Arab American University

**Faculty of Graduate Studies**

# TOWARDS BUILDING A HYBRID APPROACH FOR CONCEPTUAL-BASED CLASSIFICATION AND RANKING OF RESUMES AND THEIR CORRESPONDING JOB POSTS

By

**Abeer A. Z'aroor**

**Main Supervisor**

**Dr. Mohammed A. M. Maree**

**Co-Supervisor**

**Dr. Muath N. M. Sabha**

**This thesis was submitted in partial fulfillment of the requirements for the Master's degree in Computer Science**

**May/ 2018**

**Towards Building A Hybrid Approach for Conceptual-based Classification and Ranking of Resumes and their Corresponding Job Posts**

**By**

Abeer A. Z'aroor

This thesis was defended successfully on ……………………and approved by:

| Committee members | Signature |
|---|---|
| 1. Supervisor Name: Dr. Mohammed Maree | ............................... |
| 2. Co- Supervisor Name: Dr. Muath Sabha | ............................... |
| 3. Internal Examiner Name: Dr. Rami Hodrob | ............................... |
| 4. External Examiner Name: Dr. Rashid Jayousi | ............................... |

## Declaration

This is to declare that the thesis entitled "Towards Building A Hybrid Approach for Conceptual-based Classification and Ranking of Resumes and their Corresponding Job Posts" under the supervision of Dr. Mohammed A. M. Maree and Dr. Muath N. M. Sabha is my own work and does not contain any unacknowledged work or material previously published or written by another person, except where due reference is made in the text of the document.

Date: 12/5/2018

Name: Abeer A. Z'aroor Signature:

## Dedication

This research work is wholeheartedly dedicated to my beloved parents, who have been our source of inspiration and gave us strength when we thought of giving up, who continually provide their moral, spiritual, emotional, and financial support.

To my brother, sisters, relatives, friends, and classmates who shared their words of advice and encouragement to finish this thesis.

## Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor Dr. Mohammed Maree for the continuous support of my master thesis and research, for his patience, enthusiasm, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

My sincere thanks also goes to my co-supervisor Dr. Muath Sabha for the useful comments, remarks and engagement.

Huge thanks, love and appreciation go to my family: My father Mr. Abdullah Zaroor, my beloved mother Mrs. Nuha Haji, my one and only my brother Ahmad Zaroor and to all of my sisters: Shaheera, Ala', Reema, Rania, Shahed for their belief in me and my work and for supporting me spiritually throughout my life.

A very special gratitude goes out to all of my friends who have supported me over the last few years: Leena Abu khamies, Shafaq Abulhouf, Rudina Subaih, Reema Ewais, Sanabel Fqees, Wesal, Maysan Omar, and Azhar Hamdan.

And finally, last but by no means least, a huge thanks to my fiancée Eng. Ahmed Abu Mwais for his support, patience, and understanding.

Thanks for all your encouragement

## Abstract

With the increasing growth in online recruitment, traditional hiring methods are becoming inefficient. This is due to the fact that job portals receive enormous numbers of unstructured resumes - in diverse styles and formats - from applicants with different fields of expertise and specializations. Therefore, the extraction of structured information from applicant resumes is needed not only to support the automatic screening of candidates, but also to efficiently route them towards their corresponding occupational categories. This process assists in minimizing the effort required by employers to manage and organize resumes, as well as to screen out irrelevant candidates.

Several techniques and approaches have been proposed to address the issue of automatic matching between resumes and job postings. However, little attention has been paid to address problems associated with classification of resumes and job posts, automatic ranking of applicant resumes, and automatic profile generation from applicants' resumes. In this research work, we present a Job Post and Resume Classification system that exploits an integrated knowledge base for carrying out the classification task. Unlike conventional systems that attempt to search globally in the entire space of resumes and job posts, the proposed approach matches resumes that only fall under their relevant occupational categories. In addition, our proposed system attempts to exploit the extracted information from applicants' resumes to automatically generate user profiles that can be further used for recommending jobs to job seekers. In this context, our proposed approach attempts to push job post notifications that satisfy job seekers' preferences and skills. To demonstrate the effectiveness of the proposed system, we have conducted several experiments using a real-world recruitment dataset. Additionally, we have evaluated the efficiency and

effectiveness of proposed system against state-of-the-art online recruitment systems and the results were published in two well-recognized international conferences in 2017.

# Table of Contents

## List of Figures

## List of Tables

## List of Abbreviations

**HR –** Human Resources

**SVM** – Support Vector Machine

**HMM** – Hidden Markov Model

**SVM-kNN** – Support Vector Machine- k Nearest Neighbor

**NLP –** Natural Language Processing

**CV –** Curriculum Vitae

**NER -** Named Entity Recognition

**JRS -** Job Recommender System

**O*NET -** Occupational Information Network

**ICT** - Information and Communication Technologies

**HS -** Hiring Solved

**POS-Tagging** - Part-of-Speech Tagging

**tf–idf** - term frequency–inverse document frequency

**FE -** Feature Extraction

**CBR -** Content-based Recommendation

**KBR -** Knowledge-based Recommendation

**CFR -** Collaborative Filtering Recommendation

**HyR -** Hybrid Recommendation

# 1. Introduction

In the recent years, online job portals have started to receive enormous numbers of resumes in diverse styles and formats from job seekers who have different academic backgrounds, work experiences and skills (Kmail et al., 2015b, Faliagka et al., 2014). Finding and hiring the right talent from a wide and heterogeneous range of candidates remains one of the most important and challenging tasks of Human Resource (HR) departments in many organizations (Chen et al., 2015, Hauff and Gousios, 2015).

To address this challenge, many companies have shifted to exploit e-recruiting platforms (Mehta et al., 2013, Faliagka et al., 2014, Schmitt et al., 2016, Brandão et al., 2017). These platforms aim at reducing the cost, time and effort required for manually processing and screening applicant resumes (Sivabalan et al., 2014). As stated in (Al-Otaibi and Ykhlef, 2012a), there were more than 40,000 e-recruitment sites in 2012 for helping job seekers and employers worldwide. According to the International Association of Employment Web Sites (IAEWS)[1] the number of e-recruitment systems has become more than 60,000 in 2017.

These systems employ different methods and approaches to address the challenges associated with screening, matching, and classifying resumes and job posts. For instance, (Senthil Kumaran and Sankar, 2013, Kmail et al., 2015a) employ automatic matching methods to assign relevance scores between candidate resumes and their corresponding job offers. The main goal of these systems is achieving high precision ratios i.e. finding the best candidates for a given job post, while ignoring the cost (run time complexity) of the

[1] http://www.icmaonline.org/international-association-of-employment-web-sites

matching process. Other systems have attempted to reduce the cost issue by first segmenting the content of both resumes and job posts and finding matches between important relevant segments accordingly. For instance, (Yu et al., 2005, Kessler et al., 2009) propose using machine learning algorithms: Support vector machines (SVM) (Kessler et al., 2009) and Hidden Markov Models (HMM) (Yu et al., 2005) to automatically extract structured information from job posts and resumes by annotating the segments of job posts and resumes with the appropriate features and topics. While (Amdouni and abdessalem Karaa, 2010, Kmail et al., 2015a) use Natural Language Processing (NLP) techniques to implement the segmentation and information extraction module. Although these approaches have proved to be more efficient in carrying out the matching task (Kmail et al., 2015a), every newly obtained resume still needs to be matched with all job offers in the corpus.

To tackle this issue, other researchers propose utilizing machine learning-based techniques to classify job posts and resumes into occupational categories before conducting the matching task (Clyde et al., 1995, Javed et al., 2015, Zhu et al., 2016). However, as pointed in (Neculoiu et al., 2016), these techniques suffer from the high cost of manually classifying large amounts of job posts.

Starting from this position, we propose building a hybrid approach to classify resumes and their corresponding job posts by utilizing an integrated occupational categories knowledge base. The exploited knowledge base is utilized for i) classifying resumes and job offers based on their corresponding occupational categories, and ii) for automatically ranking applicants that best match the announced offers. In addition, the proposed approach attempts to exploit the extracted information from applicants' resumes to dynamically generate user profiles to be further used for recommending jobs to job seekers.

The remainder of this chapter is organized as follows. The background and motivations behind our proposed system are presented in Section 1.1. In Section 1.2, we present the drawbacks and limitations that are associated with existing online recruitment systems. Section 1.3 provides the research methodology. Section 1.4 defines our contributions and clarifies the obstacles that we attempt to overcome through the proposed system. Section 1.5 presents our recently accepted publications. Finally, the structure of our thesis is introduced in section 1.6.

## 1.1 Background and Motivations

Over the last few years, online recruitment has expanded significantly (Strohmeier and Piazza, 2013, Senthil Kumaran and Sankar, 2013). This expansion has led to a continuous growth in the number of job portals and hiring agencies on the Internet. It has also led to a constant increase in the number of job seekers searching for new career opportunities (Sivabalan et al., 2014, Buga et al., 2017). As a consequence, Finding and hiring qualified individuals who have all necessary skills and meet all job requirements became one of the most important, yet difficult, tasks for any HR department.

Several approaches have been proposed to support the automatic matching between applicant CVs and their corresponding job offers (Senthil Kumaran and Sankar, 2013, Hauff and Gousios, 2015, Kmail et al., 2015a). Other approaches have attempted to automate the extraction of structured segmented information from both job posts and resumes to be later used in the matching and classification processes (Amdouni and abdessalem Karaa, 2010, Chen et al., 2015). Although these approaches produce high precision ratios in selecting candidates to fill a vacant position, they still suffer from efficiency drawbacks since every newly uploaded resume needs to be matched with all job offers. To overcome this problem, other researchers utilize machine learning techniques to first classify job posts and resumes under their relevant occupational categories (Javed et al., 2015, Zhu et al., 2016). Although these techniques have proven to be more efficient (i.e. have low run time complexity), they suffer from the high cost of the manual classification of large amounts of job title data (Neculoiu et al., 2016).

Motivated by these observations, we present a hybrid approach that employs conceptual-based classification of resumes and job postings and automatically ranks candidate resumes (that fall under each occupational category) to their corresponding job postings. Furthermore, and unlike conventional recruitment systems (that are generally oriented towards assisting employers in finding the candidates that best match their job requirements) our proposed approach aims to pay more attention to candidates through automatically generating profiles from their submitted CVs in order to assist them by recommending job offers that best match their skills and preferences.

### 1.2 Problem Statement and Research Questions

As we have discussed in the previous section, many precision-oriented techniques have been proposed to find matches between candidate resumes and their corresponding job offers. However, little attention has been paid to addressing problems associated with automatic resumes and job posts classification (Bekkerman and Gavish, 2011, Javed et al., 2015, Zhu et al., 2016). For instance, when an employer seeks a “Web Developer” that falls under “Web Development” occupational category, conventional e-recruitment systems search globally in the entire space of resumes for finding applicants that best match the offered position. In this context, each and every resume in the resumes collection will be matched to the offered job posts instead of matching only those that fall under their corresponding occupational category ("Web Development" in our example). To address this issue, many approaches and techniques have been proposed to reduce the run time complexity and the cost required for hiring new employees (Yu et al., 2005, Javed et al., 2015, Zhu et al., 2016). Even though these approaches succeeded in enhancing the efficiency of the matching process, they still suffer from several limitations associated with their underlying methods and techniques as presented in section 1.1.

In this section, we present the research questions that we endeavor to examine and address during our research work.

- What are the strengths and weakness of the methods and techniques that are employed by current e-recruitment systems?

In order to answer this question, we have conducted a comprehensive comparative analysis between current online recruitment systems and studied the features of the implemented methods /techniques by each system.

- How to reduce the high cost (run time complexity) of the matching process between resumes and their corresponding job posts?

To tackle this issue, we propose a job resume classification system that employs an integrated occupational categories knowledge base to efficiently route both job posts and resumes to their corresponding occupational categories.

- Can we process unstructured resumes to generate automatic user profiles?

In order to answer this question, we utilize feature extraction techniques to convert unstructured resume to semi-structured format. After that, we employ parsing techniques to convert the semi-structured resumes to structured format. Finally, we anticipate a statistical based concept relatedness measures and integrated knowledge base to further enrich the applicant profiles.

### 1.3 Research Methodology

The following points introduce the main steps that we carried out during our research work:

- **Section-based Segmentation and Low-weighted Terms Removal**

At this phase, resume segments such as Education, Experience, Loyalty and other Employment information such as Company name, Applicant's Role in the company, Date of designation, Date of resignation and Loyalty are automatically extracted. During this step, unstructured resumes are converted into segments (semi-structured document) based on employing Natural Language Processing (NLP) techniques and rule-based regular expressions. More details on this phase are provided in section 4.1.1.

Once resumes are converted into semi-structured documents, the list of candidate matching concepts (skills set) is identified, extracted, and filtered. To carry out this task, we utilize the term frequency–inverse document frequency (tf-idf) weighting scheme (Belkin and Croft, 1992). In this context, concepts that belong to the list of pre-defined expressions and terms (e.g. contact information, birth of date (BoD), country name, address) or have low tf-idf weights are removed from the lists of candidate concepts. Accordingly, the candidate concepts list is refined and filtered to be further processed during the classification phase as detailed below.

- **Conceptual Classification of Resumes and Job Posts**

During this phase, each skill in the skills set, that are extracted from both resumes and job posts, is sequentially submitted to an integrated occupational categories knowledge base in order to enrich it with a list of semantically-relevant occupational categories. As a result, a list of occupational categories is obtained wherein they are sorted based on their semantic relevance strength. After classifying job posts and resumes, category-based matching is conducted in order to match resumes with their corresponding job posts that fall under the same space as detailed below.

- **Matching Resumes to their Corresponding Job Postings**

Unlike conventional approaches that carry out the matching task between resumes and job posts based on searching globally in the entire space of resumes and job posts, our goal of this step is to minimize the matching space through assigning relevance scores between resumes that only fall under the same occupational category(ies) of each given job posts. Moreover, we employ multiple semantic resources and concept-based relatedness measures to construct semantic networks for both job posts and resumes. These semantic networks are used to find resume-to-job offer matches based on an edit distance function that returns measures of semantic closeness between the semantic networks.

- **Automatic Profile Generation**

At this phase, we utilize the extracted segments from resumes in order to automatically create profiles. To do that, first we employ document parsing techniques to convert the semi-structured resume into structured format. Then, we further enrich the profiles by automatically adding new skills obtained from our integrated knowledge base and concept relatedness measures techniques related to their experience field. After that, we exploit these profiles in order to push job recommendation to job seekers based on their qualifications and experience as detailed below.

- **Recommending Job Posts to Job seekers**

Skills in the newly generated applicant profiles are used to push job post notifications that satisfy the required skills by job seekers. To accomplish this task, we employ an edit distance function. This function returns measures of similarity between job seeker's profile and their corresponding job posts that fall under the same space. In this context, the higher the similarity score is, the more a job post is considered relevant to the applicant's qualifications.

### 1.3.1 System Evaluation

To validate the efficiency and effectiveness of the proposed online recruitment system, we use state-of-the-art indicators. Moreover, we compare the produced results by the proposed system with one of the state-of-the-art systems. We collect a data set of 2000 resumes and we used 10,000 different job posts obtained from different online recruitment systems. The collected resumes are unstructured documents in different formats such as (.pdf) and (.doc), and we considered job posts as structured documents having the following segments (job title, job description, required skills, years of experience, required education qualifications and additional desired requirements).

## 1.4 Contributions

We summarize the main contributions of our research work as follows:

1. Proposing an automatic occupational category based classification of resumes and job postings system that exploits an integrated knowledge base for carrying out the classification task. Accordingly, and unlike traditional online recruitment systems that didn't take into consideration the run time complexity of the matching process, our proposed system attempts to minimize the searching space by classifying both job offers and resumes to appropriate occupational categories. By doing that, only resumes that fall under the same space of the job offers will be considered in the matching process.
2. Proposing an automatic profiles generation from applicants' resumes by employing a section-based segmentation heuristic that exploits natural language processing (NLP) techniques, regular expressions and parsing techniques. Then we use our integrated knowledge base and concept relatedness measures techniques to further enrich the applicant profiles with additional skills related to their experience field.
3. Proposing a user Profile Dependent Recommender system that exploits the enriched applicants' profiles to provide more precise job post recommendations that satisfy their preferences and qualifications.

## 1.5 Publications

In this section, we list our recently accepted publications in the field of classification and matching in online recruitment systems. In the first paper, we propose a job post and resume classification online recruitment system that integrates the following modules: Section-based Segmentation Module, Filtration Module, Classification Module and

Category-based Matching Module. While in the second paper, we extend the experiments of our proposed prototype online recruitment system to validate its efficiency and effectiveness.

- Zaroor, A., Maree, M., & Sabha, M. (2017, June). A Hybrid Approach to Conceptual Classification and Ranking of Resumes and Their Corresponding Job Posts. In International Conference on Intelligent Decision Technologies (pp. 107-119). Springer, Cham.
- Abeer Zaroor, Mohammed Maree and Muath Sabha .“JRC: A Job Post and Resume Classification System for Online Recruitment”. Accepted for publication in the Proceedings of the 29th IEEE International Conference on Tools with Artificial Intelligence (ICTAI2017).

## 1.6 Structure of the Thesis

The rest of this thesis is organized as follows. In Chapter 2, we present a background about the evolution of recruitment. In addition, we introduce a comprehensive comparative analysis of existing online recruitment systems. A general overview of the architecture of our proposed system is presented in chapter 3. We introduce a detailed description of the techniques and methods that we utilize in the proposed system in Chapter 4. Chapter 5 presents the conducted experimental evaluation of the efficiency and effectiveness of the proposed system. In this chapter, we also compare between the result produced by our system and one of the state-of-the-art online recruitment systems. Finally, in Chapter 6, we discuss the conclusions and outline the future extensions of our research work.

