smaller tables that makes it easier to sort, index and do search. Many small tables also implies clustered indexes that will ease the query tuning, which is vital for the system performance for the CRS. Techopedia [3] puts this forward in more detail for the curious reader.

3.4.2 Tables and ER-Model

The database consists of 18 tables that maps candidates and job queries with respective attributes. Figure 3.9 presents a table structure that contains 14 tables, and is therefore simplified because the relationship between the "JobAdvertisementQuery" table and the attribute tables such as Skills, has the same table structure as the one that maps Candidates to attributes. Their relationships have therefore been presented as list inside the JobAdvertisementQuery table.

Some tables from the figure ended up being excluded for the experiment, but are worth their mention. They derives important fundamentals for future work that we investigate further in chapter 8 Conclusion and Future Work. One example of an attribute exclusion is candidates personality and integrity tests. They should orgininally be stored in the main table "Test" that has the relational table "CandidateTest". The reason for its exclusion is due to the nature of how we ended up conducting this experiment, explained in Chapter 5 Experiment. Ensuring that data fields from the harvested CV's, introduced in 3.4.3 Blog Storage, reflects the data input gathered in the Data Collection Module, introduced in 3.6.1 was of utter importance to derive a consolidated experiment. However, having the ability to collect test scores from the candidates through Highered's platform, we opted

[3] https://www.techopedia.com/definition/22561/third-normal-form-3nf

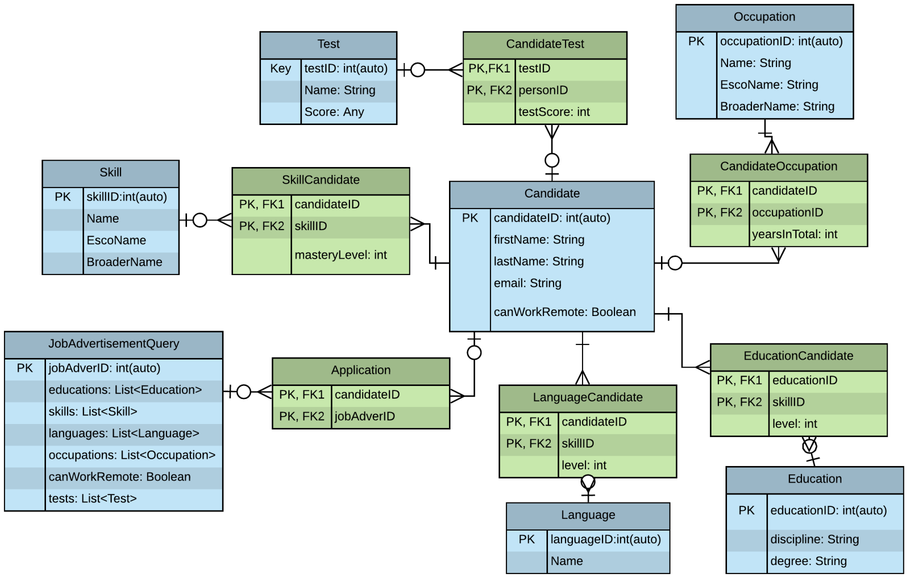


Figure 3.9: ER-Model over the Database Architecture

to exclude. From a rigorous examination of different available CV's we learned that a majority of users either didn't have their test score presents or conducted ones.

The Skill table consists of a name, EscoName and a BroaderName. The reason for two name fields is because we can separate distance measure to be used only on fields that are registered that hasn't been utilized with ESCO. This means that names present in the EscoName can have a binary nature of its similarity. The Broader name is being stored if the respective skill is an element of a familiy of related skills. The relation between candidates and skills are being maped from the "SkilLCandidate" table. This table stores the degree of mastery a candidate subjectively possesses of that skill. The same structure also applies for the Occupation relation, with the objective to store the users previous working experience.

The Application table is responsible for mapping the relevant candidates that have made their interest in a specific job announcement. We could therefore make sure that we tested a relevant set of candidates that could be contacted by the firm in the experiments aftermath. In addition, we make sure that the magnitude of uninterested candidates was excluded.

As presented in 3.9, candidates have a candidateID and an external user id. The candidateID is an internal key that is auto generated when the user stores

3.4.3 Storage Container

To be able to conduct our experiment explained in chapter 5, it was a necessity to collect sufficient input data through the DCM Module, as we discuss in 3.6.1. In addition to the input data, we also prompted the users to upload a separate CV. These Cv's are stored in a container that is located on a storage server in Azure, which is Microsofts cloud and computing platform service[4]. These CV's are named with the generated User ID we receive from Highered when a user uploads a CV using the DCM. Figure A.1 in Appendix displays a screendump from the storage container showing CV files being stored named with the user id. Figure 3.21 displays the interaction when a user access the backend Controller which handles the credentials and authorization for storing the CV file in the blob storage.

There are two general reasons for having a distinction between the storage location for CV's and input data. First, when we allow users to upload a CV that can be created from a vast amount of different file types such as PDF and DOCX, it seemed more wisely store these in a separate file system. Since this data is not directly used by CRS but required for the experiment, we could therefore drop dealing with file to byte conversion to be able to store these files in the database. The input data to the CRS should in principle reflect the data from the CV's. Therefore it would be excessive to store this data in the database because the CRS already works on the data stored in the database.

3.5 CBR Architecture

This section presents the CBR Architecture that encompasses the CBR-System from Figure 3.1. In this section we will explain how CBR is incorporated into the CRS.

3.5.1 Introduction

The CBR system was implemented as suggested in the MyCBR documentation. Three of the four CBR cycles from figure 2.1 were implemented in Java at the server side. The remaining cycle **Revise** was implemented client-side using JavaScript.

3.5.2 Framework

In our project we chose to use the MyCBR Framework. We landed at this decision after spending some time testing the available frameworks and discussing the possibility of implementing CBR without a framework. The main advantage of using a framework is the time saved by utilizing existing code, however depending on the framework, this time could be lost learning how to utilize the framework. Our project does not easily conform to a flat knowledge model; therefore, a significant part of the work would be in designing and testing several representation models. This was something that we kept in our mind while performing the preliminary testing using each of the available frameworks jColibri and MyCBR. Below we will list up some of the advantages and disadvantages to these frameworks in relation to our use case.

- **MyCBR** is open source and implemented in java, we used MyCBR version 3.1 in this review. MyCBR comes with the MyCBR Workbench that we introduced in chapter 2. The MyCBR Workbench was a particularly good fit for our project due

[4] https://azure.microsoft.com/en-us/

to it supporting modelling of object-oriented representations with part-of and is-a relations supported out of the box. Another big advantage to us was the fact that one of our professors Kerstin Bach is one of the main contributors to MyCBR. During testing we found that the MyCBR Workbench was both intuitive and effective when designing knowledge models if we forgive the odd bug popping up now and then. The SDK and its API were not up to the same standard and somewhat less usable, this could to some degree be forgiven due to the tutorial slides made available on the MyCBR website. The tutorial slides and all the examples available only cover flat knowledge structures, therefore leaving out using other concepts and sets as attributes.

- **jColibri**, presented in chapter 2, also has knowledge modelling capabilities. The knowledge modelling in jColibri is implemented as a plugin for the open source IDE Eclipse. Modelling knowledge in eclipse is not as intuitive and slightly limited when compared to the MyCBR workbench. The jColibri codebase has not seen much change in the last 10 years, this is also true for the Eclipse plugin Colibri Studio. This could be the reason why the authors stumbled across some compatibility issues with Colibri Studio using the latest version of Eclipse. jColibri does shine when it comes to documentation and examples, their latest development being their template engine. The templates as discussed in chapter 2 allows you to quickly setup and test various forms of CBR systems. However in the main release there mostly exist CBR templates for recommender systems.
- **No framework**, the third option, we would have to follow the CBR paradigm and implement the CBR cycle by ourselves, tailored for our specific application. While this might reduce the time spent learning to leverage a SDK, there is also the risk of the development time increasing beyond this.

3.5.3 Case based reasoning

The Case-based reasoning module is the brains behind our candidate ranking system. Although our architecture is loosely coupled, the CBR module takes what could be called the center position of the system. In this section we will go over our CBR model, which representations we considered and how we implemented the different CBR steps.

From figure 3.10 you can see that we chose to implement retrieval, reuse and retain serverside, the reuse cycle is handled in the adaptor class. Due to the amount of MyCBR instances that had to be constucted and built up we chose to implement an InstanceBuilder class to handle the creation and adding of properties to an instance. The model package is used to create a consistent data-structure accross our entire architecture, the model files themselves can be parsed to JSON and sent across socket connections. CBREngine is a static class containing the code for loading and storing the MyCBR project to disk.

3.5.3.1 Case representation

A variety of representations were considered before arriving at our current object-oriented representation, among these an ontology-based representation. The ontology-based representation was considered due to its flexibility with different relations and how this might

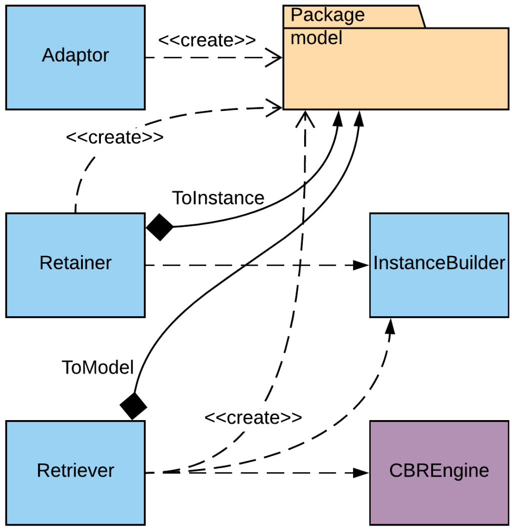


Figure 3.10: A diagram of the Java part of the CBR system

make it easier to take advantage of the domain specific ontologies available online such as ESCO. This would both enable the system to produce more specific explanations as to why certain solutions were recommended. The ontology-based representation was however sat aside, this was mostly due to the authors finding the basic relations provided with MyCBR to be sufficient.

The chosen representation has full support in the MyCBR Framework and is based on two main types of relations (is-a, part-of). Using these two relations an object-oriented case representation was built. The representation is divided into concepts, attributes and relations, in our representation we have six concepts: Skill, Language, Education, Occupation, Candidate and Job. Each of these concepts have their own set of attributes.

The object-oriented representation allows for concepts to have other concepts as their attributes, this also works for one to many relations were one concept can have a set of another concept's instances as an attribute. In our model we have the Candidate concept, this concepts attributes consist of four sets, one set of each of the concepts (Skill, Language, Education, Occupation). The Candidate concept itself is used as an attribute in the Job concept describing the ideal candidate for the job in terms of (Skill, Language, Education, Occupation).

The Job concept is the main concept of our CBR System, also called the problem description. Each instance of the Job concept can also contain a list of solution attributes. The Solution concept holds a weight, and a link to an attribute of one of the following concepts (Skill, Language, Education, Occupation).

The concepts used to build the case representation reflect the main part of the information that is gathered from the candidates and information that is available as part of the ESCO Ontology described in chapter 2. As an example, each instance of the Skill concept

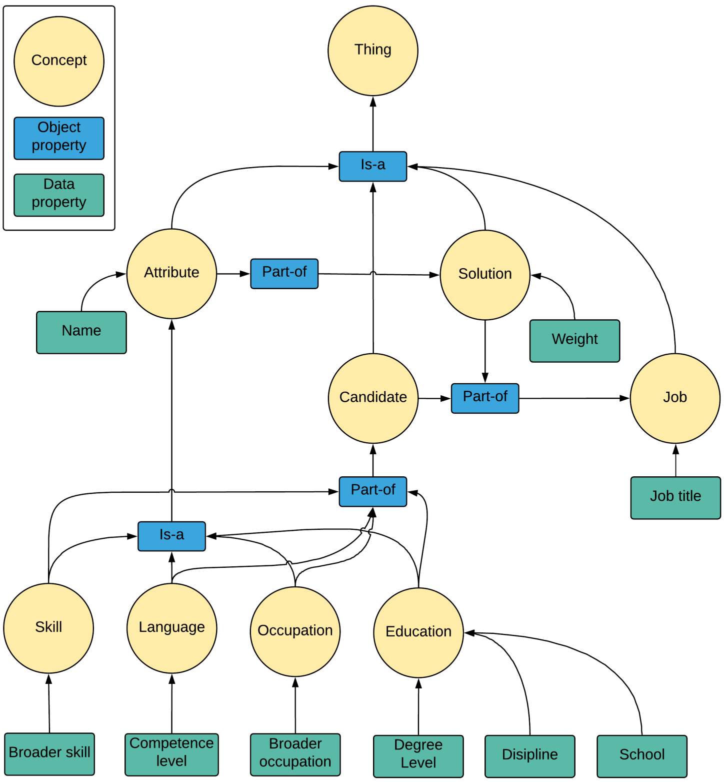


Figure 3.11: A diagram of the case representation

will have been reported in as part of our data collection and it will have a link to a ESCO skill resource uri, if the skill exist in the ESCO Ontology.

One of the main considerations when designing the case representation and the CBR system was compliance with GDPR Council of European Union (2016). The regulations aim to give more control to the owner of the data, that is the user. This includes the control over who, where and for how long their data can be stored. As a result, the entire CBR system has been designed around the idea of learning the best criterias to use when determining the relevance of a candidate. The actual sorting of the list based on the criterias is handled by the RankingSorter 2.

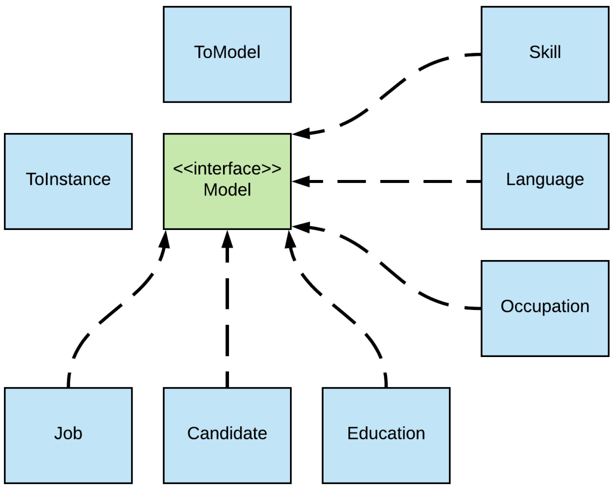


Figure 3.12: A diagram of the model structure in the java codebase

3.5.3.2 Case retrieval

The sequential retrieval technique in MyCBR was used and a set of similarity measures were designed. Sequential retrieval as described in chapter 2 is a robust and simple retrieval technique, however the technique is not the most efficient one. We decided to use the sequential retrieval technique mainly due to its support of arbitrary similarity measures. This was important during testing and development as several similarity measures had to be considered. As a prototype system the case base is not likely to reach such a large size that the sequential retrieval method would become too slow. The main source of latency is expected to be the network communication. In the paragraphs below we will describe the different similarity measures used.

Similarity Measures

This section will present the similarity measures used in the CBR system. The case representation consists of several concepts, each of these concepts has a set of attributes. These attributes are of varying importance when it comes to identifying the concept itself. Due to this, each concept has its own global similarity measure, also called an amalgamation function, discussed in chapter 2. Each of these amalgamation functions will be used to collect the similarity of each of the concept's attributes into a global similarity measure.

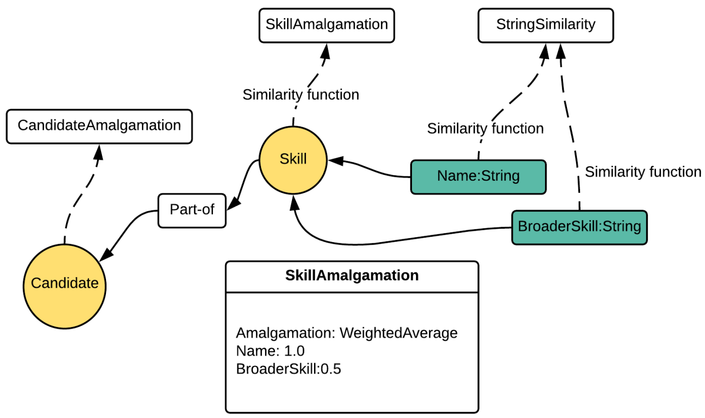


Figure 3.13: A diagram displaying an amalgamation function combining local similarity measures using weighted average

In figure 3.13 you can see the amalgamation function of our Skill concept. The purpose of this function is to combine the two local similarity function of Skills attributes. Skill contains two string attributes, each of these use a cosine string similarity function. From the SkillAmalgamation function you can see that the skill attributes are weighted differently, with name being twice as important as broaderSkill. From ESCO we have that each Skill can have a broader-skill, this broader-skill property is shared by many similar skills, some more similar than others, therefore the broaderSkill attribute gets a lower weight.

Local Similarities

The concepts in our case representation (figure 3.11) each have their own set of attributes. The attributes consist of the types String, Symbol, Integer and using a concept as an attribute. The attributes with Integer as type is part of the solution concept, in our current iteration we do not include the solution itself into the similarity measure during retrieval. The other attribute types are String, Symbol and Concept, String attributes use a string similarity function based on cosine similarity with trigrams. Concepts used as attributes will convey their local similarity measures through their amalgamation function.

For some of our attributes we have chosen to setup and configure unique symbolic similarity functions. This is true for the degreeLevel attribute of the education concept, a screendump of this symbolic function can be seen in figure 3.14.

We structured the symbolic function for the degree level to prefer job queries looking for lower degrees when a similar level is not available. This decicion was made by HigherEd's recruiting expert. In the event that a company is looking for a PHD to fill a vacancy they will generally have a slightly different approach compared to when looking

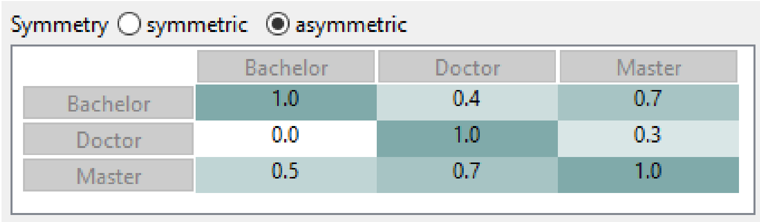


Figure 3.14: A screendump of the MyCBR Workbench modeling the degree level symbolic function

for a Master's candidate. For instance: grades and academic accomplishments will usually not be used to differ between PHD level candidates, while this can still be important when looking at candidates at the Master's level. We decided to reflect this in our symbolic function by making it asymmetric and weighting Bsc and Msc closer compared to the PHD level.

Similarity	Similarity function
String similarity	Cosine
Symbolic similarity	Binary
Language similarity	Custom symbolic
Degree level similarity	Custom symbolic

Table 3.1: A table of the local similarities used in this project

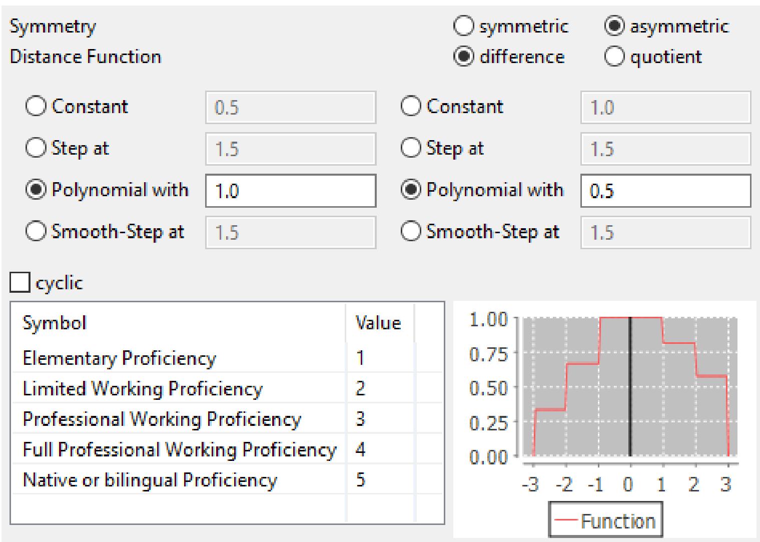


Figure 3.15: A screendump of the MyCBR Workbench modeling the language level symbolic function

When modeling the symbolic similarity function of the level of language competency it decided to give solutions with a higher competency criteria a slight bias, should the queried level not exist. This is reflected in the symbolic function seen in figure 3.15.

Global Similarities

All the concepts in our representation that has more than one attribute requires some form of global similarity measure. The global similarity measure is used to combine the local similarities, in this project weighted average has been used with adequate results. Each of the concepts will have their own set of tweaked weights, this weight is based on the current knowledge of the authors as well as some testing in MyCBR Workbench.

Attribute	**Weight**
Disipline	1.0
Degree level	0.5
School	0.25

Table 3.2: Amalgamation function for the Education concept

Lets take the Education concept, it has three attributes, (degree level, discipline, school). Discipline was found to be the most important attribute, with degree level coming in second. School could be important for some companies, therefore it is left in the model. Figure 3.2 displays a table of the weights in the education amalgamation function.

Attribute	**Weight**
Name	1.0
BroaderSkill	0.5

Table 3.3: Amalgamation function for the Skill concept

In figure 3.3 you can a table of our skill amalgamation function. We decided to give the BroaderSkill attribute one third of the total weight. This way a similar skillname will have much higher priority than one that only matches a broaderskill and the instance where both name and broader match will have the highest similarity. Both the skill name and the broaderskill use our default string similarity function.

Attribute	**Weight**
Name	1.0
LanguageLevel	0.75

Table 3.4: Amalgamation function for the Language concept

Attribute	**Weight**
Name	1.0
BroaderOccupation	0.5

Table 3.5: Amalgamation function for the Occupation concept

Retriever

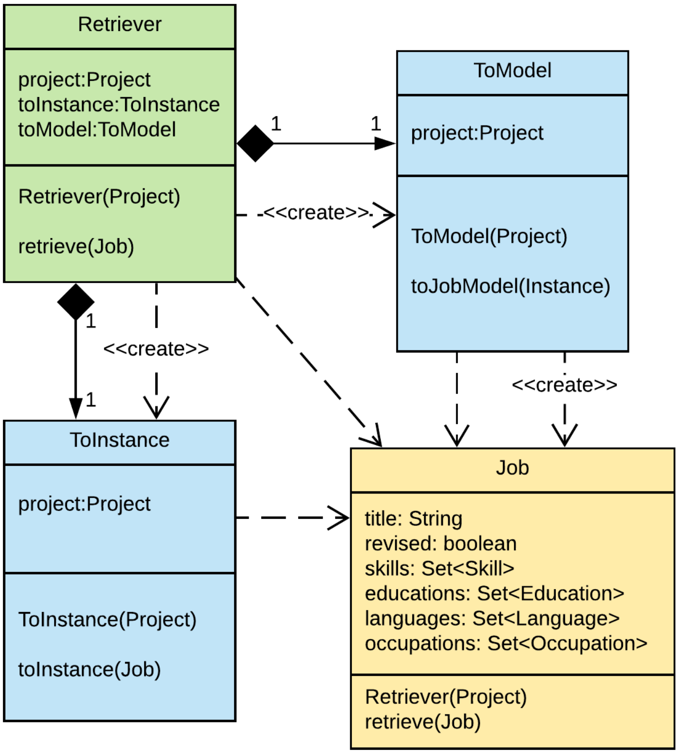


Figure 3.16: A UML Class diagram of the retriever class

Our retrieval step is implemented in java using the MyCBR library. The retrieval logic has been placed in a java class called **Retriever**. The retriever class has a function called retrieve. This function takes query data in the form of a model, the model can be either of the implementations of model, as seen in figure 3.12. When the retrieve function is called with a model it will use the ToInstance class to convert the model into a InstanceBuilder. The InstanceBuilder is then given a mycbr.retrieval's query-instance. The build function is then called and the mycbr.retrieval's query-instance is then populated with the correct attributes. When all the attributes have been added, the mycbr.retrieval is initiated using its start method. When the retrieval procedure has completed, the best matching case is converted back to "Job" model and returned see figure 3.16.

3.5.3.3 Case reuse

Like we talked about in chapter 2, cases can be reused in a multitude of ways. The simplest form being direct reuse of former cases and the more advanced forms often involving rule-bases used to adapt cases on an attribute level. In our implementation we used an approach that resembles direct reuse but with some simple adaption rules.

Adaption

Our adaption mechanism uses this retrieved model and the query model in order to apply its adaptation rules. The and the general behavior of the adaption step is described in figure 3.17. When the adaptation starts the first step involves stripping the case for superfluous attributes. As an example, if we query for a job with the skills ”Programming” and ”User interface design”, if the closest case also contains a entry for Language ”English, Native or bilingual proficiency” then the adaption will remove this superfluous language attribute. This leaves the two skills and their respective weights. The downside of this approach is that it restricts the benefits of sparse queries in that only the attributes related to the query will be returned. However in our application and for the way that our queries are structured this trade-off is not as bad, the consequences of getting a ranking-strategy that rewards undesired attributes outweigh the benefits of a ranking-strategy that correctly rewards some left-out important attributes.

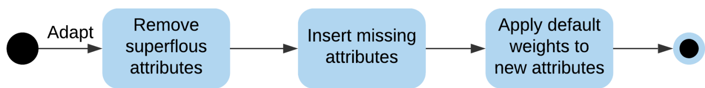


Figure 3.17: A UML State diagram of the adaption behaviour

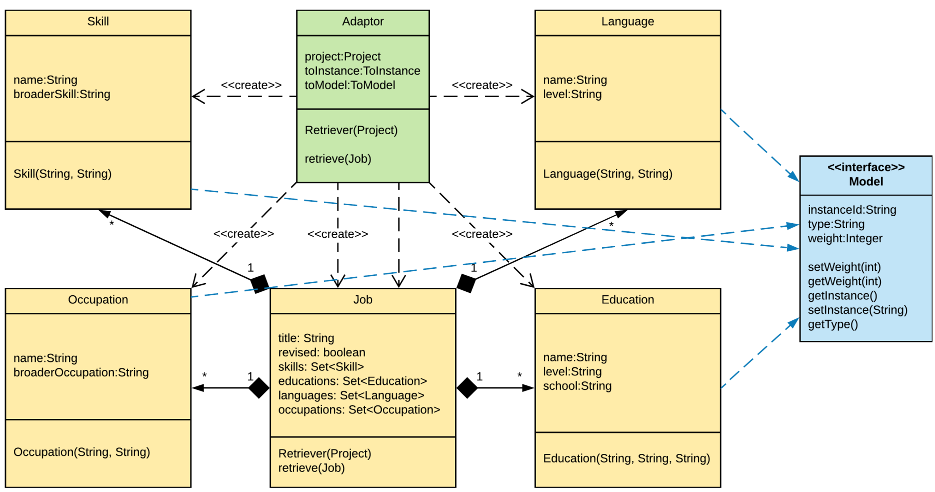


Figure 3.18: A UML Class diagram of the adaptor class

The adaption mechanism has been built around to support a mechanism for training/evolving the default weights from the case base itself. A recursive adaption approach was also considered, where the full query would return a case with most of the attributes and where the remainder of attributes could be composed into a separate query for a case with these specific attributes. However due to the project time-frame and other parts of the system being prioritized the potential gain from implementing this did not outweigh the costs.

The **Adaptor class** deal with models, it takes as an input the model used as the query and the model returned by the retriever. The adaptor class follows the behavior in figure 3.17. If the case being adapted has a low of a similarity, below 50% then the adaptor attempts to query esco for suggestions. This is only possible if the job query has an esco occupation as part of the experience job criteria

3.5.3.4 Revision

The revision phase of our CBR system has been separated out from the java implementation and is implemented client-side in JavaScript. The reasoning behind doing this is due to the delay caused by the round-trip time when communicating over long distances. The revision process itself has borrowed some ideas from the Conversational CBR retrieval/query building phase, we hypothesized that this could be a good solution considering that in a sense our cases are ”filter settings/ranking strategies” and to improve/edit a ranking strategy requires some visual feedback in how the list of ranked candidates change depending on the strategy used. Therefore a ”Conversational” approach was implemented to guide the users into creating/editing a case, and the case being the strategy/weights used to rank the candidates for the relevant job position. One of the most important considerations when making the revision UI, was that it should not be too tedious for the user.

3.5.3.5 Case retention

The case bases are not expected to become large enough to cause any issues during the training and experiment phase of this project. Our approach has therefore been to store all revised/validated cases that are not already 98% equal to an already existing case.

Retaining a large number of cases could negatively impact the retrieval phase of the CBR system, especially given the sequential retrieval technique we use. For systems using very large case bases an approach using less computational complexity and more storage complexity could be more appropriate like kd-trees.

Another way could be to reduce the amount of cases needed. This however usually requires more sophisticated case representations and/or adaption techniques.

