A Hybrid Approach to Conceptual Classification and Ranking of Resumes and Their Corresponding Job Posts

Abeer Zaroor[1], Mohammed Maree[2](✉), and Muath Sabha[2]

[1] Computer Science Department, Faculty of Engineering and Information Technology, The Arab American University, Jenin, Palestine
abeerzaroor.aauj@gmail.com

[2] Information Technology Department, Faculty of Engineering and Information Technology, The Arab American University, Jenin, Palestine
{mohammad.maree,muath.sabha}@aauj.edu

Abstract. Due to the constant growth in online recruitment, job portals are starting to receive thousands of resumes in diverse styles and formats from job seekers who have different fields of expertise and specialize in various domains. Accordingly, automatically extracting structured information from such resumes is needed not only to support the automatic matching between candidate resumes and their corresponding job offers, but also to efficiently route them to their appropriate occupational categories to minimize the effort required for managing and organizing them. As a result, instead of searching globally in the entire space of resumes and job posts, resumes that fall under a certain occupational category are only those that will be matched to their relevant job post. In this research work, we present a hybrid approach that employs conceptual-based classification of resumes and job postings and automatically ranks candidate resumes (that fall under each category) to their corresponding job offers. In this context, we exploit an integrated knowledge base for carrying out the classification task and experimentally demonstrate - using a real-world recruitment dataset- achieving promising precision results compared to conventional machine learning based resume classification approaches.

Keywords: Online recruitment · Concept-based classification · Job matching

1 Introduction

In recent years, online recruitment has expanded significantly [1, 2]. This expansion has led to a continuous growth in the number of job portals and hiring agencies on the Internet [3, 4]. It has also led to a constant increase in the number of job seekers searching for new career opportunities [5, 6]. Accordingly, online job portals are starting to receive thousands of resumes (in diverse styles and formats) from job seekers who have different fields of expertise and specialize in different domains [7]. Several approaches have been proposed to support the automatic matching between candidate resumes and their corresponding job offers [8–10]. Examples on these approaches are automatic keyword-based resume matching techniques [8] machine

I. Czarnowski et al. (eds.), *Intelligent Decision Technologies 2017*,
Smart Innovation, Systems and Technologies 72, DOI 10.1007/978-3-319-59421-7_10

leaning based approaches [9], and semantics-based techniques [10]. The main goal of these approaches is achieving high precision ratios i.e. finding the best candidates for a given job post ignoring the cost (run time complexity) of the matching process. Other systems have attempted to reduce the cost issue by segmenting the content of both resumes and job posts and finding matches between important segments in both instead of matching between the content of the whole resumes and job posts. For instance, the authors of [11, 12] propose using machine learning algorithms (Support vector machines (SVM) [11] and Hidden Markov Model (HMM) [12]) 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 the authors of [8] 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 [8], every newly obtained resume still needs to be matched with all job offers in the corpus. To overcome this issue, other researchers propose utilizing machine learning-based techniques to classify job posts and resumes into occupational categories prior conducting the matching task [13]. However, as pointed in [2], these techniques suffer from low precision ratios and produce high error rates. To address the issues associated with the previously highlighted techniques, 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. We summarize the contributions of our work as follows:

- Automatic Occupational Category based Classification of Resumes and Job Postings.
- Employing a Section-based Segmentation heuristic by exploiting an integrated occupational categories knowledge base.

The remainder of this paper is organized as follows. In Sect. 2, we introduce the work related. Section 3 describes the overall architecture of the proposed system. Experimental validation of the effectiveness and efficiency of the proposed system is presented in Sect. 4. In Sect. 5, we discuss the conclusions and outline future work.

