Taxonomy development and its impact on a self-learning e-recruitment system

Evanthia Faliagka[1], Ioannis Karydis[3], Maria Rigou[1], Spyros Sioutas[3], Athanasios Tsakalidis[1], and Giannis Tzimas[2]

[1]Computer Engineering and Informatics Dept., University of Patras, Patras, Greece
{faliagka,rigou}@ceid.upatras.gr
[2]Dept. of Applied Informatics in Management & Economy, Technological Educational Institute of Messolonghi, Messolonghi, Greece
{tsak,tzimas}@cti.gr
[3]Dept. of Informatics, Ionian University, 49100, Kerkyra, Greece
{karydis,sioutas}@ionio.gr

Abstract. In this work we present a novel approach for evaluating job applicants in online recruitment systems, using machine learning algorithms to solve the candidate ranking problem and performing semantic matching techniques. An application of our approach is implemented in the form of a prototype system, whose functionality is showcased and evaluated in a real-world recruitment scenario. The proposed system extracts a set of objective criteria from the applicants' LinkedIn profile, and compares them semantically to the job's prerequisites. It also infers their personality characteristics using linguistic analysis on their blog posts. Our system was found to perform consistently compared to human recruiters, thus it can be trusted for the automation of applicant ranking and personality mining.

Keywords: e-recruitment; personality mining; recommendation systems; data mining

1 Introduction

In the recent years an increasing number of people turn to the web for job seeking and career development while a lot of companies use online knowledge management systems to hire employees, exploiting the advantages of the World Wide Web [1]. The information systems used to support these tasks are termed e-recruitment systems and automate the process of publishing position openings and receiving applicant CVs, thus allowing Human Resource (HR) agencies to target a very wide audience at a small cost. At the same time this situation may as well prove overwhelming to HR agencies that need to allocate human resources for manually assessing the candidate resumes and evaluating the applicants' suitability for the positions at hand. Automating the process of analyzing the applicant profiles to determine the ones that best fit the specifications of a given job position could lead to a significant gain in terms of efficiency. For example, it is indicative that SAT Telecom India reported 44% cost savings and a drop in average time needed to fill a vacancy from 70 to 37 days [2] after deploying an e-recruitment system.

Several e-recruitment systems have been proposed with an objective to speed-up the recruitment process, leading to a better overall user experience. *E-Gen*

system [3] performs analysis and categorization of unstructured job offers (i.e. in the form of unstructured text documents), as well as analysis and relevance ranking of candidates. In contrast to a free text description, the usage of a common "language" in the form of a set of controlled vocabularies for describing the details of a job posting would facilitate communication between all parties involved and would open up the potential of the automation of various tasks within the process [4]. Another benefit from having postings annotated with terms from a controlled vocabulary is that the terms can be combined with background knowledge about an industrial domain. Job portals could offer semantic matching services which would calculate the semantic similarity between job postings and applicants' profiles based on background knowledge about how different terms are related. For example, if *Java* programming skills are required for a certain job and an applicant is experienced in *Delphi*, the matching algorithm would consider this person's profile a better match than someone else's who has the skill *SQL*, since *Delphi* and *Java* are more closely related than *SQL* and *Java*. This approach allows for comparison of job position postings and applicants' profiles using background knowledge instead of merely relying on the containment of keywords, like traditional search engines do.

CommOn framework [5] applies Semantic Web technologies in the field of HR Management, while *HR-XML* can partly support the "standardized" representation of competency profiles [6]. In this framework the candidate's personality traits, determined through an online questionnaire which is filled-in by the candidate, are considered for recruitment. In order to match applicants with job positions these systems typically combine techniques from classical IR and recommender systems, such as relevance feedback [3], semantic matching in job seeking and procurement tasks [7], Analytic Hierarchy Process [8, 9] and NLP technology used to automatically represent CVs in a standard modeling language [10]. These methods, although useful, suffer from the discrepancies associated with inconsistent CV formats, structure and contextual information. In addition approaches that incorporate ontological information for determining the degree of position-to-applicant matching face significant complexity problems concerning the development of the required ontological structure and associations. This problem appears even when trying to reuse available ontologies (ontology discovery through evaluation to ontology integration and merging), a task that requires considerable manual work [11]. What's more, these methods are unable to evaluate some secondary characteristics associated with CVs, such as style and coherence, which are very important in CV evaluation.