Retainer

The retainer class takes a job model and uses the toInstance class to generate a an instanceBuilder with the job model as its specification. The instanceBuilder is then supplied a query instance from a mycbr.retrieval object and the build function is called. The query instance is then populated by all the attributes described by the job model. The start function of the mycbr.retrieval object can then be called and a sequential retrieval is performed. The similarity of the best match from the mycbr.retrieval is then checked. The current implementation retains all the cases where the similarity is less than 98% as a new case. When there exists a case with similarity of 98% or more then the existing case is revised. Each revised case also has a history of the last three revisions and their corresponding queries, this was included to facilitate a machine learning approach of learning revision/adaption rules in the future. Approaches combining cbr with other ai methods such as artificial neural networks have shown improvements in some applications, Lees and Corchado (1999); Chen and Burrell (2001); Craw et al. (2006).

3.6 Application Modules

This section will take a deep dive into the different modules that aggregates the system user functionality. We will examine their needs, architectures and functionality. There are three modules in the CRS. We start by introducing the Data Collection Module before we take a look at the Training and Job Query Module. Figure 3.19 presents an use case diagram presenting the main cases over the user functionality that the system provides.

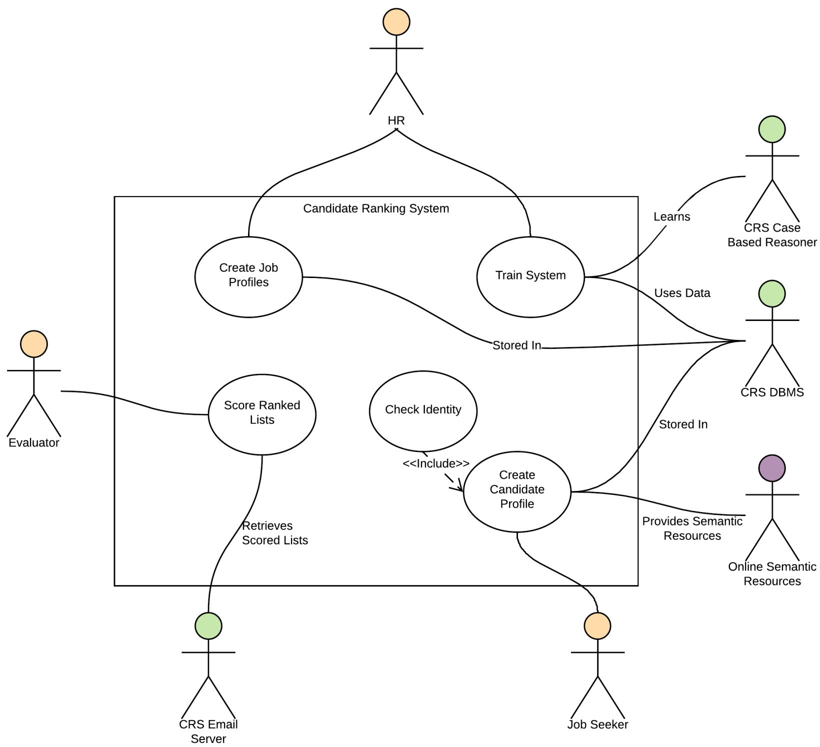


Figure 3.19: Use Case Diagram presenting the Overview for the Application

3.6.1 Data Collection Module

The information that already existed on Highered's platform was limited to name, email, language and education. Due to the inadequate data available to make reasonable decision from, the DCM was constructed. To figure out the most important fields in a hiring process, we went through different CV's to both learn and provide a pattern for a common structure. We also set up a meeting with Highered and psychologist that had a specialty in occupational psychology.

From this meeting we were told that the most important fields in a hiring process is regarded personality tests combined with interests and skills. Since Highered had a partnership with Cut-E [5], a company specializing in the creation of personality tests, we considered this as a great opportunity to gather good input for our CRS.

However, after some discussions we came to the conclusion that these tests perhaps may provide a greater value in a post screening stage. We therefore excluded these tests as input-data for this experiment. To make comparisons with similar studies and having an objective experiment, we ought that the CRS should have a larger influence on the screening process itself, and instead produce potential candidates to be chosen for further evaluation by a manager. The DCM therefore incorporates the fields written in 3.6.1.2 Data Fields.

3.6.1.1 DCM Architecture

The DCM consists of an interaction view that fetches data from different fields through APIs. This is explained further in 3.6.1.2. The DCM Architecture is separated in a frontend view and a backend view. Figure 3.20 illuminates an overview for the main components for the frontend side. The Candidate-Collection Component is the main component that interacts with the different views and it is responsible for communication and interaction between views and services. requesting data from APIs. The Candidate Service receives the newly created candidate that is supplied with new information and sends this data secure over HTTPS to the Backend Server. Figure 3.21 presents the backend part for the DCM.

3.6.1.2 Data fields

This section describes the different data sections that are included in the DCM to collect necessary data about the candidates. Figure

Personal Details, Education and Languages

Since the DCM module is conducted on a different platform, the DCM was embedded into Highereds platform through an Iframe. When their users logs in and accesses the DCM, the DCM receives a token from Highered's platform that our CRS can use to require user details such as firstname, lastname, language and education that the candidate has already registered on Highered's platform. If the user wants to apply additional information, the DCM fetches proposed schools and languages through Highered's existing REST api. The

[5] https://www.cut-e.no/nettbasert-testing/oeve-paa-tester/

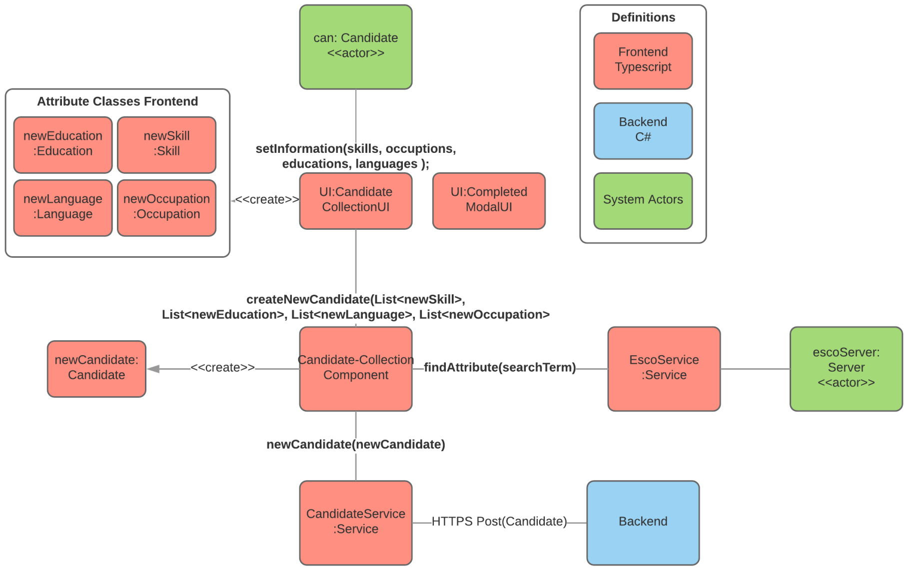


Figure 3.20: Communication Diagram showing the overview of the Frontend Components in the DCM

Personal details section also includes a CV uploader. This CV however, was an important step in the experiment process. This is discussed further in Chapter 5 Experiment.

Skills and Occupation

Both Skills and Occupation sections are incorporated into the DCM by utilizing the ESCO API, described in 2.4.7.1. The DCM allows candidates to search through this API to find their skills and previous working experience. When the user has found their respective skills and previous occupations, the DCM tries to requests for further information on behalf of the user. More precise, the DCM tries to fetch the broader skill and the broader occupation for each attribute. However, since the ESCO API is constantly under development and names of skills and occupations may not occur per now, the DCM also allows users to write in these categories manually as welll, EU-Commission (2017a).Providing both options is important when the CRS shall rank candidates where a scenario could be that attributes are included in the CV, but not present for the CRS, if relying on the ESCO API isolated. Typos and errors cold eventually be solved at a later date using NLP and machine learning.

Figure 3.22 illustrates a logical view over skills and occupations respectively. It is worth to mention that broader occupations may not be present through the traditional ESCO standard, and therefore have to be fetched through the ISCO standard instead.

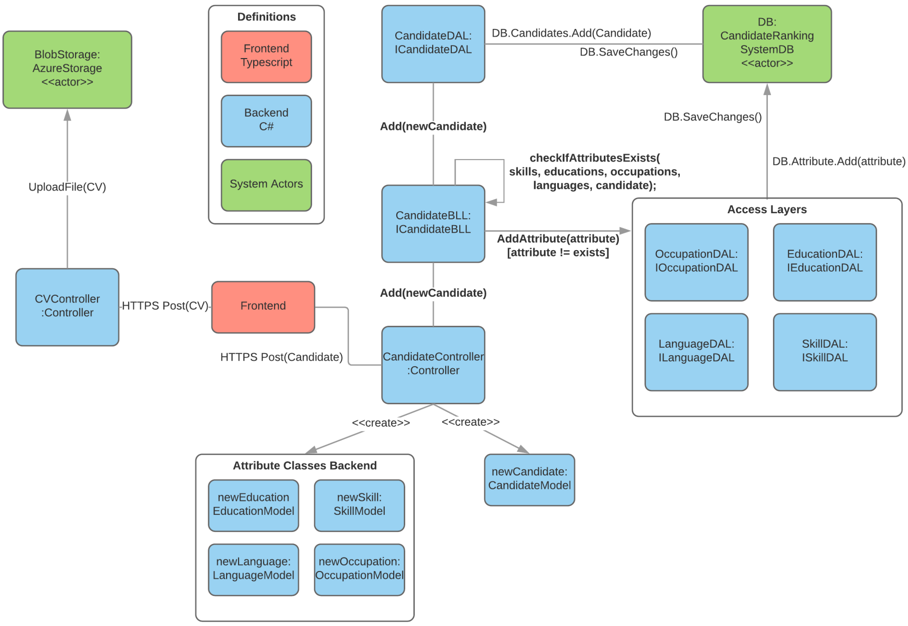


Figure 3.21: Communication Diagram showing the overview of the Backend Components in the DCM

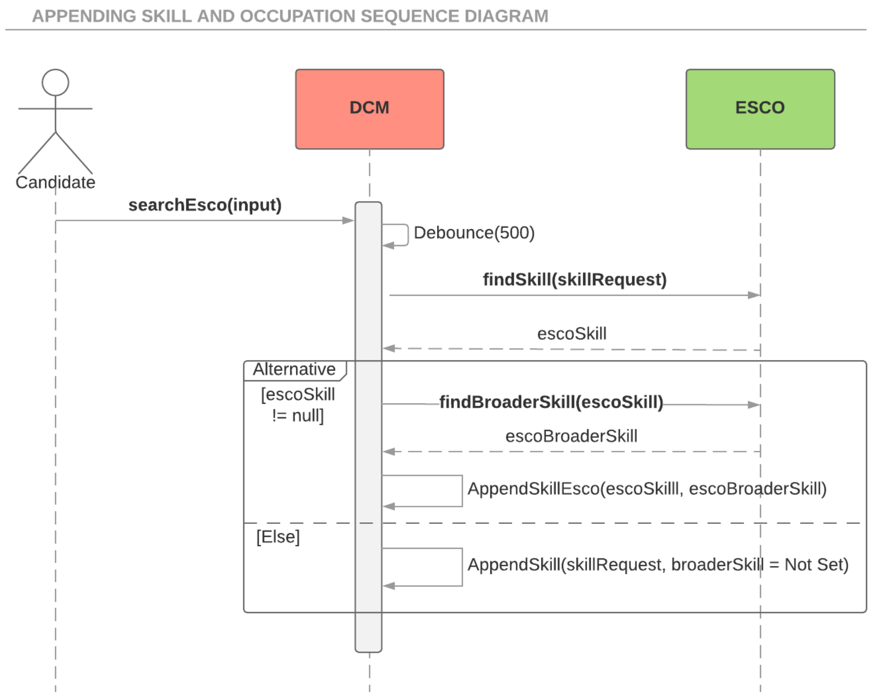


Figure 3.22: Showing the sequence between DCM and ESCO

3.6.2 Training Module

This section explains the Training module that derives the revision part of the CBR paradigm introduced in 2.3.2. This section will further explain its architecture through a class diagram, revision with activity diagrams, and present the ranking algorithm that shows how candidates are ranked in the CRS.

3.6.2.1 Motivation

To be able to cover the revise step in the CBR Cycle discussed in section 2.3.2, a training module was vital to implement. This is due to the nature of how the system should learn from the cases and how the experiment should be conducted, discussed further in chapter 5. Seperating the revision step from MyCBR and Java onto a customized environment for the revision felt natural. The revision therefore encompasses the Frontend and Web API layers.

Like the DCM introduced in section 3.6.1, the Training Module consists of both a frontend and a backend layer. The frontend layer is vital for handling the user interaction, and the backend layer is required to feed the CRS with new input, retrieve desired cases, and relevant job candidates.

3.6.2.2 Revision and CCBR

As explained in 3.6.2.1, the revision step is the motive for having a training module. Figure 3.23 presents an activity diagram that shows the core steps. First the user is prompted with the option to select a specific type of job posting to train. When specified, the retrieval process finds the best case considered. The CRS then finds the relevant set of candidates to apply the case on. When applied, a ranked list will be presented, discussed in 3.6.2.3 It is from this step, the revision is being managed.

If the user is satisfied with the ranking, no revision is required. The case can therefore be considered trained. If the ranking is not satisfactory, the user should consolidate the case. In order to train the case and receive necessary input, the system has been inspired by a conversational case based reasoning approach. Discussed in section 2.3.6, CCBR prompts the user with a single question at the time, and re ranks the list thereafter. The system has followed the guidelines that is discussed in 2.3.6 and suggested in Aha and Breslow (1997).

3.6.2.3 Ranking Algorithm

The Ranking algorithm encompasses the presentation part for the Revision step. Its responsibility includes updating the ranked list of candidates that uses a solution strategy to provide a candidate score match. Figure 3.24 presents the main process in the ranking. The Ranking Algorithm presented in algorithm 1 derives a drill down in the form of a functional pseudo code that illustrates the overriding process. The first method named StartProcedure kickstarts the process by requesting the relevant candidates for the job post. Thereafter, a suited solution case strategy is retrieved from the case base. When this strategy is retrieved, a normalization process is being conducted. This is important so that the sum of all weights that are being retrieved equals to 100%. The last step in the chain is

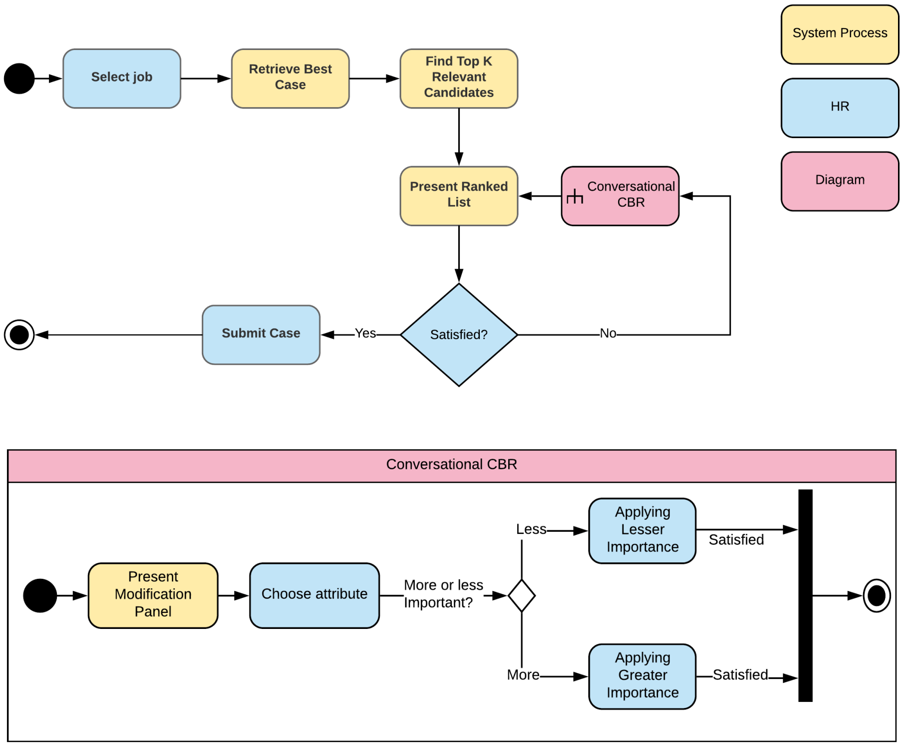


Figure 3.23: Activity Diagram showing the core steps for the Training module

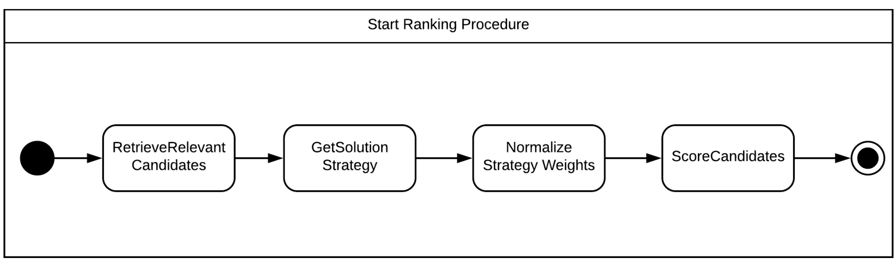


Figure 3.24: Activity Diagram showing the Ranking Process

scoring the candidates. Depending on the attribute type different measures are being applied. From the algorithm we can see that that a broader fraction is sent to the method if the attribute is a instance of either skill or occupation. If for instance a skill share a broader category with another skill, the candidate should retrieve a portion of that skill score, hence the name broaderWeight. By including this, we make sure that we do not rely isolated on a binary match between attributes in case an attribute existence is absent.

From the algorithm in 1 we can also see that a DistanceMeasureApproved if the CRS could not find the attribute's existence. From the algorithm, this distance measure is primarily used to review if the attribute has any typos in its naming convention. For instance a Levensthein approach can be applied, discussed in section 2.3.4.4.

3.6.3 Job Query Module

The Job Query module shares similarities with the DCM introduced in 3.6.1. In this section we will explain its motive and usage.

3.6.3.1 Motivation

In our preliminary study we presented a model where companies submitted job posts to the Highered platform where we could receive information based on which candidates that had applied to a certain job post. After a discussion with Highered where we conducted a review over their harvested platform statistics, we wrote down a couple of concerns:

- Based on companies that submits job posts to Highered, there's a risk that participants available to conduct the experiment represents a narrow discipline sector.
- There is a risk that the creation of a candidate set based on whom has applied to which job post may turn out to be insufficient. This with respect to the magnitude of candidates the CRS needs to make reasonable decisions.
- Storing unstructured job posts from Highered might end up with inconsistent data in terms of loss or bad conversion. This may be potentially become a time consuming job that includes techniques in field of natural language processing that we opted to avoid if possible.
- Based on the nature of how we want to conduct the experiment, it might produce invalid data with respect to the research.

Based on the three bullet points above, we opted to implement what we call the Job Query Module abbreviated further as JQM. The JQM provides the possibility to write a job posts that reflects a company need that can be stored structurally.

3.6.3.2 Data Fields

Adopted from the DCM introduced in 3.6.1, the JQM utilizes the ESCO API that provides the ability to give suggestions based on semantic resources introduced in 2.4.7.1. The personal details field section is not included. Instead there's a section with company details and sector. To read more on how the Skills and Occupation is being fetched, read section 3.6.1.2.

```
Function StartProcedure():
    candidates ← RelevantCandidates(jobPost)
    solutionCase ← GetSolution(jobPost)
    NormalizeWeights(solutionCase)
    for c ∈ candidates do
        Score(sol, c)
    end

Function NormalizeWeights(sc: solutionCase):
    WeightsumTotal ← totalSumWeights(solutionCase)
    attributeLists ← ListsOf(sol, Skill, Education,Occupation, Language)
    for attrList ∈ attributeLists do
        for el ∈ attrList do
            normalizedWeight ← Weight(el)/WeightsumTotal * 100
            SetWeight(el, normalizedWeight)
        end
    end

Function Score(can: Candidate, sol: SolutionCase)::
    SetTotalScore(can, 0)
    SetbroaderSkillWeight(can, 0)
    SetBroaderOccupationWeight(can,0)
    lists ← AttributeLists(sol)
    for attrList ∈ lists do
        if attrList typeof (ListType(Skill) OR ListType(Occupation)) then
            score ← TotalScore(can) + scoreAttributes(can, attributes,
              broaderFraction = 4)
            UpdateTotalScore(can, score)
        end
        else
            score ← TotalScore(can) + scoreAttributes(can, list)
            UpdateTotalScore(can, score)
        end
    end

```

Algorithm 1: Candidate Ranking Algorithm part 1/2

```
Function ScoreAttributes(can, attributes, ?broaderFraction):
    attrDictionary ← ListOf(can,attributes)
    for el ∈ attributes do
        attribute ← GetAttribute(attrDictionary, el)
        if Exists(attribute) OR DistanceMeasureApproved(attrDictionary, el) then
            if AttributeType(el) typeOf Education) then
                attribute ← HighestEducation(GetAttribute(attrDictionary, el), el)
            end
            UpdateWeight(attribute, el.weight)
        end
        else if (AttributeType(el) typeof (Occupation OR Skill) AND (
          HasReceivedBroaderAttribute(can, AttributeType(el)) OR
          BroaderWeight(can, AttributeType(el)) < Weight(el)/broaderFraction)
          then
            for attr ∈ attrDictionary do
                if el equals attr then
                    broaderWeight ← Weight(el)/broaderFraction
                    UpdateBroaderWeight(can, AttributeType(el), broaderWeight)
                end
            end
        end
    end
```

Algorithm 2: Candidate Ranking Algorithm part 2/2

3.6.3.3 Storage

All job posts that is registered in the JQM is storred with relationship in the JobAdvertisementQuery-table shown in figure 3.9

Chapter 4

Integration and Deployment

The purpose of this chapter is to address the considerations regarding collection, storage and processing of the data needed in order for the system described in chapter 3 Architecture to be production ready.

4.1 Introduction

The system described in 3 uses indirectly personal information in the form of resumes. This in the context of our research goal of developing a production ready system requires a set of considerations. Among these considerations we have information security, robustness and compliance with laws and regulations. For a system to be production ready, compliance with laws and regulations is extra important. A risk management iteration was performed in section 4.3, with a focus on the system's gathering/processing and storage of personal data in relation to laws and regulations.

The laws and regulations regarding consent requires that the consenting party is sufficiently informed. This information includes, the processing involved, the purpose and what data is required. To facilitate this, data-requirements were established. These requirements can be found in section 4.2.

4.2 Data requirements

In general, an AI system is usually only as good as its data. However in practice, feeding any system too much irrelevant data will eventually slow it down. As described in 3, the CBR system itself will not store any personal data. This decision was made in order for the design to be viable in production under the restrictions imposed by GDPR.

4.2.1 Candidate data

The experiment described in 5 requires us to collect resumes. This is to make sure that the recruiters participating in the experiment get about the same information that they would have in an ordinary scenario.

4.2.1.1 Screening criterion's

The data used by HigherEd when screening is listed below. In the first iteration of the system only a subset of these will be included. This decision was made in order to make the whole submission process less tedious.

- Name
- School email
- Private email
- Nationality
- Male, Female, Other
- Phone number
- Photo
- Video
- Social profiles
- Achievements
- Projects / work samples
- Areas of interest (work related)
- Land of interest to work in
- School name
- Degree
- Discpline
- End date
- Grades
- Languages
- Skills/Interest
- Course name, duration, direction
- Name of tests, and test results

4.3 Privacy and data protection

The collected data will inevitably include some personal details that could be used to identify the person. In Norway, research projects need to notify the Norwegian Centre for Research Data (NSD) and get approved before data gathering can start. An application to NSD has to be formed and certain requirements regarding storage and processing need to be met in order to get an approval. The new EU regulations GDPR went into effect 25 of may, the regulations impose a higher requirement for information security.

4.4 Risk management

Risk management was found to be necessary due to the nature of the data that was collected. The requirements imposed by NSD and the GDPR had to be met and in order to properly meet and document this a full iteration of risk management commissioned. The ISO/IEC 27005 framework from International Organization for Standardization and International Electrotechnical Commission (2011) was chosen to be used as a guide for the risk management procedure supplemented with the book by Whitman and Mattord (2017).

4.4.1 Context Establishment

Following ISO 27005 one first needs to establish a context in which one can perform risk analysis and if needed mitigation. The context consists of the scope and boundaries as well as the basic criterion's to be used when evaluating risks.

4.4.1.1 Scope and boundaries

The scope and boundaries set during the context establishment will determine the coverage of the risk management. However having too lose boundaries could result in too much resources and time required. If the time spent inventorying assets is too large then the gathered data might become obsolete before the inventory process is done.

We have limited our scope to include the assets directly related to the information gathered. A full list of the assets included in our scope can be seen in figure 4.5. From the figure you can see that the candidate information itself is considered the main asset and also the asset with the highest priority. Secondary assets include our business-partners and the DCM module. The candidate ranking system itself was scoped out of the context due to the limited time.

4.4.1.2 Basic criteria

The basic criterion's usually includes a risk acceptance criterion, risk evaluation criterion as well as a impact criterion. In our case the risk acceptance criterion is strictly set by the requirements imposed by us by NSD and GDPR.

Evaluation of risks will be done using a qualitative scale. A risk evaluation matrix will be used to calculate risk based on the likelihood of a threat and the impact 4.4. The controls will be factored into the impact and likelihood values.

Risk Control Strategy	Categories Used by NIST SP 800-30	Categories Used by ISACA and ISO/IEC 27001	Others
Defend	Research and Acknowledgement	Treat	Self-protection
Transfer	Risk Transference	Transfer	Risk transfer
Mitigate	Risk Limitation and Risk Planning	Tolerate (partial)	Self-insurance (partial)
Accept	Risk Assumption	Tolerate (partial)	Self-insurance (partial)
Terminate	Risk Avoidance	Terminate	Avoidance

Figure 4.1: A table taken from Whitman and Mattord (2017), comparing control categories from different frameworks.

The control strategy categorization proposed in Whitman and Mattord (2017) will be used during the risk evaluation and treatment. The categorization contains five main control strategies: Defend, Transfer, Mitigate, Accept and Terminate. Figure 4.1 compares the categories used by different frameworks. The Defend category includes controls used to minimize the likelihood of a threat by reducing the vulnerability exposure. The Transfer category is used to describe controls gained by transferring/outsourcing parts of the information system, effectively handing over some of the risk management responsibility. Mitigation is used to describe control strategies that attempt to reduce the impact of a threat. An example of a mitigation strategy is encryption, the encrypted data could be stolen, however, given that the encryption is strong enough, the confidentiality might not be breached. Other examples include incident/disaster response planning. Terminate strategies includes avoiding/removing the activities/venues that introduce vulnerabilities. Some systems such as control systems in nuclear plants are critical enough to warrant termination strategies such as keeping them disconnected from the internet.

Likelihood evaluation critera	
Qualitative	**Description**
RARE	The event is not likely to occur within 10 years
UNLIKELY	The event could occur once a year
POSSIBLE	The event occurs several times each year
LIKELY	The event could happen happen on a monthly basis or more frequently

Figure 4.2: A table containing the likelihood evaluation criteria.

Impact evaluation critera			
Breach of	**Impact**		
	MINOR	MODERATE	MAJOR
Confidentiality	Minor annoyance, no potential for economical loss or bad publicity	Could cause limited bad publicity and/or economical loss.	Bad publicity and/or economical loss
Integrity	Minor or no effect on other assets, individuals or company.	Limited effect on other assets, individuals or company.	Serious effect on other assets, individuals or company.
Availability	Minor or no concequence.	Limited concequence for other assets.	Serious concequences for other assets, individuals or company.

Figure 4.3: A table containing the impact evaluation criteria matched to each of the concepts in the CIA triad.

Risk evaluation matrix					
	Impact				
Likelihood	L\I	ZERO	MINOR	MODERATE	MAJOR
	RARE	VERY LOW	VERY LOW	LOW	MEDIUM
	UNLIKELY	VERY LOW	LOW	MEDIUM	HIGH
	POSSIBLE	LOW	MEDIUM	HIGH	VERY HIGH
	LIKELY	MEDIUM	HIGH	VERY HIGH	VERY HIGH

Figure 4.4: A risk evaluation matrix combining impact and likelihood to estimate a qualitative risk.

4.4.1.3 Acceptable risk

Using the evaluation criteria detailed in figures (4.2, 4.3, 4.4), we are willing to accept a **LOW** risk.

4.4.2 Risk identification

The risk identification process is divided into phases beginning with identifying assets. After assets have been identified and prioritized the next phase can begin, this phase involves identifying relevant threats. When both threats and assets have been identified and prioritized, the process of finding vulnerabilities linking assets to threats can start. The process itself is quite time consuming, therefore the top prioritized assets should be analyzed first. The last step involves identifying the controls currently in use.

4.4.2.1 Identifying assets

The three most important assets were picked and prioritized from a larger list of assets identified. The first step involved identifying all assets that could fit within the scope and boundaries. The resulting list contained 21 assets, many of which were low priority and some of which had direct relations to other assets. From this list a set of assets were removed due to low priority. A set of three assets were defined from the remaining list. These three are listed in figure 4.5.