2 Related Work

Many techniques have been proposed to precisely match between candidate resumes and their corresponding job offers [14, 15]. However, little attention has been paid to addressing problems associated with automatic resumes and job posts classification [16]. For instance, when an employer seeks a “Web Developer” that falls under “Web Development” occupational category, the conventional 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 post instead of matching only those that fall under the corresponding occupational category (“Web Development” in this scenario). To address this issue, the authors of [12] have proposed resume Information Extraction (IE) with Cascaded Hybrid Model. This system employs HMM and SVM classification algorithms in order to annotate segments of resumes with the appropriate category, taking the advantage of the resume contextual structure where the related information units usually occur in the same textual segments. Accordingly, resumes pass through two layers; where in the first layer a HMM is applied to segment the entire resume into consecutive blocks and each block is annotated with a category such as Personal Information, Education, and Research Experience. After that, in the second layer SVM is applied in order to extract the detailed information from the blocks that have been labeled with Education and Personal Information respectively. However, a large fraction of the produced results by this system suffer from low precision since the information extraction process passes through two loosely-coupled stages. Another system (E-gen) [17] has been built in order to automate the recruitment process by segmenting and classifying job posts. First, job posts are transformed into vector space representations. Then, SVM classification algorithm is utilized to annotate segments of job posts with the appropriate topics and features. A correction algorithm is further applied because during the classification process some segments were incorrectly classified [17]. The main drawback of this system is the time needed to pre-process and post-process job posts in order to minimize the error and maximize the matching probability. On the other hand, JobDiSC system [18] attempts to classify job openings automatically by employing a standard classification scheme called Dictionary of Occupational Titles (DOT). The proposed system comprises three main modules: (1) Parser/Analyzer: which creates an unclassified job opening for each job listings captured from electronic forms prepared by employers. (2) Learning System to automatically generate classification rules from a set of pre-classified job openings and (3) Classifier that assigns one or more class for each job post depending on its confidence level for any potential class assigned to it. 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 [19].

3 Overview of the Architecture of the Proposed System

In this section, we present an overview of the architecture of the proposed system and discuss its main constituents.

As shown in Fig. 1, when a user submits a CV, the system directs it to the Section-based Segmentation module which is used to extract personal, education, experience information and employment history, in addition to a list of candidate matching concepts. After that, the Filtration module refines the concept lists by removing concepts that have low tf-idf [8] weights and concepts that don't contribute to the semantics of each segment. Next, the Classification module takes a set of skills from both resumes and job posts as an input in order to classify job posts and resumes to their corresponding occupational categories. At this step, we exploit an integrated knowledge base which combines two main semantic resources: DICE[1] and O*NET[2]. More details on these resources will be provided in Sect. 3.1. Then, the Category-based

[1] http://www.dice.com/skills.

[2] http://www.onetcenter.org/taxonomy/2010/list.html.

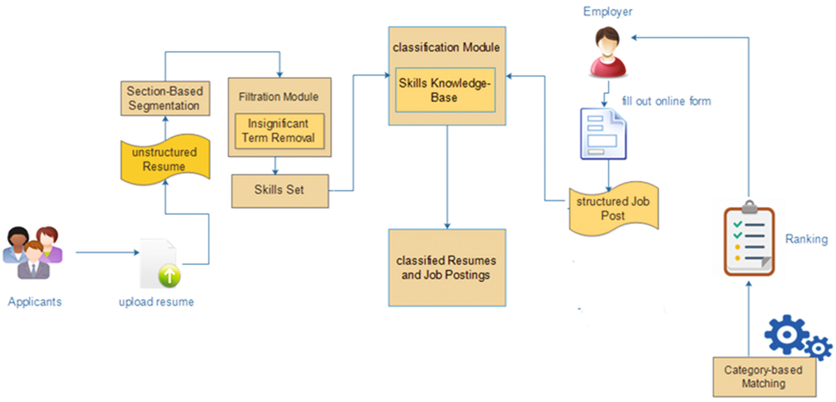

Fig. 1. Overall architecture of the proposed system

Matching module takes lists of concepts from both resumes and job posts to construct semantic networks by deriving the semantic relatedness between them using semantic resources. Finally, the matching algorithm takes the semantic networks as an input and produces the measures of semantic closeness between them as an output. The following sections detail the steps carried out in each module of the system.

3.1 Section-Based Segmentation and Conceptual Classification Modules

During this phase, an automatic extraction of segments such as Education, Experience, Loyalty and other Employment information such as Company name, Applicant Role in the company, Date of designation, Date of resignation and Loyalty is performed. In this context, the system matches segments of resumes to their relevant segments of job posts instead of matching the whole resumes and job posts. During this phase, unstructured resumes are converted into segments (semi-structured document) based on employing Natural language processing techniques (NLP) and rule-based regular expressions. As detailed in [7], the NLP steps are: document splitting, n-gram tokenization, stop word removal, part-of-Speech-Tagging (POST) and Named Entity Recognition (NER). Table 1 shows an example that illustrates the process of segmenting a sample resume.