Such approaches attempt to match terms found in CV descriptions to job position descriptions. In this work a different approach is adapted in the sense that the semantic matching primarily concerns applicant skills as denoted in the respective LinkedIn profile descriptions. Applicant skills are then semantically associated with equivalent concepts from job descriptions as specified by the recruiter, who constructs a list of required job position skills using a predefined IT skills hierarchy. Hierarchy skills are contained in the LinkedIn skills but also the hierarchy integrates even broader skills ending up to the root of "IT skills".

The system described in this work, attempts to solve the candidate ranking problem by applying a set of supervised learning algorithms in combination with a semantic skills matching mechanism, for automated e-recruitment. It is an integrated company oriented e-recruitment system that automates the candidate pre-screening and ranking process. Applicant evaluation is based on a predefined set of objective criteria, which are directly extracted from the applicant's LinkedIn profile. What's more, the candidate's personality characteristics, which are automatically extracted from his social presence [12], are taken into account in his evaluation. Our objective is to limit interviewing and background investigation of applicants solely to the top candidates identified from the system, so as to increase the efficiency of the recruitment process. The system is designed with the aim of being integrated with the companies' Human Resource Management infrastructure, assisting and not replacing the recruiters in their decision-making process.

2 System Overview

In this work, we have implemented an integrated company oriented e-recruitment system that automates the candidate evaluation and pre-screening process. Its objective is to calculate the applicants' relevance scores, which reflect how well their profile fits the position's specifications. In this Section we present an overview of the proposed system's architecture and candidate ranking scheme.

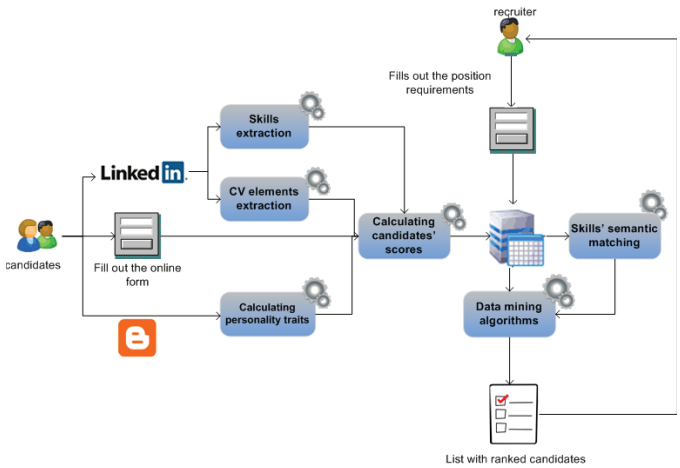

Fig. 1. System's Architecture.

2.1 Architecture and implementation

The proposed e-recruitment system implements automated candidate ranking based on a set of credible criteria, which will be easy for companies to integrate with their existing Human Resources Management infrastructure. In this study we focus on 5 complementary selection criteria, namely: Education (in years of formal academic training), Work Experience (in months of related experience), Loyalty (average number of years spent per job), Extraversion and skills. The

system's architecture, which is shown in Figure 1, consists of the following components:

- *Job Application module*: Implements the input forms that allow the candidates to apply for a job position.
- *Personality mining module*: If the candidate's blog URL is provided, applies linguistic analysis to the blog posts deriving features reflecting the author's personality.
- *Semantic matching*: Calculates the semantic distance between candidate skills and prior experience, as extracted from the respective LinkedIn profile and job position requirements.
- *Applicant Grading module*: Combines the candidate's selection criteria to derive the candidate's relevance score for the applied position. The grading function is derived through supervised learning algorithms.

The proposed e-recruitment system was fully implemented as a web application, in the Microsoft .Net development environment. Job applicants are given the option to authenticate using their LinkedIn account credentials to apply for one or more of the available job positions. This allows the system to automatically extract the selection criteria required for candidate pre-screening from the applicants' LinkedIn profile, so the user experience is streamlined. As part of the job application process, candidates are asked to fill-in the feed URI of their personal blog. This allows our system to syndicate the blog content and calculate the extraversion score with the personality mining technique presented in Section 2.3.