Asset list				
Asset	**C**	**I**	**A**	**Priority**
Candidate data	MAJOR	MODERATE	MINOR	1
Evaluator data	MINOR	MODERATE	MINOR	2
Data collection module	ZERO	MODERATE	MINOR	3

Figure 4.5: A prioritized list of assets included in the risk evaluation.

In the final asset list we have the candidate data as the asset with the highest priority. This is due to the personal nature of the information and the importance of this information in relation to the success of our project. The candidate data asset includes the information gathered by the data collection module as well as the information contained in the candidates' cv. This asset will be used to analyze the threats and risks involved when transferring and storing the collected candidate data. The initial gathering of the candidate data will be analyzed in relation to the data collection module.

The data needed to evaluate the system has been given the second highest priority. The success of the project is highly dependant on us being able to evaluate the system. The evaluator data includes the ranked lists of cvs provided by human experts as well as the resulting data from the experiment. The consequences involved in a breach of confidentiality is not high in this asset, however the a breach of integrity could have a negative impact on the project itself. The data defined as candidate data also has a relation to the evaluation, however due to the nature of the candidate data we decided to separate it out into its own asset.

The data collection module takes the last place on our list of top 3 most critical assets. The data collection module is used to collect the candidate data. Because of this, the data collection module can be seen as an attack vector for the candidate data, in some cases a vulnerability. However because the data collection module will be integrated into the platform of our partner company we decided to include it as an asset itself. The data collection module will therefore carry some value as to the image of the company that we will integrate it into. The data collection module itself and its human machine interface also has the potential of exposing vulnerabilities.

4.4.2.2 Identifying threats

From ISO270005 we have the following definition of a threat ”A potential cause of an incident, that may result in harm of systems and organization”. Some definitions also include a threat agent that causes a threat, an example being a hacker initiating a software attack or nature causing a fire. We will use the ISO270005 definition in our threat identification and therefore only consider the ”threats/threat actions” themselves.

Threat list	
Threat	**Priority**
Breach of laws and regulations	1
Software attacks	2
Sabotage or vandalism	3
Human error or failure	4
Technical failures	5

Figure 4.6: A prioritized list of threats included in the risk evaluation.

A large list of threats were collected and supplemented with the list found here [1]. From this list we prioritized the threats and selected the five most relevant. These threats can be seen in figure 4.6. The threat with the highest priority is ”Breach of laws and regulations”. We consider a breach of law and or regulation to be relevant because of the restrictions

[1] https://advisera.com/27001academy/knowledgebase/threats-vulnerabilities/

and requirements imposed on us by NSD and to a lesser degree, the GDPR. A breach of these regulations, especially the ones covered in the personal data act, would require us to file a incident report to the Norwegian Data Protection Authority. An incident response would then be initiated and overseen by research institute in dialog with the Norwegian Data Protection Authority and NSD.

The assets are exposed to a number of possible vulnerabilities due to them being online and accessible on the internet. The ”software attack” threat will be used to identify all the technical vulnerabilities and attack vectors relevant to our assets. The ”sabotage or vandalism” threat will cover the less technical ”attacks” and will be related to vulnerabilities in UI and the interaction design. Failure due to lacking protocols, training or service level agreement are possible vulnerabilities connected to our last two threats.

4.4.2.3 Identifying vulnerabilities

ISO 27005 defines a vulnerability as ”A weakness of an asset or group of assets that can be exploited by one or more threats”. For each asset, a set of vulnerabilities were identified linking the asset to some of the threats identified in above. In addition to identifying the vulnerabilities we also determined the type of impact the vulnerability could cause the asset using the CIA triangle.

Candidate data

The vulnerabilities identified for the asset ”Candidate data” has been listed in figure 4.7. A lack of controls and or organizational policy was considered as an vulnerability in regards to the threat of breaching regulations. Insufficient consent was also added as a vulnerability. Without proper consent the processing and storage of the data would be a breach of its confidentiality. From the Norwegian Personal Data Act this would also be unlawful processing of personal data linked to a natural person. Making sure to collect the proper consent and not to act outside the consent collected will be important in order to adhere to the laws imposed by the EU and the Norwegian Personal Data act. In our application to the NSD we have agreed upon a period of which the data processing shall occur. At the end of this period, at july 20th, all the personal data has to be deleted.

The data collection module can be a vulnerability for software attacks on the candidate data. A software attack on the data collection module itself could affect the integrity and confidentiality of the candidate data. If the candidate collection module were to be disabled by the attack then this could result in no candidate data being collected, thus affecting the availability of the candidate data. If improperly secured, the collection module could open up for man in the middle attacks or SQL Injections. Insufficient user authentication during data collection could give malicious code the ability to automatically post fake/flawed data.

A number of vulnerabilities can be associated with storage, some of these can be used in software attacks. When picking a storage provider, we will have to consider where the data is stored physically as well as the service level agreement, the access control, latency and ddos protection among others. Much of the risks can be entirely outsourced to the storage company, including but not limited to physical security and data backup. We are however responsible for vetting and selecting a storage solution and agreement provides an acceptable level of risk.

Candidate data		
Threat	**Vulnerability**	**CIA**
Breach of laws and regulations	Insufficient organizational policy	CIA
	Insufficient controls	CIA
	Insufficient concent	C
Software attacks	Data collection module • Man in the middle • SQL Injection • Insufficient user authentication	CI
	Data storage • Insufficient ddos protection • Insufficient access control • Unencrypted storage	CIA
	Data transfer • Man in the middle • Insufficient organizational control	C
Sabotage or vandalism	Posting of lacking/bad data • Input validation • Required fields • Authentication	I
Human error or failure	Insufficient backup solution	IA
	Data collection module • bad interaction design • Too complex user interface	I
Technical failures	Loss of hosting service	A
	Failure/Bug during collection	I

Figure 4.7: A list of vulnerabilities identified between the candidate data and threats.

Software attacks can also happen during the transfer of the candidate data itself. Missing and or insufficient organizational policy could also make the transfer vulnerable to software attacks. An example here could be someone sending the data using an encrypted email, this could be vulnerable to a software attack called ”man in the middle”.

The candidate data could be sabotaged or vandalized during collection. This could be done by intentionally posting flawed data which could have an adverse effect on the integrity of the candidate data and might also influence the results of the project itself.

Human error or failure could affect both the availability and integrity of the candidate data. If the data has an insufficient backup procedure/solution then simple user error could potentially permanently damage or delete data. Other vulnerabilities include the interaction design, if the interaction design is poor then users might enter invalid information. If the user interface is too cumbersome/complex the user might opt out of the study itself.

Technical failures would most likely affect the availability and/or integrity of the candidate data. Should the hosting/storage service be lost then the data would become unavailable. Other failures could be caused by bugs, if a bug where to occur during collection and storage then this could have negative consequences on the candidate data, for example some SQL columns could be missing due to a bug in the data tier.

Candidate Collection Module

The vulnerabilities found for the candidate collection module are detailed in figure 4.8. In our application to NSD we have agreed to inform our participants about the study and the purpose of the collection. Failing to do this would potentially mislead the participants and could result in a improper consent. We have considered this to have an impact on the availability of the collection module, as it would have to be fixed and would therefore be unavailable during the repairs.

Candidate collection module		
Threat	**Vulnerability**	**CIA**
Breach of laws and regulations	Inadequate information about: • The data collected • Purpose of the data • The data processing • Deletion date	A
	Insufficient collection of a users concent	A
Software attacks	SQL Injection	CIA
	Cross site/frame scripting	CI
	Denial of Service	A
	Man in the middle	CI
Human error or failure	Bugs and design flaws	IA
Technical failures	Loss of hosting service	A
	Loss of suggestion API's Loss of authentication service	IA

Figure 4.8: A list of vulnerabilities identified between the data collection module and threats.

The candidate module also has some vulnerabilities that would enable software attacks. Improperly escaped input fields combined with SQL could enable an SQL Injection attack, this type of attack could potentially affect both confidentiality, integrity and availability of whatever is stored in the database. The availability of the candidate collection module could be affected by a denial of service attack. Cross frame/site scripting and click-jacking could potentially be used to attack users visiting the module.

The collection module itself could be prone to human error, or technical failures. While implementing the module the authors might introduce bugs or even design flaws, this could potentially affect the integrity and availability of the collection module itself. The module will be dependant on a set of API's, losing communication with these will severely limit the module's functionality, perhaps even break it. The module will be hosted at a third party, a technical failure there would cause the module to become unavailable.

Evaluator data

The evaluator data vulnerabilities are listed in figure 4.9. Breach of laws and regulations can occur if the candidate data used in the evaluations are not properly anonymized. This could also potentially have an adverse effect on the evaluation data itself, as described in Cole et al. (2004) a bias regarding information such as gender might occur when screening resumes. The data collected from the evaluators should also be collected along with a consent of its processing and storage.

Evaluator data		
Threat	**Vulnerability**	**CIA**
Breach of laws and regulations	Insufficient anonymization of candidates	CIA
	Insufficient concent	C
Software attacks	Man in the middle	CI
	Infected personal equipment	CIA
Human error or failure	Insufficient instructions	IA

Figure 4.9: A list of vulnerabilities identified between the evaluator data and threats.

During the evaluation procedure the anonymized candidate data will most likely be transferred via mail. This opens up for software attacks such as a man in the middle attack. Other vulnerabilities include infected personal equipment of either the evaluators of the authors.

The evaluation process itself will determine the integrity of the evaluator data. Human error of failure could occur due to insufficient instructions. This could in the worst case make the evaluator data useless.

4.4.2.4 Identifying controls

The controls currently in use have been identified and listed relation to the asset/threat combos most affected. The controls can be found in the figures (4.10, 4.11, 4.12). The effectiveness of the controls are discussed the risk evaluation phase.

Candidate data	
Threat	**Controls**
Breach of laws and regulations	Anonymization of data
	Pop-up with terms and conditions
Software attacks	Data collection module • Linq statements • Outsourced user authentication
	Data storage • Azure tenant SQL • Azure tenant Blob storage • IP-Whitelist and user authentication
Sabotage or vandalism	Posting of lacking/bad data • Clientside input validation • Required fields • Prefilled fields • Single use token
Human error or failure	Code review and version control
	Data collection module • Design discussions • Interaction testing • Third party design framework
Technical failures	Azure SLA 99.9% monthly uptime

Figure 4.10: A list of controls identified between the candidate data and threats.

The control strategies used are mainly defence and transfer with some mitigation. Azure has been used for app hosting and storage, this also transfers a set of the risk management to Microsoft, including but not limited to physical security and technical failure. Several controls in the defend category has been identified, using LinQ queries towards databases for instance. Data anonymization as a control strategy would categorize as a mitigation strategy, it mitigates the impact of a breach, not the likelihood.

4.4.3 Risk evaluation

The risks are evaluated using the evaluation criteria defined during the context establishment. The impact has been evaluated based on the type of exposure (CIA) related to the vulnerability and the impact evaluation matrix found in figure 4.3. The likelihood has been estimated based on the likelihood evaluation matrix found in figure 4.2. Existing controls are factored in as estimated reductions in likelihood and/or impact. The resulting estimations of likelihood and impact found in the figures (4.13, 4.14, 4.15) are the result with the current controls applied.

Candidate collection module	
Threat	**Controls**
Breach of laws and regulations	Pop-up collecting participant consent
Software attacks	C# .NET Backend • Data layer based on LinQ
	Restrictive CORS configuration
	Outsourced authentication control
	Single use, time limited token
Human error or failure	Design meetings
	Version control
	Code review
Technical failures	Azure tenant App SLA 99% uptime
	Staged deployment of versions • Development stage • Testing stage and environment • Production

Figure 4.11: A list of controls identified between the collection module and threats.

Evaluator data	
Threat	**Controls**
Breach of laws and regulations	Anonymization of candidate data
	Textual consent over email
Human error or failure	Instructional brochure

Figure 4.12: A list of controls identified between the evaluator data and threats.

4.4.3.1 Candidate data risk evaluation

The figure 4.13 details the vulnerabilities and their contribution to the threat exposure of the asset. The risk of a breach of laws and regulations due to insufficient organizational policy has been evaluated to be medium. The data anonymization is estimated to mitigate the impact of a breach of confidentiality. The risk due to insufficient controls is also estimated to be medium.

Candidate data risk evaluation				
Vulnerability	**Control**	**Impact**	**Likelihood**	**Risk**
Insufficient organizational policy	Data anonymization	MODERATE	UNLIKELY	MEDIUM
Insufficient controls		MODERATE	UNLIKELY	MEDIUM
Insufficient concent	Pop-up with terms and conditions			
Data collection module • Man in the middle • SQL Injection • Insufficient user authentication	Data collection module • Linq statements • Outsourced user authentication • SSL	MODERATE	RARE	LOW
Data storage • Insufficient ddos protection • Insufficient access control • Unencrypted storage	Data storage • Azure tenant SQL • Azure tenant Blob storage • IP-Whitelist and user authentication • Data anonymization	MODERATE	UNLIKELY	MEDIUM
Data transfer • Man in the middle • Insufficient organizational control		MODERATE	UNLIKELY	MEDIUM
Posting of lacking/bad data • Input validation • Required fields • Authentication	Clientside input validation Required fields Prefilled fields Single use token	MINOR	UNLIKELY	LOW
Insufficient backup solution		MAJOR	UNLIKELY	HIGH
Data collection module • bad interaction design • Too complex user interface	Design discussions Interaction testing UI framework	MINOR	POSSIBLE	MEDIUM
Loss of hosting service	Azure SLA 99.9% monthly	MINOR	RARE	LOW
Failure/Bug during collection	Code review Test driven development	MODERATE	RARE	LOW

Figure 4.13: Estimated risks for the candidate data.

The vulnerabilities to software attacks, introduced through the data collection module have been evaluated to pose a low risk. The likelihood of a successful man in the middle attack is greatly reduced by using SSL with official certificates. The likelihood of a significant impact SQL Injection is reduced by utilizing LinQ for all database queries. LinQ will automatically escape and utilize prepared statements. The candidate module uses a third party authentication service.

For data storage the risk has been evaluated to medium. The impact of a breach is considered moderate, lowered from major due to data anonymization. The likelihood of breach due to the vulnerabilities has been estimated to be unlikely. The same impact and likelihood has been estimated for the vulnerability to a software attack during data transfer.

The risk of sabotage and vandalism has been evaluated as low. Input validation and required fields are checked both client side and server-side. The submission token used is a single use token, this means that every new submission will require the user to obtain a new token from the authentication service.

The current risk level for human error or failure is high. The vulnerability of not having a backup solution for the storage combined with human error or failure is considered to have a potentially major impact, the likelihood however is evaluated as unlikely. This results in a high risk for the current lack of a backup solution. Human error can have an impact on the integrity of the candidate data, the vulnerability here is the UI and interaction design of the data collection module. The controls limit the impact, however because human errors are still possible the risk is evaluated as medium.

Technical failures have been evaluated with a low risk. The impact of a technical failure affecting the availability is considered minor, and the likelihood is considered rare due to a the Azure SLA guaranteeing 99.9% uptime. A failure or a bug in the collection module could potentially impact the integrity on a large scale. The impact is limited to moderate and the likelihood is evaluated to be rare, this is due to test driven development and code reviews.

4.4.3.2 Collection module risk evaluation

The candidate collection module has one unacceptable risk, figure 4.14. Due to inadequate information about the study, the consent collected from the participants could have been given on a false premise. The likelihood has been evaluated as possible and the impact as moderate, this makes the risk high. The vulnerability of insufficient collection of consent, deals with the way the consent is collected. The current control involves a pop up modal prompting the user to accept or decline to give his consent.

4.4.3.3 Evaluation data risk evaluation

The evaluation data risk evaluation can be seen in figure 4.15. The existing controls have been effective in reducing the risk level for most of the threats, however the current controls do not sufficiently protect against a man in the middle attack. This results in a medium risk level.

4.4.4 Risk treatment

The main cause of the high risks involved in with the candidate data was due to impact should the confidentiality be breached. This impact was mitigated by anonymizing the data, however to get the risks down to an acceptable level the data storage should also be encrypted. The risk exposed by insufficient organizational policy can be defended against by implementing a processing policy. A similar strategy can be used to defend

Candidate collection module risk evaluation				
Vulnerability	**Control**	**Impact**	**Likelihood**	**Risk**
Inadequate information about: • The data collected • Purpose of the data • The data processing • Deletion date		MODERATE	POSSIBLE	HIGH
Insufficient collection of a users concent	Pop-up collecting participant consent	MODERATE	RARE	LOW
SQL Injection	C# .NET Backend • Data layer based on LinQ	MINOR	RARE	LOW
Cross site/frame scripting	Restrictive CORS configuration	MINOR	UNLIKELY	LOW
Denial of Service	Azure DDOS protection	ZERO	UNLIKELY	LOW
Man in the middle	SSL	ZERO	UNLIKELY	LOW
Bugs and design flaws	Design meetings Code review Version control Staged deployment of versions • Development stage • Testing stage and environment • Production	MODERATE	RARE	LOW
Loss of hosting service	Azure tenant App SLA 99% uptime	MINOR	RARE	LOW
Loss of suggestion API's Loss of authentication service		MINOR	UNLIKELY	LOW

Figure 4.14: Estimated risks for the collection module.

Evaluator data risk evaluation				
Vulnerability	**Control**	**Impact**	**Likelihood**	**Risk**
Insufficient anonymization of candidates	Anonymization by redaction of • name • age • gender • location • ethnicity • sexual orientation • pictures	MODERATE	RARE	LOW
Insufficient concent	Textual consent over email	MODERATE	RARE	LOW
Man in the middle		MODERATE	UNLIKELY	MEDIUM
Infected personal equipment		MINOR	UNLIKELY	LOW
Insufficient instructions	Instructional brochure	MODERATE	RARE	LOW

Figure 4.15: Estimated risks for the evaluation.

against insufficient controls, setting up a dedicated processing environment. A dedicated vm environment for processing the candidate data would give a better overview of the controls and vulnerabilities involved.

Candidate data risk treatment				
Vulnerability	**Control**	**Risk**	**Treatment**	**New Risk**
Insufficient organizational policy Insufficient controls	Data anonymization	MEDIUM	Defend • Processing policy • Dedicated processing environment	LOW
Data storage • Insufficient ddos protection • Insufficient access control • Unencrypted storage	Data storage • Azure tenant SQL • Azure tenant Blob storage • IP-Whitelist and user authentication • Data anonymization	MEDIUM	Mitigate • Storage Encryption	LOW
Data transfer • Man in the middle • Insufficient organizational control		MEDIUM	Mitigate • Email Encryption • SSL	LOW
Insufficient backup solution		HIGH	Mitigate • Azure cloud backup	LOW
Data collection module • bad interaction design • Too complex user interface	Design discussions Interaction testing UI framework	MEDIUM	Mitigate • Instruction video	LOW

Figure 4.16: Treatments for the risks related to the candidate data.

Azure cloud backup has been suggested as a treatment for the threat of human error or failure. This will reduce the impact of a failure and the resulting risk after treatment is therefore low.

The treatments suggested for the candidate collection module and the evaluator data can be seen in figures (4.17 and 4.18). A set of articles containing information about the study and the life cycle of the data collected will ensure that the participants have good overview of what they are participating in. These articles can be found in the appendix. Using encrypted emails when transferring evaluation data will lower the impact should a man in the middle attack occur.

Candidate collection module risk evaluation				
Vulnerability	**Control**	**Risk**	**Treatment**	**New Risk**
Inadequate information about: • The data collected • Purpose of the data • The data processing • Deletion date		HIGH	**Defend** Articles describing the lifecycle and purpose of the data and the study.	LOW

Figure 4.17: Treatments for the risks related to the collection module.

Evaluator data risk evaluation				
Vulnerability	**Control**	**Risk**	**Treatment**	**New Risk**
Man in the middle		MEDIUM	**Mitigate** Email encryption	LOW

Figure 4.18: Treatments for the risks related to the Evaluation data.

4.4.5 Risk acceptance

In our context establishment we decided that an acceptable risk level would be low. With the new controls suggested in the risk treatment section implemented the risks are now at an acceptable level.

Chapter 5

Methods and Experiment

This chapter details the methods that will be used to evaluate the candidate ranking system described in chapter 3. We have divided this chapter into sections each explaining a method used.

5.1 Introduction

In chapter 1 we defined a set of research questions and mentioned the methods we would use to answer them, the main method being design and creation. In this chapter we will go into how the resulting artifact will be evaluated, the artifact being the candidate ranking system. We will take a look at the data generation methods that will be employed, as well as how we plan on using the resulting data in the evaluation of the system.

The main parts of our plan for evaluating the system includes training the system, testing it on real world recruitment data and comparing the results with the results of a human expert. This method of evaluation has been used with other studies Faliagka et al. (2012a, 2014, 2015); Menon and Rahulnath (2016); Gil et al. (2016), all of these focusing on ranking/screening candidates in recruitment. Mentioned by Kmail et al. (2015), the weakness of this is that candidates selected by the human expert does not necessarily represent the best list. To explore this further, an informal experiment experiment has been designed, slightly inspired by the Touring test Turing (2009).

5.2 Evaluation methods

The ranking system will be evaluated against data collected from real and simulated recruitment scenarios. The system will be compared against the information retrieval ranking algorithm Okapi BM25, as suggested in Gil et al. (2016). The BM25 would be a good baseline for determining the performance of learning techniques as this is one of the common bag of words based information retrieval techniques.

5.2.1 Data generation

A set of candidates and a job description will be used by the system to generate a list of ranked candidates to the job. A list will also be selected and sorted by the relevance of the candidates to the job, this will be done by a human expert. As a baseline, a list will also be generated by the Okapi BM25 algorithm.

5.2.2 Precision and recall

The list generated by our system and the list generated by Okapi BM25 will both be measured against the list generated by the human expert. Precision and recall will be determined and from these the e-measures will be calculated. The result will indicate how well the system was able to reproduce the content of the list. A section about precicion and recall can be found in chapter 2.

5.2.3 Mean Average Precision

The mean average precision will be used in the experiment to calculate average precision score for the precision for the set of job positions.

5.2.4 Precision@K

Precision at k looks at the precision level after certain hits in order to evaluate how good the rankings are after a given ranking order.

5.2.5 E-Measure

Based on the calculated precision and recall, an E-Measure that favors precision will be utilized as mention in the precision and recall section above.

5.2.6 Spear-man coefficient

The system will also be measured for its ability to produce a correctly ordered list of candidates. The difference between the order of two lists will be measured using the spearman coefficient. More information about the spearman coefficient can be found in chapter 2.

5.2.7 Discounted cumulative gain

Oates (2006) explains that even though precision and recall are broadly used, it only gives us binary relevance assessments and might be heavily influenced by relevant documents that we found late in the ranking. This might blur the distinction between a ranking system that retrieves highly relevant candidates at the top of the ranking and a ranking system that retrieves only the middle important ones at the top of the ranking. Even though we opted to use the Spear-man Correlation coefficient, it does not distinguish the distance between two positions further down the ranking and in the top of the ranking. We therefore felt it was important to include the DCG. More about the DCG is explained in section2.5.2.6.

From 2.5.2.6 we learned that the DCG allows for several shades of relevance in the ranking. We have therefore created a relevance Rank based on the top-25, where the ten first are considered relevant. These ten ranking positions was given a weight in descending order from 10 to 1 and all the rest was given 0. Chapter 6 section 6.4.3.2 will present this further and the results.

5.2.8 Okapi BM25 as a baseline

As described in 5.2, BM25 has been used in similar studies was considered by Gil et al. (2016) to be the best available baseline. This is due to the long track record of the BM algorithms in information retrieval and earlier systems for ranking candidates based on a wordbag model have used it.

We use the Okapi BM25 implementation found in Apache Lucene [1]. The resumes of the candidates are first indexed. The position description is then cleared parsed, leaving only behind the criteria keywords. These are then used by Okapi to score the resumes. The output is a list of resumes ordered by the scores assigned to them by Okapi BM25.

5.3 Experiment

This section will explain the evaluation methods conducted in the experiment as well as the limitations regarding variables and control needed to account for.

5.3.1 Introduction

The experiment we have devised cannot be considered a formal experiment as major parts of it is uncontrolled. One of the biggest factors being the participants themselves, their unexplored biases regarding uncontrolled variables in the CV's. However, we believe that the results from the experiment can provide indications.

As mentioned initially, the experiment itself has been slightly inspired by the touring test Turing (2009). The idea is simple, a human recruiter is given a job description and some lists of candidates. They are then asked to rank these lists in the order of the best matching list to the worst. The inspiration from the touring test is that the recruiters themselves do not know which of the lists are created by an algorithm. We will use this approach to gather data about the performance of both our system and the BM25 algorithm compared to the performance of a human recruiter.

The main benefit of this experiment type is that it could potentially be used to identify an algorithm performing better than a singular human expert. This would require it to be conducted in a controlled manner and on a large sample of participants. This also assumes that the performance of a human expert can be judged by his peers.

[1] https://lucene.apache.org/

5.3.2 Variables and Controls

In the study Arnulf et al. (2010), it was found that the layout of CV's had a measurable degree of influence on the decision of recruiter. They found that for the same candidate, when their resume was structured in a formal way the candidate was twice as employable as compared to when their resume had a more creative layout. Because our system does have access to the same layout information we have decided to remove resumes that do not follow the formal layout described in Arnulf et al. (2010).

We have also decided to remove information about gender, age and ethnicity to limit the amount variables. The field study Cole et al. (2004) found indications of a gender bias in resume evaluation. Findings from the study Eva et al. (2012) indicates the existence of an ethnicity bias in the screening of resumes.

5.3.3 Recruiters and Evaluators

When referring to recruiters and evaluators, we have to differentiate the meaning of these two roles. The set of recruiters that had the responsibility to evaluate the experiment are referred to further as "Evaluators". The recruiters that created their top five lists are referred to as "HR", which is an abbreviation for Human Recruiter.

In order to have expertise evaluation, we contacted and chose a variety of recruiters that had experience with screening and scoring candidates. However, there's not a single answer key for each scoring. We therefore asked each of the evaluators to write down which list they found most relevant. In addition, they should also write a comment to justify their selection. The last step we asked them for was to give us their subjectively opinion of how many of the candidates from each list they found to be relevant for the job position. From the evaluation their task was therefore to:

- Tell which list the found to be best
- Comments to why they chose as they did
- Tell us from 1-5 how relevant each list was where 1 means 1 and 5 means all candidates in the list was relevant.

5.3.4 Training the system

A set of 40 jobs where taken from HigherEd's platform. These jobs were selected within categories matching the most common education fields among the candidates collected through HigherEd, see figure 5.1. From the set of 40 jobs, a subset of 20 was chosen by a recruiter from HigherEd. The training module was then used by this recruiter to create job queries and revise these according to the criteria found in the 20 job ads.

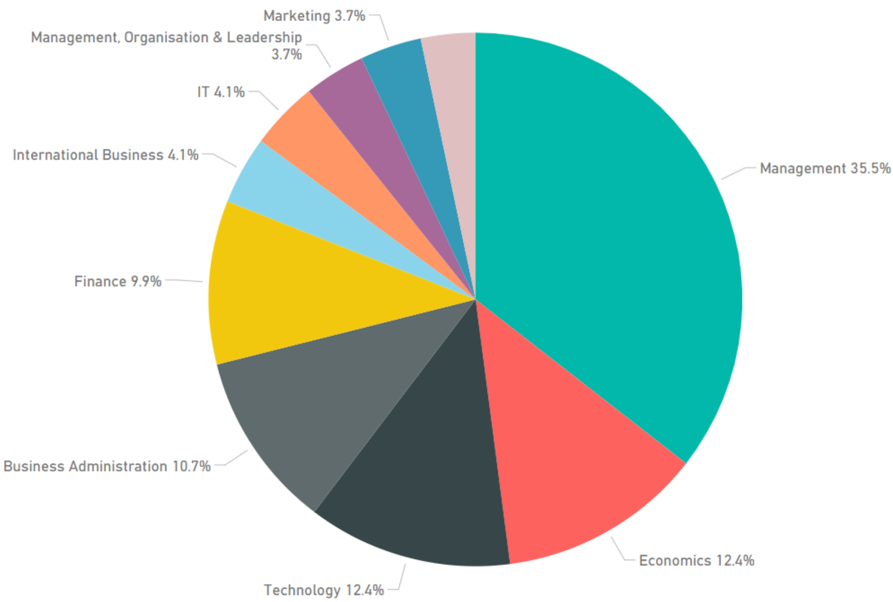


Figure 5.1: The 10 most common education fields among the candidates gathered through HigherEd

5.3.5 Data set

The data-set used consists of four job positions with their description and criteria. These four jobs where selected by the authors from the list of 20 jobs in subsection 5.3.4 that were not used in training the system. For each job position we picked 30 applicants out of the candidates harvested through HigherEd. From each of these lists of 30 applicants a set of five top 5 lists was screened. One of the five lists was screened by a human recruiter, one list by the CRS, one by BM25, one is a copy of the list screened by the human expert rearranged by the CRS and the last list is generated by random.

5.3.6 Setup

The experiment had a few prerequisites, among these the four sets containing a job and 30 candidates outlined in section 5.3.5. During the setup phase of the experiment this data needed to be processed into what we call scenario packs. Each scenario pack will contain a job description, 30 candidates and a questionnaire. The questionnaire contained some questions about how one would weight the criteria outlined in the job description and a section where the top 5 candidates can be filled in.