# 2. Literature Review

We start this chapter with a background about the recruitment process, discuss the e-recruitment problem and present the state-of-the-art solutions oriented to candidates/job matching, information extraction and classification of resumes and job posts. Then, we provide a comprehensive comparative analysis of existing techniques/approaches that are employed in the automatic recruitment domain and highlight their strengths and weaknesses.

## 2.1 Background

Recruiting process is a core function of human resource management that aims to find and hire the best-qualified candidates whom are valuable for the company, in a timely and cost effective manner (Färber et al., 2003, Sivabalan et al., 2014, Schmitt et al., 2016). In the past, recruitment depended extensively on word-of-mouth and face-to-face meetings were many organizations hire employees through collecting a handwritten CVs and walk-in applications after posting a job vacancy on a traditional media such as bulletin boards, newspapers, magazines, and job agencies (Pande, 2011, Barber, 2006). After attracting job seekers to that job, recruiters select candidate applicants by screening their resumes. Finally, candidates are invited for meetings to validate their language competencies, and communication skills. Figure 1 illustrates the stages of the manual recruitment process.

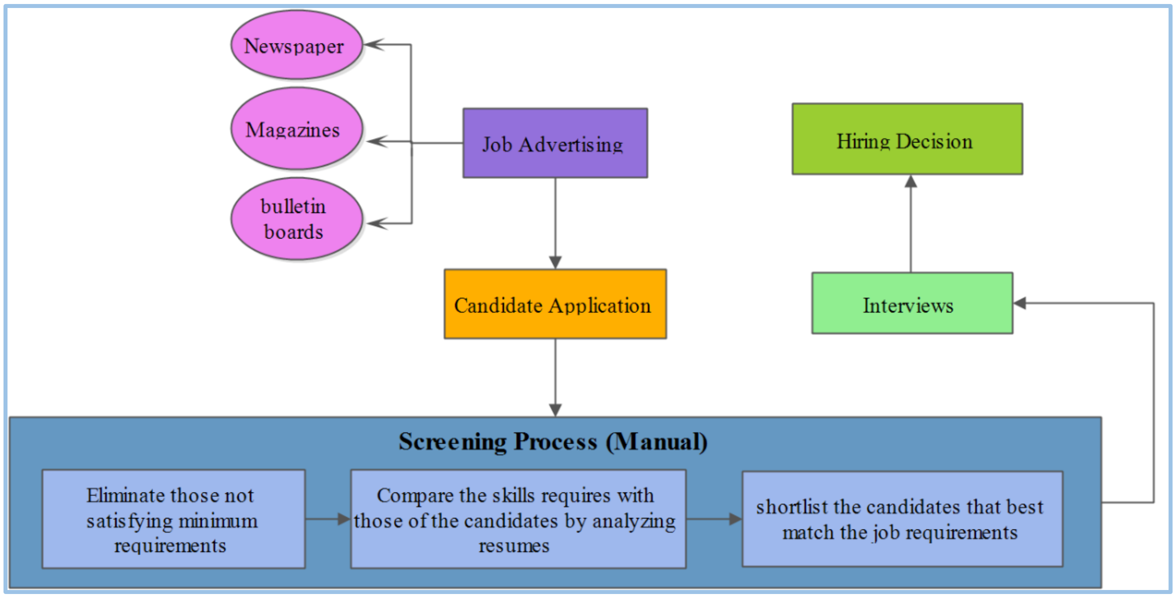


**Figure 1. Manual recruitment process Stages (Barber, 2006)**

Although manual recruitment performs well in screening out unqualified applicants, it still has the following limitations (Kerrin and Kettley, 2003, Keim et al., 2005, Malinowski et al., 2006, Lee, 2007a):

- Insufficient storage of information: as the hiring agencies would have to store files and archives of masses of CVs written on paper, making applications difficult to access and sort through.
- Reduction in sharing information and customer services: customer queries can be difficult to respond to as information is stored in different place.
- Costly and time consuming process, in addition to the high efforts required for screening out irrelevant applicants.
- Duplication of data entry: since tracking of documents, files and transactions is a tedious task for employees, the same data might get repeated many times.
- Inconsistency in data entry, room for errors, mistaken information since data might get misplaced during manual filing.

With the rise of the internet and modern technology, many companies have shifted to use automatic online recruitment systems to place online job advertisements, improve the quality of recruiter's candidate search, and to acquire information about skills and competencies of individuals (Colucci et al., 2003, Malinowski et al., 2006, Sharon, 2011). On the other hand, job-seekers use them to publish their profiles (Lee and Brusilovsky, 2007), and to search for job vacancies that satisfy their preferences and qualifications (Yi et al., 2007). In 2003, it was reported that 45% of job seekers confirmed having used the online job-portals as part of their job search (Lee, 2005). By 2006, a survey conducted by the Society for Human Resource Management showed that the number of job seekers who used online job-portals in their job search has raised to become 96% (Fazel-Zarandi and Fox, 2009). Consequently, a huge volume of job descriptions and candidate resumes are becoming available online (Parry and Tyson, 2008, Patil et al., 2017). However, the amount of available information is increasing steadily and thus the ability to process and track a large number of applications in a fast and cost-effective manner requires a huge cognitive effort (Lang et al., 2011, Barber, 2006). By reviewing state-of-the-art online recruitment systems, we can clearly find that they have employed different techniques and approaches to cope with this information overload problem. In the following section, we provide more details about these techniques and approaches.

## 2.2 Approaches/ Techniques Utilized by Online Recruitment Systems

Many approaches and techniques have been proposed for addressing the e-recruitment challenges (Lee, 2007b, Faliagka et al., 2011). In this context, some approaches attempt to overcome issues associated with the matching process between candidate resumes and their corresponding job offers, while others attempt to classify resumes and job posts prior to starting the matching process. In this section, we classify these techniques and discuss the major drawbacks and limitations that are associated with each technique.

### A. Information Extraction-based Techniques

A group of e-recruitment systems have employed Information Extraction (IE) techniques in an attempt to solve the resumes and job posts information overload problem. Example of these techniques are machine learning (Kessler et al., 2007, Yu et al., 2005), Natural Language Processing (NLP) (Amdouni and abdessalem Karaa, 2010, Kmail et al., 2015b), and knowledge management (Feldman and Sanger, 2007). Among the systems that employ IE techniques are (Finn and Kushmerick, 2004, Kessler et al., 2007). These systems utilize machine learning algorithms (Support Vector Machines (SVM) (Kessler et al., 2007) and SVM using Sequential Minimal optimization (SMO) (Finn and Kushmerick, 2004)) in order to annotate segments of resumes with the appropriate features and topics, taking the advantage of the resume's contextual structure where related information units usually occur in the same textual segments. However, the main drawback of these approaches is that a large fraction of the produced results suffer from low precision since the information extraction process passes through two loosely-coupled stages. In addition, the time needed to pre-process and post-process job posts (in order to minimize the error and maximize the classification accuracy) is huge (as detailed later in Chapter 5). In the work that is presented in (Yu et al., 2005), the system has been built based on employing Hidden Markov Model (HMM) and Support Vector Machine (SVM) classification algorithms in order to annotate segments of resumes with the appropriate occupational categories. Accordingly, the resumes pass through two layers. In the first layer, the HMM is applied to segment the entire resume into consecutive blocks where each block is annotated with a category of general information label such as Personal Information, Education, and Research Experience. After that, in the second layer both the HMM and SVM are applied in order to extract the detailed information from the blocks that have been labeled with Education and Personal Information respectively instead of searching globally in the entire resume. To evaluate the proposed system, an experimental instantiation is conducted on a data set of 1200 resumes. Although the system shows good precision (81.71%), and recall ratios (71.34%), the proposed approach suffers from error propagation since the information extraction process passes through two loosely-coupled stages. In the context of our work, we define these measures as follows:

Precision (P): is the Percentage Difference between the manually assigned relevance scores and those automatically generated by the system.

Recall (R): is the fraction of the resumes that are relevant to the job post that are successfully retrieved.

Amdouni and his colleagues (Amdouni and abdessalem Karaa, 2010) propose an approach based on NLP techniques to model the semantic content of unstructured resumes which are in different styles and formats (DOC, PDF, RTD, etc.). To carry out the information extraction process, the authors employ the information extraction tool **ANNIE plugin** (**A** **N**early-**N**ew **IE** system) which consists of the following components:

- **Sentence Splitter:** this module is used to identify and annotate the beginning and end boundaries of each sentence.
- **Tokenizer:** after identifying sentences in the processed resumes, the ANNIE Tokenizer splits the text into very simple tokens for example (punctuation, numbers, symbols and different types).

- **Part Of Speech (POS)Tagger:** each token is assigned to its part of speech tag such as verb, noun, adjective, etc. To accomplish this task, GATE uses the Brill-style POS tagger, this component produces a tag to each word or symbol.
- **Gazetteer:** it is a system of lexicons **(**user defined dictionary) that creates annotation to provide information about entities such as persons, organizations, job titles, list of cities name, time identifiers etc.
- **Named Entity Transducer:** The named entity recognition is the most important task in the information extraction process. During this step, each token is assigned an annotation based on a set of pre-defined annotations which are defined in Jape rules and Gazetteer lists.
- **OrthoMatcher:** this module performs entity tracking, by recognizing relations between entities. Furthermore, it assigns annotations to the unclassified tokens, based on the co-reference with existing tokens.

Finally, during the last step, they generate an XML file containing all annotations after cleaning GATE output by removing unnecessary tags. To validate the proposed process, an empirical study is conducted over a CV corpus that contains 150 CVs in different domains from a Tunisian recruitment firm. Although The results of the evaluation phase were satisfactory, the proposed module didn’t take into consideration the job posts analysis to be further used to automatically match job offers to CVs. Additionally, since a small number of resumes is used to validate the effectiveness of the proposed system, we believe that the system’s efficiency will degrade when handling real-world recruitment datasets that comprise hundreds of resumes and job offers.

### B. Automatic Resumes to Job Posts Matching Models

Many approaches/ techniques have been utilized by online recruitment systems to assist employers in screening out irrelevant resumes. Examples of these techniques are Boolean-based approaches (Belkin and Croft, 1992, Gebser et al., 2009), relevance based models (Yi et al., 2007) and Semantics-based techniques (Mochol et al., 2007, Fazel-Zarandi and Fox, 2009, Senthil Kumaran and Sankar, 2013, Hauff and Gousios, 2015, Kmail et al., 2015b, Martinez-Gil et al., 2016). For instance, Martinez-Gil et al. (Martinez-Gil et al., 2016) propose an approach for the automatic matching, learning and efficient querying of information from the HR domain. To carry out the matching process, the proposed approach exploits DISCO[2], ISCO[3] and ISCED[4] taxonomies to achieve better matching results than traditional techniques that simply look for overlapping keywords between the content of job posts and the applicant's resume (Al-Otaibi and Ykhlef, 2012b). The proposed system starts by utilizing the above mentioned taxonomies in an attempt to calculate the transformation cost (i.e. insertions, deletions or substitutions) of a given applicant's profile into a requested job post, so that profiles with higher transformation cost will be ranked lower than those with lower cost. After producing the matching results, automatic reports will be delivered to the applicants who have high transformation cost in order to help them to assess their weaknesses and strengths when applying for similar occupations later on. While the authors claim to be able to achieve better matching results, it is not clear how this is achieved, especially given the fact that existing ontologies suffer from missing background knowledge problem. In addition, they may suffer from

[2] http://disco-tools.eu/disco2_portal/
[3] http://www.ilo.org/public/english/bureau/stat/isco/isco08/index.htm
[4] http://www.uis.unesco.org/Education/Pages/international-standard-classification-of-education.aspx

contradictory semantic relations problem since the same concept might be defined in those taxonomies. Another drawback is the high computational time needed for delivering each result.

Hauff and Gousios (Hauff and Gousios, 2015) propose an approach to match job advertisements to developers based on the information available on their GitHub[5] profiles. In this context, a DBPedia Ontology (Auer et al., 2007) is utilized to extract relevant information from job advertisements and from ReadMe files of the user's GitHub projects in order to match users to jobs. We can summarize the main modules of the proposed approach:

- A module to extract candidate concepts from job advertisements and GitHub developer profiles using DBPedia Ontology.
- A module for weighting concepts using TF-IDF weighting scheme which gives a low weight to concepts that are not very informative (i.e. appear in many documents).
- A module to match job descriptions and GitHub developer profiles.

According to the authors, to validate the proposed system's effectiveness, experiments are carried out on a dataset that consists of 5,000 job advertisements and 1,000 GitHub user's profiles. Even though the authors claim that the produced results are satisfying in matching job offers with the user's profiles, the information was extracted only from ReadMe files (i.e. limited to code elements of a particular programming language) ignoring the other requirements that may be required in the job advertisements such as degree of education in a specific field.

[5] https://github.com/

MatchingSem (Kmail et al., 2015a, Kmail et al., 2015b) is a semantics-based automatic recruitment system that employs multiple semantic resources and concepts relatedness measures techniques to match resumes with their corresponding job offers. The proposed system first attempts to extract lists of candidate concepts from both resumes and job posts. After that, the lists of candidate concepts are submitted to an existing semantic resources to construct semantic networks by deriving the semantic relatedness between them. When a concept is not defined in the exploited semantic resources, statistical-based concept-relatedness measures are used to enrich the constructed semantic networks. Finally, the resulting networks are automatically matched and different relevance scores are produced by the system.

To measure the effectiveness of the proposed system, the authors have instantiated and used - in a precision-recall based empirical framework - a data set of 500 resumes and ten different job posts. Although the proposed system produced satisfactory matching results, it suffers from low precision and recall ratios for job posts that require specific years of experience. In addition to the huge cost (run time complexity) of the matching process (Zaroor et al., 2017).

**C. Classification of Job Posts based on their Corresponding Occupational Categories**

As reported in (Javed et al., 2015), in the online recruitment domain, classification of large datasets consisting of hundreds of thousands of job posts and resumes to their corresponding occupational categories is important to facilitate and reduce the huge cost (run time complexity) of the matching process, and improves the applicant/employer search and recommendation of jobs. To design a large-scale job title classification system, many systems leverage standard classification schemes (Clyde et al., 1995, Zaroor et al.,

2017) and machine learning algorithms (Javed et al., 2015). For example, Clyde et al. (Clyde et al., 1995) proposed JobDiSC classification system, the goal of the proposed system is to classify job openings automatically by employing a standard classification scheme called Dictionary of Occupational Titles (DOT)[6]. Accordingly, JobDiSC has 3 main modules:

- Parser-Analyzer: which creates an unclassified job opening for each job listings captured from electronic forms prepared by employers.
- Learning System: The learning system automatically generates classification rules from a set of pre-classified job openings
- Classifier: In this module, the classifier utilizes the rules generated by the learning system to classify new job openings. In the context of this process, the classifier attempts to assign one or more class for each job opening depending on its confidence level for any potential class assigned to it. If the classifier is able to assign a job opening to at least one class, the job opening becomes a classified job opening; otherwise it becomes an unclassifiable job opening and a human expert will later provide a manual classification.

To evaluate JobDiSC system, the classifier is initially trained with a set of pre-classified job openings of about 8000. The results of experiments show high classification accuracy (89%). However, the main drawback of this system is that DOT's usefulness has waned since it doesn't cover the occupational information that is more relevant to the modern workplace (O*NET, 2018a).

[6] https://occupationalinfo.org/

On the other hand, Javed and his colleagues (Javed et al., 2015) propose Carotene which is a machine learning-based semi-supervised job title classification system. Carotene has a classifier architecture composed of a Support Vector Machine- k Nearest Neighbor (SVM-kNN) cascade, as well as a clustering component that is used in taxonomy discovery. The proposed system starts by analyzing and pre-processing job titles by removing extraneous markup characters, stop-words and noise. Then, machine learning algorithms are utilized to classify job posts to a major occupational categories obtained from Standard Occupational Classification (SOC). After that, the kNN classifier is employed to assigns the most relevant minor occupational category obtained from the job title set which discovered by the clustering component.

An Experimental instantiation of the proposed system has been installed to validate its accuracy in the classification process. Although the multi-stage classifiers have shown good classification results, there are several limitations and drawbacks to the proposed system. The first disadvantage is the expense of data acquisition for training. With many thousands of groups of job titles, manually classifying large amounts of job title data becomes prohibitively expensive. A second disadvantage is its deficiency of corrigibility i.e. once a classification error has been discovered or a new example has been added to a class, the main alternative to enhance the framework is to retrain the entire classifier with the new sample added to the correct class in the training set (Neculoiu et al., 2016).

**D. Recommender System Techniques**

As stated in (Lee and Brusilovsky, 2007), the recommender system techniques can be utilized in order to tackle the problem of information overload by prioritize the delivery of information for job seekers based on their learned qualifications and preferences. As such, the recommender system approaches are classified into four main categories as reported in (Al-Otaibi and Ykhlef, 2012b, Hong et al., 2013):

- Collaborative filtering Recommendation (CFR) (Huang et al., 2007, Adomavicius and Tuzhilin, 2005, Bobadilla et al., 2011, Luo et al., 2013): also known as the user-to-user correlation recommendation. In this recommendation, the system finds applicants who have the same job interest (taste) with the target applicant and recommends jobs based on what the similar applicants prefer.
- Contend-based filtering Recommendation (CBR) (Pazzani and Billsus, 2007, Lops et al., 2011, Shalaby et al., 2018): this method suggests job posts that have similar content information to the corresponding applicants.
- Knowledge-based Recommendation (KBR) (Burke, 2002, Gupta and Garg, 2014): the principle of a KBR is to suggest jobs based on inferences about applicant's needs and preferences based on rules and patterns obtained from the functional knowledge of how a specific job meets the requirement of a particular applicant.
- Hybrid approaches Recommendation (HyR) (Burke, 2007, Lee and Brusilovsky, 2007, Hong et al., 2013): all the above mentioned recommendation approaches have their own drawbacks and limitations. To overcome these limitations, HyR recommendation attempts to combine these approaches to get better performance.

