JRC: A Job Post and Resume Classification System for Online Recruitment

Conference Paper · November 2017

DOI: 10.1109/ICTAI.2017.00123

3 authors:

Abeer Zaroor
Arab American University

Mohammed Maree
Arab American University

Muath Sabha
Arab American University

2017 International Conference on Tools with Artificial Intelligence

JRC: A Job Post and Resume Classification System for Online Recruitment

Abeer Zaroor
Computer Science Department
The Arab American University
Jenin, Palestine
abeer.zaroor@aauj.edu

Mohammed Maree *
Information Technology Department
The Arab American University
Jenin, Palestine
Corresponding author:
mohammad.maree@aauj.edu

Muath Sabha
Information Technology Department
The Arab American University
Jenin, Palestine
muath.sabha@aauj.edu

***Abstract*— Due to 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 specialization. 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 to their corresponding occupational categories. This assists in minimizing the effort required by employers to manage and organize resumes, as well as to screen out irrelevant candidates. In this paper, we present JRC - 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, JRC matches resumes that only fall under their relevant occupational categories. 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.**

Keywords—Conceptual Matching; Resume Ranking; Online Recruitment; Knowledge base Assisted Classification

I. Introduction

In the recent years, online job portals have started to receive an enormous number of resumes in diverse styles and formats from job seekers who have different academic backgrounds, work experiences and skills [1, 2]. Finding and hiring the right talent from a wide and heterogeneous range of candidates remains one of the most important and challenging tasks of the HR department in any organization [3, 4]. To address this challenge, many companies have shifted to exploiting e-recruiting platforms [5, 6]. These platforms reduce the cost, time and effort required for manually processing and screening applicant resumes. As stated in [7], there were more than 40,000 e-recruitment sites in 2012 for helping job-seekers and recruiters worldwide. According to the International Association of Employment Web Sites (IAEWS) [8], 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 candidate resumes. For instance, one of the employed methods addresses the automatic matching between candidate resumes and their corresponding job offers [9, 10, 11]. 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 [12, 13]. Although these approaches produce high precision ratios in finding candidates to fill a vacancy [9], they give less attention to the run time complexity of the matching process i.e. every job offer will be matched with every resume in the corpus instead of matching resumes that are only related to their occupational category. Other researchers try to overcome this problem by utilizing machine learning techniques to first classify job posts and resumes under their relevant occupational categories [14]. Although these techniques have proven to be more efficient (i.e. have low run time complexity), they suffer from high error rates and low classification accuracy [15].

To overcome the abovementioned limitations, we present a hybrid approach to classify resumes and their corresponding job post by utilizing an integrated occupational categories knowledge base. The exploited knowledge base assists in i) classifying resumes and job offers under their corresponding occupational categories and ii) automatically ranking applicants that best match the announced offers. We summarize the contributions of our work as follows:

- Automatic Integrated Knowledge based Occupational Category Classification of Resumes and Job Postings.
- Employing a Section-based Segmentation heuristic by exploiting Natural Language Processing (NLP), Concept-relatedness techniques and regular expressions.

The remainder of this paper is organized as follows. In section 2, we introduce the work related. Section 3 describes an overview of the proposed system's architecture. In section 4, we provide the details of the proposed matching steps. Experimental validation of the effectiveness and efficiency of the proposed system is presented in section 5. In section 6, we discuss the conclusions and outline future work.