The scenario packs was sent out by the authors and our business contacts in order to search for recruiters to help us in the evaluation.

A set of flyers were designed and sent around in order to collect experiment participants. These flyers can be seen in Appendix B. From the appendix, we originally wanted the recruiters to participate in both of the steps shown in B.1 and B.1. However, due to

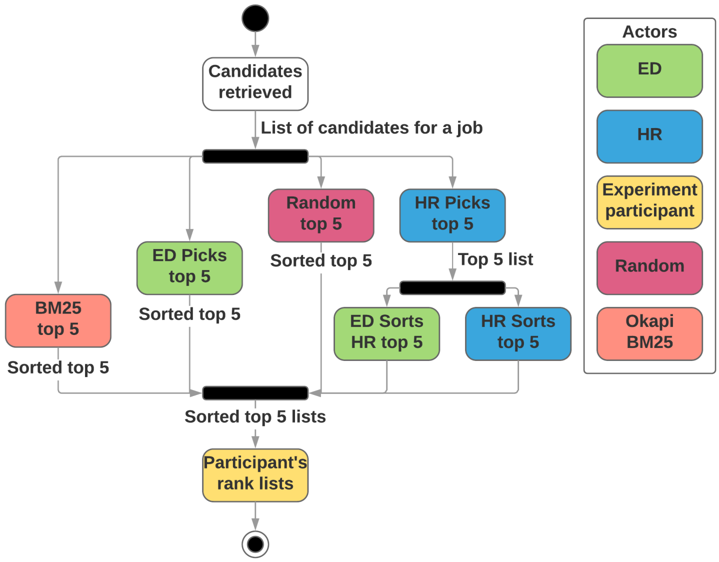


Figure 5.2: A UML Activity diagram detailing the experiment plan

the time consumption, we received feedback that some could participate with certain parts, and some could participate in both. The original plan however, is outlined in appendix B.5. This showed that the same recruiter would be mapped to evaluate a different job position.

When we received the responses for the scenario packs, we could start with the facilitation for the evaluation packs.

The evaluation packs consisted of a set of top 5 lists of candidates matched with a job-description. One list will have been made by a recruitment professional, one by our system, one by BM25, one random and one by another recruitment professional reordered by our system.

5.3.7 Plan

In figure 5.2 we have diagrammed the experiment plan. In the first phase of the plan we have a set of CV's applying to some job. In the next phase, a HR expert will pick his top 5 CV's for the job and rank them in order of preference. Another set of top 5 CV's will be picked and ranked by our CBR algorithm. The next part of the experiment involves an experiment participant rating these top 5 lists. This would however only tell us something if the CBR system could consistently outperform the human recruiter. As a control we added a random generated top 5 list as-well, that we predicted would be be ranked worst. The fourth top 5 list was a copy of the list selected by the HR expert, except it was re-ranked by the CRS. This was done to gain insight into the systems ability to produce a good order.

A fifth list of CV's generated by the fully automated Okapi BM25 algorithm that we in-

n	Exact	Cum	Inverse
0	0.37283	0.37283	1.00000
1	0.44384	0.81667	0.62717
2	0.16140	0.97806	0.18333
3	0.02105	0.99912	0.02194
4	0.00088	0.99999	0.00088
5	0.00001	1.00000	0.00001

Table 5.1: Probability distribution 30 choose 5, with 5 relevant

troduced in 2.5.2.1. As explained in section 2.5.2.1 this algorithm has been recommended as a baseline for evaluating new ranking methods (Gil et al., 2016).

5.3.8 Evaluation procedure

The top 5 lists produced as part of the experiment will be used both individually to measure precision and correlation between the CRS and HR using Okapi BM25 as a baseline.

$$E(X) = \sum_{k=1}^{n} k * \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}} \tag{5.1}$$

In table 5.1 you can see a hypergeometric probability distribution for N=30, K=5 and n=5. From the table you can see that when picking at random you can expect to get one candidate correctly about 44.4% of the time. From our 4 packs of 30 candidates, given that they contain 5 relevant candidates, from 5.1 we expect that the random list on average will have 0.88 relevant candidates, from the expected value of a hypergeometric distribution. If the evaluators consistently assign the random list a high number of relevant candidates then this might indicate that the list of 30 candidates for that specific job contains more than 5 relevant candidates.

The evaluator will for each position pick the list that they consider the best. As explained in the introduction 5.3, we are not able to get enough evaluators to determine whether a recruiter on average will prefer one list over the other. However we will still be able to say something about the recruiters in our sample. The amount of wins will be evaluated as a whole and per position, we use the null hypothesis that there is a uniformly distributed probability over the lists. With this hypothesis calculate the probability of a list getting picked several times. We will use an $\alpha = 0.05$ when determining statistical significance. Recall, precision and a spearman's correlation coefficient will also be calculated for CRS and BM25 in relation to HR.

The (MEAP) mean average precision will be calculated accross all the positions, for both BM25 and the CRS. Each position will be treated as a query where the relevant candidates are given by the HR list. The average precision at recall will also be calculated.

We will also use the binominal distribution to measure the likelihood of getting the amount of wins if randomly selected. The binominal distribution formula is given in formula:

k	**p=1/5**
0	0.035
1	0.132
2	0.231
3	0.250
4	0.188
5	0.103
6	0.043
7	0.014
8	0.003
9	0.001
10	0.000
11	0.000
12	0.000
13	0.000
14	0.000
15	0.000

Table 5.2: Binominal distribution showing the probability for win by random given 15 attempts and five lists

$$P(X = k) = \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k} \tag{5.2}$$

where n is the amount of tries, X is the amount of times that an event occurs. p is the probability that the event occurs, and k is the amount of times the p occurs. From table 5.2, shows the probability of selecting random x times among the five lists in the experiment given the amount 15 tries.

5.4 Real world scenario

In this section we will explain the evaluation procedure, dataset and the evaluation methods regarding the real world scenario. The difference between this scenario and the experiment, is that all the candidates used in this set had applied to the specific job. This mean that instead of finding a top 5 list based on a variety of data set, it should now rank a set of 25 candidates where the differences may be smaller. By running the system in a completely different environment than the one it was trained in could also give indications of how robust the solution is.

5.4.1 Introduction

Our business contact, Ingraphic agreed to facilitate a real world recruitment scenario for us. To this end they gave us access to the applicants for a position that they were hiring for, the job description and the shortlist of top 25 candidates that resulted after screening the 170 applicants. The job in question is a ”Business Intelligence Consultant” position.

5.4.2 Setup

A 6 month old recruitment scenario where Ingraphic hired a Data Analyst was used to train the candidate ranking system. Where the authors manually filled in the old resumes into the data collection module.

And 4 months later ingraphic was hireing again, this time for a Business Intelligence Consultant. The authors manually filled in the candidates CV’s into the data collection module.

5.4.3 Evaluation Procedure

Originally, we wanted to calculate the precision and recall from the top-25 based on the entire data-set of 170 applicants. But to anonymize the entire data-set and register each user into the CRS manually turned out to be a time consuming job that may not provide us with the new knowledge we wanted. Since this real world scenario contained a more narrowed set, it was more interesting to match the correlation to see how precise our CRS could match the human recruiter. We therefore opted to utilize the Spearman Rank Correlation Coefficient to measure the distance for each specific position in the top-25 rank, and compare the results with what Okapi.

Based on the feedback from the human recruiter in Ingraphic that provided the top-25 list, we were told that its top ten was especially noteworthy. From this information, we decided to facilitate the real world scenario by measuring the precision and recall. Also we wanted to use the P@K - Precision at K explained in 5.2, to measure the precision at certain selection levels. In addition we also used the the discounted cumulative gain to penalize the ranking algorithms that had high-ranked candidates in lower positions of the rank.

5.4.4 Data set

25 candidates based on the recruiter’s top 25 out of the 170 initially was used to measure the two approaches.

5.5 Candidate collection module

An evaluation of the collection method and procedure was proposed in order to gain some insights into both the quality of the data and the factors involved in users pulling out.

5.5.1 Evaluation method

This was done using the power bi analytics tool on the collected data and a questionnaire was sent out to some of the participants.

5.5.1.1 Questionnaire

A self administered questionnaire containing opinions was prepared to gather evaluation data about the DCM module and the quality of the data gathered. According to Oates (2006) there are several benefits of using a self-administrated questionnaires:

- It will save researchers time
- Respondents are less likely to try to please the researchers
- We as researchers can be confident that all respondents saw same questions
- A self-administered Questionnaire to answer the position criteria.
- More people can be asked to complete the questionnaire

Appendix B.3 displays the questionnaire containing closed questions and one open question for giving feedback on a dream job. The questionnaire was developed with the main focus of being short and therefore more compelling to fill in.

The questionnaire was created using Google forms. A small set of three questions were set as required and placed at the very top of the questionnaire. The remaining questions were marked as optional. This enabled us to formulate the email distributing the questionnaire in a way that presumably would make it more compelling for those short on time.

Chapter 6

Results

This chapter presents our empirical results gathered using questionnaires and evaluation procedures introduced in 5. We start by presenting the results obtained for and from the data collection module. Afterwards we present the raw data obtained from running the experiment and the data gathered from running the system in a real world scenario.

6.1 Results Data gathering

The collection module was integrated into the platforms of two companies, Highered and Ingraphic. Combined a total of 476 candidates were collected using the collection module, distributed in two seprate databases. In this section we present some statistics on the collected candidates, and some of our findings regarding the candidate selction.

6.1.1 Candidates

For the deployment of the CRS onto Higherd's platform, we montiored and created several analytics.

6.2 Results Collection module

A questionnaire was prepared and issued to the participants collected from Highered. The purpose of which was to evaluate the collection module itself, this questionnaire recieved about 100 respondents. The questionnaire can be found in the appendix B.3.

6.2.1 Usability & Look and Feel

We asked the respondents how they would rate the DCM Module between 1-5 based on the Usability and look and feel of using the system. 1 Represented bad and 5 great. These results are presented in figure 6.2. The results showed that the majority found the DCM

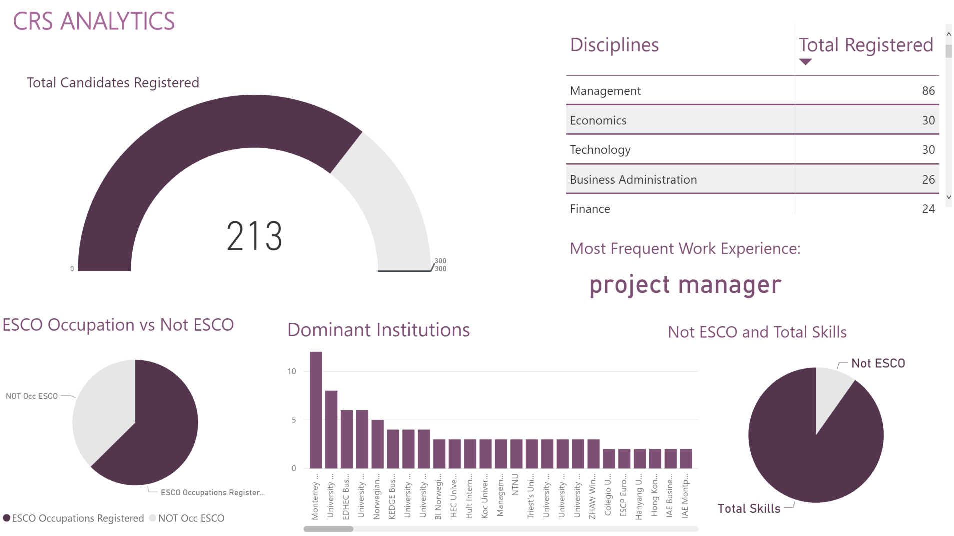


Figure 6.1: A PowerBI report showing the analytics monitoring data from the data collection in the DCM deployed to Highered's platform

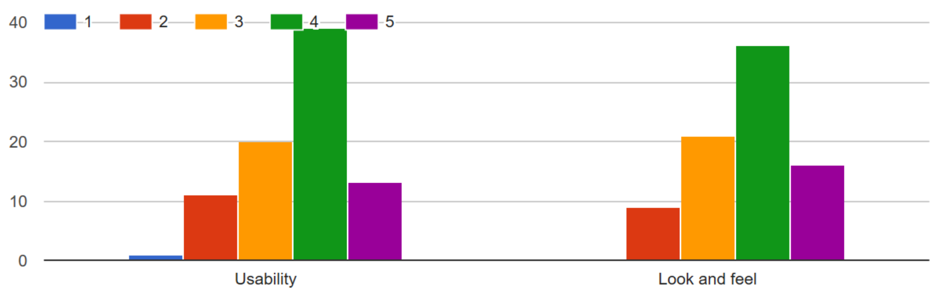


Figure 6.2: Chart showing the spread between Usability and Look and Feel

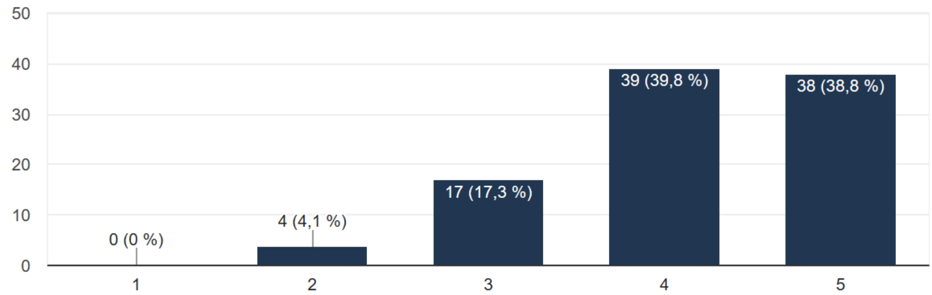


Figure 6.3: Chart presenting the similarity between the CV and DCM Profile

Module easy to use and had a welcoming interface.

6.2.2 Similarity between DCM and CV

We asked the respondents to answer How similar the data in their CV's was compared to the data sent to the CRS with respect to skills, languages, educations and occupations. Grade 1 represents that there was no similarity and 5 identical.

From Figure 6.3 we see that the majority respondent that the attributes was identical or almost identical with a combined percentage of 78.6%.

6.2.3 Evaluating ESCO

We asked the respondents if they felt the suggestions for skills and occupations was sufficient. Based on 100 candidates 59.2% answered that it was sufficient and 40.8% felt it lacked some options.

6.2.4 ESCO Sufficiency

The respondents were asked how sufficent they found using the CRS API presented in figure 3.22 to connect to ESCO discussed in section 2.4.7.1 in order to retrieve occupations and skills.

The results shows that the majority found it sufficient, but a large amount may have missed some important or less important attributes.

6.2.5 Effort Spent Each Section

We wanted to provide some metrics on how time consuming each attribute section was. This may say something about the threshold for filling out in each attribute. 1 meaning a small amount of time spent and 5 meaning much time spent.

The results indicates that the majority spent equally the same amount of time on filling out each section.

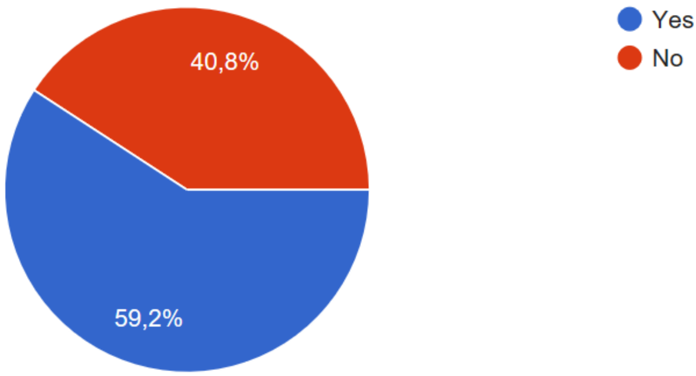


Figure 6.4: Pie Chart showing the sufficiency of using ESCO for finding skills and previous work experience

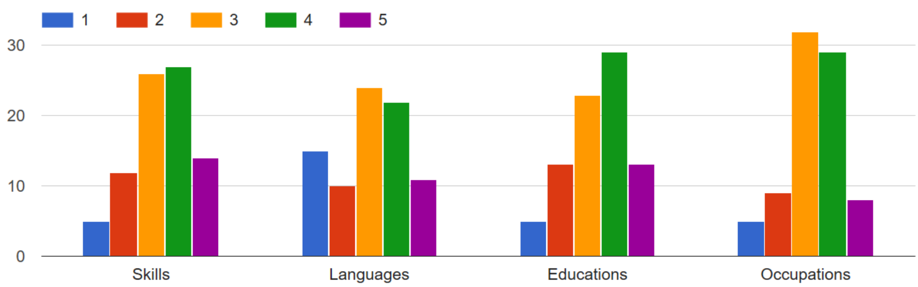


Figure 6.5: Chart showing the effort spent distribution among the attributes

6.3 Results from Experiment

This section presents the results obtained from the experiment conducted with human recruiters. The four positions are collected from Highered's platform and the results from each position are listed in each subsection. We anonymized the lists in advance so it shouldn't be clear which lists whom had made. We also changed the order for each position in case the recruiter evaluated several positions and certain patterns may be recognized. Each Evaluation-pack contained as mentioned previously: a random list, a list produced by our CRS, a reranked list from the CRS based on the list of the recruiter and a list produced by utilizing the Okapi BM 25 algorithm.

6.3.1 List Creation

This section presents the feedback from the recruiters that participated in the creation of top-5 lists for the four given positions. There was a total of 10 evaluators who participated in creating lists and 4 out 10 was randomly chosen to be used in the experiment. The tables below presents their given relevance ranking for each of the positions in the top-5 list they made from 1-5 where 5 meant the candidate could be called in for an interview.

Rank Position	How Relevant
First Place	5
Second Place	5
Third Place	4
Fourth Place	4
Fifth Place	4

Table 6.1: Relevance feedback by the recruiter at Experis for position Accounting Analyst top-5 list

Rank Position	How Relevant
First Place	5
Second Place	5
Third Place	5
Fourth Place	5
Fifth Place	5

Table 6.2: Relevance feedback by the recruiter at Highered for position Industrial Placement top-5 list

6.3.2 Results from Job Position Marketing and Sales Consultant

This section presents the result for the position Marketing and Sales Consultant. The position is retrieved from a company called BASF, which is the largest chemical producer in the world. The criterias for the list can be seen in appendix B.4.

Rank Position	How Relevant
First Place	5
Second Place	5
Third Place	5
Fourth Place	2
Fifth Place	2

Table 6.3: Relevance feedback by the recruiter at AmroDelphi for position Marketing and Sales Consultant top-5 list

Rank Position	How Relevant
First Place	4
Second Place	3
Third Place	2
Fourth Place	2
Fifth Place	1

Table 6.4: Relevance feedback by the recruiter 2 at Experis for position Financial Accountant

6.3.2.1 List Preprocessing

Table 6.5 below shows the mapping between the list number and the participant that created the list in the evaluation pack: The table below shows the results produced by each

List Number	Participants
List 1	Random
List 2	CRS
List 3	CRS-Reranked
List 4	Okapi BM 25
List 5	Recruiter AmroDelphi

Table 6.5: Showing the mapping between list number and participant for position Marketing and Sales Consultant

participant.

Rank	HR	CRS-Reranked HR	CRS	Okapi-Reranked HR	Okapi	Random
1	16686	16686	16686	40397	40397	22748
2	40397	40397	22748	16686	58214	50168
3	43451	13045	20313	46221	16686	38938
4	46221	46221	45215	13045	22748	58112
5	13045	43451	58214	43451	23513	58069

6.3.2.2 Evaluators Results

This section presents the results obtained for each Evaluator that has given feedback for the position Marketing and Sales Consultant

Evaluator1 - Emilie Recruiter at Ingraphic

Scored: List2 as the best list among the 5.
Comment from Evaluator: ”I chose this list because it contained most relevant candidates. I mean that the top 2 from list 2 was very relevant compared to the top candidates in other lists”.
Relevance Rank: Table 6.10

Evaluator2 - Maiken Recruiter at Princess

Scored: List2 as the best list among the 5.
Comment from Evaluator: ”All the candidates in list 2 fit well with the given criterias. They had both relevant education and experience. They all had analytical capabilities, leadership, marketing and sales experiment. They all also possesed the language skills of both english and german, that was listed as a positiv advantage. The only one that lacked some marketing and sales experiment was the one in the fourth place position. However, this candidate had both good leader and analytical experience and project experience”.
Relevance Rank: Table 6.10

Evaluator3 - Arve Recruiter at ICD

Scored: List2 as the best list among the 5.
Comment from Evaluator: ”Lists 2,3,5 all had the same amount of relevant candidates. However, list number 2 seemed to consist of candidates supporting a higher proficiency in German. Lists 3 and 5 seem to contain the same candidates, however here I preferere the arrangement of list 5.”.
Relevance Rank: Table 6.10

Evaluator4 - Sofie Recruiter at ICD

Scored: List5 as the best list among the 5.
Comment from Evaluator: ” I am unsure if it required both english and german from the description. From the criterias it may seem so. The lists 2 and 5 had many relevant candidates. For me, it seemed that list 5 had most experienced, but list 2 had most with german knowledge”
Relevance Rank: Table 6.10

Evaluator5 - Gustaf recruiter at HigherEd

Scored: List2 as the best list among the 5.
Comment from Evaluator: ”List 2 was most oriented towards sales and marketing background and interest. The others was more random.”
Relevance Rank: Table 6.10

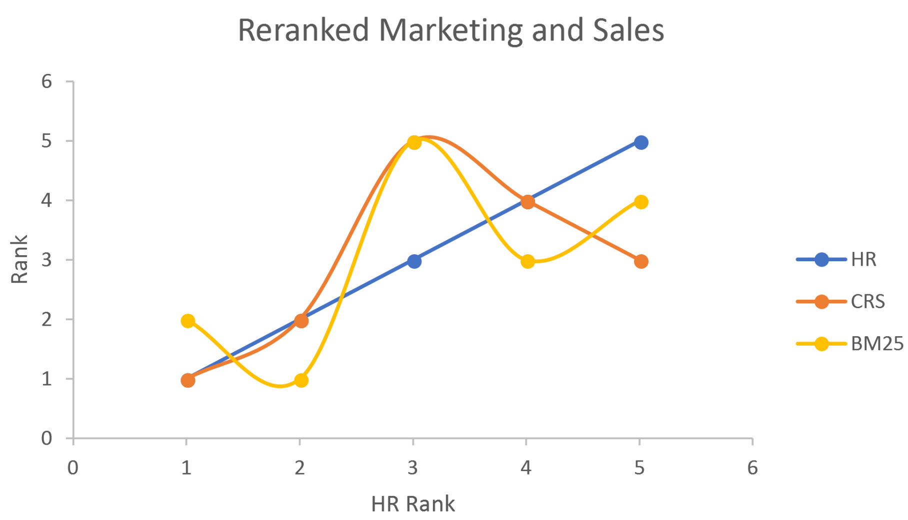


Figure 6.6: Graph displaying the HR top 5 list reordered by Okapi BM25 and CRS

Evaluator6 - Martine newly educated recruiter

Scored: List2 as the best list among the 5.
Comment from Evaluator: " I felt that the second list had the strongest candidates with respect to the education and experience needed for the position criterias."
Relevance Rank: Table 6.10

6.3.2.3 Correlation and precision

The following paragraphs will be used to present the calculated correlation coefficient and the level of precision and recall. As described in chapter 5, in these calculations the HR list is used as the reference to compare the CRS against BM25.

Spearman's correlation coefficient

The spearman coefficients and the corresponding z, P(z > Z), P(z < Z) values for a normal distribution can be found in table 6.6.

-	CRS	BM25
ρ	0.6	0.6
z-score	-1.2	-1.2
P(Z < z)	0.885	0.885
P(Z > z)	0.115	0.115

Table 6.6: Spearman's coefficient ρ, and Z score

Precision

The recall for the CRS on the HR list, found in table 6.7 has a cumulative probability of P(X>=1) = 0.62717, not statistically significant. BM25 found in table 6.8 has a P(X>=2) = 0.18333.

HR	CRS	Relevant	Recall	Precision
16686	16686	1	0.2	1
40397	22748	1	0.2	0.5
43451	20313	1	0.2	0.32
46221	45215	1	0.2	0.25
13045	58214	1	0.2	0.2

Table 6.7: Precision and recall calculated for CRS, with the HR list as relevant for Marketing and Sales Consultant.

HR	BM25	Relevant	Recall	Precision
16686	40397	1	0.2	1
40397	58214	1	0.2	0.5
43451	16686	2	0.4	0.67
46221	22748	2	0.4	0.5
13045	23513	2	0.4	0.4

Table 6.8: Precision and recall calculated for BM25, with the HR list as relevant

CRS	P(X>=5)
5	0.15136%

Table 6.9: Times favoured and the probability for this by random chance

Relevance ranks

Participants	Maiken	Emilie	Arve	Sofie	Gustaf	Martine	Average
HR	3	4	4	3	2	4	3.5
HR-CRS	4	4	4	2	2	4	3.5
CRS	5	5	4	3	3	5	4.25
OKAPI	3	3	3	2	2	3	2.75
RANDOM	1	0	1	0	0	1	0.5

Table 6.10: A table of the relevance ranks given assigned by evaluators for the position Marketing and Sales

6.3.3 Results from Job Position Accounting Analyst

this section presents the result for the position Marketing and Sales Consultant. The position is retrieved from a company called BASF, which is the largest chemical producer in the world. The criterias for the list can be seen in appendix B.4.

6.3.3.1 List Preprocessing

Table 6.11 below shows the mapping between the list number and the participant that created the list in the evaluation pack:

List Number	Participants
List 1	CRS
List 2	Random
List 3	Recruiter Experis (HR)
List 4	Okapi BM 25
List 5	CRS Re-Ranked HR

Table 6.11: Showing the mapping between list number and participant for position Accounting Analyst

Rank	Recruiter	CRS-Reranked HR	CRS	Okapi-Reranked HR	Okapi	Random
1	37538	7926	7926	17214	17124	57466
2	9237	9237	9237	9237	9237	51648
3	46171	37538	26343	37358	38938	9237
4	7926	46171	37538	46171	39385	23626
5	17124	17124	54675	7926	37538	52287

6.3.3.2 Evaluators Result

Evaluator1 - Gustaf recruiter at HigherEd

Scored: List1 as the best list among the 5.
Comment from Evaluator: ”To be honest, non of the lists where really relevant in my eyes, and I would hardly invite any of them for interviews. However, I still see that list 1, 4, 5 contains quite similar candidates, so this might lead me to belive that these are the most relevant of the selection.”.
Relevance Rank: Table 6.16

6.3.3.3 Correlation and precision

Spearman’s correlation coefficient

The spearman coefficients and the corresponding z, P(z >Z), P(z < Z) values for a normal distribution can be found in table 6.12.

-	CRS	BM25
ρ	0.3	-0.1
z-score	-0.6	0.2
P(Z < z)	0.726	0.420
P(Z > z)	0.274	0.579

Table 6.12: Spearman's coefficient ρ, and Z score

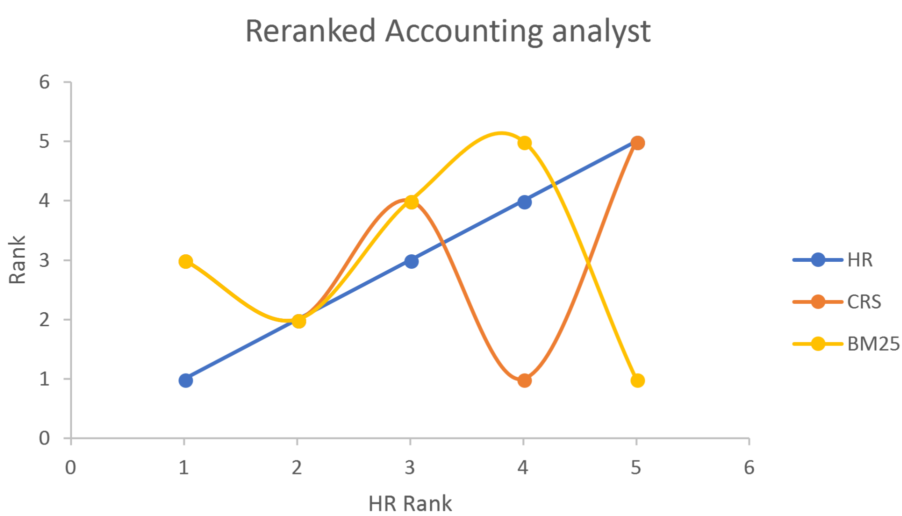


Figure 6.7: Graph displaying the HR top 5 list reordered by Okapi BM25 and CRS

Precision

The recall for the CRS on the HR list, found in table 6.13 has a cumulative probability of P(X>=3) = 0.02194. BM25 found in table 6.14 has a P(X>=3) = 0.02194. Both of these results are statistically significant in that the probability of them happening by chance is below $\alpha = 0.05$.