In order to classify both resumes and job posts, we utilize an integrated knowledge-base which combines Dice skills center (henceforth stated as DICE) and a standardized hierarchy of occupation categories known as the Occupational Information Network (O*NET) (henceforth stated as O*NET). In this context, we use DICE to classify skills that belong to Information and communication technologies (ICT), and economy field because we empirically found 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. Table 2 shows a comparative analysis between Dice and O*NET classification.

Table 1. Results of the Section-based Segmentation Module

Input CV	Segmentation Result	Candidate Terms	Filtered List
Input CV: I have 2 years of experience as a web designer And I have the following skills android, ios, PHP, HTML, JavaScript, CSS, JQuery, VB, .NET and c# Education: B.Sc. in CS. Employment Details I worked as web designer in ASAL Company from 2008 to 2010.	<ApplicantData> <Experience> <Years>2</Years> <Field>web designer</Field> </Experience> <Education> <Degree> B.Sc.</Degree> <Field>CS</Field> </Education> <EmploymentHistory> <role> web designer</role> <companyName>ASAL</companyName> <FromDate>2008</fromDate> <ToDate>2010</ToDate> <loyalty> 2</loyalty> </EmploymentHistory> <Skills> android, ios, PHP, HTML, JavaScript, CSS, JQuery, VB, .NET , c# </Skills> </ApplicantData>	android ios HTML Javascript Jquery designer Asal Web php skills experience company VB .NET c# CSS	android ios HTML Javascript Jquery designer web php VB .NET c# CSS

Table 2. Comparative Analysis between DICE and O*NET Classification

	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 2, 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.

3.1.1 Skill-Based Resume Classification Module

In this module each skill in the skills set is submitted to the exploited skills knowledge base sequentially in order to obtain a list of candidate job categories. As shown in Fig. 2, the skill "android" is first submitted to the exploited skills knowledge base. For this skill, the knowledge base returns one occupational category, that is "Software

Development/Mobile Development". Next, the rest of the skills in the skills set are submitted to the exploited knowledge base. 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). Accordingly, for the skills set that we have obtained from CV1, the occupational category "Web Development" gets the highest weight, followed by "Software Development/Application Development" and then "Soft-ware Development/Mobile Development" respectively.

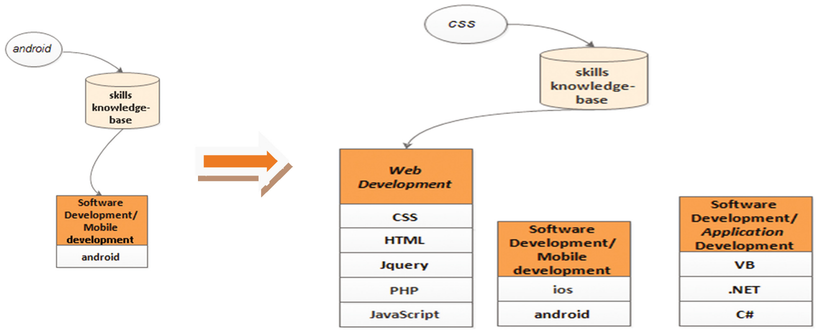
android
skills knowledge-base
Software Development/ Mobile development
android
css
skills knowledge-base
Web Development
CSS
HTML
Jquery
PHP
JavaScript
Software Development/ Mobile development
ios
android
Software Development/ Application Development
VB
.NET
C#