2375-0197/17/31.00 ©2017 IEEE
DOI 10.1109/ICTAI.2017.00123

II. RELATED WORK

Many approaches and techniques have been proposed for addressing the e-recruitment challenges. 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 [16, 17, 18, 20, 13]. For instance, the authors of [16] have proposed an approach for the automatic matching and querying of information in the human resources domain. The proposed approach exploits DISCO, ISCO and ISCED 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 ignoring the hidden semantic dimensions in the text of both documents [2]. The authors of [17] have proposed an ontology-based hybrid approach that matches job seekers and job advertisements through utilizing a similarity-based approach to rank applicants. The proposed system exploits semantic technologies in order to improve the matching process. However, the main drawback of this approach is the huge cost (run time complexity) of the matching process. On the other hand, JobDiSC system [18] attempts to classify job advertisements automatically by employing a standard classification scheme called Dictionary of Occupational Titles (DOT). The proposed system automatically generates classification rules from a set of pre-classified job openings and assigns one or more class for each job post. The main drawback of this system is that DOT doesn't cover the occupational information that is more relevant to the modern workplace [19]. Other systems utilize machine learning algorithms in order to annotate segments of resumes with the appropriate category, taking the advantage of the resume's contextual structure where related information units usually occur in the same textual segments [13, 20]. 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 to the time needed to pre-process and post-process job posts in order to minimize the error and maximize the classification accuracy.

III. OVERVIEW

In this section, we present an overview of the proposed system's architecture and discuss its main modules. As shown in Figure 1, the proposed 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 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[1] and O*NET[2]. More details on these resources will be provided in Section IV. 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 [2]. 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.

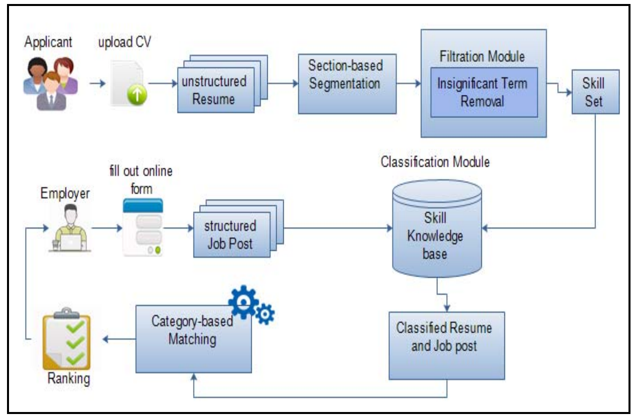

Fig. 1. Overall Architecture of the Proposed System

IV. DETAILS OF THE PROPOSED MATCHING STEPS

In this section, we detail the steps of the proposed system as follows:

A. Section-Based Segmentation Module

During this module, an automatic extraction of important segments such as: Education and Experience and other employment information such Company name, Applicant's Role in the company, Date of designation, Date of resignation and Loyalty is carried out. Accordingly, unstructured resumes are converted into segments (semi-structured documents) based on employing Natural Language Processing (NLP) techniques and rule-based regular expressions. As detailed in [9], the NLP steps are: document splitting, n-gram tokenization, stop word removal, part-of-Speech-Tagging (POST) and Named Entity Recognition (NER). To do this, first, we divide the text of a given resume into segments in order to process each paragraph separately. Then, each segment of the resume is split into tokens where we remove tokens that appear to be of little value in the classification and matching process. After that, we utilize the StanfordCoreNLP POSTagger 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:

[1] http://www.dice.com/skills
[2] http://www.onetcenter.org/taxonomy/2010/list.html

Example 1: Resume Segmentation- Sample of a job seeker's resume (CV1):

I have 3 years of experience as a web developer. And I have the following skills: PHP, HTML, CSS, JQuery, Ajax, android, ios
Education: Bachelor of Science (BSc) in Computer Science.
Employment Details
I worked as a Front-End developer in SaFa Company from 2007 to 2011.

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

```
<Applicantdata>
<Experience>
<Years>3</Years>
<Field> web developer </Field>
</Experience>
<Education>
<Degree> Bachelor of Science (BSc) </Degree>
<Field> Computer Science </Field>
</Education>
<EmploymentHistory>
<role> Front-End developer </role>
<companyName> SaFa Company </companyName>
<FromDate>2007</FromDate>
<ToDate>2011</ToDate>
<loyalty> 4</loyalty>
</EmploymentHistory>
<skills> PHP, HTML, CSS, JQuery, Ajax, android, ios
</skills>
</Applicantdata>
```

Once unstructured resumes are converted into semi-structured document, the list of candidate concepts is identified, extracted, and filtered using the filtration module. Table 1 shows the results of this step.