An example of recommender systems is Proactive (Lee and Brusilovsky, 2007) which is an adaptive job recommender system that provides four different interfaces to capture the job seeker's tastes and preferences. First, the most recent jobs are presented to the job seekers. Then, the job seekers can search for jobs via an advanced interface or they can specify their preferences, like location, the desired salary etc. After receiving the applicant information about the preferred jobs and transition history in the advance search, the recommendations of jobs are made to meet the applicant's preferences using CBR and KBR approaches. Although the proposed system utilized a Hybrid Recommendation approach to enhance the effectiveness of recommender systems, it suffers from several limitations. For instance, it provides a One Way Recommendation (OWR) i.e. it doesn't take into consideration to recommend candidates to potential employers. Furthermore, Proactive requires an explicit feedback in order to give an accurate recommendation to the job seekers.

iHR is another job recommender system proposed by Hong et al (Hong et al., 2013). The system clusters candidates according to their activity within the system into three major groups:

- Proactive users have clear image of the work they want to do and proactively search for a vocation.
- Passive users have only an ambiguous idea and therefore retain a more passive attitude
- Moderate users are neither particularly active nor passive.

As a result, iHR has the capability of choosing the appropriate recommendation approaches according to user's characteristics and suitability. For instance, for proactive users, the system uses a CBR approach in the same manner as with Proactive system. As for the passive users, the system uses a CFR approach, and for the moderate users the system uses a hybrid approach.

To evaluate iHR effectiveness, the authors collected personnel information of 100 job-seekers. The personnel information consists of user's personal information, login information, and historical behaviors. Although iHR addressed the challenges caused by employing a single recommendation approach in a Job Recommender System (JRS), it suffers from several limitations. As Proactive recommender system, iHR provides a one-way recommendation, in addition to the data sparsity problem. Furthermore, as (Xie et al., 2014, Shalaby et al., 2018) reported since iHR depends on user's activity and history behavior, it may suffer from privacy-leaking problems especially when an external personnel participates in the construction of recommendation systems.

The following table summarizes the comparative analysis that we have introduced in the previous sections.

**Table 1. Classification of the studied online recruitment systems**

| Index | System/Approach | Category | Goal of the system | Implementation techniques/approaches | Produced output |
|---|---|---|---|---|---|
| 1 | Resume information extraction with cascaded hybrid model | Information Extraction-based Techniques | annotate segments of resumes with the appropriate features and topics | Hidden Markov Model (HMM) and Support Vector Machine (SVM) classification algorithms | Semi-structured resumes |
| 2 | Web-based recruiting | Information Extraction-based Techniques | model the semantic content of unstructured resumes | NLP Techniques | Semi-structured resumes |
| 3 | A smart approach for matching, learning and querying information from the human resources domain. | Automatic Resumes to Job Posts Matching Models | Matching resumes to job offers | Semantics based methods | List of qualified applicants |
| 4 | Matching GitHub developer profiles to job advertisements | Automatic Resumes to Job Posts Matching Models | Matching job advertisements to developers' GitHub profiles | Semantics based methods | List of qualified applicants |
| 5 | MatchingSem | Automatic Resumes to Job Posts Matching Models | Matching resumes to job offers | Semantics based methods and statistical-based techniques | List of qualified applicants |

| | | | | | |
|---|---|---|---|---|---|
| 6 | JobDiSC | Classification of Job Posts based on their Corresponding Occupational Categories | Routing job posts to their corresponding occupational categories | Standard classification scheme (DOT) | Classified job posts |
| 7 | Carotene | Classification of Job Posts based on their Corresponding Occupational Categories | Routing job posts and advertisements to their corresponding occupational categories | Machine learning algorithms | Classified job posts |
| 8 | Proactive | Recommender System Techniques | Pushing job recommendations that meet the applicant's preferences | CBR and KBR approaches | List of recommended job posts to job seekers |
| 9 | iHR | Recommender System Techniques | Pushing job recommendations that meet the applicant's preferences | CBR approach<br>CFR approach<br>hybrid approach | List of recommended job posts to job seekers |

## 2.3 Summary

The aim of this chapter was to introduce a literature review about the online recruitment systems. We have explained the approaches/ techniques utilized by online recruitment systems including information extraction techniques, automatic resumes to job posts matching models, classification of job posts and resumes based on their occupational categories and recommender system techniques. Then, we have conducted a comprehensive comparative analysis of these techniques/approaches and highlighted their strengths and weaknesses.

# 3. Conceptual Classification and Ranking of Applicant Resumes and Their Corresponding Job Posts

## 3.1 Introduction

In this chapter, we introduce our proposed online recruitment approach. First, we present a general overview of our proposed online recruitment system. Then, we clarify the overall architecture of the proposed system. Finally, we summarize this chapter in section 3.3.

## 3.2 General Overview of the Proposed System

In this section, we present a general overview of our proposed system. In our research work, we utilize an integrated knowledge base for carrying out the classification task. Furthermore, we combine statistical-based concept-relatedness measures and the integrated knowledge base to automatically generate user profiles to be further used for recommending jobs to job seekers. The proposed system comprises three major components (Conceptual Classification, Automatic Profile Generation and Job Recommendation). In the next sections, we present the details of each component.

### *3.2.1 Conceptual Classification Component*

We have constructed the first component (named as JRC Job Resume Classifier) of our proposed online automatic recruitment system based on employing an integrated knowledge base to efficiently match between candidate resumes and job offers. Figure 2 describes the overall architecture of the Conceptual Classification Component.

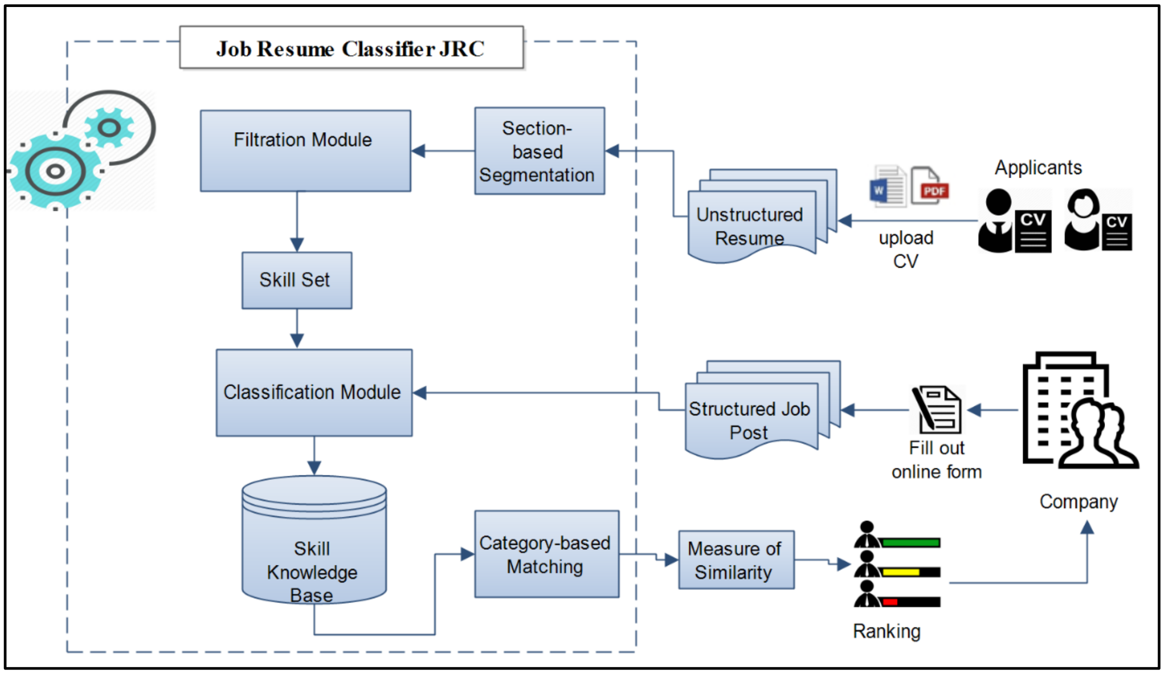


**Figure 2. General architecture of the proposed conceptual classification component**

As shown in Figure 2, the first component of our system comprises several modules that are organized as follows:

First, a ***Section-based Segmentation module*** is used to extract a list of candidate matching concepts, in addition to information such as personal, education, experience and applicant's employment history. Next, the ***Filtration module*** refines the concept lists by removing insignificant terms that don't contribute in the matching process. The third module of the proposed system, namely the ***Classification Module*** takes a *set of skills* extracted from both resumes and job posts as input in order to classify them under their corresponding occupational categories. At this step, we exploit an integrated occupational categories knowledge base which combines two main classification schemes: DICE and O*NET. Then, the ***Category-based Matching module*** takes the lists of skills from both resumes and job posts to construct semantic networks by deriving the semantic relatedness between their concepts in the same fashion as presented in (Kmail et al., 2015b). Finally, the matching algorithm takes the semantic networks as input - as long as they are in the same space - and produces the measures of semantic closeness between them as an output.

### *3.2.2 Automatic Profile Generation Component*

In this component, we utilize the extracted information from applicants' resumes to automatically generate profiles to support routing, management, as well as recommendation of jobs to their relevant candidate job seekers.

By doing this, we overcome many issues associated with the manual and tedious task of filling resume forms as listed below.

1) The process of creating, updating and maintaining the online profile requires applicants to perform a lot of manual operations, as a result, many applicants find it difficult, time consuming and tedious task to fill out such forms and therefore, they may leave parts of their profiles incomplete.
2) When employers seek to hire applicants, they may not be able to find the best candidates for a desired job post. This is mainly because the online profiles of a potential job candidate may not accurately reflect all their qualifications and skills.
3) In general, job seekers prefer to create their resumes in a way that they meet with their design preferences. In addition, job seekers find it difficult to adhere to the guidelines and rules of filling application forms manually. As such, job seekers often prefer to create their own (.doc) or (.pdf) resumes instead of filling application forms.
4) Generally, job seekers have to fill application forms according to the rules and settings that are prepared by the hiring agencies. Therefore, this will require

applicants to fill form that belongs to each and every hiring agency which is a very tedious task.

Figure 3 depicts the overall architecture of the proposed system after integrating the new components for the automatic profile generation.

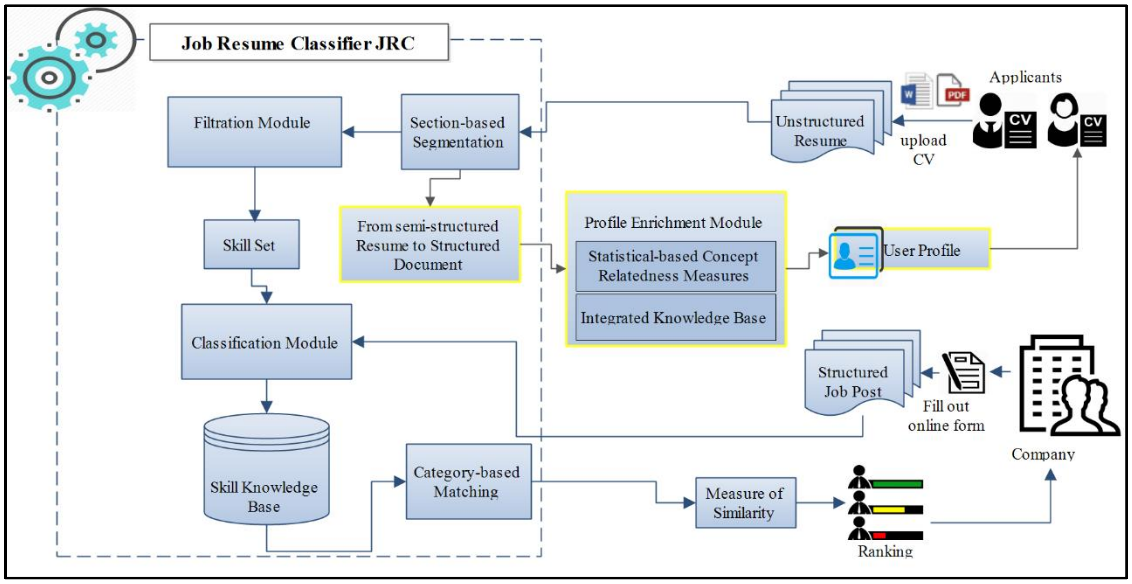


**Figure 3. General architecture of the proposed automatic profile generation component**

As shown in Figure 3, the new component - ***From Semi-Structured Resume to Structured Document*** - is integrated within the structure of our proposed system. This component is exploited in order to convert resumes from semi-structured formats into structured formats. As a result, a user profile that contains Personal information (Applicant name, birth of date, Nationality, Marital Status, Mobile Number, Email, Language, etc.), Educational Background, Experience Field, and Employment History, in addition to the Skills is automatically created. After that, skills that are extracted from each applicant's resume are then submitted to the ***Profile Enrichment Module*** wherein we utilize Statistical-based Concept Relatedness Measures (Hiring Solved (HS) dataset) (SOLVED, 2018) and our Integrated Knowledge base to enrich the applicant's profile with additional skills that aren't included in the applicant resume to increase the precision of creating applicant's profiles. To validate the effectiveness of this module, experimental validations were conducted by comparing the automatic scores generated by our approach with the users' evaluation for their automatic profiles. Chapter 5 provides more details on the conducted experiments.

### *3.2.3 Job Recommendation Component*

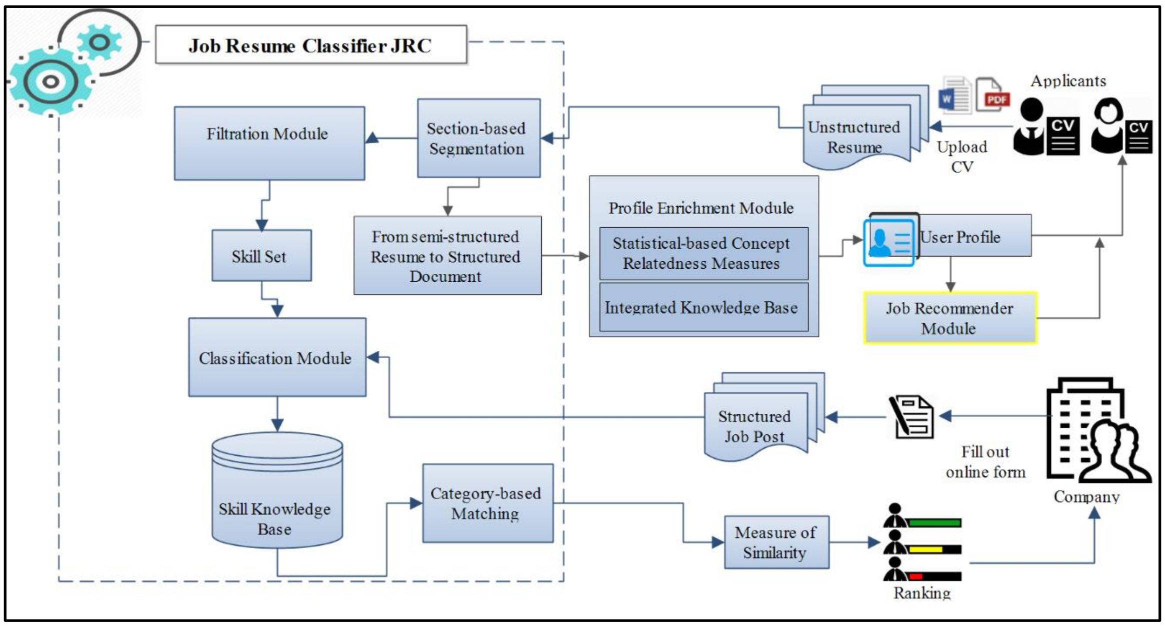


**Figure 4. General architecture of the proposed system job recommendation component**

Traditionally, the focus of online recruitment systems has been oriented towards assisting employers in finding the best candidates for their job requirements. In this context, applicants are normally given less attention because of ignoring to recommend job offers that best match their skills and preferences. To overcome this drawback, we have integrated a new component called ***Job Recommender Component*** that exploits the user's profile in order to push job post notifications that satisfy job seekers qualifications and skills. For instance, when an applicant with "php and html" skills seeks a job post, only job posts that fall under "Web Development" category will be considered as potential candidates for that applicant. Experimental instantiation of the proposed system has been installed to demonstrate the effectiveness of this component. We provide more details on the conducted experiments in chapter 5.

### 3.3 Summary

Our goal in this chapter was to provide a general overview of our proposed online recruitment system and to clarify the overall architecture of the proposed system. Also, we elaborated that our proposed system has three main components (Conceptual Classification, Automatic Profile Generation and Job Recommendation). In addition, we clarified the role for each component in assisting the applicant and the employer throughout the recruitment process.

# 4. Detailed Steps of the Proposed Online Recruitment System

In this chapter, we present the implementation details of our proposed online recruitment system. First, we present the details that pertain to the first version of the system's prototype in section 4.1. Then, in section 4.2, we describe the updated version of the system's prototype and detail the features of the new components. Finally, we summarize this chapter in section 4.3.