Fig. 2. List of occupational categories generated for the sample CV

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

Algorithm 1. A weighted List of Resume category

```
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 ← ⟨ ⟩;
3:  categoryList ← ⟨ ⟩;
4:  for each skill ∈ skills [$s_1$, $s_2$, …,$s_n$] do
5:      categoryList ← GET_FROM_KB (skill);
6:      while categoryList ≠ 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 skills knowledge-base respectively. 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

Table 3. Occupational categories returned for the CV used Example 1

Job category	Skills
Software Development/Mobile Development	Android, ios
Software Development/Application Development	.Net, c#, VB
Web Development	CSS, html, php, Javascript, jquery

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 3 shows each occupational category assigned to its corresponding skills.

3.1.2 Job Post Classification Module

In the Job Post Categorization 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 skill 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 Sect. 4.2.

3.2 Matching Resumes and Their Corresponding Job Postings

In the same fashion as proposed by the authors of [7], we use the same semantic resources (WordNet ontology [20] and YAGO2 ontology [21]) and statistical concept-relatedness measures to derive the semantic aspects of resumes and job posts. It is important to mention that we have considered additional weighting parameters such as: loyalty parameter (degree of devotion to the company that the candidate worked or currently working in) in order to increase the effectiveness of the matching process. It is also important to point out that we use a dynamic threshold value to fairly handle the loyalty parameter as shown in the following scoring formulae:

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

Where:

- **S:** is the relevance score result.
- **Sr:** is the set of applicant's skills.
- **RSj:** the required skills in the job post.
- **Er:** is the set of concepts that describe applicant educational information.
- **REj:** is the set of concepts from the required educational information in job post.
- **Xr:** set of concepts that describe applicant experience information.
- **RXj:** concepts that represent the required experience information in the job post.

- **Yw:** the total number of employment years.
- **Cw:** 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%. The results of using the scoring formula are detailed in the next section (Sect. 4.3).

4 Experimental Results

This section describes the experiments that we have conducted to evaluate the efficiency and the effectiveness of the proposed system. In order to evaluate the accuracy of the proposed system, we collected a data set of 2000 resumes downloaded from:

- http://www.amrood.com/resumelisting/listallresume.htm and
- http://www.indeed.com/resumes

and used 10,000 different job posts obtained from:

- http://jobs.monster.com,
- http://www.shine.com/job-search and
- http://www.careerbuilder.com.

The collected resumes are unstructured documents in different document formats such as (.pdf) and (.doc) and we considered job posts as structured document having sections (job title, description, required skills, required years and field of experience, required education qualification and additional desired requirement). The experiments of our system prototype show that the classification process for the training data of resumes and job posts took 6 h on average on a PC with dual-core CPU (2.1 GHz) and (4 GB) RAM. The used operating system is Windows 8.1.

4.1 Execution Time for Matching Resumes with Corresponding Job Post with/Without Classification

In this section, we compare the results produced by our system with MatchingSem system [7] which is semantics-based automatic recruitment system. As shown in Fig. 3, our system Job Resume Classifier (JRC) was able to achieve better results than MatchingSem System. And this is due to the fact that, unlike MatchingSem, we only match job posts with corresponding resumes that fall under the same occupational category instead of searching globally in the entire space of resumes. For instance, "Front-End Developer" job post costs 6 h of execution time for finding the best candidate using MatchingSem, while it only took 1 h in JRC since resumes that fall under "Web Development" category were considered in the matching process.

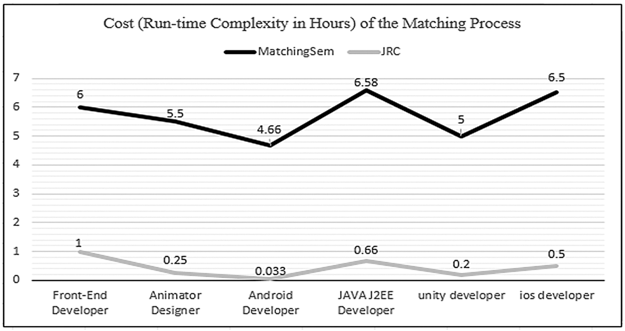

Fig. 3. Cost (Run-time complexity in hours) of the matching process

4.2 Experiments of Job Post Classification