HR	CRS	Relevant	Recall	Precision
37538	7926	1	0.2	1
9237	9237	2	0.4	1
46171	26343	2	0.4	0.67
7926	37538	3	0.6	0.75
17124	54675	3	0.6	0.6

Table 6.13: Precision and recall calculated for CRS, with the HR list as relevant

HR	BM25	Relevant	Recall	Precision
37538	17124	1	0.2	1
9237	9237	2	0.4	1
46171	37358	2	0.4	0.67
7926	46171	2	0.4	0.5
17124	7926	3	0.6	0.6

Table 6.14: Precision and recall calculated for BM25, with the HR list as relevant

CRS	P(X>=0)
1	0.2

Table 6.15: Times favoured and the probability

Relevance ranks

Participants	Gustaf
HR	2
HR-CRS	2
CRS	2
OKAPI	2
RANDOM	0

Table 6.16: A table of the relevance ranks given assigned by evaluators for the position Accounting analyst

6.3.4 Results from Job Position Industrial Placement

This section presents the result for the position Industrial Placement. The position is retrieved from a company called Alphasights, a worlwide service firm The criterion for the list can be seen in appendix B.4.

6.3.4.1 List Preprocessing

Table 6.17 below shows the mapping between the list number and the participant that created the list in the evaluation pack:

List Number	Participants
List 1	Recruiter HigherEd
List 2	CRS Re-Ranked Recruiter
List 3	Okapi BM25
List 4	CRS
List 5	Random

Table 6.17: Showing the mapping between list number and participant for position Industrial Placement

Rank	Recruiter	CRS-Reranked HR	CRS	Okapi-Reranked HR	Okapi	Random
1	35231	19230	57466	20172	44663	47395
2	15809	15809	19320	58214	20172	51637
3	19230	58214	48451	15809	57975	15809
4	20172	20172	15809	35241	58214	35794
5	58214	35241	45215	19230	35794	43620

Table 6.18: Participants ranking for position Industrial placement

6.3.4.2 Evaluators Result

Evaluator1 - Erik recruiter at ICD

Scored: List4 as the best list among the 5.
Comment from Evaluator: ”It would seem that all the lists contained aleast one candidate not currently in a education program. List number 4 had this candidate closer to the bottom of the list. Lists 1 and 2, while containing good candidates, they also give too much attention to a particular candidate that ended his education in 2013. This seems to me not like a likely candidate for a industry placement/internship position. Industry placement is an opportunity for a student to gain professional development during the scope of education.”.
Relevance Rank: Table 6.23

Evaluator2 - Emilie recruiter at Ingraphic

Scored: List3 as the best list among the 5.
Comment from Evaluator: " I weighted education and that they could speak english. Further I saw that everyone had relevant work experience. I therefore felt that wouldn't have a large impact on the diversity. The criteria that they should had good grades was very hard to evaluate, since most of them didn't have this criteria present in their CV. It was a tie between list 3 and list 4, but list 3 won since it seemed as the most relevant with the total magnitude of relevant candidates in correct position. List 4 however was a strong candidate, except for 1 of the candidates having a background in real estate. Therefore, list 3 won this for me. However, I found this type of position with the given criterias and the dataset hard to evaluate."
Relevance Rank: Table 6.23

Evaluator3 - Arve recruiter at ICD

Scored: List2 as the best list among the 5.
Comment from Evaluator: " The three first candidates of list 4 were more relevant than the three first of both list 1 and 2. However, concidering all 5 candidates then list 1 and 2 seems to be most relevant." **Relevance Rank:** Table 6.23

Evaluator4 - Sofie recruiter at ICD

Scored: List2 as the best list among the 5.
Comment from Evaluator: " From my understanding, the industrial placement is the same as placement at a firm. List 1 and 2 had many students that fits well here, but also a large amount of irrelevant that comes in second place. List 3 had total of 3 relevant, whereas two of these scored in top of the list. That being said, it was a large amount of candidates that had too much experience among the candidates." **Relevance Rank:** Table 6.23

6.3.4.3 Correlation and precision

Spearman's correlation coefficient

The spearman coefficients and the corresponding z, P(z > Z), P(z < Z) values for a normal distribution can be found in table 6.19.

-	CRS	BM25
ρ	-0.2	-0.6
z-score	0.4	1.2
P(Z < z)	0.344	0.115
P(Z > z)	0.655	0.885

Table 6.19: Spearman's coefficient ρ, and Z score for position Industrial Placement

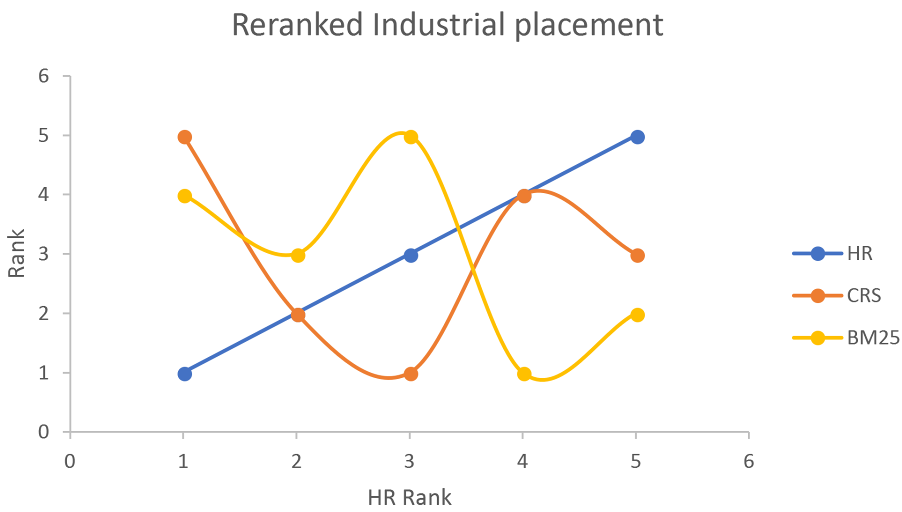


Figure 6.8: Graph displaying the HR top 5 list reordered by Okapi BM25 and CRS

Precision

The recall for the CRS on the HR list, found in table 6.20 has a cumulative probability of P(X>=2) = 0.18333. BM25 found in table 6.21 has a P(X>=2) = 0.18333. Both of these results are not significant.

HR	CRS	Relevant	Recall	Precision
35241	57466	0	0	0
15809	19320	1	0.2	0.5
19230	48451	1	0.2	0.32
20172	15809	2	0.4	0.5
58214	45215	2	0.4	0.4

Table 6.20: Precision and recall calculated for CRS, with the HR list as relevant

HR	BM25	Relevant	Recall	Precision
35241	44663	0	0	0
15809	20172	1	0.2	0.5
19230	57975	1	0.2	0.32
20172	58214	2	0.4	0.5
58214	35794	2	0.4	0.4

Table 6.21: Precision and recall calculated for BM25, with the HR list as relevant

CRS	P(X>=1)
1	1.596%

Table 6.22: Times favoured and the probability

Relevance ranks

Participant	Arve	Emilie	Erik	Sofie	Average
HR	4	5	4	4	4.25
HR-CRS	4	5	4	4	4.25
CRS	4	5	5	2	4
OKAPI	3	5	3	3	3.5
RANDOM	1	2	1	0	1

Table 6.23: A table of the relevance ranks given assigned by evaluators for the position Industrial placement

6.3.5 Results from Job Position Financial Accountant

This section presents the result for the position Financial Accountant. The position is retrieved from a company called Yara, a global firm specializing in agricultural products and environmental protection agents. The criterias for the list can be seen in appendix B.4.

6.3.5.1 List Preprocessing

Table 6.24 below shows the mapping between the list number and the participant that created the list in the evaluation pack:

List Number	**Participants**
List 1	Okapi BM25
List 2	CRS
List 3	CRS-Reranked Recruiter(HR)
List 4	Recruiter Experis
List 5	Random

Table 6.24: Showing the mapping between list number and participant for position Financial Accountant

6.3.5.2 Evaluators Result

Evaluator1 - Gustaf recruiter at Highered

Scored: List4 as the best list among the 5.
Comment from Evaluator: ”This list was most relevant because I felt that all candidate was to some degree relevant.” **Relevance Rank:** Table 6.29

Rank	Recruiter	CRS-Reranked HR	CRS	Okapi-Reranked HR	Okapi	Random
1	47894	9237	9237	59083	34391	3312
2	9237	22599	22599	9237	59083	58254
3	59083	47894	35634	47894	9237	57981
4	22599	59083	47894	22599	38340	56462
5	48451	48451	59083	48451	26340	37438

Figure 6.9: Participants ranking for position Financial Accountant

Evaluator2 -Per recruiter at Panamera-Search

Scored: List4 as the best list among the 5.
Comment from Evaluator: "The top two on this list was the best for me."
Relevance Rank: Table 6.29

Evaluator3 - Emilie recruiter at Ingraphic

Scored: List4 as the best list among the 5.
Comment from Evaluator: " Based on the list I felt that the criterions I did not feel that the candidates matched 100%. However, despite that many of them had correct education, there was many that didn't posses the 3 years of relevant work experience criteria. That being said, three of the lists, list2,3 and 4 had the relevant candidates, but I chose list 4 as the favourite because I felt that its top 2 had the strongest candidates for the position."
Relevance Rank: Table 6.29

6.3.5.3 Correlation and precision

Spearman's correlation coefficient

The spearman coefficients and the corresponding z, P(z ¿ Z), P(z ¿ Z) values for a normal distribution can be found in table 6.25.

-	CRS	BM25
ρ	0.5	0.6
z-score	-1	-1.2
P(Z $<$ z)	0.841	0.115
P(Z $>$ z)	0.159	0.885

Table 6.25: Spearman's coefficient ρ, and Z score

Precision

The recall for the CRS on the HR list, found in table 6.26 has a cumulative probability of P(X>=4) = 0.00088. BM25 found in table 6.27 has a P(X>=2) = 0.18333. The result from CRS is statistically significant with P(X>=4) = 0.00088 beeing less than $\alpha = 0.05$.

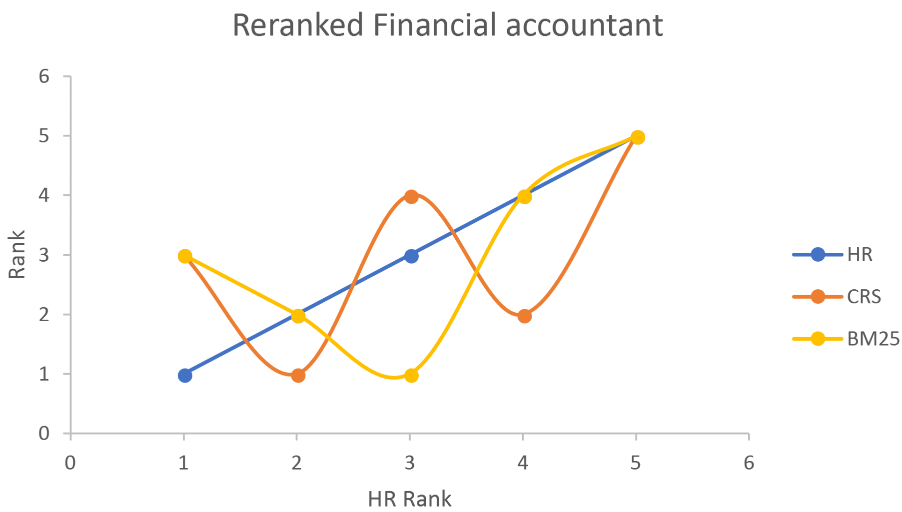


Figure 6.10: Graph displaying the HR top 5 list reordered by Okapi BM25 and CRS

HR	CRS	Relevant	Recall	Precision
47894	9237	1	0.2	1
9237	22599	2	0.4	1
59083	35634	2	0.4	0.67
22599	47894	3	0.6	0.75
48451	59083	4	0.8	0.8

Table 6.26: Precision and recall calculated for CRS, with the HR list as relevant

HR	BM25	Relevant	Recall	Precision
47894	34391	0	0	0
9237	59083	1	0.2	0.5
59083	9237	2	0.4	0.67
22599	38340	2	0.4	0.5
48451	26340	2	0.4	0.4

Table 6.27: Precision and recall calculated for BM25, with the HR list as relevant

CRS	P(X>=0)
0	

Table 6.28: Times favoured and the probability

Participants	Gustav	Per	Emilie	Average
HR	4	4	5	4.333333333
HR-CRS	3	4	5	4
CRS	4	3	5	4
OKAPI	2	3	3	2.666666667
RANDOM	1	1	0	0.666666667

Table 6.29: A table of the relevance ranks given assigned by evaluators for the position Financial Accounting

Relevance ranks

6.3.6 Positions combined

6.3.6.1 Hyper-geometric probability distribution

HR	HR-CRS	CRS	OKAPI	RANDOM
4	2	7	1	0

Table 6.30: Sum of times favoured over all four positions

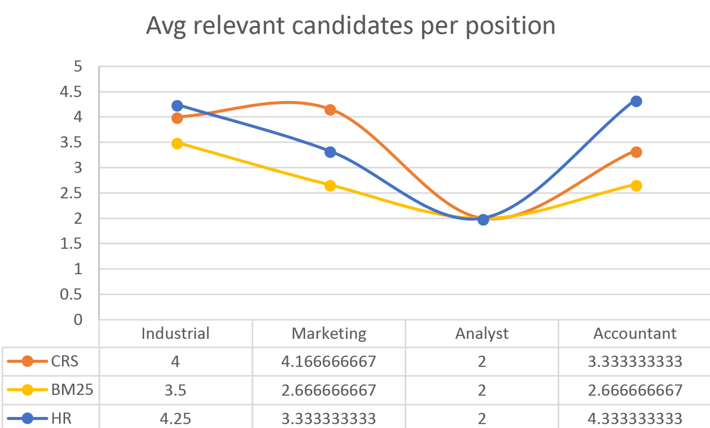


Figure 6.12: Average relevant candidates reported by evaluations per position

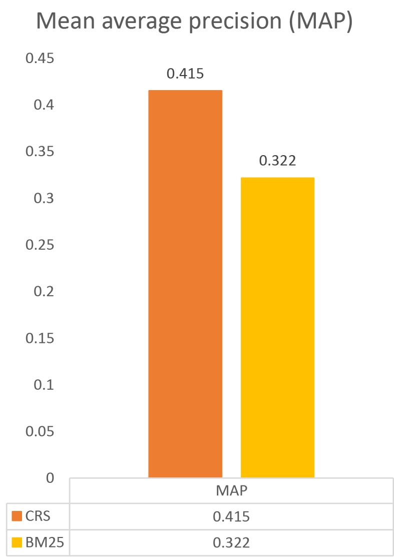


Figure 6.11: Caption

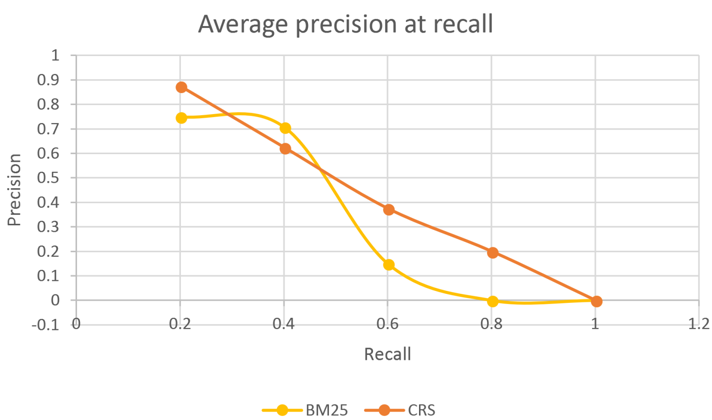


Figure 6.13: Caption

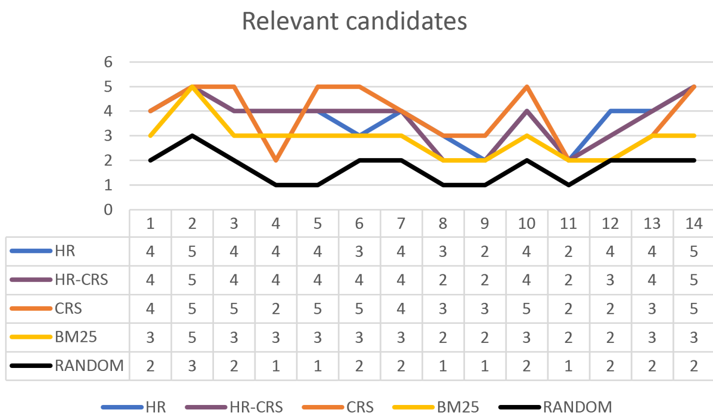


Figure 6.14: Amount of relevant candidates per list for each evaluation

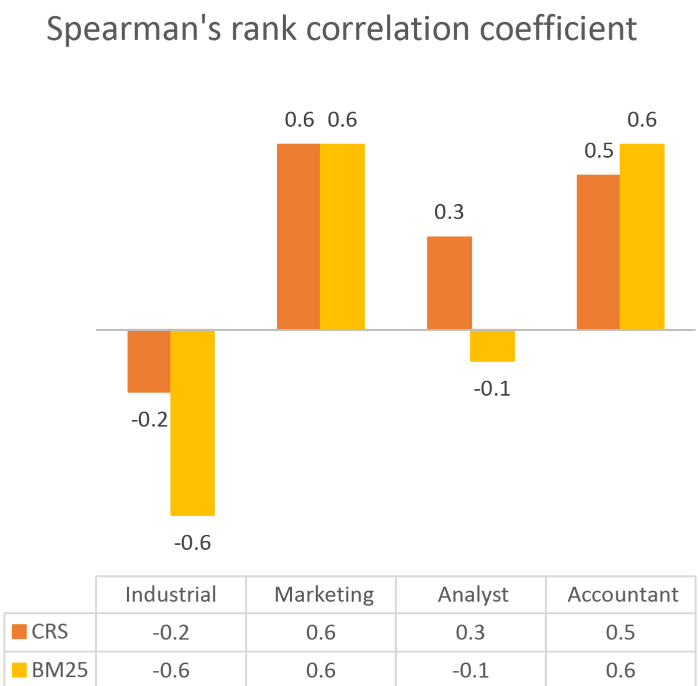


Figure 6.15: Caption

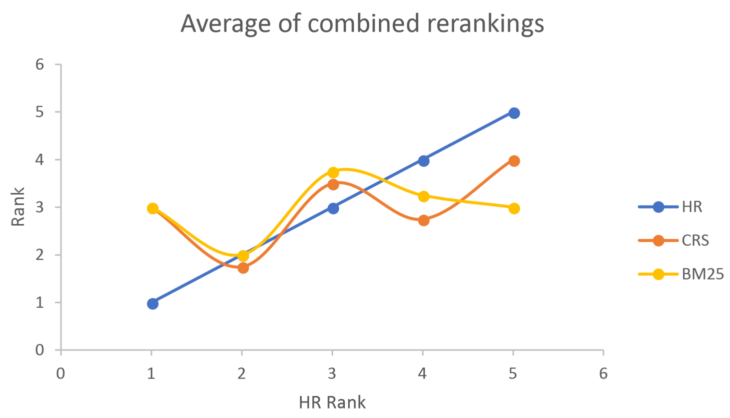


Figure 6.16: a figure

6.4 Real World Scenario

In this section we present the results obtained from the real world scenario, discussed in 5.4. Here we test the correlation between Okapi BM25 and a recruiter, and the candidate ranking system and the recruiter.

6.4.1 Spearman Coefficient results

In this section we will present our

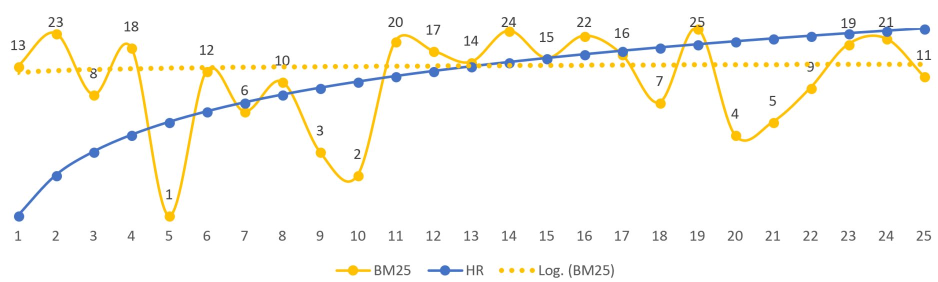


Figure 6.17: Showing The Correlation between Okapi BM25 and the recruiter for position Business Intelligence

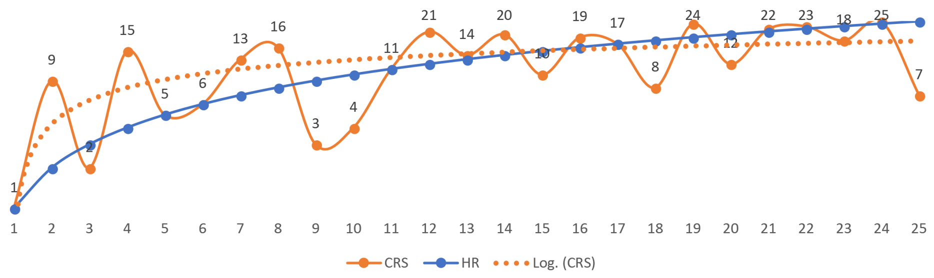


Figure 6.18: Showing The Correlation between CRS and the recruiter for position Business Intelligence

6.4.1.1 Spearman Correlation between Okapi BM25 and Recruiter

$$\sum d^2 = 2266, n = 25$$
$$r_s = 1 - \frac{6\sum d^2}{n(n^2-1)} => 1 - \frac{6*2266}{25(625-1)} = 1 - \frac{13596}{15600} = 0.1284$$

The Spearman Coefficent correlation between Okapi BM25 and Recruiter for 25 rankings based on the position ”Business Intelligence” is 0.132.

6.4.1.2 Spearman Correlation between CRS and Recruiter

$$\sum d^2 = 900, n = 25$$
$$r_s = 1 - \frac{6\sum d^2}{n(n^2-1)} => 1 - \frac{6*900}{25(625-1)} = 1 - \frac{5400}{15600} = 0.65$$

The Spearman Coefficent correlation between Okapi BM25 and Recruiter for 25 rankings based on the position ”Business Intelligence” is 0.65.

6.4.2 Precision, Recall and E-Measure

In this section we will first calculate the precision and recall for Okapi BM25 and CRS respectively. Table 6.33 presents the recall and precision results for Okapi BM25.Table 6.34 presents recall and precision results for the CRS.

Thereafter, we present precision at full recall. The first test presents results at the recruiter’s top 5 list. Thereafter, we test precision at full recall given that the top-10 are the relevant. In the end we will measure the E-Measure metric which favors the precision for each of the tests.

6.4.2.1 Precision and Recall Okapi BM25

This section presents the precision and recall obtained with Okapi BM25 for the top 10 out of the 25 from the real world recruiting scenario, and precision at full recall at 5. Full Recall was obtained at 22 selections.

Precision@5: From table 6.33, The Precision at 5 is $\frac{5}{5} = 1$
Precision@10: From table 6.33, The Precision at 10 is $\frac{6}{10} = 0.6$
Precision@15: From table 6.33, The Precision at 15 is $\frac{9}{15} = 0.6$

Candidate ID	Recruiter Rank	Okapi BM25	Distance	Distance Squared
32	10	2	8	64
33	9	3	6	36
56	8	10	2	4
58	7	6	1	1
105	25	11	14	196
107	24	21	3	9
108	23	19	4	16
110	22	9	13	169
113	21	5	16	256
117	20	4	16	256
119	6	12	-6	36
121	19	25	-6	36
126	18	7	11	121
127	5	1	4	16
128	17	16	1	1
130	16	22	-6	36
132	15	15	0	0
137	4	18	-14	196
138	11	20	-9	81
148	3	8	-5	25
149	12	17	-5	25
151	13	14	1	1
152	14	24	-10	100
163	2	23	-21	441
175	1	13	-12	144

Table 6.31: Ranking Correlation between Okapi and Recruiter

Candidate ID	Recruiter Rank	CRS Rank	Distance	Distance Squared
32	10	4	6	36
33	9	3	6	36
56	8	16	-8	64
58	7	13	-6	36
105	25	7	18	324
107	24	25	-1	1
108	23	18	5	25
110	22	23	1	1
113	21	22	1	1
117	20	12	8	64
119	6	6	0	0
121	19	24	-5	25
126	18	8	10	4
127	5	5	0	0
128	17	17	0	0
130	16	19	-3	9
132	15	10	5	25
137	4	15	-9	81
138	11	11	0	0
148	3	2	1	1
149	12	21	-9	81
151	13	14	-1	1
152	14	20	-6	36
163	2	9	7	49
175	1	1	0	0

Table 6.32: Ranking Correlation between CRS and Recruiter

CandidateID	Rank	Relevant	Recall	Precision
32	1	REL	1/10 = 0.10	1/1 = 1
33	2	REL	2/10 = 0.20	2/2 = 1
127	3	REL	3/10 = 0.30	3/3 = 1
56	4	REL	4/10 = 0.40	4/4 = 1
126	5		4/10 = 0.40	4/5 = 0.8
148	6	REL	5/10 = 0.50	5/6 = 0.833
110	7		5/10 = 0.50	5/7 = 0.714
105	8		5/10 = 0.50	5/8 = 0.625
58	9	REL	6/10 = 0.60	6/9 = 0.666
149	10		6/10 = 0.60	6/10 = 0.60
175	11	REL	7/10 = 0.70	7/11 = 0.636
132	12		7/10 = 0.70	7/12 = 0.583
137	13	REL	8/10 = 0.80	8/13 = 0.615
119	14	REL	9/10 = 0.90	9/14 = 0.642
151	15		9/10 = 0.90	9/15 = 0.6
138	16		9/10 = 0.90	9/16 = 0.5625
128	17		9/10 = 0.90	9/17 = 0.53
117	18		9/10 = 0.90	9/18 = 0.5
130	19		9/10 = 0.90	9/19 = 0.474
108	20		9/10 = 0.90	9/20 = 0.45
107	21		9/10 = 0.90	9/21 = 0.43
163	22	REL	10/10 = 1	9/16 = 0.45

Table 6.33: Presenting precision and recall levels till full recall for Okapi BM25

Figure 6.19 presents the precision graph for the CRS and figure 6.20 presents the recall graph for the CRS.

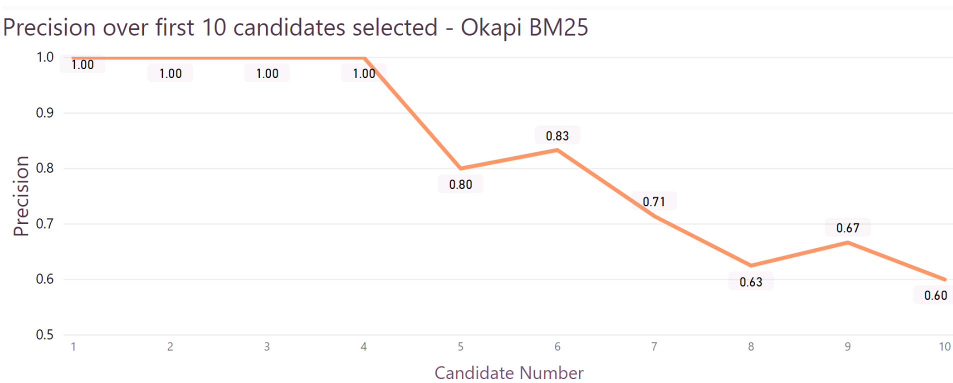


Figure 6.19: Precision for Okapi BM25 at ten selections

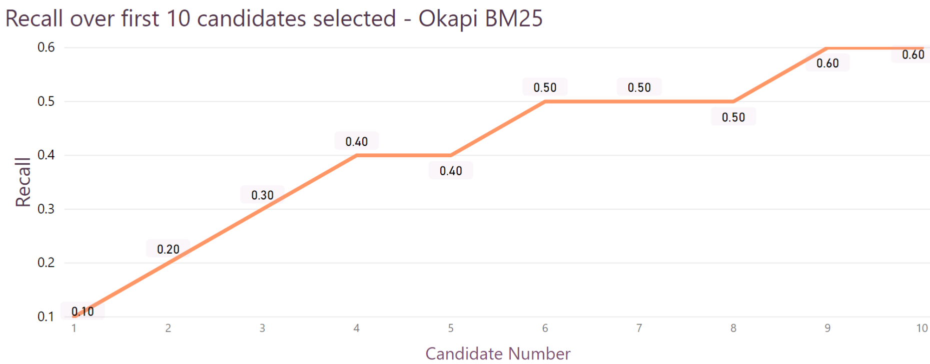


Figure 6.20: Recall for Okapi BM25 at ten selections

6.4.2.2 Precision and Recall CRS

This section presents the precision and recall obtained with the Candidate Ranking System for the top 10 out of the 25 from the real world recruiting scenario.