On the recruiter's side, the system after authentication provides access rights to post new job positions and evaluate job applicants. In the "rank candidates" menu, the recruiter is presented with a list of all available job positions and the candidates that have applied for each one of them. Upon the recruiter's request, the system estimates applicants' relevance scores and ranks them accordingly. This is achieved by calling the corresponding Weka classifier, via calls to the API provided by Weka software [13]. The recruiter can modify the candidate ranking, by assigning new relevance scores to the candidates. This will improve the future performance of the system, as the recruiter's suggestions are incorporated in the system's training set and the ranking model is updated. It must be noted here that the ranking model is initialized as a simple linear combination of the selection criteria, until sufficient input is provided from the recruiters to build a training set.

2.2 Semantic matching

In the previous version of the system [14] it was found that except from senior positions that required domain experience and specific qualifications, our system performed consistently with a Pearson's correlation of up to 0.85. The present expanded version of the system tackles the problem of specific qualifications and experience in senior positions and demonstrates improved accuracy (as will be presented in Section 3) by deploying semantic matching technologies.

The data exchange between employers, applicants and job portals in a Semantic Web-based recruitment scenario is based on a set of vocabularies which

provide shared terms to describe occupations, industrial sectors and job skills [15]. Semantic matching is a technique which combines annotations using controlled vocabularies with background knowledge about a certain application domain. In our case, the domain specific knowledge is represented by a taxonomy of IT skills (Figure 2). A taxonomy is defined as a set of categories or terms organized into a hierarchy with parent-child relationships and implied inheritance, meaning that a child term (ie, `C`) has all of the characteristics of its parent term (ie, `Structured`). A taxonomy only contains broader and narrower relationships.

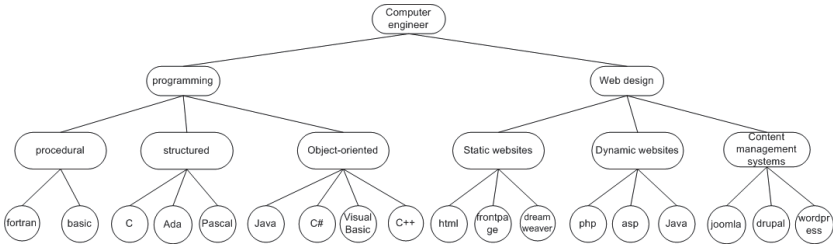

Fig. 2. Part of the implemented IT skills taxonomy.

The implemented taxonomy serves a dual role:

1. Matches the applicants' skills as stated in the respective LinkedIn profile and the job position requirements as specified in the job description and rejects all candidates that don't fulfill the requirements.
2. Searches the text of job title and job description of the job experience section in the applicant's LinkedIn profile and identifies terms corresponding to skills required by the recruiter. Thus, in the current system version, the calculation of the job experience criterion takes into account only the job experience that concerns relative competencies.

It is important to clarify that in both cases we do not use a simple keyword search but a concept search. First, for the specific job position a skills search is applied to the candidate skills, as specified in the respective LinkedIn profile (Figure 3). In most cases a recruiter does not ask for specialized competencies but resorts to more general qualifications, such as object-oriented programming (as opposed to Java or C#). In this case the proposed algorithm searches the hierarchy tree and identifies the leaves with the node of the skill required by the recruiter as their lowest (nearest) common ancestor (for instance, object-oriented programming). Next, the identified leaves are examined to determine if there is a match with the skills stated by the candidate. In the case that these is no match then the candidate is excluded from the ranking process.

For those candidates that were found to have the necessary skills a second search is conducted to determine whether one or more of the candidate's past work experience belongs to the same domain of expertise as the job position of interest. The algorithm applied for this purpose can be briefly described as follows: Let $S1$ be the skills corresponding to a past job position $E1$ as stated in

the work experience section (title or description text) of the respective LinkedIn profile. Also, let S be the skills required by the current job position, corresponding to the job position domain and may be found at any level of the hierarchy. If there is an overlap between S and $S1$ ($S \cap S1 \neq \emptyset$), the past job position $E1$ is regarded as relevant and thus is taken into account in the relevant job experience calculations.

Fig. 3. LinkedIn skills example.