In this section we present job post classification. As mentioned in Sect. 3.1.2 we have used job title and required skills in the classification process. In Table 4, we have used 7 job posts in order to clarify the classification process.

Table 4. Job post classification results

Job title	Required skills	Job classification	Weight
Front End Web Developer	Backbone, CSS3, HTML5, JSON, JQuery, JavaScript, Angular	Software Development/Web Development	**100%**
Unity Developer	3D, Unity	Software Development/Interactive Multimedia	**85%**
	Objective-C, Xcode	Software Development/Mobile development	15%
Animator Designer	3D, texture, compositing, lighting, animation	Software Development/Interactive Multimedia	**86.6%**
	Illustrator, After Effects, Premiere Pro, Maya	Design/Multimedia Design	13.4%
IT Support Specialist	printers, network routing protocols, TCP/IP, smartphones, VOIP, laptops, VLANs, desktops	IT Administration/Technical Support	**94%**
	SQL Express, SQL Server	Data/Databases	6%
IT Technician	DHCP, DNS, Troubleshooting, Installation, Configuration	IT Administration/Technical Support	**100%**

(*continued*)

Table 4. (*continued*)

Job title	Required skills	Job classification	Weight
Database Administrator	Oracle Data Guard, Oracle RMAN, RDBMS, Oracle RAC, ASM, Oracle 11 g, PeopleSoft DBA	Data/Databases	**92.5%**
	EMC Storage, Symmetrix, DMX	IT Administration/Technical Support	7.5%
Data Entry Assistant	Typing, Filing, Clerk	Industry-specific/General skills	**79%**
	Drupal, HTML, CSS	Software Development/Web Development	9%
	MS Word, Excel, PowerPoint, Outlook	Industry-specific/Microsoft Office	12%

As shown in Table 4, we can see that "Front End Web Developer" job post falls under "Software Development/Web Development" 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, "Unity Developer" job post falls under "Software Development/Interactive Multimedia" space with weight 85%, 70% for job title under "Software Development/Interactive Multimedia" category. When we submit "3D and unity" skills we see that they fall under the same space as job title with weight 15%, but "Objective-C, Xcode" skills fall under "Software Development/Mobile Development" space with weight 15%. And the same for "Data Entry Assistant" job post, that falls under three categories: "Industry-specific/General skills" with weight 79%, "Industry-specific/Microsoft Office" with weight 12%, and "Software Development/Web Development" with weight 9%.

4.3 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 corresponding job post.

As shown in Table 5, we match job posts with their corresponding resumes that fall under the same occupational categories. For instance, "Android Developer" job post is matched only with resumes that fall under "Mobile Development" category. As such, CV1 and CV3 are matched with "Android developer" and "Web developer" job posts. And this is because these CVs exist in both "Mobile Development" and "Web Development" categories. However, the matching score differ from job post to another.

Table 5. Precision results of matching resumes with corresponding job post by the system

Occupation category	Job title	Resume index	Manual score	Auto score	Precision
Software Development\ Mobile development	Android developer	CV1	0.80	0.80	1.00
		CV2	0.80	0.76	0.95
		CV3	0.05	0.09	0.55
Software Development/Web development	Web developer	CV4	0.42	0.53	0.79
		CV3	0.70	0.75	0.93
		CV1	0.40	0.47	0.85
Software Development/Interactive Multimedia	Unity developer	CV5	0.45	0.40	0.89
		CV6	0.47	0.61	0.77
		CV7	0.57	0.66	0.86

For instance, CV3 achieved a very low matching score when matched with Android Developer job post (0.05 manual score, 0.09 automatic score), but CV1 achieved better score for the same job post (0.8 manual score, 0.8 automatic score). On the other hand, CV3 achieved better results than CV1 when it was matched with "Web developer" job post (0.70 manual score, 0.75 automatic score) and this is because CV3 falls under "Web developer" with weight 80% and falls under "Mobile Development" with weight 35%.

In order to validate our proposal and evaluate the quality of the produced results by our system, we have compared our system with one of the previously proposed systems, called MatchingSem [7]. Table 6, shows the results produced by our system compared to MatchingSem system.

Table 6. Comparison with MatchingSem System

Job title	Resume index	JRC precision	MatchingSem precision
Android Developer	**CV1**	**1.00**	0.91
	CV2	**0.95**	0.84
	CV3	**0.55**	0.50