TABLE I. RESULT OF THE FILTERATION MODULE

Candidate terms extracted from resume	Filtered Concepts List from resume
PHP	android
HTML	ios
CSS	HTML
JQuery	CSS
Ajax	Jquery
Android	web
SaFa	php
Web	Ajax
php	php
skills	
experience	
company	
ios	

As shown in Table 1, concepts that belong to the list of pre-defined terms (e.g. contact info, address, birth date, country name) or have low tf-idf weights [2] are removed from the lists of candidate concepts.

B. Conceptual Classification Module

In our previous work [2, 23], we have utilized an integrated knowledge-base which combines Dice skills center (henceforth stated as DICE) and Occupational Information Network (O*NET) (henceforth stated as O*NET) to classify resumes and job posts. In this context, we use DICE to classify skills that belong to Information and Communication Technologies (ICT), and Economy field because we noticed that O*NET is not scalable enough for our classification needs. 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. For instance, "JPA" which refers to "Java Persistence" is classified under "Accountants" category by O*NET, but classifies correctly under "Software Development" in DICE. However, there are 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.

1) Skill-Based Resume Classification Module

In this module, each skill in the skills set is submitted to the exploited knowledge base sequentially in order to obtain a list of candidate occupational categories. As a result, a list of weighted occupational categories is obtained and sorted by the highest weight (as one skill may return zero, one, or more than one occupational category). For instance, as shown in Figure 2, when the skill "android" is submitted to the skills knowledge base, "Software Development/ Mobile Development" occupational category is obtained first. Then, using this procedure, a list of additional weighted categories is obtained and sorted according to their highest weight.

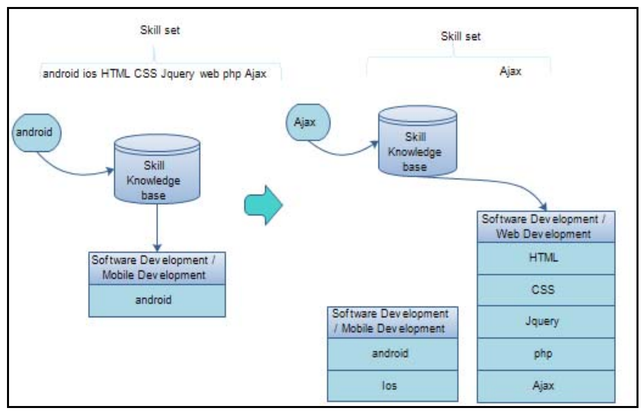

Fig. 2. List of obatined occupational categories for CV1

Table 2 shows each occupational category assigned to its corresponding skills.

TABLE II. SKILLS TO OCCUPATIONAL CATEGORIES MAPPING

Job category	skills
Software Development/ Mobile Development	Android, ios
Software Development/ Web Development	CSS, html, php, Ajax, jquery

2) Job Post Classification Module

In the Job Post Classification module, we use both the job title and the required skills from the structured job post for classification purposes. First, the job post is pre-processed and filtered through removing noisy information such as: city names, state and country acronyms that appear in the job title or job details. After that, we use the skills knowledge base to classify job posts in the same manner as we do for classifying resumes. Accordingly, we assign weights (Job Title=70% and Required Skills=30%) since we believe that the job title is more significant than the required skills and guides to better matching results. More examples on the results of this module are presented in Section V.B.

C. Matching Resumes and their Corresponding Job Postings

Inspired by the work developed in [2], we employ multiple semantic resources to derive the semantic aspects of resumes and job posts. These are WordNet ontology [21] and YAGO2 ontology [22]. 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 semantic resources. 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. By loyalty we mean the degree of devotion to the company that the applicant is working or worked in. The formula for calculating the scoring percentage is as follows:

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

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 weighting values:
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).