CandidateID	Rank	Relevant	Recall	Precision
175	1	REL	1/10 = 0.10	1/1 = 1
148	2	REL	2/10 = 0.20	2/2 = 1
33	3	REL	3/10 = 0.30	3/3 = 1
32	4	REL	4/10 = 0.40	4/4 = 1
127	5	REL	5/10 = 0.50	5/5 = 1
119	6	REL	6/10 = 0.60	6/6 = 1
105	7		6/10 = 0.60	6/7 = 0.86
126	8		6/10 = 0.60	6/8 = 0.75
163	9	REL	7/10 = 0.70	7/9 = 0.77
132	10		7/10 = 0.70	7/10 = 0.70
138	11		7/10 = 0.70	7/11 = 0.636
117	12		7/10 = 0.70	7/12 = 0.583
58	13	REL	8/10 = 0.80	8/13 = 0.615
151	14		8/10 = 0.80	8/14 = 0.571
137	15	REL	9/10 = 0.90	9/15 = 0.6
56	16	REL	10/10 = 1	10/16 = 0.625

Table 6.34: Presenting precision and recall levels till full recall for CRS

Figure 6.21 presents the precision graph for the CRS and figure 6.22 presents the recall graph for the CRS.

Precision@5: From table 6.34, The Precision at 5 is $\frac{5}{5} = 1$
Precision@10: From table 6.34, The Precision at 10 is $\frac{7}{10} = 0.7$
Precision@15: From table 6.34, The Precision at 1 is $\frac{9}{15} = 0.6$

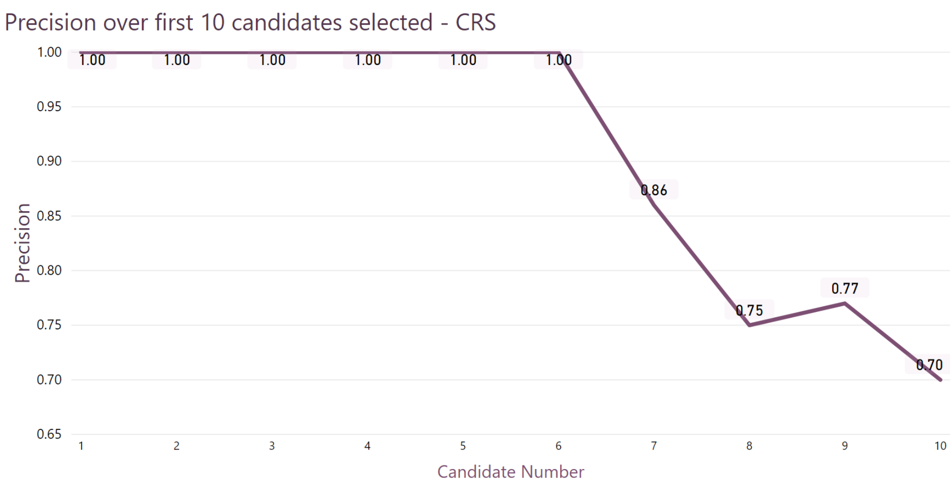


Figure 6.21: Precision for CRS at ten selections

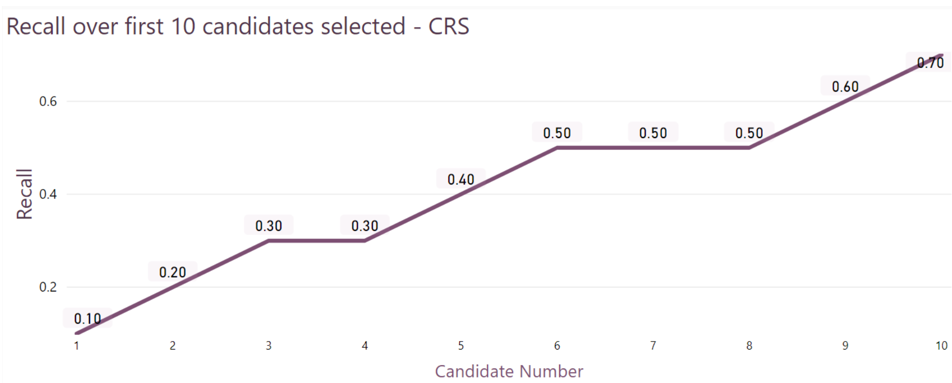


Figure 6.22: Recall for CRS at ten selections

6.4.3 E-Measure for CRS and Okapi BM25

Based on the Precision and Recall calculated from section above, we calculated E-Measure 2.19 based on each precision@K. Since we found it more important with precision rather than Recall, we used the E-Measure introduced in 2.5.2 we can set Beta value greater than 1 for the E-Measure. Formula is presented in 2.19.

$$E = \frac{(\beta^2)PR}{\beta^2 P + R} = \frac{(1+\beta^2)}{\frac{\beta^2}{R} + \frac{1}{P}} \tag{6.1}$$

E-Measure Okapi BM25 - 5 selections
Total Relevant in the corpus is 10. With Recall = 4/10 = 0.4, P = 4/5 = 0.8, Beta = 1.2 the result leads to:

$$E_{OkapiBM25} = \frac{(1+1.2^2)}{\frac{1.2^2}{0.4} + \frac{1}{0.8}} = 0.503 \tag{6.2}$$

E-Measure Okapi BM25 - At 9 selections
With R = 6/10 = 0.6, P = 6/9 = 0.666, Beta = 1.2 the result leads to

$$E_{OkapiBM25} = \frac{(1+1.2^2)}{\frac{1.2^2}{0.6} + \frac{1}{0.666}} = 0.625 \tag{6.3}$$

E-Measure Okapi BM25 - At Full Recall for recruiter top 10
With R = 10/10 = 1, P = 9/16 = 0.45, Beta = 1.2 the result leads to

$$E_{OkapiBM25} = \frac{(1+1.2^2)}{\frac{1.2^2}{1} + \frac{1}{0.45}} = 0.666 \tag{6.4}$$

E-Measure CRS - 5 selections
Total Relevant in the corpus is 10. With Recall = 5/10 = 0.5, P = 5/5 = 1, Beta = 1.2 the result leads to:

$$E_{CRS} = \frac{(1+1.2^2)}{\frac{1.2^2}{0.5} + \frac{1}{1}} = 0.628 \tag{6.5}$$

E-Measure CRS - 9 selections
Total Relevant in the corpus is 10. With Recall = 7/10 = 0.7, P = 7/9 = 0.778, Beta = 1.2 the result leads to:

$$E_{CRS} = \frac{(1+1.2^2)}{\frac{1.2^2}{0.7} + \frac{1}{0.778}} = 0.73 \tag{6.6}$$

E-Measure CRS - At Full Recall for recruiter top 10
Total Relevant in the corpus is 10. With Recall = 1 = 0.7, P = 10/16 = 0.778, Beta = 1.2 the result leads to:

$$E_{CRS} = \frac{(1+1.2^2)}{\frac{1.2^2}{0.7} + \frac{1}{0.778}} = 0.895 \tag{6.7}$$

Top-5 is the relevant

REL	Recruiter Rank	Okapi BM25	IDCG	DCG	CIDCG	CDCG	NDCG
10.00	1.00	13.00	10.00	1.00	10.00	1.00	10.00
9.00	2.00	23.00	9.00	1.00	19.00	2.00	10.53
8.00	3.00	8.00	5.05	1.89	24.05	3.89	16.19
7.00	4.00	18.00	3.50	0.50	27.55	4.39	15.95
6.00	5.00	1.00	2.58	4.31	30.13	8.70	28.87
5.00	6.00	12.00	1.93	0.39	32.07	9.09	28.34
4.00	7.00	6.00	1.42	1.78	33.49	10.87	32.45
3.00	8.00	10.00	1.00	0.33	34.49	11.20	32.47
2.00	9.00	3.00	0.63	2.52	35.12	13.72	39.08
1.00	10.00	2.00	0.30	2.71	35.42	16.43	46.39
1.00	11.00	20.00	0.29	0.29	35.71	16.72	46.83
1.00	12.00	17.00	0.28	0.28	35.99	17.00	47.24
1.00	13.00	14.00	0.27	0.27	36.26	17.27	47.63
1.00	14.00	24.00	0.26	0.26	36.52	17.53	48.01
1.00	15.00	15.00	0.26	0.26	36.78	17.79	48.37
1.00	16.00	22.00	0.25	0.25	37.03	18.04	48.72
1.00	17.00	16.00	0.24	0.24	37.27	18.29	49.06
1.00	18.00	7.00	0.24	0.96	37.51	19.24	51.30
1.00	19.00	25.00	0.24	0.24	37.75	19.48	51.60
1.00	20.00	4.00	0.23	1.62	37.98	21.10	55.55
1.00	21.00	5.00	0.23	1.37	38.21	22.47	58.80
1.00	22.00	9.00	0.22	0.45	38.43	22.91	59.62
1.00	23.00	19.00	0.22	0.22	38.65	23.14	59.85
1.00	24.00	21.00	0.22	0.22	38.87	23.35	60.08
1.00	25.00	11.00	0.22	0.22	39.09	23.57	60.30

Table 6.35: Table of calculations for DCG with relevance minimum 1, on the list produced by BM25

REL	Recruiter Rank	CRS Rank	IDCG	DCG	CIDCG	CDCG	NDCG
10.00	1.00	1.00	10.00	10.00	10.00	10.00	100.00
9.00	2.00	9.00	9.00	2.00	19.00	12.00	63.16
8.00	3.00	2.00	5.05	5.68	24.05	17.68	73.51
7.00	4.00	15.00	3.50	0.50	27.55	18.18	65.99
6.00	5.00	5.00	2.58	2.58	30.13	20.76	68.91
5.00	6.00	6.00	1.93	1.93	32.07	22.70	70.78
4.00	7.00	13.00	1.42	0.36	33.49	23.05	68.83
3.00	8.00	16.00	1.00	0.33	34.49	23.39	67.80
2.00	9.00	3.00	0.63	2.52	35.12	25.91	73.77
1.00	10.00	4.00	0.30	2.11	35.42	28.02	79.09
1.00	11.00	11.00	0.29	0.29	35.71	28.31	79.26
1.00	12.00	21.00	0.28	0.28	35.99	28.59	79.42
1.00	13.00	14.00	0.27	0.27	36.26	28.86	79.58
1.00	14.00	20.00	0.26	0.26	36.52	29.12	79.72
1.00	15.00	10.00	0.26	0.26	36.78	29.37	79.87
1.00	16.00	19.00	0.25	0.25	37.03	29.62	80.00
1.00	17.00	17.00	0.24	0.24	37.27	29.87	80.13
1.00	18.00	8.00	0.24	0.72	37.51	30.59	81.54
1.00	19.00	24.00	0.24	0.24	37.75	30.82	81.65
1.00	20.00	12.00	0.23	0.23	37.98	31.05	81.77
1.00	21.00	22.00	0.23	0.23	38.21	31.28	81.87
1.00	22.00	23.00	0.22	0.22	38.43	31.51	81.98
1.00	23.00	18.00	0.22	0.22	38.65	31.73	82.08
1.00	24.00	25.00	0.22	0.22	38.87	31.95	82.18
1.00	25.00	7.00	0.22	0.86	39.09	32.81	83.93

Table 6.36: Table of calculations for DCG with relevance minimum 1, on the list produced by CRS

6.4.3.1 Statistical Significance

6.4.3.2 Comparison with Discounted Cumulated Gain

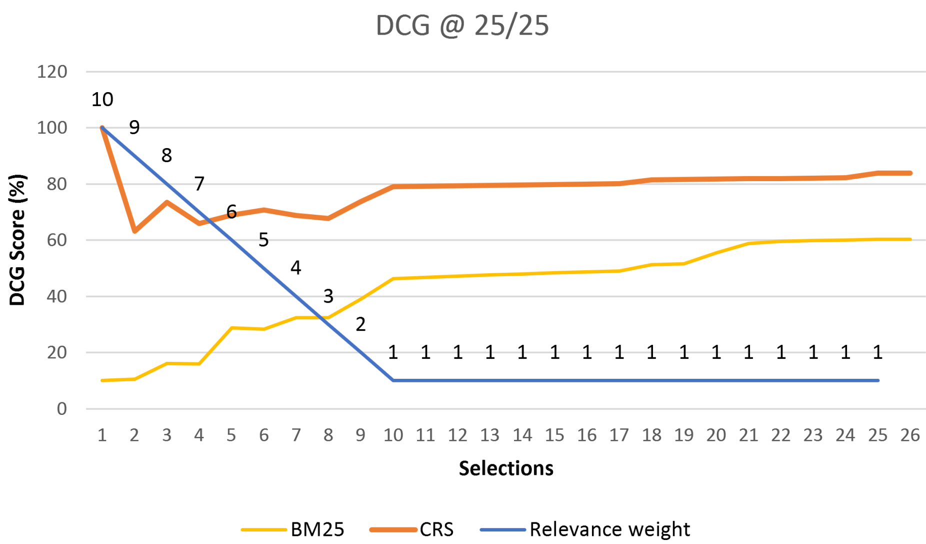


Figure 6.23: Plot of the NDCG values of table 6.35 and table 6.36. The DCG calculated for each step down the list.

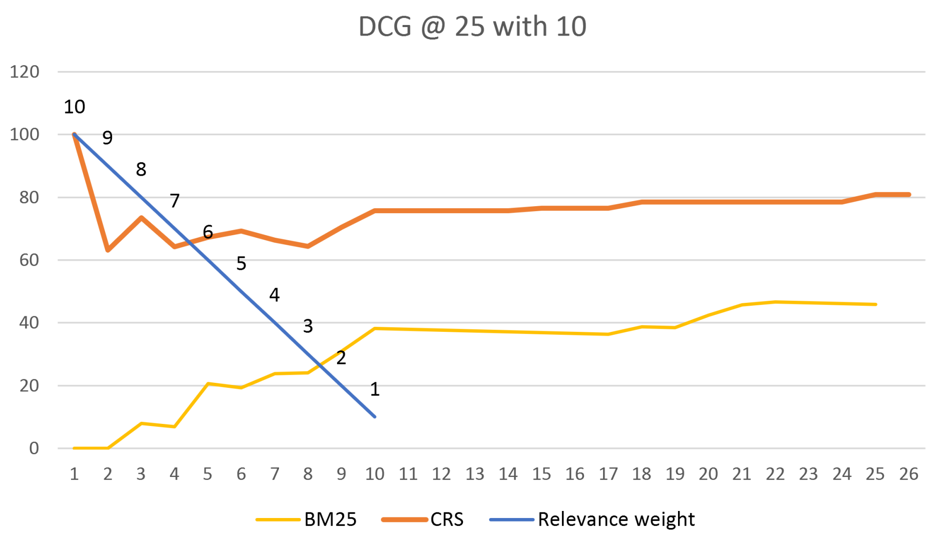


Figure 6.24: Plot of the NDCG values of table 6.37 and table 6.38. The DCG calculated for each step down the list.

REL	Recruiter Rank	Okapi BM25	IDCG	DCG	CIDCG	CDCG	NDCG
10.00	1.00	13.00	10.00	0.00	10.00	0.00	0.00
9.00	2.00	23.00	9.00	0.00	19.00	0.00	0.00
8.00	3.00	8.00	5.05	1.89	24.05	1.89	7.87
7.00	4.00	18.00	3.50	0.00	27.55	1.89	6.87
6.00	5.00	1.00	2.58	4.31	30.13	6.20	20.57
5.00	6.00	12.00	1.93	0.00	32.07	6.20	19.33
4.00	7.00	6.00	1.42	1.78	33.49	7.98	23.83
3.00	8.00	10.00	1.00	0.33	34.49	8.31	24.10
2.00	9.00	3.00	0.63	2.52	35.12	10.84	30.86
1.00	10.00	2.00	0.30	2.71	35.42	13.55	38.24
	11.00	20.00	0.00	0.00	35.42	13.55	38.24
	12.00	17.00	0.00	0.00	35.42	13.55	38.24
	13.00	14.00	0.00	0.00	35.42	13.55	38.24
	14.00	24.00	0.00	0.00	35.42	13.55	38.24
	15.00	15.00	0.00	0.00	35.42	13.55	38.24
	16.00	22.00	0.00	0.00	35.42	13.55	38.24
	17.00	16.00	0.00	0.00	35.42	13.55	38.24
	18.00	7.00	0.00	0.96	35.42	14.51	40.95
	19.00	25.00	0.00	0.00	35.42	14.51	40.95
	20.00	4.00	0.00	1.62	35.42	16.13	45.52
	21.00	5.00	0.00	1.37	35.42	17.49	49.38
	22.00	9.00	0.00	0.45	35.42	17.94	50.65
	23.00	19.00	0.00	0.00	35.42	17.94	50.65
	24.00	21.00	0.00	0.00	35.42	17.94	50.65
	25.00	11.00	0.00	0.00	35.42	17.94	50.65

Table 6.37: Table of calculations for DCG with relevance minimum 0, on the list produced by BM25

REL	Recruiter Rank	CRS Rank	IDCG	DCG	CIDCG	CDCG	NDCG
10.00	1.00	1.00	10.00	10.00	10.00	10.00	100.00
9.00	2.00	9.00	9.00	2.00	19.00	12.00	63.16
8.00	3.00	2.00	5.05	5.68	24.05	17.68	73.51
7.00	4.00	15.00	3.50	0.00	27.55	17.68	64.17
6.00	5.00	5.00	2.58	2.58	30.13	20.26	67.25
5.00	6.00	6.00	1.93	1.93	32.07	22.20	69.22
4.00	7.00	13.00	1.42	0.00	33.49	22.20	66.28
3.00	8.00	16.00	1.00	0.00	34.49	22.20	64.36
2.00	9.00	3.00	0.63	2.52	35.12	24.72	70.39
1.00	10.00	4.00	0.30	2.11	35.42	26.83	75.74
	11.00	11.00	0.00	0.00	35.42	26.83	75.74
	12.00	21.00	0.00	0.00	35.42	26.83	75.74
	13.00	14.00	0.00	0.00	35.42	26.83	75.74
	14.00	20.00	0.00	0.00	35.42	26.83	75.74
	15.00	10.00	0.00	0.26	35.42	27.08	76.46
	16.00	19.00	0.00	0.00	35.42	27.08	76.46
	17.00	17.00	0.00	0.00	35.42	27.08	76.46
	18.00	8.00	0.00	0.72	35.42	27.80	78.49
	19.00	24.00	0.00	0.00	35.42	27.80	78.49
	20.00	12.00	0.00	0.00	35.42	27.80	78.49
	21.00	22.00	0.00	0.00	35.42	27.80	78.49
	22.00	23.00	0.00	0.00	35.42	27.80	78.49
	23.00	18.00	0.00	0.00	35.42	27.80	78.49
	24.00	25.00	0.00	0.00	35.42	27.80	78.49
	25.00	7.00	0.00	0.86	35.42	28.66	80.92

Table 6.38: Table of calculations for DCG with relevance minimum 0, on the list produced by CRS

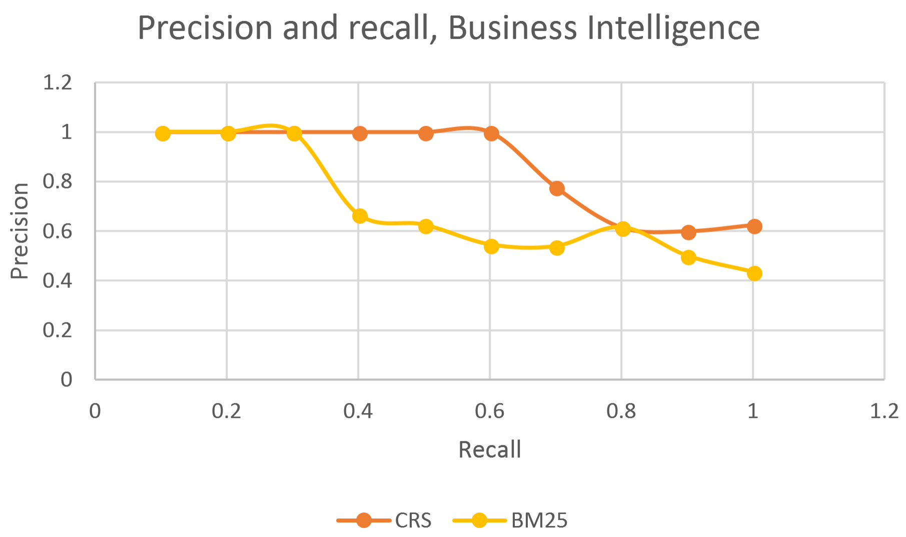


Figure 6.25: Caption

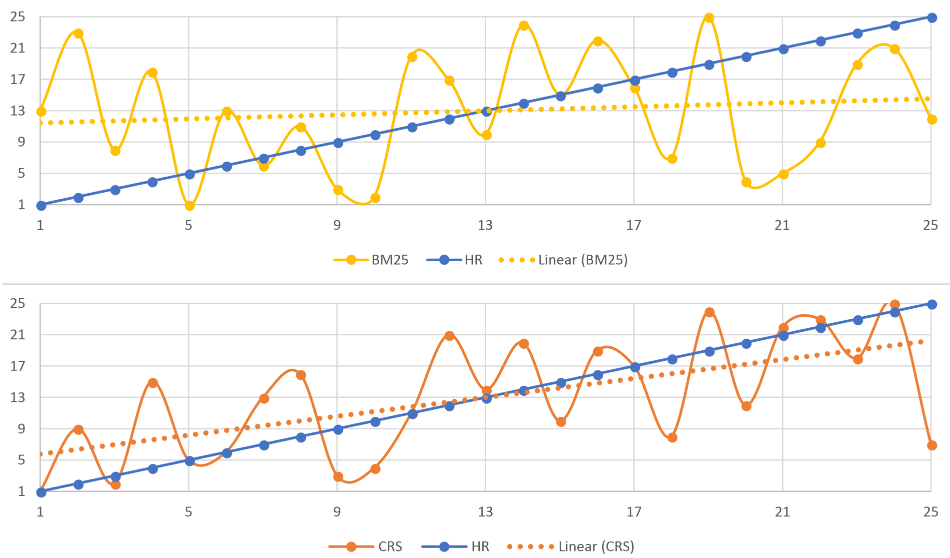


Figure 6.26: Showing the precision and recall tradeoff between CRS and BM25 for the position Business Intelligence based on ten hits

Chapter 7

Discussion

This chapter is dedicated to discuss the research questions in relation to the results presented in chapter 6. The first three section encompasses the research questions, the last two is dedicated to discuss the limitations and some of the preparation.

7.1 Experiment

Chapter 6 presents the result based on fourteen evaluations packages that encompassed the four job positions. In this section we analyze and discuss the results in relation to the research questions below.

RQ:1 How does the lists produced by the Candidate Ranking System compare to those made by Okapi BM25 and human recruiters?
RQ:1.2 when evaluated by other experts, how does the CRS top-5 list compare to The top-5 list made/generated by human recruiters and Okapi BM25 in four job position scenarios with respect to ranking, precision and recall?
RQ:1.3 When wish to juxtapose Okapi and CRS to the list produced by a list considered to be the most precise, namely those that are made by human experts

7.1.1 Analyzing the Results

In this section we analyze the results obtained from the fourteen evaluations. We start by looking at the experiment as a whole before we drill down into the evaluations for each position.

7.1.1.1 The Overall favourite and Most relevant

The combined results from all the evaluations can be seen in figure 7.1. As we expected, the random list did not get favoured in any evaluation, we also expected the HR list and the HR-CRS list to be relatively close. The combined results though not statistically significant, seem to indicate that human recruiters would not necessarily prefer lists created by other recruiters when stacked up against the CRS and okapi.

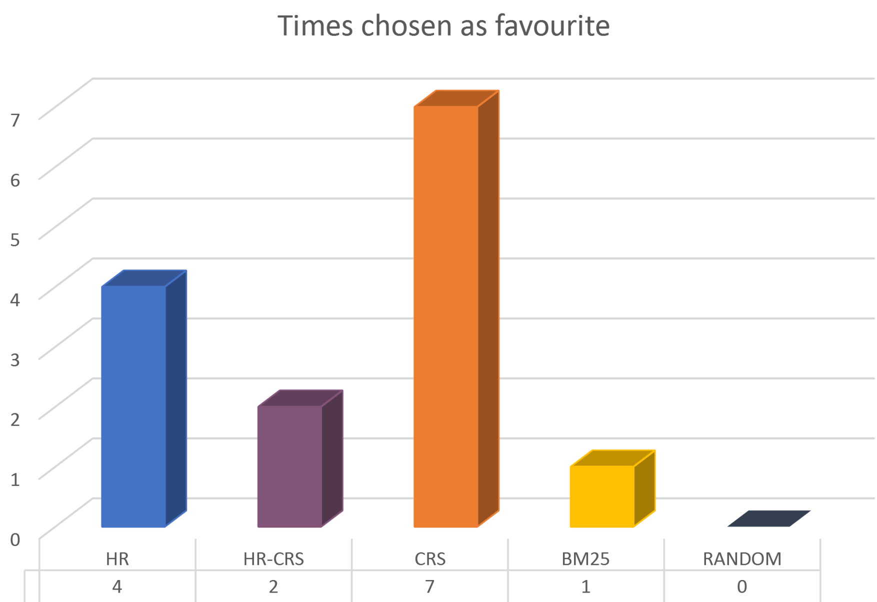


Figure 7.1: Showing the standings after fourteen evaluations

We see that our CRS won 7 out of 14 evaluations. In second place, we had the HR. It seemed as the CRS provided better top-5 lists than Okapi here. However, most of the CRS wins came from the job position Marketing and Sales Consultant. This was the position containing most evaluations. Evaluations for each position will be analyzed and discussed further in the sections below.

By looking on the evaluators favourites may say something about which list was better than the other for certain positions given the specific data-set and criterion. That being said, is also important to look at the amount of relevant candidates that each of the evaluators felt the lists contained. Figure 7.2 presents the floating average of relevant candidates over the 14 evaluations. The CRS line colored in orange is above Okapi for all the evaluations. The CRS also ties and even beats the recruiters in several cases. An interesting observation of the raw data, is that the evaluators dont't agree on the amount of relevant candidates in certain lists for given positions. This shows that there is not necessarily a single answer for every solution. In these evaluations we see a clear difference between Okapi and the CRS. This might indicate that the CRS outperforms Okapi.

A trend of more relevant candidates in the lists created by the CRS and HR can be seen in the cumulative relevance, figure 7.3. The total relevant candidates ended up with a total of 53 for the CRS, slightly in front of HR with 52. BM25 came out a bit lower with a total of 40 relevant.

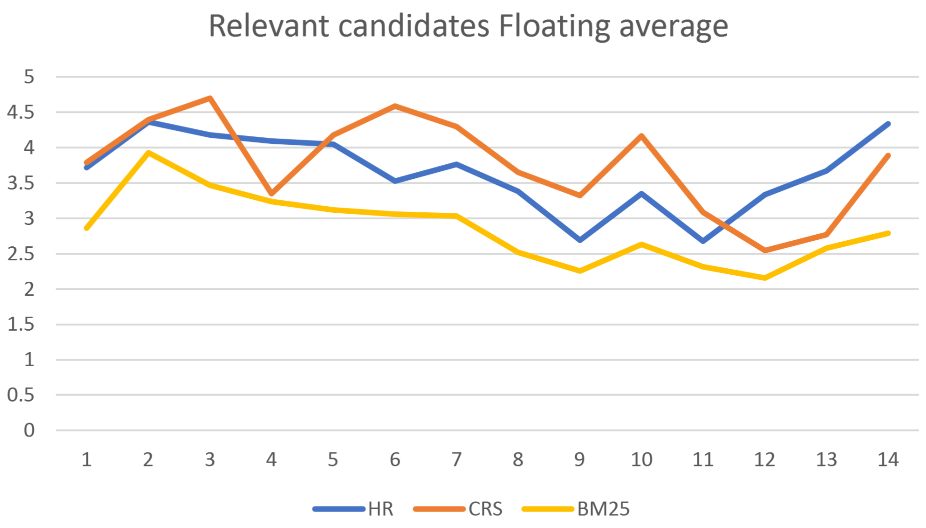
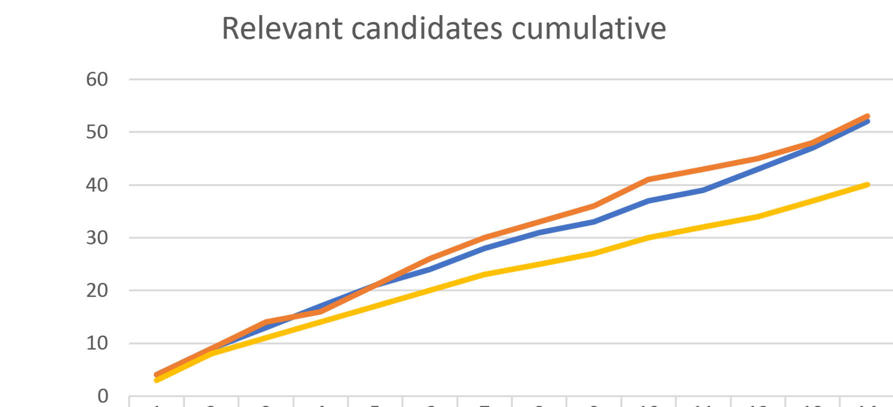


Figure 7.2: Showing the floating average of relevant candidates based on the fourteen evaluations



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
HR	4	9	13	17	21	24	28	31	33	37	39	43	47	52
CRS	4	9	14	16	21	26	30	33	36	41	43	45	48	53
BM25	3	8	11	14	17	20	23	25	27	30	32	34	37	40

Figure 7.3: Cumulative Relevance Representation

	Industrial placement				Marketing and sales			
	CRS	OKAPI	HR	HR-CRS	CRS	OKAPI	HR	HR-CRS
Recall	0.4	0.4	1	1	0.2	0.4	1	1
Favoured	0.25	0.25	0	0.5	0.834	0	0.167	0
Spearman	-0.2	-0.6	1	-0.2	0.6	0.6	1	0.6
Relevant	4	3.5	4.25	4.25	4.17	2.67	3.34	3.34
	Accounting analyst				Financial accounting			
	CRS	OKAPI	HR	HR-CRS	CRS	OKAPI	HR	HR-CRS
Recall	0.6	0.6	1	1	0.8	0.4	1	1
Favoured	1	0	0	0	0	0	1	0
Spearman	0.3	-0.1	1	0.3	0.5	0.6	1	0.5
Relevant	2	2	2	2	3.34	2.67	4.32	4

Table 7.1: Combined data for each position.