As shown in Table 6, for the job title "Android Developer" and the three CVs, the quality of the produced results (namely, the precision) by our system is higher than MatchingSem system. The reason behind this is that – 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 Conclusions and Future Work

In this research work, we have introduced a hybrid approach that employs conceptual-based classification of resumes and job postings and automatically matches candidate resumes to their corresponding job postings that fall under each occupational category. The proposed system first utilizes NLP techniques and regular expressions 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. 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. Faliagka, E., Iliadis, L., Karydis, I., Rigou, M., Sioutas, S., Tsakalidis, A., Tzimas, G.: On-line consistent ranking on e-recruitment: seeking the truth behind a well-formed CV. Artif. Intell. Rev. **42**(3), 515–528 (2014)
2. Kessler, R., Béchet, N., Roche, M., Torres-Moreno, J., El-Bèze, M.: A hybrid approach to managing job offers and candidates. Inf. Process. Manage. **48**(6), 1124–1135 (2012)
3. Chen, J., Niu, Z., Fu, H.: A novel knowledge extraction framework for resumes based on text classifier. In: Proceedings of the International Conference on Web-Age Information Management, pp. 540–543. Springer International Publishing (2015)
4. Schmitt, T., Philippe C., Michele, S.: Matching jobs and resumes: a deep collaborative filtering task. In: Proceedings of the 2nd Global Conference on Artificial Intelligence, pp. 1–14 (2016)
5. Hauff, C., Georgios G.: Matching GitHub developer profiles to job advertisements. In: Proceedings of the 12th Working Conference on Mining Software Repositories, pp. 362–366 (2015)
6. Pimplikar, R., Singh, A., Varshney, R., Visweswariah, K.: Efficient multifaceted screening of job applicants. In: Proceedings of the 16th International Conference on Extending Database Technology, pp. 661–671. ACM (2013)
7. Kmail, A., Maree, M., Belkhatir, M., Alhashmi, S.: An automatic online recruitment system based on exploiting multiple semantic resources and concept-relatedness measures. In: Proceedings of the IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 620–627 (2015)
8. Kmail, A., Maree, M., Belkhatir, M.: MatchingSem: online recruitment system based on multiple semantic resources. In: Proceedings of the 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 2654–2659. IEEE (2015)
9. Hong, W., et al.: A job recommender system based on user clustering. J. Comput. **8**(8), 1960–1967 (2013)
10. Kumaran, V.S., Sankar, A.: Towards an automated system for intelligent screening of candidates for recruitment using ontology mapping EXPERT. Int. J. Metadata Semant. Ontol. **8**(1), 56–64 (2013)

11. Kessler, R., Béchet, N., Torres-Moreno, J.M., Roche, M., El-Bèze, M.: Job offer management: how improve the ranking of candidates. In: Rauch, J. et al.(eds.) Foundations of Intelligent Systems, pp. 431–441. Springer, Heidelberg (2009)
12. Yu, K., Guan, G., Zhou, M.: Resume information extraction with cascaded hybrid model. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 499–506. Association for Computational Linguistics (2005)
13. Javed, F., et al: Carotene: a job title classification system for the online recruitment domain. In: Proceedings of the IEEE First International Conference on Big Data Computing Service and Applications (BigDataService), pp. 286–293 (2015)
14. Yi, X., Allan, J., Croft, W.B.: Matching resumes and jobs based on relevance models. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 809–810. ACM, Amsterdam (2007)
15. Faliagka, E., et al: Application of machine learning algorithms to an online recruitment system. In: The Seventh International Conference on Internet and Web Applications and Services, ICIW 2012, pp. 215–220 (2012)
16. Bekkerman, R., Gavish, M.: High-precision phrase-based document classification on a modern scale. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 231–239. ACM (2011)
17. Kessler, M., et al: 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, pp. 985–995. Springer, Aguascalientes (2007)
18. Clyde, S., Zhang, J., Yao, C.-C.: An object-oriented implementation of an adaptive classification of job openings. In: Proceedings of the 11th Conference on Artificial Intelligence for Applications, pp. 9–16. IEEE (1995)
19. About Occupational Information Network (O*NET). https://onet.rti.org/about.cfm. Date Visited 5 Feb 2016
20. Miller, G.A.: WordNet: a lexical database for English. Comm. ACM **38**(11), 39–41 (1995)
21. Hoffart, J., et al.: 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, pp. 229–232. ACM, Hyderabad (2011)