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

Where y_d is the weight for the degree d in the applicant 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 satisfied the educational requirement for the job post ($\frac{y_{BSc}}{x_{BSc}}$=perfect match). However if the applicant has a Diploma degree he will be considered underqualified since ($\frac{y_{Di}}{x_{BSc}}$=under qualified). If the applicant has a Master or PhD degree 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:

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

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

- If $y_r = x_j$ the applicant will be a qualified match.
- If $y_r < x_j$ the applicant will be underqualified.
- If $y_r > x_j$ the applicant will be 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 JPs the employer specifies a number of years while other JPs they specify the number of years in a specific field. For example: +2 years of experience in java development.

And let JS be an applicant who applies for JP has a set of qualification $(Ed_{JS}, Ex_{JS}, S_{JS})$ where Ed_{JS} the education degree JS has, Ex_{JS} is the amount of experience JS has, S_{JS} S_{JP}: list of skills, $\sum_{i=1}^{n} S_{JS_i}$. A qualified match donates 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}= $Ed_{JP} \cap Ed_{JS}$

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

V. Experimental Evaluation

This section describes the experiments that we have carried out to evaluate the techniques of the proposed system. In order to evaluate the efficiency and the effectiveness of the proposed system, we collected a data set of 2000 resumes downloaded from Amrood1[3], indeed[4], and we used 10,000 different job posts obtained from monster[5], shine[6] and careerbuilder[7]. The collected resumes are unstructured documents in different document 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). 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 [2] 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 3 shows the run-time complexity of the matching process between them.

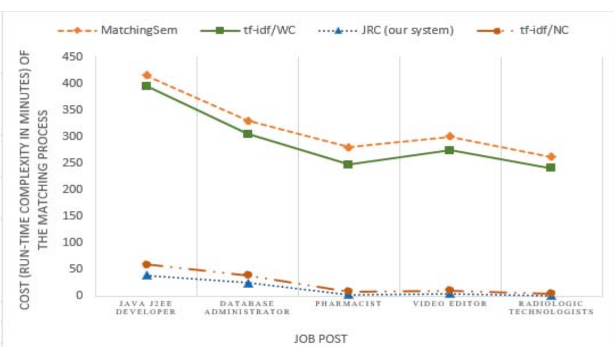

Fig. 3. Cost (Run-time Complexity in Hours) of the Matching Process

As shown in Figure 3, our system (JRC) was able to achieve higher precision 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 in tf-idf/WC and 40 min in JRC since only resumes that fall under "software Development/Web architecture" category were considered in the matching process i.e. we only match 148 resumes instead of matching 2000 resume. Furthermore, our system provides better result than tf-idf/WC since JCR attempt to reduce the cost issue by segmenting the content of both resume and job post and finding matches between important segments in both instead of matching between the content of the whole resumes and job posts. For instance, "video editor" job post cost 5 min of execution time and 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 4 shows the run-time complexity of the Matching Process between JRC and MatchingSem on the minimized dataset.

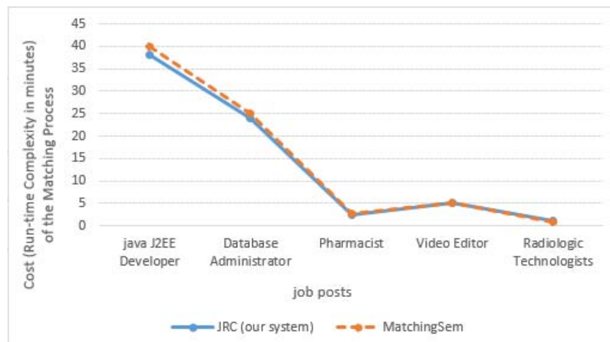

Fig. 4. Cost (Run-time Complexity in Hours) of the Matching Process between JRC and MatchingSem

As we can see from Figure 4, 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 job post classification. As mentioned in section IV.B.2, we have used job title and required skills in the classification process. In Table 3, we have compared the results of the classification process with weighted zone scoring (henceforth stated as WZS) and without weighted zone scoring (henceforth stated as NWZS).