A concern raised in Kmail et al. (2015); Kessler et al. (2012) is that lists created by human recruiters are not necessarily the best. Gil et al. (2016) found in their study that the human recruiter had been precise for the top of the list and that the quality of the ordering dropped further down. From our results we found that, on average, the evaluators only preferred the HR list for the job Financial accountant.

7.1.1.2 Analyzing the Results from Industrial Placement Evaluation

For the Industrial Placement Evaluation, we received a total of 4 evaluations. In this position both the CRS and Okapi won once, and the CRS re-rank of the recruiter list won twice. This position was particularly interesting as the CRS would not take into considerations that students or newly educated should be preferred over candidates that possessed more work experience.

Looking isolated on the favourite does not necessarily say much. From table 6.23 we see that on an average basis, the CRS scored right below the recruiter with respect to the amount of relevant candidates. An interesting observation showed that the re-order of the HR list won over the HR itself in two out of four cases. From analyzing the comments in section 6.3.4, we observe that this may be related to the recruiter-list favours experienced candidates. Based on the criterion's from the B.4, there was not mentioned work experience in particular. This means that the CRS may have adapted its previous case by removing work experience when retaining the new case for Industrial Placement.

By analyzing the comments, there was an agreement formed that the list produced by the CRS was a strong candidate, that may indicate that it managed to find good candidates, despite loosing in some cases.

When focusing isolated on the correlation with the human recruiter, the figure 6.19 presents the z score that is based the probability to have overlap with HR based on the hypergeometric selection from the 30 candidate set given that five was relevant and 25 unrelevant. Here we can see that there was a 34.4% chance that a random selected list would produce these results and 11,5% for it to produce Okapi BM25's result.

From table 6.20 and 6.21 we have 2 in recall for both the systems.

In figure 6.8 we have the results of having okapi and the CRS rearrange the HR list. From these results, we see that the CRS and HR agreed on the second and fourth position of the ranking. The CRS however completely disagreed on the first and fifth place. Okapi has the recruiters fourth place as the best opposed to the CRS' fifth place.

7.1.1.3 Analyzing the Results from Marketing and Sales Consultant Evaluation

This was the position that received the largest amount of "6" evaluations. This was also the position based on the four positions that the CRS performed overall better with respect to the evaluators favored lists. Based on the comments the list produced by the CRS had the largest amount of marketing oriented candidates. From the result tables in section 6.3.2 we calculated an average of 4.16 relevant candidates based on the six evaluations. The reason why the CRS performed this good for this position may be numerous. First the defined criterion in the position, see appendix B.4 might have been well defined and easy to digeste for the Job Query Module. Skills from this position also seemed more straight forward. The position preferred at least 5 years of experience. The CRS would not be able to find one with 5 years experience, but the trained case contained work experience with sufficient weights. In addition it comes very clear from the text what type of educational background wanted.

Despite that the CRS won in most of the evaluation cases, meaning it probably contained a better list than HR, we still found it important to measure the precision by comparing the list produced by the CRS with the Okapi. From table 6.7 we see that the CRS had a precision of 20%, meaning that it shared one of the candidates with HR. In this case the Okapi BM25 obtained a precision at 40%, sharing two candidates with HR, seen in 6.8.

We also investigated how the CRS would compare to the Okapi BM25 when re-ranking the human expert lists. From figure 6.6 we see that the CRS agreed with the HR in the top 2 positions, and overall agreed with 3 out of 5 positions. The BM25 however, did not agree on any of the positions but was close with the top two positions. For this positions with the given criterion, we therefore see that the CRS agreed more with the human expert when re-ranking the list for this position.

7.1.1.4 Analyzing the Accounting Analyst Position

From this position we received only one evaluation. Despite that the CRS won, found all of the lists in this position less relevant.

7.1.1.5 Analyzing the Financial Accountant Position

The recruiter reported that this list did not have any highly relevant candidate, but that he had provided a list of the least irrelevant candidates. He placed gave the top a score of 4/5 and second place as 3/5. Rest of the candidate seemed to have a lesser degree of relevance.

This raised a question regarding the quality of the 30 candidates in the data-set. The recruiter's relevance rank is presented in table 6.4. This was in particular interesting as despite being the only produced list among the four created by the recruiters where the recruiter reported the list did not contain one significantly relevant candidate, this was the only list for a given position that the recruiter won overall, being chosen 3 out of 3 tries. The evaluators did not agree completely with the relevance quality of the

Despite the relevance ranked by the recruiter, the evaluators found the list more suitable. From table 6.29, on average the three evaluators ranked the list to 4.33. Comparing to the CRS, the overall evalautors said that tWhat turned out to be interesting was that the CRS was close

Table 6.26, presents a table showing the overall precision and recall given that HR is the answer key. The probability of selecting four out five overlapping with HR is set to be 0.088%, showing statistical significance.

7.2 Real World Scenario

This section discusses our findings obtained from section 6.4. We will analyze the real world scenario where we compared the Okapi BM25 algorithm based on a ranked candidate list containing 25 participants for a specific job position. This section will help in order to evaluate following research question:

RQ:1 How does the lists produced by the Candidate Ranking System compare to those made by Okapi BM25 and human recruiters?

By take a closer look at the correlation by answering the follow:

RQ1.1 How does the CRS list correlate with human recruiters against the baseline Okapi BM25 in a top-25 list for a certain position in real world scenario?

7.2.1 Discussing the Methods

From the related work literature, we learned that experts seemed to be extremely accurate when determine the first positions in the ranking. However if they found certain candidates lesser suitable, they will not be ranked them consistently (Gil et al., 2016). As explained in 5.2.7, the spearman's ranking coefficient wont take into consideration the ranking positions. In addition, the recruiter facilitating us with the top-25 list told us that he paid particular attention to the top ten, but found it a bit harder to distinguish between positions further down the list. Based on this information, it motivated us in derive calculations for using the Discounted Cumulative Gain introduced in 5.2.7. Despite we still wanted to measure the precision and recall, these measurements evaluates only on binary preferences. We therefore hoped to achieve more rigorous insight onto how both the CRS and Okapi would correlate given the weighted positions into consideration, as we would penalize high rankings that came further down the list. Hopefully, this would work as a supplement to the Spearman's Ranking Correlation when we evaluating our research questions.

7.2.1.1 Precision, Recall and E-Measure

We also compared the top-25 list by looking at their respective precision and recall at certain levels. Based on these levels, we calculated the E-Measure in order to produce a harmonic metric for the ranking systems.We decided to let the ten first candidates work as the relevant corpus among the 25.

Table 6.33 presents the raw data for the Okapi BM25 scoring. As we can see it obtained full recall after 22 hits. From the measurements we calculated that the precision at 5 was at 100%. Further we saw that it was consistently at both 60% at after ten hits, and 60% after 15 hits. Figure 6.19 illustrates.

From table 6.34 we see that CRS achieved full recall after 16 hits, given that the recruiter's top 10 is the relevant introduced initially. Same as the Okapi, the CRS achieved a 100% precision after the five first hits. After 10 hits, the precision fell down to a percentage of 70% and 60% after 15 hits.

When comparing them based on precision, they performed almost identical, despite that the CRS performed slightly better at ten hits. We notices that both systems managed to have a high precision on the first hits. The same goes for the recall levels, where after ten hits, the CRS has a recall level of 70% against Okapis 60%.

We found it more important to evaluate the precision rather then the recall. Therefore we used the E-Measure opposed to the harmonic F-Measure. In section 6.4.3, based on the precision and recall levels explained initially, we find that the CRS had consistently had a better E-Measure values overall. At full recall, meaning that all the top candidates was present in the list and the recall level would be 1, the results shows that Okapi BM25 had an E-Measure of 0.625 and the CRS having a 0.895. These results gave us some insight into the ranking systems viabilty when it comes to find the correct candidates. This also gave us some indications for further analyzing of the correlation in the section below.

7.2.1.2 Analyzing the Correlation

Looking at the Spearman Correlation
Figure 6.17 shows the spearman correlation between the recruiter and Okapi BM25. As we can see from the figure, Okapi shows in several cases to be close for some positions. However, the coefficient turned out to be 0.1284. Comparing the results with the CRS, we see from 6.18 that there is a remarkable difference. The CRS achieved a score of 0.65. Despite the difference between them, we could only show that the CRS correlates to a certain degree. From the figure 6.18 we see that the CRS degrees with the recruiter to a greater extent in the first positions in the ranking.

Punishing Lower Ranked Values
We ran two tests by using the DCG evaluation measure. The first step shown in 6.23, we weighted all the positions with at least 1 in weight and gave the top 9 positions an ascending weight from 2 till 10. From the figure we see interesting findings that the CRS was leaning towards 80%, compared to the Okapi leaning towards 60%. This gave us an indication that the CRS consistently ranked important candidates better than Okapi.

Based on the information from the recruiter explained initially, telling us that the top-ten was the separated cliff for the wheat, we wanted to run a test where we focused isolated on the top ten out of the 25-list. This meant that no score was given for the positions 11 till 25. The results presented in figure 6.24 shows that the results was very positive towards the CRS, correlated even more with the human expert based on the top 10 rather than the whole list in total. By focusing on weighting the first ten, the improvement went from 80% till 84%. Compared to the baseline, we see that there was a dramatically fall for Okapi. Going from 60.30% to a drop of 50.65%.

Comparison

Based on the two measurements with respect to correlation, we see that the CRS excelled itself by showing promising results for the given real world scenario. The combined results showed that the CRS outperformed the Okapi measure when evaluating how well the two lists correlated with respect to Expert recruiters top ten out of a top-25 list.

7.3 Integration and Deployment

We will in this section discuss the integration, deployment and the results obtained from the application. We start by analyzing the data from the data collection before we take a closer look at integration itself.

7.3.1 Evaluating the Data Collection

As mentioned earlier, we deployed a part of the system onto Highered's platform in order to harvest sufficient candidates that could take part in our evaluation of the system. We was told in advance that majority of the users was graduate students, but it also contained some with a decent amount of users with working experience. In order to find appropriate job position that could be evaluated, we analyzed the incoming data harvested from this platform. We therefore created visualization reports using Microsoft PowerBI[1] to give us sufficient insight. We deployed the report to a web-url so we could live monitor the incoming registrations. Figure 6.1 presents these visuals in one of the reports.

Firstly, we wanted to monitor skills and occupations registered to see how large percentage of the incoming skills and occupations that was from the ESCO ontology and how many where manually submitted. The purpose of this was to see if we had to consider a distance measure in order to account for typos, such as Levensthein, explained in section 2.3.4.4. This turned out to be unecessary as only 67 of 684 skills were written manually.

Secondly, we kept track on the incoming disciplines that was registered. As mentioned above, this could help us in finding the correct job positions that encompassed the majority of the registered users. From 6.1, we see that the majority of users had the education background of Management and Business, Economics and Finance related educations and Technology. This motivated us to find and select corresponding jobs that may be appropriate with respect to the dominant disciplines. From Highered's platform, we selected the positions Marketing and Sales Consultant, Financial Accountant, Accounting Analyst and Industrial Placement. The latter was especially tailored for students since it was a one year program position.

In addition to monitor the registration, we wanted to evaluate the collection to see to what degree the data registered into the CRS reflected the information found on the candidates CV's. We also wanted the candidates to give us some feedback on the use of the data collection module.

Figure 6.3 presents the result based on 98 respondents. The majority felt it was almost identical or almost identical. 17,3 percent felt it was to some degree equivalent and around 4 percent felt that it was absent. This may have an impact, as we discuss in 7.5. We also wanted to get feedback over the usability of the deployed module. Figure 6.2 presents the

[1] https://powerbi.microsoft.com/en-us/

result obtained on the usability of the module and design. We asked the respondents from 1-5 where 5 meant perfect. From the figure we see that the majority felt that the usability and using the module was on average good.

The more important feedback we got was how sufficient they found ESCO in order to find skills and previous work experience. From figure 6.4, we see that almost 60% felt that it was sufficient. From the 40% we can interpret that the candidates perhaps wanted more suggestions.

7.3.2 Integration

The data collection module was integrated into HigherEd's talent portals. They have in total 184 different portals for universities with membership in the EFDM network. Strict rules and guidelines for the security of the module had to followed to ensure that HigherEd did not breach regulations in the process. A risk management iteration was therefore performed, this is detailed in chapter 4. The identified treatments were implemented.

Among the treatments was a secure virtual machine inside HigherEd's network that would be the only place where the candidate data would be processed without anonymization. A data processing agreement was written up, covering all the the relevant regulations in GDPR and the Norwegian Personal Data Act. After implementing the treatments the, information security consultant at HigherEd gave a green light.

7.4 The Preparation and Evaluation Basis

Finding the correct position based on a diverse candidate was not easy. There was no guarantee that our data-set contained the correct candidates for these positions and that it would provide sufficient foundation for evaluating the Candidate Ranking System. In addition, we could not allow our self to send out the entire data-set to the recruiters that created the top-5 lists, as it would be extremely time consuming for the participating recruiters. Therefore, we contacted two experienced recruiters in advance and asked them how long time they thought they would use on picking a top-5 out of 30 CV's on average. The response was that on average it could take between 40 minutes up to an hour. Based on this information we opted to pick 30 CV's randomly for each job position that we control checked would contain at least five candidates that had the relevant disciplines for each position respectively. That being said, it did not necessarily mean that the candidates was perfect for the job, but could hopefully be considered relevant. When we contacted the recruiters, we asked them to choose between which job positions they felt most comfortable evaluating for.

Based on this given option, we received from one up to three evaluations from several recruiters, with the largest amount of evaluations given for the marketing and sales consultant. Since we did not know in advance the quality of the candidates in each set, we included a section in the questionnaire for the experts to give their subjectively opinion regarding the quality. In chapter 6, we see the results from the recruiters that created list for the four positions in table6.1, 6.2, 6.3and 6.4. As an example we see from table 6.2 that all the candidates the recruiter chose as top-5 was considered very relevant for the position "Industrial Placement". This gave us indications that the position at least had the sufficient foundation to be evaluated. Based on the feedback, we felt that all positions could be used in an evaluation process. The only position standing out from the expert recruiter was the Financial Accountant position presented in table 6.4. This table had a broader variety in relevance. However, we did not want to exclude this position, but noted that evaluation results may vary. This motivated further to include a section in the evaluation questionnaire where we asked the evaluators to give us some comments on their choosing.

7.5 Limitations

In this section we summarize some of our limitations during this research.

7.5.1 The amount of evaluators for the Experiment

As discussed in 7.4, screening can be a time consuming job and especially ranking top-5. Therefore, it is not easy to get a vast amount of expert recruiters to participate. Having to dealer with a smaller amount of recruiters impacts the significance as it requires a large amount of both recruiters and evaluations to prove something statistically. However, by running a real world scenario, it would perhaps give some more interesting statistical insight by comparing with a larger ranking list produced by a recruiter.

7.5.1.1 Disagreement in Accounting Analyst

For this position we had only one evaluator that evaluated. Despite, choosing the CRS list as the favourite, the evaluator said that he would hardly invite any of them to interview. This contradicts with the list produced by the recruiter, as the recruiter ranked a higher

The evaluator also said that the recruiter list, CRS and CRS reranking the recruiter was quite similar. The result of this was that it may be hard to actually justify that our list was better than the others based on comments from these comments from one evaluator.

7.5.1.2 Data Collection

Based on the results from the candidate questionnaire, presented in figure 6.3, we knew that this could potential have an impact on the top-5 ranking as some candidates may have more data present on their CV's. There was also a possibility that they invested more time in submitting their qualities into the CRS. One of the limitations regarding this, was that the CRS may perform to a certain degree poorer as the experiments was based on the CV's, and if the information present in the CRS had a large deviation from the CV, it would perform poorer.

7.5.1.3 Format of the CV's

Despite addressing this to some extent section 5.3.2, CV's still came out in a variety of different formats. A limitation here was that important information could be hard to find in some of the CV's as they did not share the same format. This could perhaps have an effect on the time consumption of evaluating, and finding important information.

Time consuming process for the evaluators. This is so because they had to learn the format for each CV.

7.5.1.4 Utilizing ESCO

A large amount may have found it hard to find what their skill and occupation names was marked as in ESCO. For instance, we we see that ”Excel” isn't stored as it names in ESCO. However, ESCO will suggest both ”use spreadsheets” and ”use microsoft office” as alternatives. As explained in chapter 2 section 2.4.7.1, the ESCO base contains a large amount of skills. However, the amount may vary a lot between certain sectors. This chart may substantiate that ESCO is a work in progress, and does not fully provide clarity in name conventions and results.

Chapter 8

Conclusion and Future Work

This chapter concludes that our findings showed that our CRS beated the Okapi BM25 for the given real world scenario and that our CRS had a significant good performance based on the simulated scenarios.

8.1 Conclusion

In this research we have developed, deployed and evaluated our Candidate Ranking System with the given set of research questions in 1.2. We have both run simulations through a set of experiments, and facilitate it to run a real world recruitment scenario. We have evaluated the system by comparing it to the Okapi BM25 measure. In this research we set out to to answer the following 2 main research questions, with the first having three sub questions. We start by concluding the first:

RQ:1 How does the lists produced by the Candidate Ranking System compare to those made by Okapi BM25 and human recruiters?
RQ:1.2 when evaluated by other experts, how does the CRS top-5 list compare to The top-5 list made/generated by human recruiters and Okapi BM25 in four job position scenarios with respect to ranking, precision and recall?
RQ:1.3 When wish to juxtapose Okapi and CRS to the list produced by a list considered to be the most precise, namely those that are made by human experts

Our findings has shown that from running a set of experiments through simulated scenarios, inspired by the Touring approach, we was able to beat both the recruiters and the Okapi BM25 in 7 out of 14 evaluations.

However, we cannot say with any statistical significance that this would be transmissible in any other given set of job positions, other sectors, or with a different set of evaluators. However, based on a binomial probability distribution we can say with significance $\alpha = 0.05$ that this wouldn't have happened randomly.

From our calculations and discussion in the previous chapters we showed that there was a significantly small chance of getting 5 correct out of 6 tries for the given position Marketing and Sales Consultant also. This may have given some indications that our CRS can perform very well for this certain positions given similar criterion's.

In addition to the Touring approach, we also evaluated our system in a real world scenario where we wanted to look for a correlation through the given position "Business Intelligence Consultant". We wanted to juxtapose the CRS against the Okapi BM25 by measuring each of their correlations with the list of therecruiter given the Spearman's Ranking Correlation and by looking at the Dicounted Cumulative Gain .

Our findings was promising, as we showed that for this given position our CRS had a much greater correlation with the recruiter based on the Spearman Ranking Correlation. Based on the feedback from the recruiter, where we were told that the top ten was especially remarkable, we ran a test using the Discounted Cumulative Gain by weighting the top ten in descending order corresponding to the position number. By comparing the two, it showed that the CRS agreed with the recruiter to a large extent, compared to what Okapi did, with an outstanding 84% compared to Okapi's 50.65%. In addition we also measured each of the systems precision and recall levels after certain hits, showing that we obtained a full recall after less hits and consistently had a greater value of precision.

Based on these findings we can conclude RQ1 that with the given information available in the case base, our CRS managed to have a good correlation, especially when weighting the top ten opposed to Okapi for a given job position and by running several experiment scenarios. We also showed that we managed to even beat the recruiters in several cases based based on evaluation from other recruiters. This gave us some interesting indications and directions for future work.

We also measured how good each of the two was on precision and recall after certain hits, and also to check how long it would take to obtain full recall, meaning all top ten out of 25 was received. Also here our CRS turned out to be better than Okapi with respect to both precision and recall.

RQ2: How does system ensure compatibility in order to be deployed in the real world with respect to the new General Data Protection Regulations imposed by the European Union and ensuring that users find necessary attributes to reflect their competency?

The system was evaluated to have a low risk during the risk management procedure outlined in 4. In this risk management procedure, breaking laws and regulations was the threat with the highest priority. From the questionnaire we sent out in relation to the data collection we had that 60% felt that they could like more suggestions when adding attributes. However, only 4% reported that their reported data was not reflective of their CV. For the current case representation, we conclude that using ESCO has been sufficient to obtain good results.

8.2 Future work

As we now have laid what we call the first stone for our Candidate Ranking System and concluded the first step discussed in 8.1 exciting questions regarding future improvement lies ahead.

8.2.1 Case Based Reasoning

The system as implemented in this paper has relied on basic adaptation techniques and perhaps an oversimplified case representation. Comparing different adaption techniques using a set of case bases of varying size could give insight into which techniques would be most suitable given a smaller case base.

8.2.1.1 Better adaption mechanism

For future work we would like to see a comparison between different adaption algorithms.For instance a recursive adaption technique could potentially be an interesting future direction for improvement.

8.2.1.2 Expanding the case description

The case description used in our implementation does not consider directly the seniority of the position or the company hiring. Expanding the case description would increase the system's ability to locally optimize. For instance, adding company to the description would open up for the system treat companies differently.

8.2.2 Experiment

8.2.2.1 More experiment participants

In order to get more statistically significant data a set of participants would have to be selected and their population would have to be estimated.

8.2.2.2 CV Layouts

To better control variables and remove chance that some CV's that stand out too much as we discussed in section 5.3.2, the CV's could be generated. This way all the CV's would have the same layout.

8.2.2.3 Working directly with the information in the CVs

The current implementation requires integration in the recruitment process. As a part of future work an approach using NLP to extract information from CV's could be used. As mentioned during the discussion, the current approach creates a divide between the reported data and the information found in the CVs. An approach relying solely on the CVs could benefit the applicants, as this would motivate them to reuse and refine their CVs. This would also address the limitation discussed in 7.5.1.2.

Bibliography

Aamodt, A., 2004. Knowledge-Intensive Case-Based Reasoning in CREEK. Springer Berlin Heidelberg, Berlin, Heidelberg, Ch. 1, pp. 1–15.
URL `https://doi.org/10.1007/978-3-540-28631-8_1`

Aamodt, A., Plaza, E., 1994. Case-based reasoning; foundational issues, methodological variations, and system approaches. AI COMMUNICATIONS 7 (1), 39–59.

Aha, D. W., Breslow, L. A., 1997. Refining conversational case libraries. In: Leake, D. B., Plaza, E. (Eds.), Case-Based Reasoning Research and Development. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 267–278.

Arnulf, J. K., Tegner, L., yunn Larssen, 2010. Impression making by rsum layout: Its impact on the probability of being shortlisted. European Journal of Work and Organizational Psychology 19 (2), 221–230.
URL `https://doi.org/10.1080/13594320902903613`

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., Patel-Schneider, P. F. (Eds.), 2003. The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, New York, NY, USA.

Baeza-Yates, R., Ribeiro-Neto, B., 2008. Modern Information Retrieval, 2nd Edition. Addison-Wesley Publishing Company, USA.

Bergmann, R., Wilke, W., Vollrath, I., Wess, S., 1996. Integrating general knowledge with object-oriented case representation and reasoning.

Berners-Lee, T., Hendler, J., 05 2001. Publishing on the Semantic Web. Vol. 410. pp. 1023–4.

Booch, G., 1994. Object-oriented Analysis and Design with Applications (2Nd Ed.). Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA.

Chen, D., Burrell, P., Dec 2001. Case-based reasoning system and artificial neural networks: A review. Neural Computing & Applications 10 (3), 264–276.
URL `https://doi.org/10.1007/PL00009897`

Codd, E. F., 8 1971. Further normalization of the data base relational model. Tech. Rep. RJ909, IBM.

Cole, M. S., Feild, H. S., Giles, W. F., Nov 2004. Interaction of recruiter and applicant gender in resume evaluation: A field study. Sex Roles 51 (9), 597–608.
URL https://doi.org/10.1007/s11199-004-5469-1

Consortium, T. W. W. W., 2004. *Resource Description Framework (RDF)*. https://www.w3.org/RDF/, [Retrieved: 12.06.2017].

Council of European Union, may 2016. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). Official Journal of the European Union L119, 1–88.

Craw, S., Wiratunga, N., Rowe, R. C., 2006. Learning adaptation knowledge to improve case-based reasoning. Artificial Intelligence 170 (16), 1175 – 1192.
URL http://www.sciencedirect.com/science/article/pii/S0004370206000798

El-Sappagh, S. H., Elmogy, M., 2015. Case based reasoning: Case representation methodologies. International Journal of Advanced Computer Science and Applications(IJACSA).

EU-Commission, 2017a. Continuous improvement workflow for esco. EU Publications.
URL https://ec.europa.eu/esco/portal/document/en/f834e202-0ebf-461a-9249-a00e91d86e94

EU-Commission, 2017b. ESCO handbook. EU Publications.
URL https://publications.europa.eu/en/publication-detail/-/publication/ce3a7e56-de27-11e7-a506-01aa75ed71a1

European-Parliament, 2008. European qualifications framework. Official Journal of the European Union.
URL http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32008H0506(01)

Eva, D., Marie, R. A., D., N. H., 2012. Multiple categorization in resume screening: Examining effects on hiring discrimination against arab applicants in field and lab settings. Journal of Organizational Behavior 33 (4), 544–570.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/job.769

Faliagka, E., Iliadis, L., Karydis, I., Rigou, M., Sioutas, S., Tsakalidis, A., Tzimas, G., Oct 2014. On-line consistent ranking on e-recruitment: seeking the truth behind a well-formed cv. Artificial Intelligence Review 42 (3), 515–528.
URL https://doi.org/10.1007/s10462-013-9414-y

Faliagka, E., Karydis, I., Rigou, M., Sioutas, S., Tsakalidis, A., Tzimas, G., 2012a. Taxonomy Development and Its Impact on a Self-learning e-Recruitment System. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 164–174.
URL https://doi.org/10.1007/978-3-642-33409-2_18

Faliagka, E., Ramantas, K., Tsakalidis, A., Tzimas, G., 2012b. Application of machine learning algorithms to an online recruitment system.

Faliagka, E., Rigou, M., Sirmakessis, S., 2015. An e-Recruitment System Exploiting Candidates' Social Presence. Springer International Publishing, Cham, pp. 153–162.
URL https://doi.org/10.1007/978-3-319-24800-4_13

Faliagka, E., Tsakalidis, A., Tzimas, G., 2012c. An integrated e-recruitment system for automated personality mining and applicant ranking. Internet research 22 (5), 551–568.

Finnie, G., Sun, Z., 2003. R5 model for case-based reasoning. Knowledge-Based Systems 16 (1), 59 – 65.
URL http://www.sciencedirect.com/science/article/pii/S0950705102000539

GAIA, 2012. Colibri studio.
URL http://gaia.fdi.ucm.es/research/colibri/colibristudio

Gil, J. M., Paoletti, A. L., Pichler, M., 2016. A novel approach for learning how to automatically match job offers and candidate profiles. CoRR abs/1611.04931.
URL http://arxiv.org/abs/1611.04931

Hamming, R. W., 1950. Error detecting and error correcting codes. Bell Labs Technical Journal 29 (2), 147–160.

ILO, 2008. ISCO-08 Standard. ILO.
URL http://www.ilo.org/public/english/bureau/stat/isco/isco08/

International Organization for Standardization and International Electrotechnical Commission, 2011. ISO/IEC 27005:. International Organization for Standardization.
URL https://books.google.no/books?id=YD7uAQAACAAJ

Kessler, R., Béchet, N., Roche, M., El-Bèze, M., Torres-Moreno, J. M., 2008. Automatic profiling system for ranking candidates answers in human resources. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer, pp. 625–634.

Kessler, R., Béchet, N., Torres-Moreno, J.-M., Roche, M., El-Bèze, M., 2009. Job Offer Management: How Improve the Ranking of Candidates. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 431–441.
URL https://doi.org/10.1007/978-3-642-04125-9_46

Kessler, R., Bchet, N., Roche, M., Torres-Moreno, J.-M., El-Bze, M., 2012. A hybrid approach to managing job offers and candidates. Information Processing and Management 48 (6), 1124 – 1135.
URL http://www.sciencedirect.com/science/article/pii/S0306457312000416

Kessler, R., Torres-Moreno, J. M., El-Bèze, M., 2007. E-Gen: Automatic Job Offer Processing System for Human Resources. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 985–995.
URL https://doi.org/10.1007/978-3-540-76631-5_94

Kmail, A. B., Maree, M., Belkhatir, M., Alhashmi, S. M., Nov 2015. An automatic online recruitment system based on exploiting multiple semantic resources and concept-relatedness measures. In: 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI). pp. 620–627.