## 4.1 Development Details of the Conceptual Classification Component

During this phase, we have implemented the first prototype of our proposed automatic recruitment system through incorporating an integrated occupational categories knowledge base which combines two main classification schemes: DICE (Kolakowski, 2018) and O*NET(O*NET, 2018b). The knowledge base is exploited to classify both resumes and job posts according to the occupational categories that they belong to. The prototype system comprised several modules as detailed below:

### *4.1.1 Section-Based Resume Segmentation Module*

When users upload their resumes (as unstructured documents), this module is employed to automatically extract important resume segments such as: Education, Experience and other employment information such *Company name*, *Applicant's Role* in the company, *Date of designation*, *Date of resignation* and *Loyalty*. To do this, several NLP steps and rule-based regular expressions are utilized. As detailed in (Kmail et al., 2015a), the NLP steps are:

- document splitting
- n-gram content tokenization
- stop words removal
- Part-Of-Speech (POS)Tagging

- Named Entity Recognition (NER)

For each resume, we divide its textual content into segments in order to process each paragraph separately. Then, each segment is split into uni, bi, and tri gram tokens and we remove tokens (stop words and token with low tf.idf scores) that appear to be of little value in the classification and matching process. After that, we utilize the StanfordCoreNLP POSTagger (Manning et al., 2014) to assign the appropriate part of speech category for each token. Finally, we employ the NER to map tokens into categories such as names of persons, countries and locations. The following example clarifies the process of resume segmentation:

**Example 1:** Resume Segmentation

- Sample of a job seeker's resume (CV1**):**

| |
|---|
| I have 2 years of experience as a web developer.<br>And I have the following skills: PHP, HTML, CSS, JQuery, Ajax, android, Adobe Photoshop, Adobe Illustrator.<br>**Education**:<br>B.Sc. in CS.<br>**Employment Details**<br>I worked as web developer in ASAL Company from 2014 to 2016. |

In this example, we convert the CV1 from unstructured document into a section-based (semi-structured) resume as follows:

```
<Applicantdata>
<Experience>
<Years>2</Years>
<Field> web developer </Field>
</Experience>
<Education>
<Degree> B.Sc </Degree>
<Field> CS </Field>
```

```
</Education>
<EmploymentHistory>
<role> web developer </role>
<companyName> ASAL Company </companyName>
<FromDate>2014</FromDate>
<ToDate>2016</ToDate>
<loyalty> 2</loyalty>
</EmploymentHistory>
<skills> web, developer, PHP, HTML, CSS, JQuery, Ajax, android,
Adobe Photoshop, Adobe Illustrator, ASAL, Company, experience
</skills>

</Applicantdata>
```


Once unstructured resumes are converted into semi-structured document, we employ a filtration module to identify, extract, and filter candidate matching concepts. Table 2 shows the results of this step.

**Table 2. Result of the filtration module**

| Candidate terms extracted from resume | Filtered Concepts List from resume |
|---|---|
| Web | Web |
| Developer | PHP |
| PHP | HTML |
| HTML | CSS |
| CSS | JQuery |
| JQuery | Ajax |
| Ajax | android |
| android | Adobe Photoshop |
| Adobe Photoshop | Adobe Illustrator |
| Adobe Illustrator | |
| ASAL | |
| Company | |
| skills | |
| experience | |

As shown in Table 2, concepts that belong to a list of pre-defined terms (e.g. contact info, address, birth date, country name) or have low tf-idf weights are removed from the lists of candidate concepts. The outcome of this step is a list of filtered skills, referred to henceforth as the *skills set*.

### 4.1.2 *Job Post and Resume Classification Module*

In order to classify both resumes and job posts, we utilize an integrated knowledge-base which combines Dice skills center (henceforth stated as DICE) (Kolakowski, 2018) and a standardized hierarchy of occupation categories known as the Occupational Information Network (O*NET) (henceforth stated as O*NET)(O*NET, 2018b). In this context, we use DICE to classify skills that belong to Information and Communication Technologies (ICT) and Economy fields because we empirically found that O*NET is not scalable enough for our classification needs. For instance, "android" is classified under "Software Developers\ Applications" occupational category and this category is very broad. Furthermore, some skill acronyms are not classified correctly in O*NET. However, and on the contrary of Dice, O*NET is able to better classify skills that are related to the Medical and Artistic fields. Table 3 shows a comparative analysis between Dice and O*NET classifications.

**Table 3. Comparison between DICE and O*NET classifications**

| | O*NET | | DICE | |
|---|---|---|---|---|
| | **Skill** | **Classification Result** | **Skill** | **Classification Result** |
| **Submission of Acronyms** | JPA | Accountants | JPA | Software Development |
| | JCA | Nursing Assistants | JCA | Software Development |
| | J2ME | Gem and Diamond Workers | J2ME | I.T. Administration/Technical Support |
| | **Skill** | **Classification Result** | **skill** | **Classification Result** |
| **Correctness in Classification** | xcode | Coaches and Scouts | xcode | Software Development/ Mobile development |
| | Radiography | Radiologic Technicians | Radiography | NOT CLASSIFIED |
| | Medical analysis | Medical and clinical Laboratory | Medical analysis | NOT CLASSIFIED |

As shown in Table 3, some skill acronyms are not recognized by O*NET, and accordingly they are not classified correctly. For instance, JPA which refers to "Java Persistence" is classified under "Accountants" category by O*NET. However, we can see that terms such as "Radiography" and "Medical analysis" are not classified in DICE, but classified correctly under “Radiologic Technicians” and “Medical and clinical Laboratory” categories in O*NET.

### *A. Skill-Based Resume Classification Module*

In this module, each skill in the skills set is sequentially submitted to the exploited knowledge base in order to obtain a list of weighted occupational categories that each skill belongs to. Occupational categories in the produced list are sorted – in a descending manner - based on their weights (as one skill may return zero, one, or more than one occupational category). For instance, as shown in Figure 5, when the skill “android” is submitted to the occupational categories knowledge base, “Software Development/ Mobile Development” occupational category is obtained first. After that “PHP, HTML” is submitted in the same manner as we did for the previous skill, and “Software Development/ Web development” occupational category is obtained as depicted in Figure 6. Then, using this procedure, a list of additional weighted categories is obtained and sorted according to their highest weight.

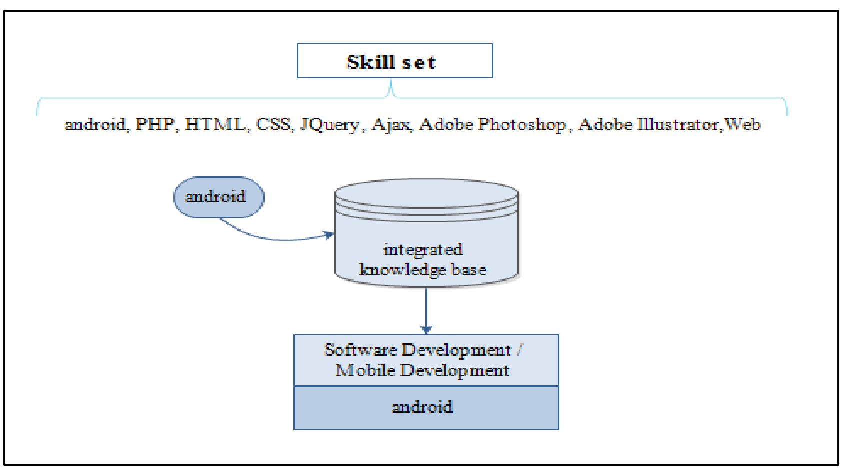


**Figure 5. Submitting "android" to the integrated knowledge base**

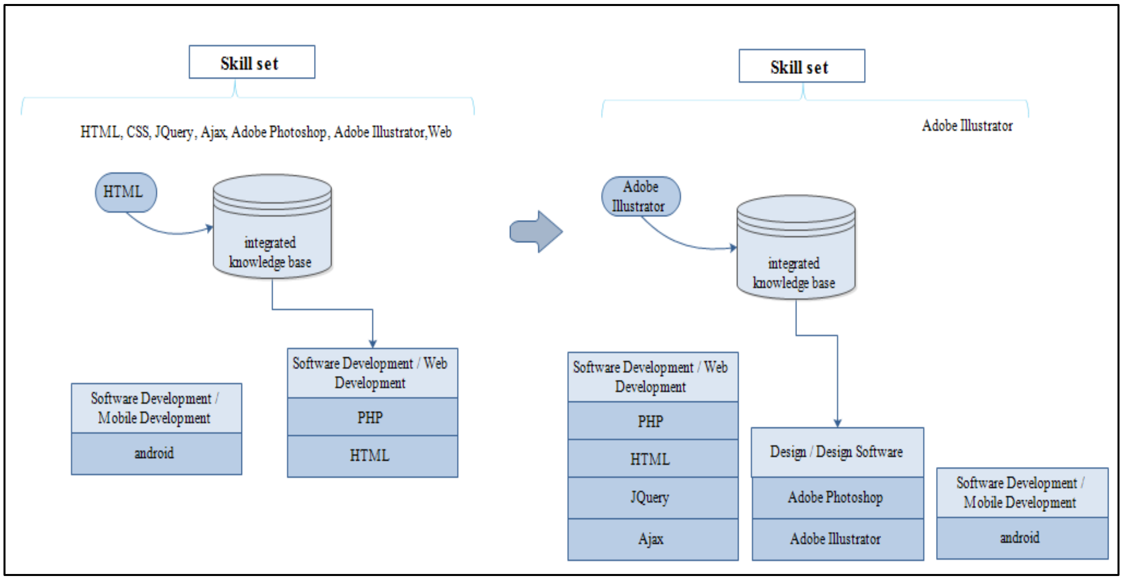


**Figure 6. List of obtained occupational categories for CV1**

To produce weights for the occupational categories, we use the following algorithm.

**Algorithm 1. Classifying Resumes According to their Corresponding Occupational Categories**

Input: skills [$s_1$, $s_2$, …,$s_n$]

Output: list of job categories sorted by the highest weight for a given resume

1: int weight=0;
2: answer $\leftarrow \langle\ \rangle$;
3: occupationalcategoriesList $\leftarrow \langle\ \rangle$;
4: **for each** skill $\in$ skills [$s_1$, $s_2$, …,$s_n$] **do**
5: occupationalcategoriesList $\leftarrow$ **GET_FROM_KB** (skill);
6: **while** occupationalcategoriesList $\neq$ NIL **do**
7: **if** category **Not IN** answer **then**
8: weight =1;
9: **ADD** (answer, $\langle category, weight \rangle$ );
10: **else**
11: weight++;
12: **Modify** (answer, $\langle category, weight \rangle$ );
13: **end if**
14: **end for**
15: **Return** answer;

In the used algorithm, skills are submitted to the integrated knowledge base respectively (Line 4). As a result, one occupational category is returned for each skill (Line 5). If the same occupational category is returned for more than one skill, the algorithm increases the weight for that particular occupational category, otherwise it sets its weight to 1. (Lines 8, 11 and 12). Finally, the algorithm returns a list of weighted occupational categories in the answer list (Line 15). Table 4 shows each occupational category assigned to its corresponding skills.

**Table 4. Skills to occupational categories mapping**

| Job category | skills |
|---|---|
| Software Development/ Mobile Development | Android |
| Design / Design Software | Adobe Photoshop, Adobe Illustrator |
| Software Development/ Web Development | CSS, html, php, Ajax, jquery |

### *B. Job Post Classification Module*

In the Job Post Classification module, we use both the job title and the required skills in each job post for classification purposes. First, the job post is pre-processed and filtered by removing noisy information such as: city names, country acronyms and state that appear in the job title or job details. After that, we use the integrated knowledge base to classify job posts in the same manner as we do for classifying resumes. In our work, we assign the following weights:

- Job Title=70%
- Required Skills=30%

as we have empirically found that the job titles are more significant than the required skills and lead to better matching results.

**Example 2:** Job Post Classification

- Sample of a job post (JP1**):**

| |
|---|
| **Job title**: JAVA J2EE Developer<br>**Required skills**: J2EE, Java EE, Struts 1, Hibernate, Tomcat 8, XML, LDAP, Eclipse, Ajax, Html |

First, the job title "JAVA J2EE Developer" is submitted to the exploited knowledge base, "Software Development/ Web Architecture" occupational category is obtained first. After that, each skill in the required skills set is submitted in the same manner as we did for the job title, and "Software Development/ Web development" occupational category is obtained when we submit "Ajax, HTML" and "Software Development/ Web Architecture" occupational category after submitting the rest of the required skills. Accordingly, this job post will be classified as "Software Development\ Web Architecture" with weight =94% and "Software Development/ Web development" with weight =6%. More examples on the results of this module are presented in Section 5.

### *4.1.3 Matching Resumes to their Corresponding Job Postings*

Inspired by the work developed by (Kmail et al., 2015b), we employ multiple semantic resources to derive the semantic aspects of resumes and job posts. These are WordNet ontology (Miller, 1995) and YAGO2 ontology (Hoffart et al., 2011). In addition, we utilize statistical concept-relatedness measures to further enrich the lists of extracted concepts from the job posts and resumes that weren't recognized by the used occupational categories knowledge base. Moreover, in order to increase the transparency and the effectiveness of the matching process, we have added an additional weighting parameter that is Loyalty parameter to the matching formula (Equation 1). By Loyalty, we mean the degree of devotion to the company that the applicant is working or worked in. It is important to point out that we have used a modified version of the candidate's Relevance Scoring (RS) formula that has been proposed in the Oracle Project Resource Management (Lavrenko and Croft, 2017) to assign relevance scores:

$$\boldsymbol{S} = \frac{|\{\boldsymbol{Sr}\}|}{|\{\boldsymbol{RSj}\}|} * \mathbf{50}\% + \frac{|\boldsymbol{Er}|}{|\{\boldsymbol{REj}\}|} * \mathbf{20}\% + \frac{|\{\boldsymbol{Xr}\}|}{|\{\boldsymbol{RXj}\}|} * \mathbf{20}\% + \frac{|\sum \boldsymbol{Yw}|}{|\sum \boldsymbol{Cw}|} * \mathbf{10}\% \quad \textbf{(1)}$$

Where:

- **S**: is the relevance score assigned between a job post and a resume.
- **Sr**: is the correspondences set of applicant's skills.
- **RSj**: are the required skills in the job post.
- **Er**: is the set of concepts that describe applicant's educational information.
- **REj**: are the concepts that represent the required educational information in the job post.
- **Xr**: is the set of concepts that describe applicant's experience information.
- **RXj**: are the concepts that represent the required experience information in the job post.
- **Yw**: is the total number of employment years.
- **Cw**: is the number of companies that the applicant worked in.

As shown in the formula, we have set the following weight:

Skills weight = 50%, Educational level weight = 20%, Job experience weight = 20% and Loyalty level weight= 10%.

In order to quantify the education parameters, as well as experience parameters, we give a weight for each field. For instance, we give a value for each educational degree (Diploma, Bachelor, Master, PhD).

$$Ed_Q = \frac{y_d}{x_d} \qquad \textbf{(2)}$$

Where $y_d$ is the weight for the degree $d$ in the applicant's resume and $x_d$ is the weight for the degree $d$ required in the job post. For example, if a job post requires a BSc degree and an applicant with a BSc degree applies for this job post; she/he will be considered a qualified applicant as he meets the educational requirement for the job post ($\frac{y_{BSc}}{x_{BSc}}$ =perfect match). However if the applicant has a Diploma degree she/he will be considered underqualified since ($\frac{y_{Di}}{x_{BSc}}$ =under qualified). If the applicant has a Master or a PhD degree, she/he will be considered overqualified for that job post since ($\frac{y_{MSc}}{x_{BSc}}$ or $\frac{y_{PhD}}{x_{BSc}}$ =over qualified). In the same fashion we quantify the experience parameters using the following formula:

$$Ex_Q = \frac{y_r}{x_j} \qquad \textbf{(3)}$$

Where $y_r$ is the years of experience the applicant has and $x_j$ represents the years of experience that are required in the job post.

- If $y_r = x_j$ the applicant will be considered as a qualified match.

- If $y_r < x_j$ the applicant will be judged as underqualified.

- If $y_r > x_j$ the applicant will be considered as overqualified.

Accordingly, assume JP be a job post with a set of requirements $(Ed_{JP}, S_{JP}, Ex_{JP})$ where,

- $Ed_{JP}$ : is the required educational degree
- $S_{JP}$ : is the list of skills, $\sum_{i=1}^{n} S_{JP_i}$
- $Ex_{JP}$ : is the required experience. It is important to mention that some for JPs, the employer specifies a number of years without specifying the field of experience (e.g. +4 years of experience), while for other JPs they specify the number of years in a specific field. For example: +2 years of experience in java development.

Let JS be an applicant who applies for JP with a set of qualifications $\left(Ed_{JS}, Ex_{JS}, S_{JS}\right)$ where $Ed_{JS}$ is the educational degree that JS has, $Ex_{JS}$ is the amount of experience that JS has, and $S_{JS}$ is a list of skills, $\sum_{i=1}^{n} S_{JS_i}$ . A qualified match denotes that a job seeker satisfies all the requirements for JP i.e. the score=100%

$$score = 20\% * E_{score} + 20\% * Ex_{score} + 50\% * Skill_{score} + 10\% * loyalty$$

where:

- $E_{score}$= $Ed_{JP} \cap Ed_{JS}$
- $Ex_{score}$= $Ex_{JP} \cap Ex_{JS}$

- $\text{Skill}_{\text{Score}} = S_{JP} \cap S_{JS}$

The following algorithm illustrates the process of finding the values of semantic closeness between the resumes and their corresponding job posts that fall under the same occupational categories.

**Algorithm 2. Finding the similarity between the resumes and their Corresponding job posts that fall under the same Occupational Categories**

```
Input: JP_list [JP1, JP2, …,JPn]
Output: Measure of similarity sorted by the highest value
similarity ←⟨ ⟩;
ResumeList ←⟨ ⟩;
for i←0; i <JP_list.length; i++
    ResumeList ← GET_Resume_List (JP_list[i].getCategory());
    for j←0; j < ResumeList.length; j++
        similarity← Relevance_Score (JP_list[i], ResumeList[j])
    end for
end for
Return similarity;
```

We would like to point out that the run-time complexity of the above mentioned Algorithm is O(n^2). In its current version, the matching process costs a similar time to other existing approaches. However, since we classify resumes and job posts before conducting the matching task, our system out-performed MatchingSem and other approaches (tf-idf scheme with/without classification) as will be explained in more details in (Chapter 5 - Experimental Evaluation).