TABLE III. Job Post Classification Results With/Without Weighted Zone Scoring

Job title	**Required skills**	**Job classification**	**Weight using WZS**	**Weight using NWZS**
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,	Data/ Databases	**96.25%**	**88.9%**

[3] http://www.amrood.com/resumelisting/listallresume.htm
[4] http://www.indeed.com/resumes
[5] http://jobs.monster.com
[6] http://www.shine.com/job-search
[7] http://www.careerbuilder.com

	database, ETL, BI			
	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 3, 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 weight 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 job title with weight 23.3% "resulting in a total of 93.3% for the Software Development/ Web Development space", 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 decrease to become 80% and "Communication/ Marketing" weight increase 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 4 shows a comparison between the classification results using two weighting scheme: Weighted zone scoring and tf-idf scheme.

TABLE IV. JOB POST CLASSIFICATION RESULTS USING WIEGHTED 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, photoshop, Adobe Audition	Design/ Multimedia Design	**100%**	**40.8%**
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 4, 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 with Corresponding Job Post

In this section we evaluate our system's effectiveness using precision indicator. For each job post, we compare between the manually assigned scores and their corresponding scores that are automatically produced by the system. Table 5, shows the precision results of matching resumes with their corresponding job post.

TABLE V. 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 CV2 CV1	0.42 0.87 0.09	0.51 0.90 0.10	0.82 0.96 0.90
Design Software / Graphics	Graphic Designer	CV1 CV2 CV3	0.67 0.12 0.81	0.70 0.20 0.81	0.95 0.60 1.00
Recruiting / Human resources	Associate HR Consultant	CV5 CV6 CV7	0.45 0.33 0.77	0.53 0.44 0.83	0.84 0.75 0.92

As shown in Table 5, 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%.

TABLE VI. 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**

As shown in Table 6, we have two 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. For instance, if we take CV1, CV3, CV6 and CV7; 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.

TABLE VII. 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 or 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 7, 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.

VI. Conclusions and Future Work

In this paper, we have proposed a job post and resume classification system (JRC) based on coupling an integrated skills knowledge base and an automatic matching procedure between candidate resumes and their corresponding job postings. The proposed system first utilizes section-based segmentation module in order to segment the resumes and extract a set of skills that are used in the classification process. Next, the system exploits an integrated skills knowledge base for carrying out the classification task. As indicated in section V, 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. In the future work, we plan to utilize the extracted information from applicants’ resumes to dynamically generate user profiles to be further used for recommending jobs to job seekers.

References

[1] E Faliagka, L Iliadis, I Karydis, M Rigou, S Sioutas, A Tsakalidis, and G Tzimas, “ On-line consistent ranking on e-recruitment: seeking the truth behind a well-formed CV,” The Artificial Intelligence Review, 42(3), 515, 2014.

[2] A Kmail, M Maree, M Belkhatir, and S Alhashmi "An Automatic Online Recruitment System based on Exploiting Multiple Semantic Resources and Concept-relatedness Measures," Proceedings of the EEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 620-627, 2015.

[3] J Chen, Z Niu, H Fu, "A Novel Knowledge Extraction Framework for Resumes Based on Text Classifier," Proceedings of the International Conference on Web-Age Information Management. Springer International Publishing, pp. 540-543, 2015.

[4] C Hauff, G Gousios, "Matching GitHub developer profiles to job advertisements." Proceedings of the 12th Working Conf. on Mining Software Repositories, pp. 362-366, 2015.

[5] T Schmitt, P Caillou, M Sebag, "Matching Jobs and Resumes: a Deep Collaborative Filtering Task," Proc. of the 2nd Global Conf. on Artificial Intelligence, pp.1-14, 2016.

[6] S Mehta, R Pimplikar, A Singh, LR Varshney and K. Visweswariah, "Efficient multifaceted screening of job applicants," Proceedings of the 16th International Conference on Extending Database Technology. ACM, pp. 661–671, 2013.