Kruchten, P., 1995. Architectural blueprintsthe 4+ 1 view model of software architecture. Tutorial Proceedings of Tri-Ada 95, 540–555.

Lee, I., 2011. Modeling the benefit of e-recruiting process integration. Decision Support Systems 51 (1), 230–239.

Lees, B., Corchado, J., 1999. Integrated case-based neural network approach to problem solving. In: Puppe, F. (Ed.), XPS-99: Knowledge-Based Systems. Survey and Future Directions. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 157–166.

Levenshtein, V. I., 1966. Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. Vol. 10. pp. 707–710.

Maniu, I., Maniu, G., et al., 2009. A human resource ontology for recruitment process. Review of General Management 10 (2), 12–18.

Menon, V. M., Rahulnath, H. A., Sept 2016. A novel approach to evaluate and rank candidates in a recruitment process by estimating emotional intelligence through social media data. In: 2016 International Conference on Next Generation Intelligent Systems (ICNGIS). pp. 1–6.

Microsoft, 2015. N-tier data applications overview. https://msdn.microsoft.com/en-us/library/bb384398.aspx, accessed: 2018-05-02.

Oates, B. J., 2006. Researching Information Systems and Computing. Sage Publications Ltd.

Pan, J. Z., Vetere, G., Gomez, P. J. M., Wu, H., 2017. Exploiting Linked Data and Knowledge Graphs in Large Organisations, 1st Edition. Springer.

Richter, M. M., Weber, R. O., 2013a. Case-Based Reasoning: A Textbook. Springer Berlin Heidelberg, Berlin, Heidelberg.
URL https://doi.org/10.1007/978-3-642-40167-1_5

Richter, M. M., Weber, R. O., 2013b. Case-Based Reasoning: A Textbook. Springer Publishing Company, Incorporated.

Robertson, S. E., 1997. Overview of the okapi projects. Journal of Documentation 53 (1), 3–7.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M., 1993. Okapi at trec-2. In: TREC. pp. 21–34.

Roth-Berghofer, T., Sauer, C., Garcia, R. A. J., Bach, K., Althoff, K.-D., Agudo, B. D., 2013. Building case-based reasoning applications with mycbr and colibri studio. Expert Update 13 (1).

Ryan, A. M., Ployhart, R. E., 2014. A century of selection. Annual Review of Psychology 65 (1), 693–717, pMID: 24050188.
URL `https://doi.org/10.1146/annurev-psych-010213-115134`

Salazar, O. M., Jaramillo, J. C., Ovalle, D. A., Guzmán, J. A., 2015. A Case-Based Multi-Agent and Recommendation Environment to Improve the E-Recruitment Process. Springer International Publishing, Cham, pp. 389–397.
URL `https://doi.org/10.1007/978-3-319-19033-4_34`

Sheriff, D. ., 2002. N-tier data applications overview. `https://msdn.microsoft.com/en-us/library/bb384398.aspx`, accessed: 2018-05-02.

Siraj, F., Mustafa, N., Haris, M. F., Yusof, S. R. M., Salahuddin, M. A., Hasan, M. R., Sept 2011. Pre-selection of recruitment candidates using case based reasoning. In: 2011 Third International Conference on Computational Intelligence, Modelling Simulation. pp. 84–90.

Siting, Z., Wenxing, H., Ning, Z., Fan, Y., July 2012. Job recommender systems: A survey. In: 2012 7th International Conference on Computer Science Education (ICCSE). pp. 920–924.

Spearman, C., July 1906. footrule for measuring correlation. British Journal of Psychology, 19041920 2 (1), 89–108.

Turing, A. M., 2009. Computing machinery and intelligence. In: Parsing the Turing Test. Springer, pp. 23–65.

Whitman, M., Mattord, H., 2017. Principles of Information Security. Cengage Learning.
URL `https://books.google.no/books?id=LtdUDgAAQBAJ`

Wu, Z., Palmer, M., 1994. Verbs semantics and lexical selection. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, pp. 133–138.

Appendix A

Candidate Ranking System

A.1 Storage Container CV

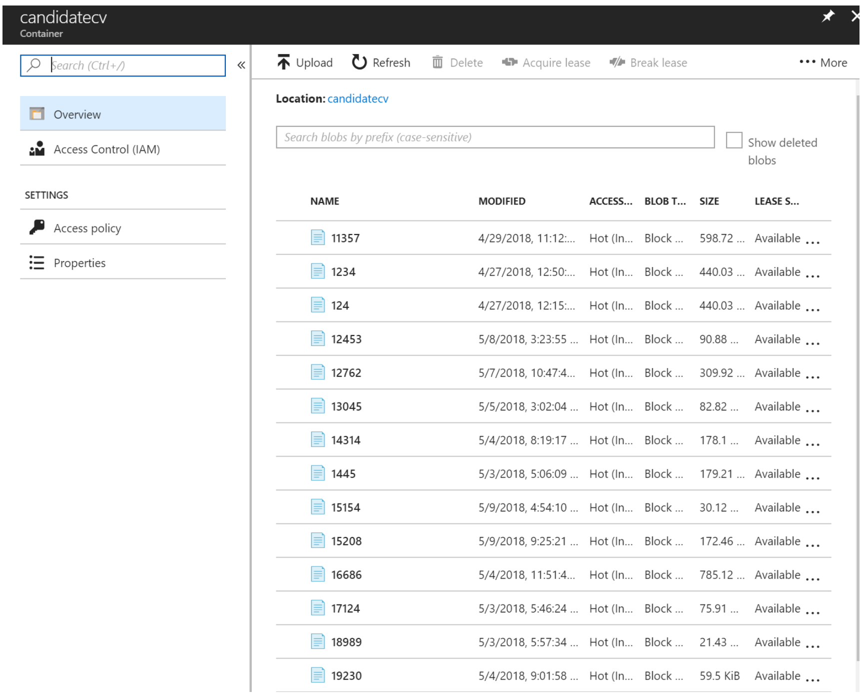


Figure A.1: screendump over CV Storage Container

A.2 Application Modules

A.2.1 CRS Dashboard

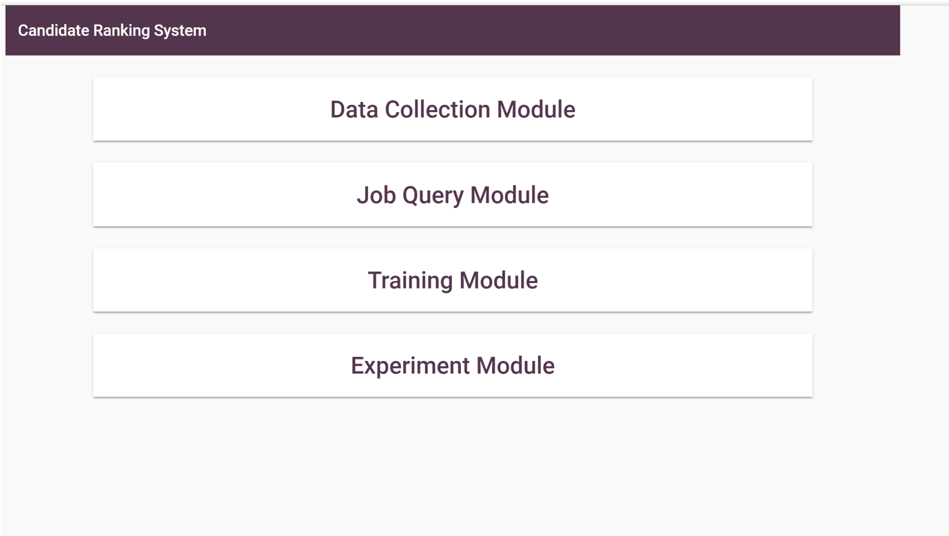


Figure A.2: Screendump from the CRS Dashboard

A.2.2 Data Collection Module

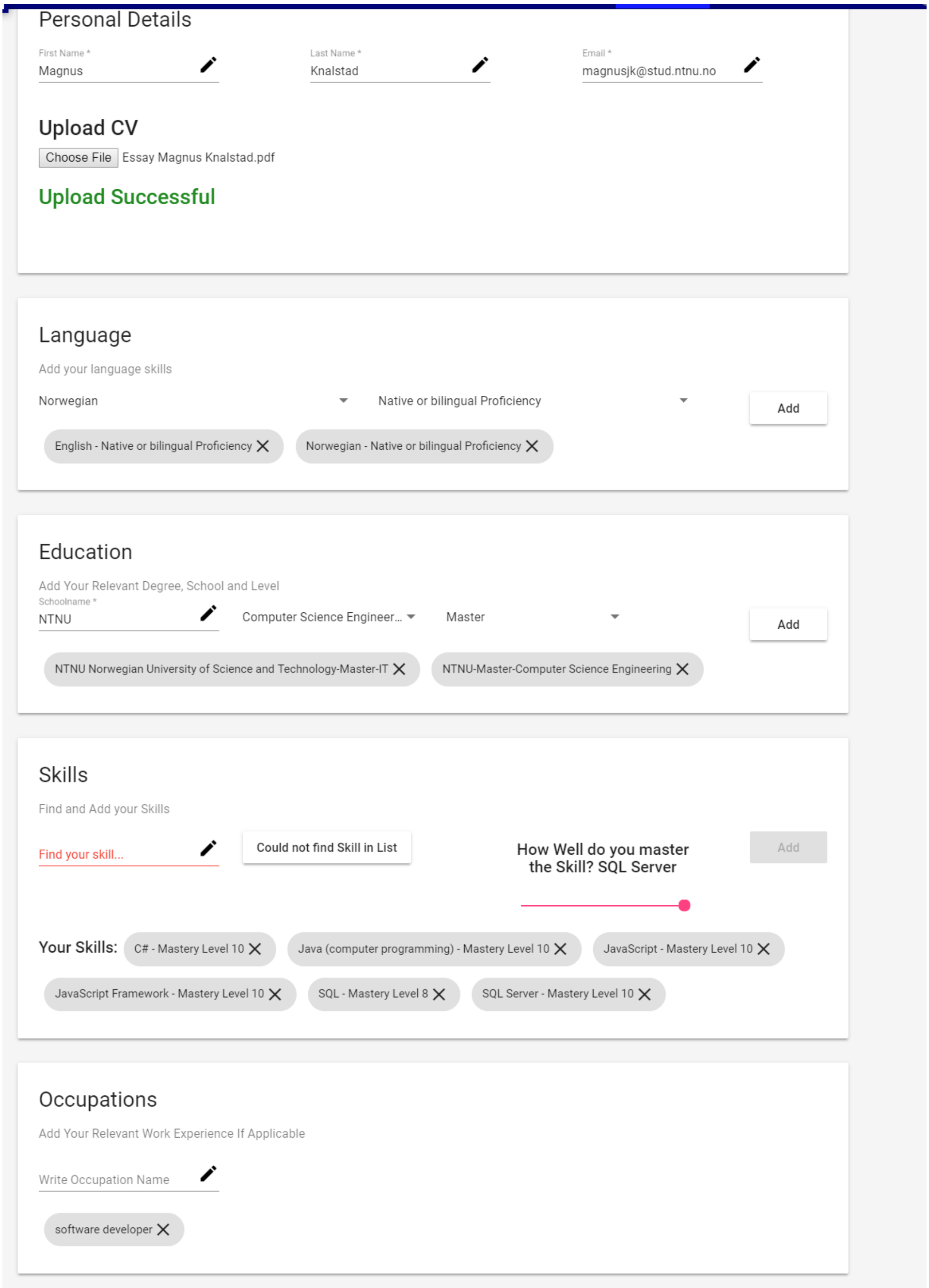


Figure A.3: Screendump from the candidate registration in the DCM view

Appendix B

Flyers and questionnaires

B.1 HigherEd flyer for experiment participants

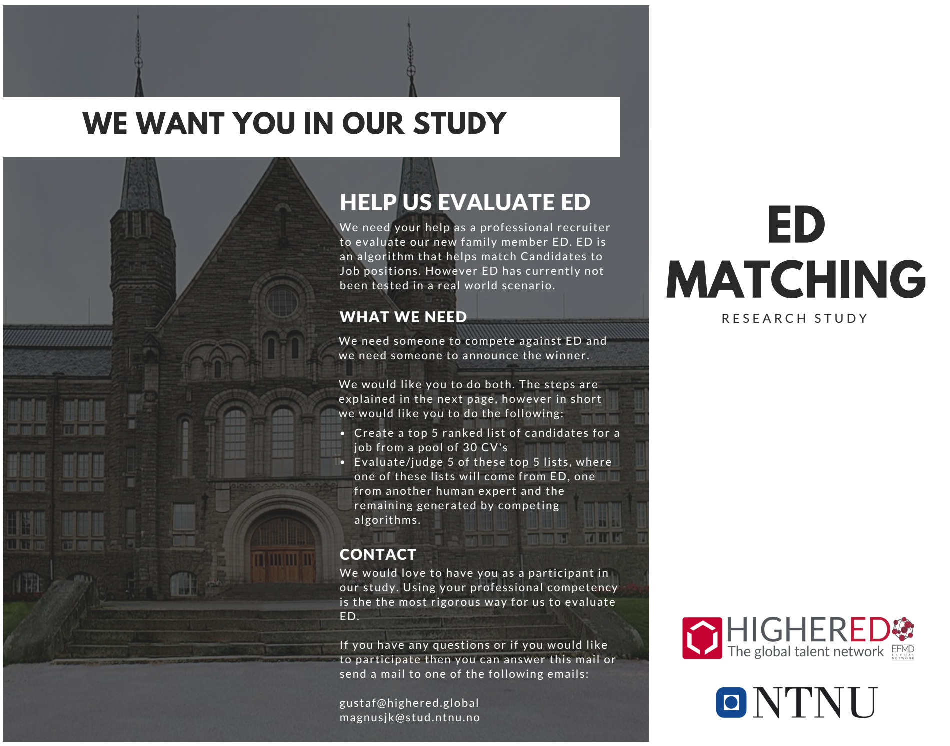


Figure B.1: Page 1 Highered experiment flyer

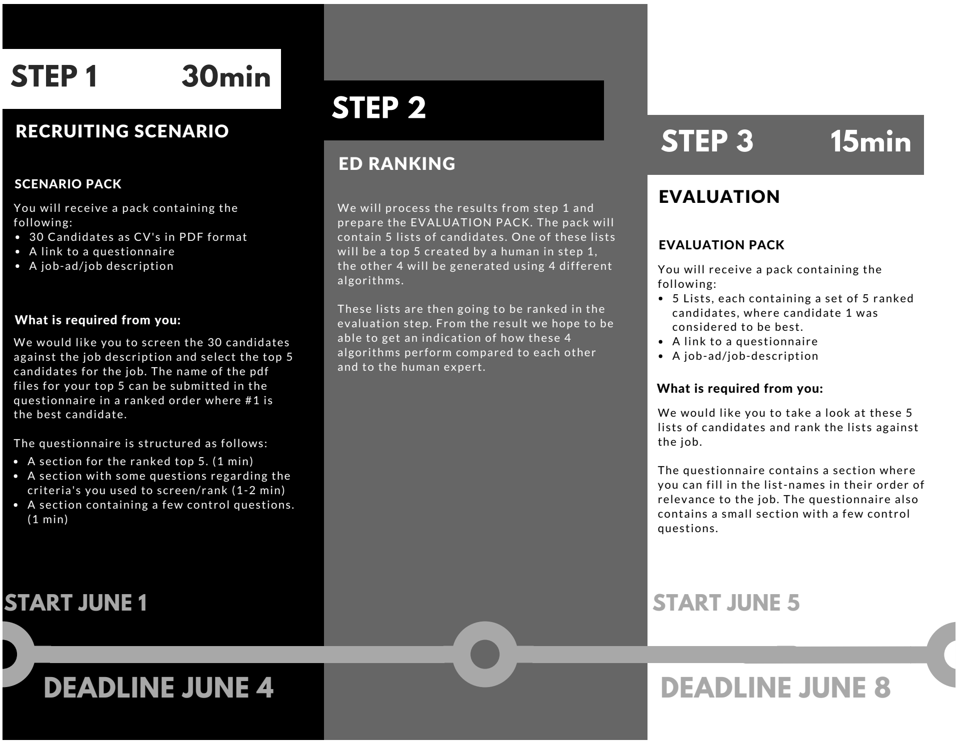


Figure B.2: Page 2 Highered experiment flyer

B.2 Ingraphic flyer for experiment participants

Figure B.3: Page 1 Ingraphic experiment flyer

B.3 Data collection evaluation questionnaire

B.3.1 Questionnaire

B.3.2 Results

B.4 Experiment data

Since this position Accounting Analyst contain a significant larger degree of text, it is listed below:

- Minimum of 3 years experience in similar or complementary roles with financial institutions and/or public accounting firms;
- Proficiency in Microsoft Windows applications, namely Excel, Word and Power Point, including advanced spreadsheet design, manipulation, and analysis skills.

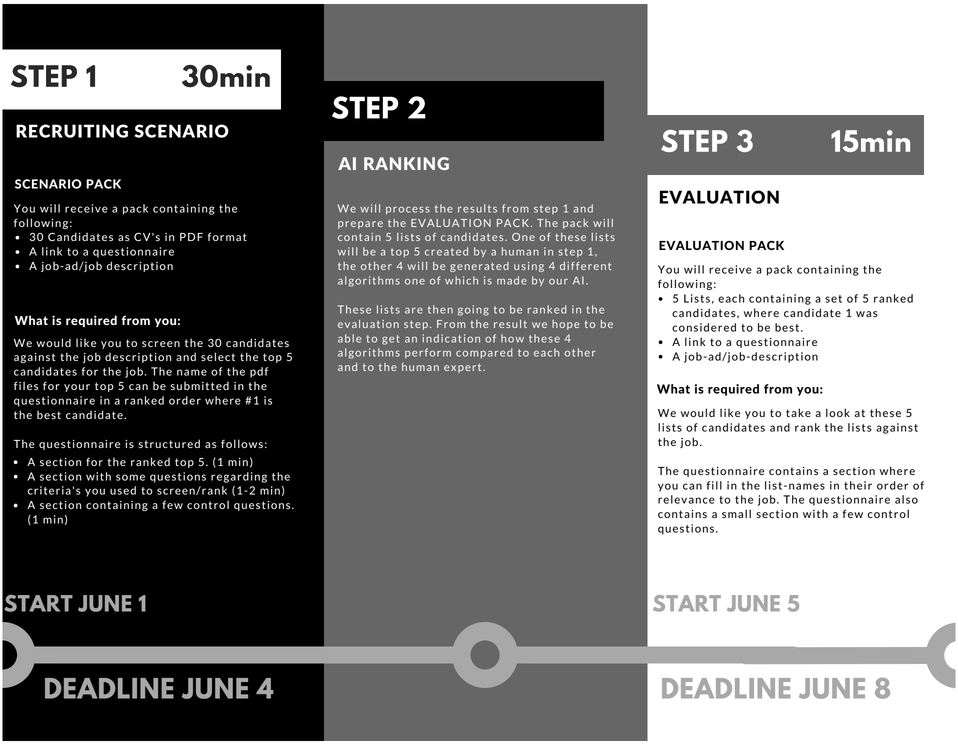


Figure B.4: Page 2 Ingraphic experiment flyer

Criteras in position Marketing and Sales Consultant at BASF
Master/mba/bachelor in economics and business administration or natural sciences
5 years of experience
Marketing & Sales Strategy Development
Portfolio Management
Pricing Excellence
Product Positioning
Market Intelligence and Wargaming
Marketing & Sales Competency Framework
Customer Focus
German and English is preferred
Visionary and analytic mindset
You also have excellent advising and project management skills
Added criterias/comments:
Advise and support the business units and regional units - perceived seniority/organizational skills important

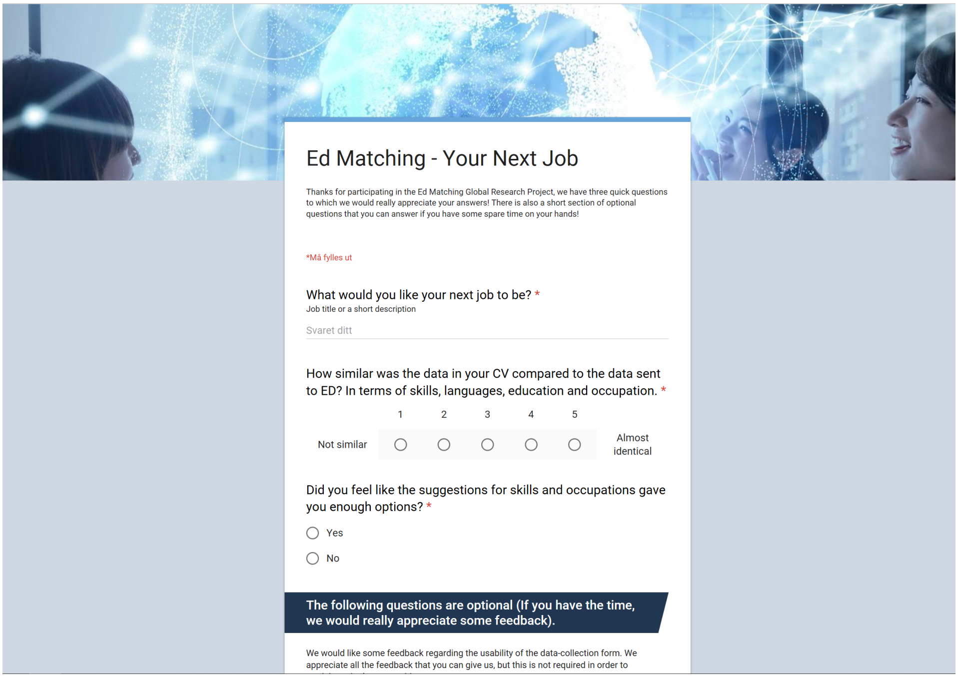


Figure B.5: DCM Evaluations Part 1/2

Criteras in position Industrial Placement at AlphaSights
Candidate must be on a course that supports one-year industry placement
Strong acamdeic credentials in either Economics or Business Management
Noteworthy Extracurricular leadership
Evidenced success in a professional and/or extracurricular field
Fluency in English is essential. Fluency in a relevant foreign language is a plus
Vast experience with multiple projects within business development
Preferred Background is Economics and/or Business Management

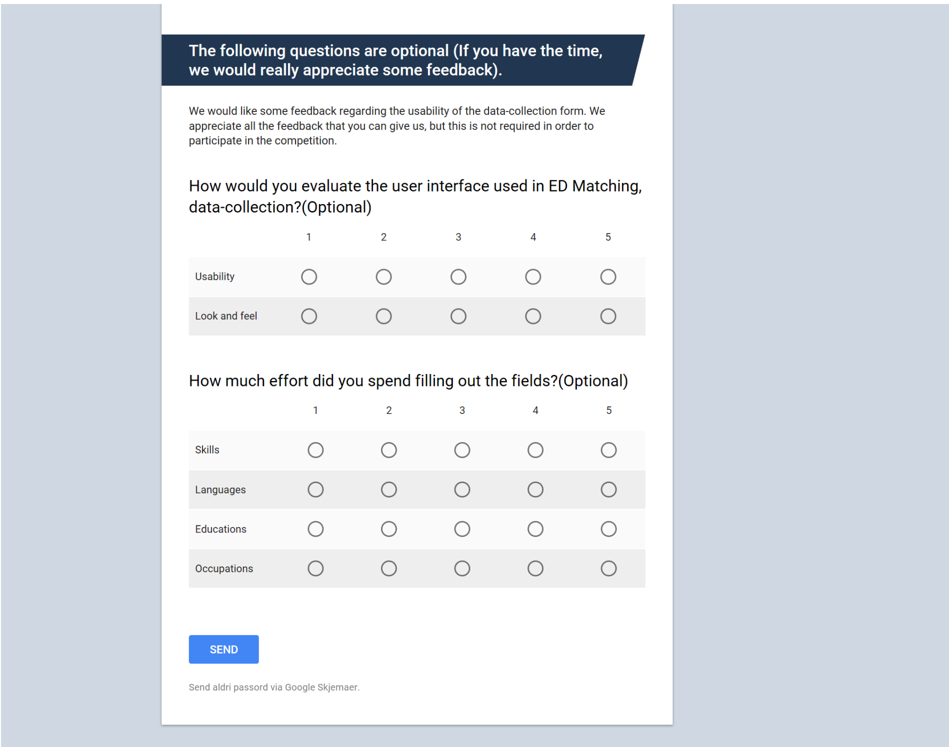


Figure B.6: DCM Evaluations Part 2/2

Criteras in position Financial Accountant
Bachelor/master/mba
Degree Qualified and Part, or fully CA/CPA
3+ Years previous accounting experience in a high volume multi-currency environment
Previous experience in SAP, hyperion & SUN is preferred
Previous experience in SAP, hyperion & SUN is preferred
Intermediate/Advanced Excel Knowledge
Previous experience in a large stock-exchange listed company
Self-starter, proactive, strong work ethic
Excellent organisational and communication skill
Added criterias/comments:
Grades still relevant
Very accounting focuse, so will be a recuirement with (at least) either extensive accounting course or experience
Looking for person with structure and attention to detail

Ability to quickly develop effective understandings of logical accounting data structures and data inter-relationships to effectively evaluate and make recommendations on process/system controls.

- In-depth, detailed understanding of accounting systems, preferably SAP FI modules and their control features and configuration capabilities/implementation at the Bank.
- Knowledge of a variety of modern procurement practices and emerging electronic commerce solutions; excellent knowledge of Bank administrative policies.
- Substantial work experience with transaction processing using integrated, automated accounting systems with a comparable level of complexity to SAP R3 enterprise resource planning system and high degree of integration.
- A proven track record of problem solving skills in an automated accounting environment; experience with SAP enterprise resource planning system R3 is desirable, but not required.
- Excellent judgment, very good written and oral communications skills (including the ability to write clear, concise and complete procedure); ability to carry out a multi-dimensional, multi-functional work program.
- Ability to deal sensitively in multi-cultural environments and build effective working relations with clients and colleagues.
- Exhibit a personality that enables the incumbent to be client focused, flexible and an excellent team player. Ability to effectively work with other staff in the department and elsewhere in the Bank.
- Self-motivated person with high degree of initiative and a strong team player.
- **Preferred Education:** Vachelor degree in Commerece, or Accounting. ACA/CPA or equivalent is desirable.
- Business Processes & Systems - Has basic understanding of business processes and systems $includingaccountingsystems$ to recommend and help implement process improvements.
- Fundamental Controllership Knowledge - Has a basic understanding of general accounting principles, risks and controls, financial concepts and financial transaction processes.
- Accounting Policy - Understands accounting principles and, under guidance of senior staff, is able to apply accounting principles to appropriately record transactions.
- Transaction Processing - Has substantial work experience with using automated systems to process transactions.
- Risk Management & Internal Controls - Applies basic understanding of internal. controls/risk management principles to document risks and controls in Bank activities.

- Client Orientation - Takes personal responsibility and accountability for timely response to client queries, requests or needs, working to remove obstacles that may impede execution or overall success.

- Drive for Results - Takes personal ownership and accountability to meet deadlines and achieve agreed-upon results, and has the personal organization to do so.

- Teamwork (Collaboration) and Inclusion - Collaborates with other team members and contributes productively to the team's work and output, demonstrating respect for different points of view.

- Knowledge, Learning and Communication - Actively seeks knowledge needed to complete assignments and shares knowledge with others, communicating and presenting information in a clear and organized manner.

- Business Judgment and Analytical Decision Making - Analyzes facts and data to support sound, logical decisions regarding own and others' work.

- Project Management - Understands the basic concepts of project management, as they relate to the execution of tasks within a project.

B.5 Evaluation Data

Recruiter	Position To Produce List	Evaluates
Recruiter 1 Experis	1- Financial Accountant	4-Marketing and Sales Consultant
Recruiter 2 Experis	2-Accounting Analyst	3-Industrial Placement
Recruiter 3 Experis	3-Industrial Placement	2-Accounting Analyst
Recruiter 4 Experis	4-Marketing and Sales Consultant	1-Financial Accountant
Recruiter 4 Experis	1-Financial Accountant	4-Marketing and Sales Consultant
Recruiter 6 Experis	2-Accounting Analyst	3-Industrial Placement
Recruiter Primepeople	3-Industrial Placement	2-Accounting Analyst
Recruiter PanameraSearch	4-Marketing and Sales Consultant	1-Financial Accountant
Recruiter yara	1-Financial Accountant	4-Marketing and Sales Consultant
Recruiter AlphaSight	2-Accounting Analyst	3-Industrial Placement
Recruiter AlphaSight	3-Industrial Placement	2-Accounting Analyst
Recruiter Amrodelphi	4-Marketing and Sales Consultant	1-Financial Accountant

Figure B.7: Original Evaluation Plan