## 4.2 Development Details of the Automatic Profile Generation and Recommendation Components

During this phase, we exploit the extracted information from applicants' resumes to automatically generate searchable user profiles to be further used for recommending jobs to job seekers. Accordingly, instead of requesting applicants to fill specific online forms prepared by employers, they will only have to upload their resumes and our system will automatically generate their profiles. In the following sections we present the details of the newly incorporated components to our system's prototype.

### *4.2.1 From Semi-Structured Resumes into Structured Profiles*

During this module, semi-structured resumes are converted to structured documents. As detailed in section 4.1.1 the unstructured resumes are converted into semi-structured format using NLP and features extraction techniques. After that, we convert the semi-structured versions of applicant resumes into structured profiles. In this scenario, employers can compare between applicant profiles easily, and to search for any resume using different search filters. For instance, they can search for applicants who have knowledge in "java programming language" or applicants who have 3 years of experience in the "Web development".

**Example 3:** Applicant's Profile Generation

- Sample of a job seeker's resume (CV2**):**

**Personal information**

My name is (sample)

Nationality: Palestinian

D.O.B: 21-09-1992

Marital Status: married and I have 3 children.

Email: Sample@sample.com

I have 5 years of experience as a java developer.

And I have the following skills: JSP, CSS, HTML, XML, Spring, Struts 1.x, java.

**Education**:

Master of Science in Computer Science in Computer Science

**Employment Details**

I worked as java developer in SAfa Company from 2012 to 2016.

As detailed earlier in section 4.2.1, CV2 will be converted into a semi-structured format as detailed below. Then, we utilize document parsing techniques in order to convert these semi-structured profiles into a searchable structured forms as shown below.

Semi-structured version of CV2


```
<Applicantdata>
<PersonalInfo>
<Name> sample </Name>
<Nationaliy> Palestinian</Nationaliy>
<DateOfBirth>21-09-1992</DateOfBirth>
<Marital Status>married </Marital Status>
<Email> Sample@sample.com</Email>
</PersonalInfo>
<Experience>
<Years>5</Years>
<Field> java developer </Field>
</Experience>
<Education>
<Degree>Master</Degree>
<Field> Computer Science </Field>
</Education>
<EmploymentHistory>
<role> java developer </role>
<companyName> SAfa Company
</companyName>
<FromDate>2012</FromDate>
<ToDate>2016</ToDate>
<loyalty> 4</loyalty>
</EmploymentHistory>
<skills> JSP, CSS, HTML, XML, Spring, Struts
1.x, java </skills>
</Applicantdata>
```


Structured version of CV2

**Personal Information**
Applicant's name: sample
Nationality: Palestinian
Date of birth: 21-09-1992
Marital Status: married
Email: Sample@sample.com

**Education**
Degree: Master in Computer Science

**Employment History**
**Work as** java developer **in** SAfa Company
**From** 2012 **to** 2016

**Experience**
5 **years of experience in** java

**Applicant Skills**
JSP, CSS, HTML, XML, Spring, Struts 1.x, java

### *4.2.2 Further Enrichment of the Applicant's Profile*

Some skills may not be explicitly mentioned in applicant resumes, for example, an applicant may mention "sql" among his/her skills set, and ignores "database" since it is a more generic term, or he/she may mention a generic term such as "Web Development" and doesn't explicitly state that the specific skills that are related to that generic skill. To address this issue, we propose to further enrich the skills set that an applicant has by automatically adding new skills obtained from our integrated skills knowledge base and Hiring Solved dataset (SOLVED, 2018) which defines a huge number of terms in the form of skills and the weights of the semantic relatedness between those skills (Kmail et al., 2015a). To carry out this step, first we find the weight for each skill in the skills set, in order to find the skills with the highest weight. After that, we use an automatic threshold value $v$ to automatically decide upon which skills should be submitted to the integrated knowledge base in order to retrieve a set of related them.

Let $\sum_{i=1}^{n} S_{JS_i}$ be a set of skills that the applicant has in his/her resume. In order to find the weight for each skills in the skills set we apply the below formula:

$$ws_{JSi} = Tf + \exp_{JSi} + corelatedness_{score} \qquad \textbf{(4)}$$

Where:

$ws_{JSi}$ : is the weight for skill $s_{JSi}$ .

$Tf$ : term frequency of skill $s_{JSi}$ .

$\exp_{JSi}$ : the amount of experience the applicant has in $s_{JSi}$ .

$corelatedness_{score}$ : represents the number of relations between each skill with the other skills in the skills set.

**Example 4:** Applicant’s Profile Enrichment

- Applicant’s skills extracted from CV2

**Applicant Skills**
JSP, CSS, HTML, XML, Spring, Struts 1.x, java

As mentioned before, we find the weight for each skill in the skills set. First, we find the co-relatedness score for each skill by submitting them to HS dataset as shown in Figure 7.

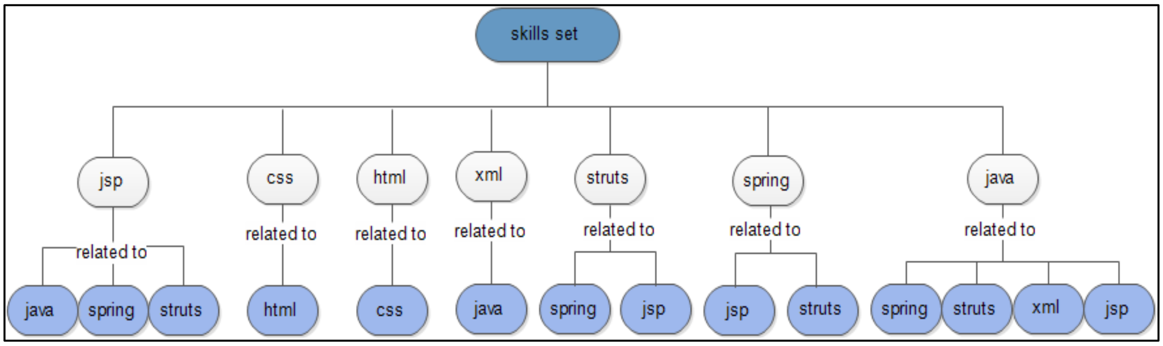


**Figure 7. Co-relatedness measures between skills**

As shown in Figure 7, after submitting the skills to HS dataset we find the co-relation score for the applicant's skills. For instance, "java" is related-to "spring, struts, xml and "jsp" and "css" is related-to "html". Accordingly, we consider the $corelatedness_{score}$ for "java" as the highest as it has more skills that are related to it in the skills set. Table 5 shows the weights for each skill in the skills set ( $ws_{JSi}$ ).

**Table 5. Weights for each skill in the skills set**

| skill | $corelatedness_{score}$ | TF | exp $_{JSi}$ | $ws_{JSi}$ |
|---|---|---|---|---|
| JSP | 0.75 | 1 | __ | 1.75 |
| CSS | 0.25 | 1 | __ | 1.25 |
| Html | 0.25 | 1 | __ | 1.25 |
| Xml | 0.25 | 1 | __ | 1.25 |
| Spring | 0.50 | 1 | __ | 1.50 |
| Struts | 0.50 | 1 | __ | 1.50 |
| Java | 1.00 | 2 | 5 | 8.00 |

As shown in Table 5, "Java" has the highest weight followed by "JSP". Accordingly, "Java" and "JSP" are submitted to our integrated knowledge base to obtain a new set of skills that are related to "java" and "JSP". Figure 8 shows the newly obtained set after submitting java and jsp to the integrated knowledge base.

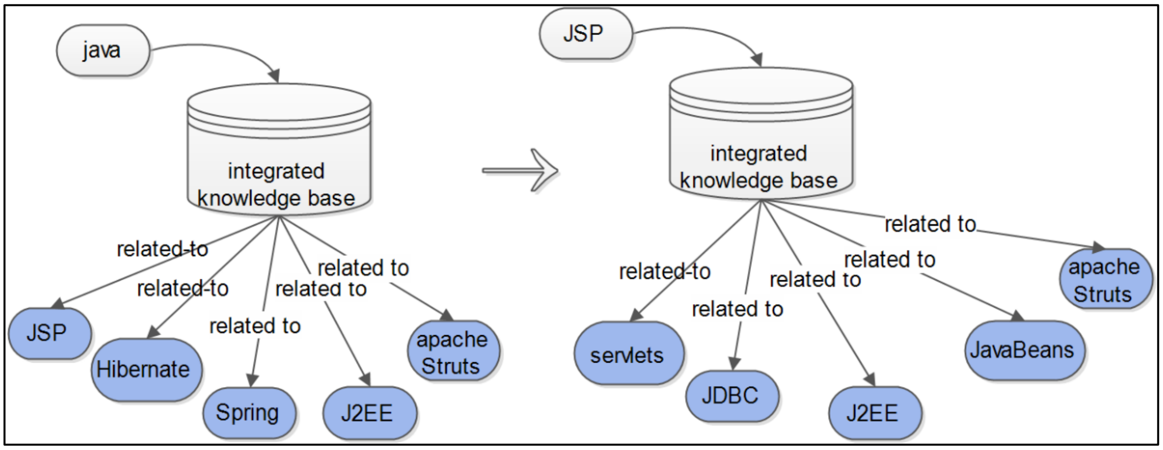


**Figure 7. New skills set obtained from integrated knowledge base**

Let $\sum_{j=1}^{m} S_{Ej}$ be the new set of skills obtained from our integrated knowledge base. To enrich the skill set of the applicant's resume $\sum_{i=1}^{n} S_{JS_i}$ with the skills of $\sum_{j=1}^{m} S_{Ej}$, we follow the following procedure:

- If $S_{Ej} \in \sum_{i=1}^{n} S_{JSi}$, this means that $S_{Ej}$ already exist in $\sum_{i=1}^{n} S_{JS_i}$, then we retain $S_{Ej}$ in the skills set.
- If $S_{Ej} \notin \sum_{i=1}^{n} S_{JSi}$, this means that $S_{Ej}$ does not exist in $\sum_{i=1}^{n} S_{JS_i}$, then we add $S_{Ej}$ to the skills set.

Accordingly, we update the applicant's profile with the skills from Example 4 as follows:

| |
|---|
| **Applicant Skills**<br>JSP, CSS, HTML, XML, Spring, Struts 1.x, java, J2EE,<br>Hibernate, Servlets, JavaBeans, JDBC |

### *4.2.3 Job Post Recommender Module*

In general, job recommendation systems are used to recommend job posts for applicants that suit their skills and qualifications based on employing a Boolean search method. However, techniques such as Boolean search method cannot be sufficient to realize the complexity of a person-job fit. As mentioned above, many applicants have incomplete online profiles due to the many manual operations required to complete their profiles. On the other hand, the online profile of a potential candidate does not accurately reflect all his/her qualifications and skills. To overcome these issues, we utilize the skills set from applicant profiles in order to assist them by recommending job offers that best match their skills and qualifications.

We summarize the steps of our proposed job recommendation modules as follows:

1) Applicant skills $\sum_{i=1}^{n} S_{JS_i}$ is extracted from the applicant's profile and then converted into a skills vector $V_{JSs}$ .

2) We convert all the required skills from the job posts that have the same space (i.e. occupational category as the applicant) to a set of job skill Vectors.

$$M_{n\times 1} = \begin{bmatrix} V_{JP1} \\ V_{JP2} \\ \\ . \\ V_{JPn} \end{bmatrix}$$

Each row represents a job post vector that contains a set of skills required for that job post.

3) The similarity is calculated between the applicant's skills vector $V_{JSs}$ and the set of job skill vectors $M_{n\times 1}$ as follows:

$$Sim_{n \times 1} = \begin{bmatrix} Sim1_{JP \times JS} \\ Sim2_{JP \times JS} \\ . \\ . \\ Simn_{JP \times JS} \end{bmatrix}$$

Where $Sim_{n \times 1}$ represents the similarity between $V_{JSs}$ and every job post in the set of job skill vectors.

4) Finally, the recommendation is generated by ranking the similarity scores to present the top-n (10 in our work) recommendations.

In order to clarify the above mentioned steps, we conduct the following example using the applicant's skills from Example 4 and a list of 4 posted job offers.

**Example 5: Job Post Recommendation**

- Applicant's skills from Example 4

| **Applicant Skills** |
|---|
| JSP, CSS, HTML, XML, Spring, Struts 1.x, java , J2EE, Hibernate, Servlets, JavaBeans, JDBC |

- Samples of job titles and the required skills

| | | |
|---|---|---|
| Job#1 | J2EE Developer | Java, Struts, Spring, HTML, MVC, TDD, CSS, JavaScript, J2EE, JDBC |
| Job#2 | Senior Java Developer | J2EE, EJB, Groovy, Spring and AJAX, Java |
| Job#3 | Web Developer | HTML, CSS, PHP, SQL and ASP.NET. |
| Job#4 | Front End Developer | HTML, XML, CSS, LESS, SASS, Grunt, Node.js |

Accordingly, $V_{JSs}$ = [JSP, CSS, HTML, XML, Spring, Struts 1.x, java , J2EE, Hibernate, Servlets, JavaBeans, JDBC].

$$M = \begin{bmatrix} \text{[Java, Struts, Spring, HTML, MVC, TDD, CSS, JavaScript, J2EE, JDBC]} \\ \text{[J2EE, EJB, Groovy, Spring, AJAX, Java]} \\ \text{[HTML, CSS, PHP, SQL, ASP.NET]} \\ \text{[HTML, XML, CSS, LESS, SASS, Grunt, Node.js]} \end{bmatrix}$$

Now we compute the similarity between the skills vector and each job vector in the matrix.

$$Sim = \begin{bmatrix} 0.58 \\ 0.25 \\ 0.17 \\ 0.25 \end{bmatrix}$$

This example aims to find job posts which best fit the qualification of applicant's profile. Based on the similarity measures, the 1$^{st}$ job post is the best candidate which fits the applicant's qualification, followed by 2$^{nd}$ and 4$^{th}$ job post. The 3$^{rd}$ job post is the least appropriate candidate for the applicant qualification.

## 4.3 Summary

The aim of this chapter was to present the methods and techniques that are used in our system. Also, we have demonstrated that the proposed system has three components. The first component incorporated four modules that mainly focused on converting unstructured resume into semi-structured format using *Section-based Segmentation Module*, then a list of candidate concepts is identified, extracted, and filtered using the *Filtration module* to be further used in the *Classification Module* to classify resumes and job posts. Finally, matching job posts and resumes that fall under the same occupational category. In the second component of the prototype system new modules were integrated to facilitate the profile generation for the applicants instead of manually creating profiles we utilize NLP and feature extraction techniques to convert the resumes from semi-structured resume into structured documents. While in the third component of the proposed system, we have integrated the "*Job recommender"* module in order to recommend job offers that best match applicant skills and preferences.

# 5. Experimental Evaluation

This chapter describes the experiments that we have carried out to evaluate the techniques of the proposed system. The evaluation process has been accomplished at three successive stages. In the first stage, we evaluate the effectiveness and the efficiency of the first component of the prototype system “Conceptual Classification Component” by comparing the results produced by this version of the system with one of the recently proposed online recruitment systems. Then, in the second stage, we evaluate the precision of the automatic profiles generated by our system. And finally, in the third stage, we validate the effectiveness of the proposed job recommender component and compare between the precision of the produced results when utilizing the enriched skills against when not utilizing them in the recommendation process. We have implemented the prototype of the proposed system using Java and JavaServer Pages (JSP) programming languages and conducted the experiments using a PC with core i5 CPU (2.1GHz) and (4 GB) RAM. The operating system is Windows 8.1.

The rest of this chapter is organized as follows. Section 5.1 presents the first stage of the experimental evaluation. The second stage of the Profile Generation evaluation is discussed in section 5.2. Section 5.3 presents the results of evaluating the job recommendation module. Concluding discussions on the conducted experiments are presented in Section 5.4.

## 5.1 Evaluating the Efficiency and the Effectiveness of JRC system

In this section, we present the experiments that we have carried out to evaluate the efficiency and the effectiveness of the first component of the prototype system. To accomplish this task, we have conducted a series of experiments on a dataset that consists of 2000 resumes downloaded from:

- Amrood (http://www.amrood.com/resumelisting/listallresume.htm )
- indeed (http://www.indeed.com/resumes )

and we used 10,000 different job posts obtained from monster (http://jobs.monster.com ) , shine (http://www.shine.com/job-search ) and careerbuilder (http://www.careerbuilder.com ). The manually constructed resumes dataset has a size of 68.3 MB of documents represented in different document formats such as (.pdf) and (.doc) and contains 71,700,480 words. However, we considered job posts as structured documents having the following segments: job title, job description, required skills, years of experience, required education qualifications and additional desired requirements.

In order to carry out the experiments, we analyzed the corpus of resumes and job offers through employing the feature extraction techniques (using NLP techniques and regular expressions) described in section 4.1.1. Then, we utilized the filtration module to refine the lists of candidate concepts. Next, we used the integrated knowledge base to classify the job offers and resumes with the appropriate occupational category. Finally, after constructing the semantic networks for job posts and resumes, we produce the measures of semantic closeness between them as long as they are in the same space.

We want to point out that, the experiments of our system's prototype show that the classification process for the resumes and job posts took 6 hours on average on a PC with dual-core CPU (2.1GHz) and (4GB) RAM.

### *A. Execution Time for Matching Resumes with Corresponding Job Post*

In this section, we compare the results produced by our system to those produced by: MatchingSem system (Kmail et al., 2015b) which is a semantics-based automatic recruitment system, tf-idf scheme without classification (henceforth stated as tf-idf/NC) and tf-idf scheme with classification (henceforth stated as tf-idf/WC). Figure 9 shows the run-time complexity of the matching process between them.