[7] S Al-Otaibi and M Ykhlef, "Job Recommendation Systems for Enhancing E-recruitment Process", in Proceedings of the International Conference on Information and Knowledge Engineering (IKE), Las Vegas Nevada, USA, pp. 433-439, 2012.

[8] The International Association of Employment Web Sites (IAEWS), avaliable from: http://www.icmaonline.org/international-association-of-employment-web-sites, Date Visited: June 20, 2017.

[9] A Kmail, M Maree, and M Belkhatir, "MatchingSem: Online recruitment system based on multiple semantic resources," Proceedings of the 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), IEEE, pp. 2654-2659, 2015.

[10] W Hong, S Zheng, H Wang, J Shi, “A Job Recommender System Based on User Clustering,” Journal of Computers, vol. 8(8), pp. 1960-1967, 2013.

[11] V.S Kumaran, and A Sankar, “Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping EXPERT,” Int. J. Metadata Semantics. Ontologies, vol. 8(1), pp. 56-64, 2013.

[12] R Kessler, N Béchet, JM Torres-Moreno, M Roche and M. El-Bèze, “Job Offer Management: How Improve the Ranking of Candidates”, in Foundations of Intelligent Systems, J. Rauch, et al., Editors. Springer Berlin Heidelberg, pp. 431-441, 2009.

[13] K Yu, G Guan, and M Zhou, "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, 2005.

[14] F Javed, Q Luo, M McNair, F Jacob, M. Zhao, and TS. Kang, "Carotene: A Job Title Classification System for the Online Recruitment Domain," Proceedings of the IEEE First International Conference on Big Data Computing Service and Applications (BigDataService), pp. 286-293, 2015.

[15] R Kessler, N Béchet, M Roche, J. M Torres-Moreno, and M El-Bèze, “A hybrid approach to managing job offers and candidates,” Information Processing & Management, 48(6), 1124-1135, 2012.

[16] J.Martinez-Gil, A.L. Paoletti, and K.D. Schewe, “A smart approach for matching, learning and querying information from the human resources domain,” In East European Conference on Advances in Databases and Information Systems, Springer International Publishing, pp. 157-167, 2016.

[17] M Fazel-Zarandi and M S Fox, “Semantic matchmaking for job recruitment an ontolgy based hybrid approach,” In Proceedings of the 3rd International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web at the 8th International Semantic Web Conference, Washington D. C., USA, 2010.

[18] S Clyde, J Zhang, and CC Yao, "An object-oriented implementation of an adaptive classification of job openings," Proceedings of the 11th Conference on Artificial Intelligence for Applications, IEEE, pp. 9-16, 1995.

[19] About Occupational Information Network (O*NET). Available from: https://onet.rti.org/about.cfm. Date Visited: February 5, 2016.

[20] R Kessler, J Torres-Moreno, and M El-Bèze, “E-Gen: automatic job offer processing system for human resources,” in Proceedings of the artificial intelligence 6th Mexican international conference on Advances in artificial intelligence, Springer-Verlag: Aguascalientes, Mexico, pp. 985-995, 2007.

[21] G.A Miller, “WordNet: a lexical database for English,” Comm. ACM, vol. 38(11), pp. 39-41, 1995.

[22] J Hoffart, FM Suchanek, K Berberich, E. Lewis-Kelham, G. De Melo, and G. Weikum, “YAGO2: exploring and querying world knowledge in time, space, context, and many languages”, in Proceedings of the 20th international conference companion on World Wide Web, ACM: Hyderabad, India, pp. 229-232, 2011.

[23] A. Zaroor, M. Maree, and M. Sabha, “A Hybrid Approach to Conceptual Classification and Ranking of Resumes and Their Corresponding Job Posts,” In: Czarnowski I., Howlett R., Jain L. (eds) Intelligent Decision Technologies 2017. IDT 2017. Smart Innovation, Systems and Technologies, vol 72. Springer, Cham.