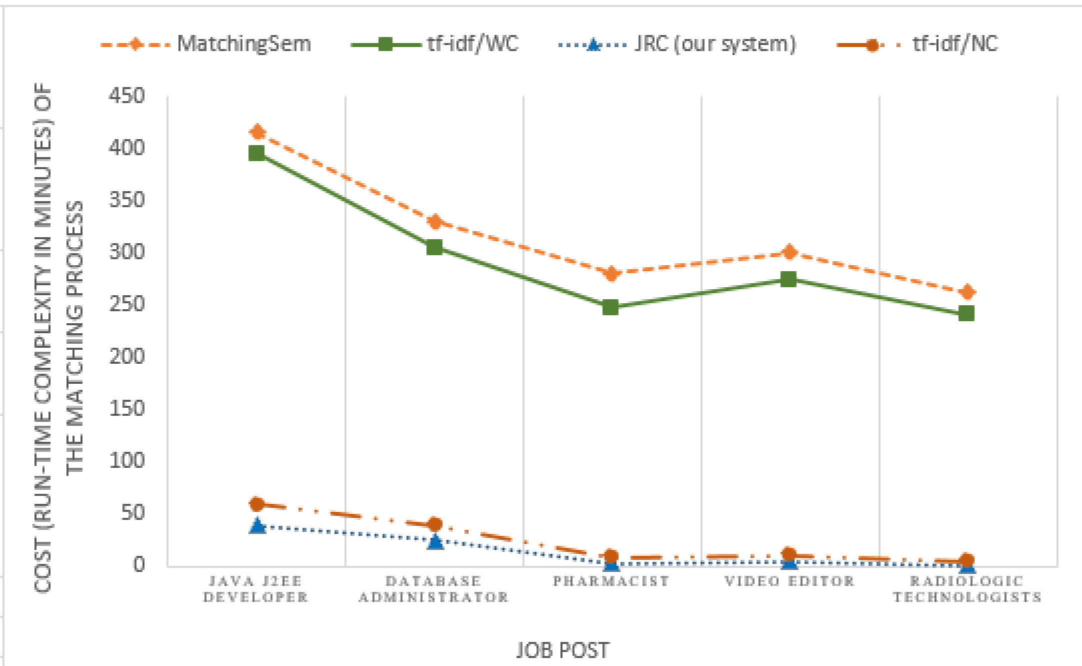


**Figure 8. Cost (run-time complexity in hours) of the matching process**

As shown in Figure 9, our system (JRC) was able to achieve higher performance results compared to the other approaches. This is due to the fact that, unlike MatchingSem and tf-idf/NC, we only match job posts with their corresponding resumes that fall under the same occupational category instead of searching globally in the entire space of resumes. For instance, “java j2ee Developer” job post costs 6 h and 55 min of execution time for finding the best candidate using MatchingSem and 6 h and 35 min using tf-idf/NC, while it only took 1 hour for tf-idf/WC and 0 h 40 min for JRC since resumes that only fall under "software Development/Web architecture" category were considered in the matching process i.e. we only match 148 resumes instead of matching 2000 resumes. Furthermore, our system provides better result than tf-idf/WC since JRC attempts to reduce the cost issue by segmenting the content of both resume and job posts and finding matches between important segments in both instead of matching between the content of the whole resumes and job posts (as performed using tf-idf/WC.) For instance, "video editor" job post costs 0 h 5 min of execution time for JRC and 0 h 11 min using tf-idf/WC.

It may be argued that it's not fair to compare MatchingSem with JRC, since MatchingSem doesn't adopt classification of job posts and resumes. Therefore, we have minimized the space of resumes and job posts to be the same number of the results produced in JRC classification results. Again, we perform the comparison but on the minimized dataset. Figure 10 shows the run-time complexity of the matching process between JRC and MatchingSem on the minimized dataset.

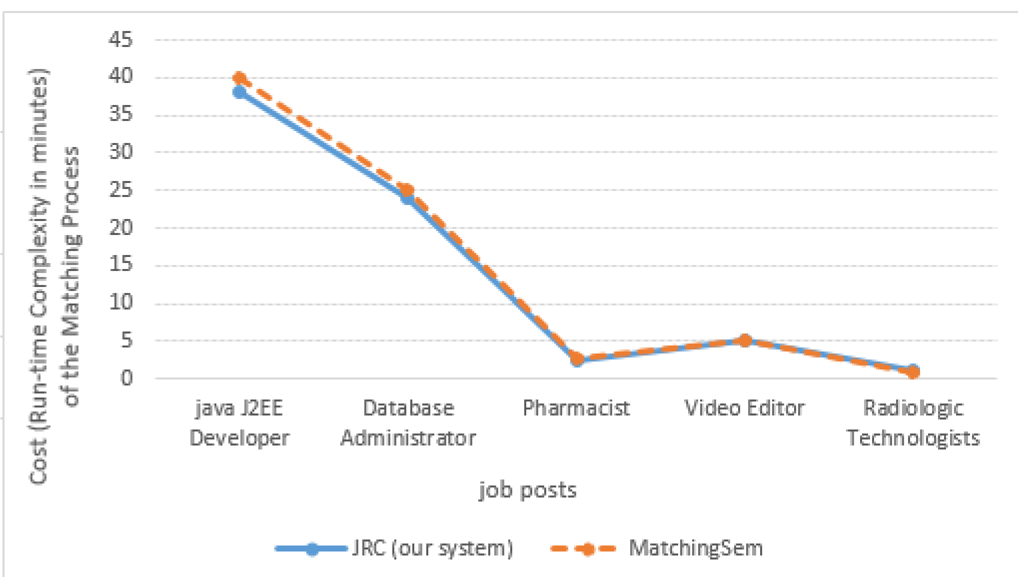


**Figure 9. Cost (run-time complexity in hours) of the matching process between JRC and MatchingSem**

As we can see from Figure 10, the run time is nearly the same especially for "video editor" and "radiologic Technologists". However, JRC produces more precise results as we demonstrate in the next sections.

### *B. Experiments of Job Post Classification*

In this section, we discuss the job post classification process. As mentioned in section 4.1.3, we have used the zones "Job Title" and the "Required Skills" in the classification process, and we have assigned weights for Job Title ( $\mathbf{JP_t}$ )=70% and the Required Skills ( $R_s$ )=30%, since we have found empirically that the job title is more significant than the required skills and guides to better matching results. In Table 6, we have compared the results of the classification process when we used weighted zone scoring (henceforth stated as $W_{ZS}$) and when we don't use weighted zone scoring (henceforth stated as $NW_{ZS}$) using the following equations:

$$W_{ZS} = JP_t * 70\% + Rs * 30\% \tag{5}$$

$$NW_{ZS} = Rs * 100\% \tag{6}$$

**Table 6. Job post classification results with/without weighted zone scoring**

| **Job title** | **Required skills** | **Job classification** | **Weight using $W_{ZS}$** | **Weight using $NW_{ZS}$** |
|---|---|---|---|---|
| Video Editor | Adobe Illustrator, After Effects, Premiere Pro, photoshop, Adobe Audition | Design/ Multimedia Design | **100%** | **100%** |
| SQL Server Developer | Sql, sql server, Redshift, Qlik view, database, ETL, BI | Data/ Databases | **96.25%** | **88.9%** |
| | MS office suite | Industry-specific / Microsoft Office | 3.75% | 11.1% |

| | | | | |
|---|---|---|---|---|
| Android Developer | Android, xcode, | Software Development/ Mobile development | **76.67%** | **30%** |
| | HTML5, CSS, javascript, ajax, jQuery | Software Development/ Web Development | 16.67% | 50% |
| | Sql server, SQL Express | Data / Databases | **6.66%** | **20%** |
| Network Technician | CAT5E, CAT6, CATV cable router, optical fiber, CCTV, BICSI | IT Administration/ Technical Support | **100%** | **100%** |
| Web Developer | Wordpress, HTML, CSS, javascript, Ajax, Jquery, Angular | Software Development/ Web Development | **93.3%** | **80%** |
| | WCM, Adobe CQ | Communication/ Marketing | **6.7%** | **20%** |
| Multimedia Developer | Adobe Creative suite, photoshop, Illustrator, After Effect, InDesign | Design/ Multimedia Design | **82.5%** | **46.1%** |
| | Ios, Android | Software Development/ Mobile development | **5.0%** | **15.4%** |
| | HTML, CSS, javascript, wordpress, Drupal | Software Development/ Web Development | 12.5% | 38.5% |

As shown in Table 6, we can see that “Web Developer” job post falls under “Software Development/ Web Development” occupational category with a weight that equals 93.3%, and this is because when we submit the job title to our skills knowledge base it returns “Software Development/ Web Development” category with a weight that equals 70%. Then we submit the required skills and we find that “Wordpress, HTML, CSS, javascript, Ajax, Jquery, Angular” skills fall under the same space as the job title with weight = 23.3%, but “WCM, Adobe CQ” skills fall under “Communication/ Marketing” space with weight = 6.7%. However, when we submit the same job post to our skills knowledge base without giving weights to the job title and the required skills; we find that “Software Development/ Web Development” occupational category weight decreases to become 80% and

“Communication/ Marketing” weight increases to become 20%. And this is because when we didn’t use weighted zone scoring we considered that the job title has the same weight as the required skills. And the same for “Android Developer” job post, that falls under three categories: “Software Development/ Mobile development” with weight 76.67%, “Software Development/ Web Development” with weight 16.67%, and “Data / Databases” with weight 6.66%. And without weighted zone scoring the weights become 30%, 50%, 20% respectively. However, we notice that the results for some job posts didn’t change like “Front End Web Developer” and “IT Technician”; and this is because these job posts fall under one job category with weight 100%.

Table 7 shows a comparison between the classification results using two weighting scheme: Weighted zone scoring and tf-idf scheme. The tf_idf weighting algorithm assigns a skills $s$ a weight in a job post $p$ as defined in Equation (7):

$$tf - idf = tf_{s,\,p} \times \log\left(\frac{N}{df_s}\right) \tag{7}$$

Where:

$tf_{s,\,p}$ :is the number of occurrences of $s$ in $p$ .

$df_s$ : represents the number of documents containing $s$

N: is the total number of documents

**Table 7. Job post classification results using weighted zone scoring and tf_idf scheme**

| **Job title** | **Required skills** | **Job classification** | **Weight using WZS** | **Weight using tf-idf scheme** |
|---|---|---|---|---|
| Video Editor | Adobe Illustrator, After Effects, Premiere Pro, | Design/ Multimedia Design | **100%** | **40.8%** |

| | | | | |
|---|---|---|---|---|
| | photoshop, Adobe Audition | | | |
| SQL Server Developer | Sql, sql server, Redshift, Qlik view, database, ETL, BI | Data/ Databases | **96.25%** | **35.72%** |
| | MS office suite | Industry-specific / Microsoft Office | 3.75% | 2.6% |
| Android Developer | Android, xcode, | Software Development/ Mobile development | **76.67%** | **30.5%** |
| | HTML5, CSS, javascript, ajax, jQuery | Software Development/ Web Development | 16.67% | 5.06% |
| | Sql server, SQL Express | Data / Databases | **6.66%** | **8.4%** |
| Network Technician | CAT5E, CAT6, CATV cable router, optical fiber, CCTV, BICSI | IT Administration/ Technical Support | **100%** | **41.9%** |
| Web Developer | Wordpress, HTML, CSS, javascript, Ajax, Jquery, Angular | Software Development/ Web Development | **93.3%** | **26.6%** |
| | WCM, Adobe CQ | Communication/ Marketing | **6.7%** | **3.85%** |
| Multimedia Developer | Adobe Creative suite, photoshop, Illustrator, After Effect, InDesign | Design/ Multimedia Design | **82.5%** | **25.7%** |
| | Ios, Android | Software Development/ Mobile development | **5.0%** | **1.5%** |
| | HTML, CSS, javascript, wordpress, Drupal | Software Development/ Web Development | 12.5% | 2.65% |

As shown in Table 7, we can see that "Video Editor" job post falls under "Design/ Multimedia Design" occupational category with weight equals 100%, and this is because when we submit the job title to our skills knowledge base it returns "Software Development/ Web Development" category with weight 70%, then we submit the required skills and we find that all of them fall under the same space with weight 30%. However, when we use tf-idf weighting the weight decreases to 40.8% and this is because the tf-idf weighting scheme deals with the job posts as a bag of words ignoring the co-relation between the different zones and the different words.

### *C. Precision Results of Matching Resumes to their Corresponding Job Posts*

In this section, we evaluate our system's effectiveness using the *Precision* indicator. To do that, we manually calculated the relevance scores between each job post and its corresponding resumes. After that, we compared the manually calculated scores to those produced by the system. Finally, we find the difference between the manually assigned scores and their corresponding scores that are automatically produced by the system. We used the following formula to measure the precision of the produced results:

$$\boldsymbol{P} = \frac{|\boldsymbol{V_m} - \boldsymbol{V_{auto}}|}{\frac{\boldsymbol{V_m} + \boldsymbol{V_{auto}}}{\boldsymbol{2}}} * \boldsymbol{100}\% \qquad \textbf{(8)}$$

Where:

- $V_m$ : is the manually assigned relevance score between each resume and job post.
- $V_{auto}$ : is the automatically calculated relevance score between each resume and job post.

Table 8 and Table 9, show the precision results of matching resumes with their corresponding job posts.

**Table 8. Precision results of matching resumes with their corresponding job posts**

| Occupational Category | Job Title | Resume index | Manual score | Auto score | Precision |
|---|---|---|---|---|---|
| **Software Development / Interactive Multimedia** | Multimedia Designer | CV4<br>CV2<br>CV1 | 0.42<br>0.87<br>0.09 | 0.51<br>0.90<br>0.10 | **0.82**<br>**0.96**<br>**0.90** |
| **Design Software / Graphics** | Graphic Designer | CV1<br>CV2<br>CV3 | 0.67<br>0.12<br>0.81 | 00.7<br>0.20<br>0.81 | **0.95**<br>**0.60**<br>**1.00** |
| **Recruiting / Human resources** | Associate HR Consultant | CV5<br>CV6<br>CV7 | 0.45<br>0.33<br>0.77 | 0.53<br>0.44<br>0.83 | **0.84**<br>**0.75**<br>**0.92** |
| | | CV8 | 0.39 | 0.30 | **0.77** |

| | | | | | |
|---|---|---|---|---|---|
| **IT Administration/ Technical Support** | Network Technician | CV9 | 0.80 | 0.80 | **1.00** |
| | | CV10 | 0.55 | 0.60 | **0.92** |
| **Software Development/ Web Development** | Web Developer | CV4 | 0.10 | 0.11 | **0.90** |
| | | CV1 | 0.46 | 0.55 | **0.83** |
| | | CV11 | 0.30 | 0.30 | **1.00** |

As shown in Table 8, we match job posts to their corresponding resumes that fall under the same occupational categories. For instance, "Graphic Designer" job post is matched only with resumes that fall under "Design Software / Graphics" category. As such, CV1 and CV2 are matched with "Graphic Designer" and "Multimedia Designer" job posts. And this is because these CVs exist in both "Design Software / Graphics" and "Software Development / Interactive Multimedia" categories. However, the matching score differ from one job post to another. For instance, CV2 achieved a very low matching score when matched with "Graphic Designer" job post (0.12 manual score, 0.20 automatic score), but CV1 achieved better score for the same job post (0.67 manual score, 0.70 automatic score). On the other hand, CV2 achieved better results than CV1 when it was matched with "Multimedia Designer" job post (0.87 manual score, 0.90 automatic score) and this is because CV2 falls under "Software Development / Interactive Multimedia" with weight 86.6% and falls under "Design Software / Graphics" with weight 13.4%.

As shown in Table 9, for each job post, we find the relevance judgment by calculating the difference between the manually assigned relevance score for each resume and its corresponding relevance score that is automatically produced by our system. We considered three job posts and four resumes for each. The first job post is "Video Editor" has the following requirements: 1 years of professional editorial experience in a video marketing environment, knowledge of Adobe Premiere, video compression, post-production, full Adobe CC suite, and experience with motion graphics and Adobe After

Effects. The second job post “Database developer” requires a Bachelor's degree in CS, knowledge in MariaDB, MySQL, Oracle DB, ASM, Oracle RAC, Oracle 11g, and 3 years of experience with SQL development.

**Table 9. Comparative evaluation – relevance judgments**

| Job title | Resume index | Manual score | Automatic score | Difference (Manual-Automatic) | Judgement |
|---|---|---|---|---|---|
| **Video Editor** | CV1 | 0.28 | 0.28 | **0.00** | **Perfect match** |
| | CV2 | 0.60 | 0.50 | **0.10** | **Under qualified** |
| | CV3 | 0.44 | 0.44 | **0.00** | **Perfect match** |
| | CV4 | 0.70 | 0.75 | **-0.05** | **Over qualified** |
| **Database Developer** | CV5 | 0.20 | 0.12 | **0.08** | **Under qualified** |
| | CV6 | 0.63 | 0.63 | **0.00** | **Perfect match** |
| | CV7 | 1.00 | 1.00 | **0.00** | **Perfect match** |
| | CV8 | 0.83 | 0.90 | **-0.07** | **Over qualified** |
| **Photographer** | CV1 | 0.22 | 0.12 | **0.10** | **Under qualified** |
| | CV9 | 1.00 | 1.00 | **0.00** | **Perfect match** |
| | CV10 | 0.55 | 0.60 | **-0.05** | **Over qualified** |
| | CV11 | 0.77 | 0.77 | **0.00** | **Perfect match** |

As we see in Table 9, if we take CV1, CV3, CV6, CV7, CV9 and CV11; we can see that the difference between the manual score and the automatic score equals “0” and this leads to the perfect match between the score assigned by the expert and the scores generated by our system. On the other hand, the difference between the manual scores and the automatic scores for CV2 and CV5 is (0.10 and 0.08) respectively, and the reason behind that is because for CV2 our system was unable to extract the Loyalty from the applicant resume, and for CV5, our system was unable to recognize “ASM” skill from the applicant resume. However, we manually enrich our knowledge base with the missing skills and re-do the experiments and the difference became "0". Finally, For CV4 and CV8 the difference between the manual scores and the automatic scores is (-0.05 and -0.07) respectively. As for CV4 that identifies an applicant with 2 years of experience in video montaging and editing, and this exceed the required experience in "video editor" job post. Furthermore, CV8 identifies an applicant with Master degree in computer science.

### *D. Comparison with State-of-the-Art Systems*

In this section, we compare the results produced by our systems JRC to those produced by MatchingSem system (Kmail et al., 2015b), and the results produced when we use Boolean matching technique i.e. tf-idf scheme. To accomplish the comparison task, we tested the three approaches against the same dataset which we collected from different online recruitments systems as we clarify in section 5.1. Then, we compared the manually calculated scores to those produced by JRC system and the other approaches.

**Table 10. Comparative evaluation – JRC vs. other approaches**

| Job title | Resume index | Manual score | Tf-idf Auto score | MatchingSem Auto score | JRC Auto score |
|---|---|---|---|---|---|
| **Back-end web developer** | CV1 | **0.38** | 0.16 | 0.30 | **0.45** |
| | CV2 | **0.26** | 0.19 | 0.19 | **0.19** |
| | CV3 | **1.0** | 0.56 | 0.70 | **1.0** |
| **Java developer** | CV4 | **0.61** | 0.35 | 0.50 | **0.65** |
| | CV5 | **0.46** | 0.35 | 0.40 | **0.46** |
| | CV6 | **0.53** | 0.21 | 0.35 | **0.54** |
| **Animator Designer** | CV7 | **0.35** | 0.20 | 0.20 | **0.35** |
| | CV8 | **0.70** | 0.61 | 0.70 | **0.75** |
| | CV9 | **0.20** | 0.20 | 0.25 | **0.25** |

As shown in Table 10, we have three job posts and for each job post we have three resumes. The first job post namely, "Back-end web developer" with the following requirements: 2+

years of experience building JPA data access layers, with Spring and Hibernate, BSc degree in CS or relevant and knowledge in Lucene, Solr, NoSQL, Riak, Cassandra SQL and Oracle. The second job post requires BS Degree in CS, SE or related field combined with 3-5 years of experience developing web applications and experience with Java in an IBM WebSphere (or similar environment). The third job post is looking for the candidate that has strong understanding of animation, timing and editing as it relates to motion graphics and can use a variety of software platforms like Photoshop, After Effects, and Cinema 4D. As we can see, the automatically calculated scores by our system (JRC) are very close to the manually assigned scores by our expert. For example, if we take the second job post "java developer" and the first applicant "CV4" who has 2+ years of experience in java programming and has BSc in computer science, we can see that the difference between the manual score and the automatic score (.04) is less than the difference between the manual score and the automatic generated by MatchingSem (0.1) and tf-idf scheme (0.26). This is because the tf-idf scheme ignores the semantic aspects of the concepts encoded in both resumes and job posts. On the other hand, – unlike MatchingSem system – we are integrating a section-based segmentation module to extract features such as educational background, years of experience and employment information from applicants' resumes. When we incorporate these features, the matching scores produced by our system are better than when using only a list of candidate concepts as proposed in MatchingSem.

## 5.2 Evaluating the Effectiveness of the Profile Generation and Job Post Recommender Components

To validate the effectiveness of the automatic profile generation and job recommendation components, we have conducted experiments on the same dataset that has been used to evaluate the effectiveness and the efficiency of JRC. In order to carry out the experiments, we started by converting semi-structured resumes into structured documents through employing document parsing techniques as described in section 4.2.1. After that, we utilized HS dataset and our integrated knowledge base to further enrich the profiles with additional skills that were not explicitly mentioned in the applicant resumes, although these skills would be rated similar by a human expert. And finally, we exploit the applicant's profiles in order to push job post notifications that satisfy job seekers qualifications and skills.

In this section, first we conduct our experiments to evaluate the accuracy of the automatic profile generation by comparing our automatically generated profiles with the manual profiles which are created at LinkedIn. To do that, we asked some applicants to fill their profiles in LinkedIn and in the same time we took their resumes to automatically generate their profiles and we compare the results using the Precision indicator. On the other hand, in order to validate the effectiveness of our job recommender module we compare between the produced results by our system when utilizing the skills enrichment module against when not using it, in order to measure the impact of using it on the effectiveness of the proposed system.

### *A. Evaluating the Automatic Profile Generation Effectiveness Compared to The Manually Generated Profile*

In this section, we compare between the profiles generated by our system against the manually constructed using LinkedIn site. We used the Precision (P) indicators in order to measure the quality of the produced results where:

$$\boldsymbol{P} = \frac{|\{\boldsymbol{relevant\ segments}\} \cap \{\boldsymbol{retrieved\ segments}\}|}{|\{\boldsymbol{retrieved\ segments}\}|} \qquad \textbf{(9)}$$

To carry out the experiments, we asked ten applicants to build their profiles manually on LinkedIn, and at the same time, we upload their resumes to our system. After that, we calculated the precision of the retrieved results by our system and compared them to the results produced when the applicants manually created their profiles.

**Table 11. Precision (P) results of automatic profile generation**

| Job seeker index | LinkedIn | | | | Our system | | | |
|---|---|---|---|---|---|---|---|---|
| | Education Precision (**$P_E$**) | Employment History precision (**$P_H$**) | Skills Precision (**$P_S$**) | Total Precision (**P**) | Education Precision (**$P_E$**) | Employment History precision (**$P_H$**) | Skills Precision (**$P_S$**) | Total Precision (**P**) |
| **JS1** | 100% | 100% | 100% | 1.00 | 100% | 40% | 100% | 0.80 |
| **JS2** | 100% | 90% | 100% | 0.96 | 100% | 0% | 100% | 0.66 |
| **JS3** | 100% | 100% | 100% | 1.00 | 100% | 100% | 100% | 1.00 |
| **JS4** | 100% | 0% | 80% | 0.60 | 50% | 60% | 100% | 0.70 |
| **JS5** | 100% | 100% | 100% | 1.00 | 100% | 100% | 100% | 1.00 |
| **JS6** | 100% | 100% | 100% | 1.00 | 100% | 100% | 100% | 1.00 |
| **JS7** | 100% | 80% | 100% | 0.93 | 100% | 100% | 100% | 1.00 |
| **JS8** | 100% | 100% | 100% | 1.00 | 100% | 50% | 100% | 0.83 |
| **JS9** | 100% | 100% | 100% | 1.00 | 100% | 70% | 100% | 0.90 |
| **JS10** | 100% | 100% | 100% | 1.00 | 100% | 60% | 100% | 0.87 |

As shown in Table 11, we can see that almost all of the profiles that the applicants created on LinkedIn have a precision =1.00, since the applicants fill all the fields that requested in LinkedIn's form structure. However, it was lower for some applicants like (*JS2*, *JS4*, *JS7*) since they have incomplete profiles as they find the manual process of creating and modifying their profiles a tedious task. For instance, "*JS4*" didn't fill out the section about his employment history, in addition, he didn't put all the skills that are mentioned in his resume. On the other hand, our system produced a satisfactory precision results in the skills section, where our system was able to extract all the skills mentioned in the applicant's resumes. Moreover, our system was able to extract the education segment correctly, but as we see for applicant *JS4* the precision for the education section ($P_E$=0.50), and this is because "*Electronics Communication Engineering*" education field couldn't be recognized by the section segmentation module. However, we enrich the NER manually through adding a new rule for this education field (Electronics Communication Engineering EDUCATION FIELD). After that, we redo our experiment and the education precision $P_E$ became 100%.

### *B. The Experiments Using Applicants Judgments*

In this section, we evaluate the system effectiveness based on comparing the automatically generated scores (using Equation 9) and the applicant's evaluation for their generated profiles by our system. After that, we calculate the difference between them.

**Table 12. The difference between system judgment and user judgment**

| Applicant Profile index | Automatic Score (AS) | Applicant judgment Score (AJS) | Difference (AJS-AS) |
|---|---|---|---|
| AP1 | 0.80 | 0.80 | **0.00** |
| AP2 | 0.66 | 0.70 | **0.04** |
| AP3 | 1.00 | 1.00 | **0.00** |
| AP4 | 0.87 | 0.80 | **-0.07** |
| AP5 | 1.00 | 1.00 | **0.00** |
| AP6 | 1.00 | 0.90 | **-0.10** |
| AP7 | 1.00 | 0.95 | **-0.05** |
| AP8 | 0.83 | 0.80 | **-0.03** |
| AP9 | 0.90 | 0.90 | **0.00** |
| AP10 | 0.87 | 0.80 | **-0.07** |

As shown in Table 12, some applicants are satisfied with the results generated by our system like (AP3, AP5, AP9), where the difference between the automatic score and the applicant's judgment score = 0. However, for some applicants their judgment was different from the results produced by our system (AP4, AP6, AP7 and AP10). For AP6, he evaluated our system 90%, and he argued his judgment that our system was able to extract his information like education, experience, skills, employment history. However, it was unable to extract his research articles. On the other hand, AP4 and AP10 evaluate our system 80% and the difference between automatic score and their judgment score =-0.07, and this is because their automatic profiles are missing the training certifications section.

### *C. The Impact of Utilizing the Profile Enrichment Module on the Effectiveness of the Job Recommender System*

In this section, we compare between the results produced by the system when we exploit the profile enrichment module against when not exploiting it. By this we mean that we compared the manually calculated scores to those produced by the system when considering profile enrichment module and when only using the skills that originally exist in the job seeker's resume. To carry out this task, we only take the skills parameter from Equation (1) which has 50% weight to calculate the relevance score. To accomplish the comparison task, we calculate the degree of similarity between resumes and recommended job posts as defined in Equation (10).

$$RS = \frac{S_{resume} \cap S_{Job}}{S_{Job}} * 50\% \tag{10}$$

Where:

- RS: is the relevance score assigned between a resume and the recommended job post.
- $S_{resume}$ : is the list of skills exist in the applicant's profile.
- $S_{Job}$ : is the list of skills required in the job post.

**Table 13. The effectiveness of the recommendation results using/not using profile enrichment module**

| Resume index | Job title | Manual score | Automatic scores without using PE Module | Automatic scores using PE Module |
|---|---|---|---|---|
| CV1 | Java developer | 0.19 | 0.20 | **0.35** |
| | JAVA J2EE Developer | 0.44 | 0.45 | **0.50** |
| | Senior java Developer | 0.30 | 0.33 | **0.46** |
| | Database Administrator | 0.15 | 0.16 | **0.16** |
| | Front End Web Developer | 0.07 | 0.07 | **0.07** |

| | | | | |
|---|---|---|---|---|
| CV2 | Network Administrator | 0.22 | 0.23 | **0.30** |
| | Network Architect | 0.15 | 0.18 | **0.25** |
| | Network Analyst | 0.46 | 0.35 | **0.50** |
| | Associate Network Engineer | 0.38 | 0.28 | **0.40** |
| | Network Admin | 0.08 | 0.08 | **0.08** |
| CV3 | Video Editor | 0.24 | 0.20 | **0.35** |
| | Animator Designer | 0.19 | 0.18 | **0.24** |
| | Multimedia Developer | 0.30 | 0.35 | **0.45** |
| | Unity Developer | 0.25 | 0.30 | **0.30** |
| | Front End Web Developer | 0.19 | 0.20 | **0.35** |
| CV4 | Front End Web Developer | 0.40 | 0.39 | **0.50** |
| | Web Developer | 0.23 | 0.23 | **0.30** |
| | Web Designer | 0.15 | 0.19 | **0.19** |
| CV5 | Android Developer | 0.35 | 0.40 | **0.50** |
| | iOS Developer | 0.12 | 0.13 | **0.13** |
| | Mobile Developer | 0.33 | 0.39 | **0.45** |
| CV6 | Video Editor | 0.28 | 0.30 | **0.40** |
| | Animator Designer | 0.20 | 0.20 | **0.20** |
| | Multimedia Developer | 0.35 | 0.40 | **0.50** |
| CV7 | Data entry | 0.28 | 0.30 | **0.30** |
| | Data Entry Coordinator | 0.34 | 0.25 | **0.35** |
| | Admin-Data Entry Clerk | 0.15 | 0.15 | **0.15** |
| C8 | Photo Editor | 0.50 | 0.50 | **0.50** |
| | Photography Producer | 0.19 | 0.22 | **0.35** |
| | Photographer | 0.30 | 0.28 | **0.28** |

As shown in Table 13, we have eight resumes, and for each resume we have a group of recommended job posts that meet the applicants experience and qualifications. For instance, the first resume (CV1) has the following skills (css, servlet, ajax, java, javascript, jsp, mysql, hrm, oracle, query, jboss, tomcat, struts, eclipse, vimal, html, mvc, sri), as well as the skills which are enriched by the system (j2ee, hibernate, jdbc, spring, NetBeans, xml). As we can see in Table 12, we observed a significant improvement on the produced results when utilizing the profile enrichment module (PE) i.e. (the relevance scores were increased for the first group of the recommended job posts “java developer”, “java j2ee developer” and “senior java developer”). For example, if we consider the first job post “Java Developer” that requires the following skills (j2ee, node.js, sdet, tomcat, unix, xml, sql, javascript, mysql, jsp, Struts, hibernate, JDBC) we can see that the relevance score when we didn’t employ the PE module were (manual score= 0.19, automatic score= 0.20). However, the automatic score increased to become (0.35). This is because when integrating the PI module, we were able to further enrich the skill list with more skills that are related to them. For CV1 (J2ee, xml, hibernate, JDBC) which were not written in the applicant resume but required in the job post.

However, for some particular results, integrating the PE module doesn’t affect the produced results. For example, when we consider the last job posts in the first group of the recommended job posts “Database Administrator” and “Front-End Web Developer”, we can see that the automatic score when we employ PE module equal the automatic score when we didn’t employ it (AutoScore without PE= 0.16, AutoScore with PE= 0.16 and AutoScore without PE= 0.07, AutoScore with PE=0.07 respectively). This is due to the fact that none of the enriched skills were required in these job posts. And hence, the enriched skills don’t affect (i.e. increase or decrease) the automatic score.

**5.3 Summary**

In this chapter, we discussed the experiments that we have conducted to validate the efficiency and the effectiveness of the proposed online recruitment system. In addition, we have compared the produced results by our system with one of the state-of-the-art systems. During this chapter, we divided the evaluation process into three successive stages. First, we evaluate the efficiency and the effectiveness of JRC based on coupling an integrated skills knowledge base and an automatic matching procedure between candidate resumes and their corresponding job postings. The conducted experiments using the exploited knowledge base demonstrate that using the proposed classification module assists in achieving higher precision results in a less execution time than conventional approaches.

Furthermore, the overall experimental evaluations for the produced matching results were promising and closely related to the manually assigned relevance scores between the job posts and their corresponding resumes.

The second and third stages evaluated the effectiveness of the *Automatic Profile Generation* and *Job Recommender Components*, and it showed that employing our integrated knowledge base has led to significant enhancements on the recommendation results due to recognize skills that were not mentioned explicitly in the applicant's resumes.

# 6. Conclusions and Future Work

This chapter summarizes our proposed approach for building an automatic online recruitment system, discusses its findings and contributions, points out the limitations and challenges that we faced in building the proposed system. Also, it outlines the future extensions for the current version of our proposed system. The chapter is divided into two sections. Section 6.1 presents a discussion of the contribution for our research work and highlights the techniques/approaches that we utilize in the proposed system. Section 6.2 discusses the future works and the other challenges that we plan to tackle in the future system updates.

## 6.1 Conclusions

In this thesis, we have proposed an automatic classification recruitment system in order to address the following issues. First, we aim to tackle the run time complexity of the matching process. Second, we aim to address the issue of the tedious task of creating manual profiles, as well as recommending job posts that satisfy job seekers qualifications and experience. To meet these goals, we summarize our contributions as follows.

Our first contribution is the development of job resume classification recruitment system (JRC) by combining feature extraction methods, integrated knowledge base and statistical concepts relatedness measures. Unlike traditional online recruitment systems, our proposal attempts to reduce the run-time complexity by minimizing the searching space through assigning job posts and resumes to their corresponding occupational categories.

The second contribution aims to automatically construct profiles for job seekers using the feature extraction techniques that convert the semi-structured resume into structured format. Furthermore, we utilize our integrated knowledge base to enrich the content of resumes with additional skills that are not explicitly mentioned in the applicant's resumes. Moreover, to facilitate the job searching task, job recommender system is applied to push job post notifications that satisfy job seekers qualifications and experience.

In order to evaluate the efficiency and effectiveness of the proposed system, we collected a dataset from different online recruitment systems (2000 resumes and 10,000 job posts). First, we compare our system performance with state-of-the-art systems, and the results produced by our system were more satfisfactory and promising. Moreover, in order to validate the effectiveness of the proposed online recruitment system, we use state-of-the-art indicator (Precision), in addition to comparing the produced results by the proposed system with state-of-the-art systems.

**6.2 Challenges and Future work**

Although the conducted experiments showed promising results, there are other potential improvements to the techniques presented in this research work. Below we discuss these improvements and outline proposals on how to achieve them in our future work:

- In the Category-based Matching Module, we attempt to match only resumes that fall under the same space as job posts. However, the run-time complexity of the matching algorithm is O($n^2$). Accordingly, we plan to improve the matching formula in order to reduce the required time for matching resumes and job posts.
- In the section-based segmentation module, we attempt to extract some information like Personal Information, Educational Background, Experience and Employment History Information. However, this module cannot extract information like Research Publication, Professional Activities and Certificates. Accordingly, we

plan to employ more rule-based and regular expression techniques to extract such information from the applicant resumes.

- In the job recommender module, we utilize both the skills extracted from resumes and the enriched related skills to compute the relevance scores between job offers and resumes and give n-top job recommendation that meets the applicant's qualifications. However, our proposed module didn't take into consideration the education or experience qualifications when computing the similarity between resumes and the recommended job posts. In order to increase the recommendation accuracy and give more personalized recommendation, we aim to integrate more parameters for example applicant behavior i.e. how they interact with an online portal. Furthermore, applicant-to-applicant correlation techniques, which finds similar applicants who have the same taste and qualification with the target applicant and recommends jobs based on what the similar applicant like.

## References

ADOMAVICIUS, G. & TUZHILIN, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering,* 17**,** pp. 734-749.

AL-OTAIBI, S. T. & YKHLEF, M. 2012a. Job recommendation systems for enhancing e-recruitment process. Proceedings of the International Conference on Information and Knowledge Engineering (IKE). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 1, pp. 433-439.

AL-OTAIBI, S. T. & YKHLEF, M. 2012b. A survey of job recommender systems. *International Journal of Physical Sciences,* 7**,** pp. 5127-5142.

AMDOUNI, S. & ABDESSALEM KARAA, W. B. 2010. Web-based recruiting. Computer Systems and Applications (AICCSA) IEEE/ACS International Conference on, 2010. IEEE, pp. 1-7.

AUER, S., BIZER, C., KOBILAROV, G., LEHMANN, J., CYGANIAK, R. & IVES, Z. 2007. Dbpedia: A nucleus for a web of open data. *The semantic web.* Springer, Busan, Korea, pp. 722–735..

BARBER, L. 2006. *E-recruitment Developments*, Institute for Employment Studies Brighton.

BEKKERMAN, R. & GAVISH, M. 2011. High-precision phrase-based document classification on a modern scale. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM-231 , .239

BELKIN, N. J. & CROFT, W. B. 1992. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM,* 35**,** 29-38.

BOBADILLA, J., HERNANDO, A., ORTEGA, F. & BERNAL, J. 2011. A framework for collaborative filtering recommender systems. *Expert Systems with Applications,* 38**,** 14609-14623.

BRANDÃO, C., MORAIS, C., DIAS, S., SILVA, A. R. & MÁRIO, R. 2017. Using Online Recruitment: Implicit Theories and Candidates' Profile. *In:* ROCHA, Á., CORREIA, A. M., ADELI, H., REIS, L. P. & COSTANZO, S. (eds.) *Recent*

*Advances in Information Systems and Technologies: Volume 3.* Cham: Springer International Publishing, pp. 293-301.

BUGA, A., FREUDENTHALER, B., MARTINEZ-GIL, J., NEMES, S. T. & PAOLETTI, L. 2017. Management of Accurate Profile Matching using Multi cloud Service Interaction. In: Proceedings of the 19th International Conference on Information Integration and Web-based Applications and Services, iiWAS 2017, Salzburg, pp. 161-165.

BURKE, R. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction,* 12**,** pp. 331-370.

BURKE, R. 2007. Hybrid web recommender systems. *The adaptive web.* Springer, pp. .377-408.

CHEN, J ,.NIU, Z. & FU, H. 2015. A novel knowledge extraction framework for resumes based on text classifier. International Conference on Web-Age Information Management. Springer, pp. 540-543.

CLYDE, S., ZHANG, J. & YAO, C.-C. 1995. An object-oriented implementation of an adaptive classification of job openings. Artificial Intelligence for Applications, 1995. Proceedings., 11th Conference on. IEEE, pp. 9-16.

COLUCCI, S., DI NOIA, T., DI SCIASCIO, E., DONINI, F. M., MONGIELLO, M. & MOTTOLA, M. 2003. A formal approach to ontology-based semantic match of skills descriptions. *J. UCS,* 9**,** pp. 1437-1454.

FALIAGKA, E., ILIADIS, L., KARYDIS, I., RIGOU, M., SIOUTAS, S., TSAKALIDIS, A. & TZIMAS, G. 2014. On-line consistent ranking on e-recruitment: seeking the truth behind a well-formed CV. *Artificial Intelligence Review,* 42**,** pp. 515-528.

FALIAGKA, E., RAMANTAS, K., TSAKALIDIS, A. K., VIENNAS, M., KAFEZA, E. & TZIMAS, G. 2011. An Integrated e-Recruitment System for CV Ranking based on AHP. WEBIST. pp. 147-150.

FÄRBER, F., WEITZEL, T. & KEIM ,T. 2003. An automated recommendation approach to selection in personnel recruitment. *AMCIS 2003 proceedings***,** 302. pp. 2329–2339.

FAZEL-ZARANDI, M. & FOX, M. S. 2009. Semantic matchmaking for job recruitment: an ontology-based hybrid approach. Proceedings of the 8th International Semantic Web Conference,.

FELDMAN, R. & SANGER, J. 2007. *The text mining handbook: advanced approaches in analyzing unstructured data*, Cambridge university press.

FINN, A. & KUSHMERICK, N. 2004. Multi-level boundary classification for information extraction. European Conference on Machine Learning. Springer, pp. 111-122.

GEBSER, M., KAUFMANN, B. & SCHAUB, T. 2009. Solution enumeration for projected Boolean search problems. International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems. Springer, pp. 71-86.

GUPTA, A. & GARG, D. 2014. Applying data mining techniques in job recommender system for considering candidate job preferences. Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on, 2014. IEEE, pp. 1458-1465.

HAUFF, C. & GOUSIOS, G. 2015. Matching GitHub developer profiles to job advertisements. Proceedings of the 12th Working Conference on Mining Software Repositories. IEEE Press, pp. 362-366.

HOFFART, J., SUCHANEK ,F. M., BERBERICH, K., LEWIS-KELHAM, E., DE MELO, G. & WEIKUM, G. 2011. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. Proceedings of the 20th international conference companion on World wide web. ACM, pp. 229-232.

HONG, W., ZHENG, S., WANG, H. & SHI, J. 2013. A job recommender system based on user clustering. *Journal of Computers,* 8**,** pp. 1960-1967.

HUANG, Z., ZENG, D. & CHEN, H. 2007. A comparative study of recommendation algorithms in e-commerce applications. *IEEE Intelligent Systems,* 22**,** pp. 68-78.

JAVED, F., LUO, Q., MCNAIR, M., JACOB, F., ZHAO, M. & KANG, T. S. 2015. Carotene: A job title classification system for the online recruitment domain. Big Data Computing Service and Applications (BigDataService), 2015 IEEE First International Conference on, 2015. IEEE, pp. 286-293.

KEIM, T., MALINOWSKI, J. & WEITZEL, T. 2005. Bridging the Assimilation Gap: A User Centered Approach to IT Adoption in Corporate HR Processes. *AMCIS 2005 Proceedings***,** 220.

KERRIN, M. & KETTLEY, P. 2003. *E-recruitment: Is it Delivering?*, Institute for Employment Studies.

KESSLER, R., BÉCHET, N., TORRES-MORENO, J.-M., ROCHE, M. & EL-BÈZE, M. 2009J. ob Offer Management: How Improve the Ranking of Candidates. ISMIS. Springer, pp. 431-441.

KESSLER, R., TORRES-MORENO, J. M. & EL-BÈZE, M. 2007. E-Gen: automatic job offer processing system for human resources. Mexican international conference on artificial intelligence. Springer, pp. 985-995.

KMAIL, A. B., MAREE, M. & BELKHATIR, M. 2015a. MatchingSem: online recruitment system based on multiple semantic resources. Fuzzy Systems and Knowledge Discovery (FSKD), 2015 12th International Conference on. IEEE, pp. 2654-2659.

KMAIL, A. B., MAREE, M., BELKHATIR, M. & ALHASHMI, S. M. 2015b. An automatic online recruitment system based on exploiting multiple semantic resources and concept-relatedness measures. Tools with Artificial Intelligence (ICTAI), IEEE 27th International Conference on, 2015b. IEEE, pp. 620-627.

KOLAKOWSKI, N. S. A. N. 2018. *Dice Skills Center* [Online]. Available: https://www.dice.com/skills [Accessed 25 Feb 2018].

LANG, S., LAUMER, S., MAIER, C. & ECKHARDT, A. 2011. Drivers, challenges and consequences of E-recruiting: a literature review. *Proceedings of the 49th SIGMIS annual conference on Computer personnel research.* San Antonio, Texas, USA: ACM, pp. 26-35.

LAVRENKO, V. & CROFT, W. B. 2017. Relevance-based language models. ACM SIGIR Forum. ACM, pp. 260-267.

LEE, D. H. & BRUSILOVSKY, P. 2007. Fighting information overflow with personalized comprehensive information access: A proactive job recommender. Autonomic and autonomous systems. ICAS07. Third international conference on, 2007. IEEE, pp. 21-21.

LEE, I. 2005. The evolution of e-recruiting: A content analysis of Fortune 100 career web sites. *Journal of Electronic Commerce in Organizations (JECO),* 3**,** pp. 57-68.

LEE, I. 2007a. An architecture for a next-generation holistic e-recruiting system. *Communications of the ACM*, 50**,** pp. 81-85.

LEE, I. 2007b. An architecture for a next-generation holistic e-recruiting system. *Commun. ACM,* 50**,** pp. 81-85.

LOPS, P., DE GEMMIS, M. & SEMERARO, G. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook.* Springer, pp. 73-105.

LUO, X., XIA, Y. & ZHU, Q. 2013. Applying the learning rate adaptation to the matrix factorization based collaborative filtering. *Knowledge-Based Systems,* 37**,** pp. 154-164.

MALINOWSKI, J., KEIM, T., WENDT, O. & WEITZEL, T. 2006. Matching people and jobs: A bilateral recommendation approach. System Sciences. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on, 2006. IEEE, 137c-137c.

MANNING, C., SURDEANU, M., BAUER, J., FINKEL, J., BETHARD, S. & MCCLOSKY, D. 2014. The Stanford CoreNLP natural language processing toolkit. Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. pp. 55-60.

MARTINEZ-GIL, J., PAOLETTI, A. L. & SCHEWE, K.-D. 2016. A smart approach for matching, learning and querying information from the human resources domain. East European Conference on Advances in Databases and Information Systems. Springer, pp. 157-167.

MEHTA, S., PIMPLIKAR, R., SINGH, A., VARSHNEY, L. R. & VISWESWARIAH, K. 2013. Efficient multifaceted screening of job applicants. Proceedings of the 16th International Conference on Extending Database Technology .ACM, pp. 661-671.

MILLER, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38**,** pp. 39-41.

MOCHOL, M., WACHE, H. & NIXON, L. 2007. Improving the accuracy of job search with semantic techniques. International Conference on Business Information Systems. Springer, pp. 301-313.

NECULOIU, P., VERSTEEGH, M. & ROTARU, M. 2016. Learning text similarity with siamese recurrent networks. Proceedings of the 1st Workshop on Representation Learning for NLP. pp. 148-157.

O*NET. 2018a. About Occupational Information Network (O*NET) [Online]. Available: https://onet.rti.org/about.cfm [Accessed 10 Feb 2018].

O*NET. 2018b. *Occupational Information Network* [Online]. Available: https://www.onetonline.org/ [Accessed 25 Feb 2018].

PANDE, S. 2011. E-recruitment creates order out of chaos at SAT telecom: system cuts costs and improves efficiency. *Human Resource Management International Digest,* 19**,** pp. 21-23.

PARRY, E. & TYSON, S. 2008. An analysis of the use and success of online recruitment methods in the UK. *Human Resource Management Journal,* 18**,** 257-274.

PATIL, N. A., WAGH, G., PATIL, L. S. & SHIROLE, B. S. 2017. A Survey on Resume Extractor and Candidate Recruitment System. *International Journal,* 2.

PAZZANI, M. J. & BILLSUS, D. 2007. Content-based recommendation systems. *The adaptive web.* Springer. pp. 325-341..

SCHMITT, T., CAILLOU, P. & SEBAG, M. 2016. Matching jobs and resumes: a deep collaborative filtering task. GCAI 2016. 2nd Global Conference on Artificial Intelligence, pp. 124–137.

SENTHIL KUMARAN, V. & SANKAR, A. 2013. Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping (EXPERT). *International Journal of Metadata, Semantics and Ontologies,* 8**,** pp. 56-64.

SHALABY, W., ALAILA, B., KORAYEM, M., POURNAJAF, L., ALJADDA, K., QUINN, S & .ZADROZNY, W. 2018. Help Me Find a Job: A Graph-based Approach for Job Recommendation at Scale. *arXiv preprint arXiv:1801.00377*.

SHARON, P. 2011. E-recruitment creates order out of chaos at SAT Telecom: System cuts costs and improves efficiency. *Human Resource Management International Digest,* 19**,** pp. 21-23.

SIVABALAN, L., YAZDANIFARD, R. & ISMAIL, N. H. 2014. How to transform the traditional way of recruitment into online system. *International Business Research,* 7**,** pp. 178-185.

SOLVED, H. 2018. *HIRINGSOLVED* [Online] .HiringSolved website [Online]. Available: https://hiringsolved.com/explorer [Accessed 7 Feb 2018].

STROHMEIER, S. & PIAZZA, F. 2013. Domain driven data mining in human resource management: A review of current research. *Expert Systems with Applications,* 40, pp. 2420-2410.

XIE, Y., WULAMU, A., HU, X. & ZHU, X. 2014. Design and Implementation of Privacy-preserving Recommendation System Based on MASK. *JSW,* 9**,** pp. 2607-2613.

YI, X., ALLAN, J. & CROFT, W. B. Matching resumes and jobs based on relevance models 2007. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 809-810.

YU, K., GUAN, G. & ZHOU, M. 2005. Resume information extraction with cascaded hybrid model. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, pp. 499-506.

ZAROOR, A., MAREE, M. & SABHA, M. 2017. A Hybrid Approach to Conceptual Classification and Ranking of Resumes and Their Corresponding Job Posts. International Conference on Intelligent Decision Technologies. Springer, pp. 107-119.

ZHU, Y., JAVED, F. & OZTURK, O. 2016. Semantic Similarity Strategies for Job Title Classification. *arXiv preprint arXiv:1609.06268*.

## الملخص باللغة العربية

# نحو بناء نهج مبني على التصنيف المصطلحي للفصل بين السير الذاتية للمتقدمين ومفاضلتهن وفق إعلانات التوظيف

مع النُموِّ المتزايدِ في مواقعِ التوظيفِ عبْرَ الإنترنت، أصْبَحت أساليبُ التوظيفِ التقليديةِ غيرُ فعَّالة، ويرجعُ ذلكَ إلى حقيقةِ أنَّ هذه المواقعَ تستقبل أعداداً هائلةً من السيرِ الذاتيةِ مختلفةِ المبْنى – مِن حيثُ النَّمط والهيكلية والمُحتوى - من الباحِثين عن العَمل من مُختلف المَجالات والخِبرات والتَخصصات. ولِذلك، فإنّ أَهميةَ استخراجِ المعلوماتِ من السيرِ الذاتيةِ لا تكمنُ فَقط في تَفعيل وتَسريع عَملية الفرزِ التلقائي للمُرشَحين، ولكن أيضا في عملية تَوجيه السيرِ الذاتيةِ نَحو الفئاتِ الوَظيفية التابِعة لَها وتَرتيبها بشكل اوتُوماتيكي بناءً على تَحقيقها للشُروط المَطلوبة في اعلاناتِ التَوظيف. وقَد اقتُرحت عدةُ أساليب وطرق لمعالجة مسألة المطابقة التلقائية بين السير الذاتية والوظائف. ومع ذلك، فَقد تم إعطاء القليلِ من الاهتِمام لِمعالجة المشاكل المُرتبطة بالتصنيف المصطلحي للسير الذاتية والوظائف، والترتيب التلقائي لمقدمي الطلبات، إضافة الى إقتراح السير الذاتية للمُوَظِّفين من خلال استغلال المعلومات التي يتم استخراجها من السير الذاتية للمتقدمين.

و بناءً على ما سبق، فإننا نقترح في هذه الرسالة نظام توظيف إلكتروني يعتمد على قاعدة معلومات موحدة لتنفيذ مهمة التصنيف المصطلحي للسير الذاتية بالإضافة لإعلانات التوظيف. على عكس الأنظمة التقليدية التي تحاول التوفيق بين متطلبات إعلانات التوظيف عن طريق البحث في كامل مساحة السير الذاتية والوظائف، يقوم نظامنا بالتوفيق بين السير الذاتية و اعلانات التوظيف التي تقع فقط ضمن الفئات المهنية والوظيفية ذات الصلة. فعلى سبيل المثال، عندما يبحث صاحب العمل عن "مطور ويب" فإن نظامنا يقوم بالبحث فقط ضمن السير الذاتية التي تقع ضمن الفئة المهنية "تطوير الويب" بدلاً من البحث في جميع الفئات المهنية مما يقلل الوقت اللازم في عملية التوفيق بين السير الذاتية واعلانات التوظيف ذات العلاقة. بالإضافة إلى ذلك، فإننا نحاولُ من خلال النظام المقترح العمل على استغلال المعلومات المستخرجة من السير الذاتية لمقدمي الطلبات لبناء صفحاتٍ شخصيةٍ خاصة بهم اتوماتكياً بحيث يتم استخدامها مرة أخرى لتوصية الوظائف للباحثين عن عمل. وفي هذا السياق، سيؤدي النهج المقترح إلى دفع إشعارات الوظائف التي تتلائم مع اهتمامات ومهارات الباحثين عن العمل. ولإثبات فعالية النظام المقترح، قمنا بإجراء عدة تجارب باستخدام مجموعة من إعلانات التوظيف والسير الذاتية. بالإضافة إلى ذلك، قمنا بتقييم كفاءة وفعالية النظام المقترح من خلال

المقارنة مع أحدث أنظمة التوظيف عبر الإنترنت، وقد تم نشر النتائج التي توصلنا اليها في اثنين من المؤتمرات الدولية المصنفة عالمياً في العام 2017.