2.3 Personality mining

Previous works have shown that by applying linguistic analysis to blog posts, the author's personality traits can be derived [16] as well as his mood and emotions [17]. The text analysis in these works is performed with LIWC (Linguistic Inquiry and Word Count) system, which analyzes written text samples and extracts linguistic features that act as markers of the author's personality. Pennebaker and King [18] have found significant correlations between these frequency counts and the author's personality traits, as measured by the Big-Five personality dimensions.

In this work, we focus on the extraversion personality trait, due to its importance in candidate selection. Extraversion is a crucial personality characteristic in positions that interact with customers, while social skills are important for team work. Specifically, the emotional positivity and social orientation of candidates, both directly extracted from LIWC frequencies, can act as predictors of extroversion trait [12]. We measured extraversion using the candidate's blog posts, which are input to the TreeTagger tool [19] for lexical analysis and lemmatization. Then, using the LIWC dictionary, our system classifies the canonical form of words output from TreeTagger in one of the word categories of interest (i.e. positive emotion, negative emotion and social words) and calculates the LIWC scores. Finally, the system estimates the applicant's extraversion score.

An expert recruiter has assigned extraversion scores to each of 100 job applicants with personal blogs, which were part of a large-scale recruitment scenario. The recruiter's scores were used to train a regression model, which predicts the candidates' extraversion from their LIWC scores in the posemo, negemo, social

categories. In what follows, a linear regression model was selected as a predictor of the extraversion score E, as proposed in [20], due to its increased accuracy and low complexity. Equation 1 corresponds to the linear model that minimizes the Mean Square Error between actual values assigned by the recruiter and predicted scores output by the model:

$$E = S + 1.335 * P - 2.250 * N \tag{1}$$

where S is the frequency of social words (such as friend, buddy, coworker) returned from LIWC, P is the frequency of positive emotion works and N is the frequency of negative emotion words.

2.4 Candidate ranking

The proposed system leverages machine learning algorithms to automatically build the applicant ranking models. This approach requires sufficient training data as an input, which consist of previous candidate selection decisions. Methods that learn how to combine predefined features for ranking by means of supervised learning algorithms are called "learning-to-rank" methods.

In the typical "learning to rank" process a training set is used that consists of past candidate applications represented by feature vectors, denoted as $x_i^{(k)}$, along with an expert recruiter's judgment of the candidates' relevance score, denoted as y_i. The training set is fed to a learning algorithm which constructs the ranking model, such that its output predicts the recruiter's judgment when given the candidates' feature vector as an input. In the test phase the learned model is applied to sort a set of candidate applications, and return the final ranked list of candidates.

In our problem, a scoring function $h(x)$ outputs the candidate relevance score, which reflects how well a candidate profile fits the requirements of a given job position. Then the system outputs the final ranked list by applying the learned function to sort the candidates. The true scoring function is usually unknown and an approximation is learned from the training set D. In the proposed system the training set consists of a set of N previous candidate selection examples, given as an input to the system (Equation 2):

$$D = \{(x_i, y_i)|x_i \in R^m, y_i \in R)\}_{i=1}^{N} \tag{2}$$

3 Experimental Evaluation

The proposed system was tested in a real-world recruitment scenario, to evaluate its effectiveness in ranking job applicants. The system's performance evaluation is based on how effective it is in assigning consistent relevance scores to the candidates, compared to the ones assigned by human recruiters.

In the recruitment scenario used in our tests, we compiled a corpus of 100 applicants with a LinkedIn account and a personal blog, as these are key requirements of the proposed system. The same corpus was used in a previous version of the system [14] for comparison reasons. The applicants were selected

randomly via Google blog search API with the sole requirement of having a technical background, as indicated by the blog metadata (list of interests), as well as a LinkedIn profile. Our corpus of job applicants was formed by choosing the first 100 blogs returned from the profile search API that fulfilled our preconditions. We also collected three representative technical positions announced by an unnamed IT company with different requirements, i.e. a sales engineering position, a junior programmer position and a senior programmer position.

he sales engineering position favors a high degree of extraversion, while experience is the most important feature for senior programmers. Junior programmers are mainly judged by loyalty (as a company would not invest in training an individual prone to changing positions frequently) as well as education. What's more, each position has its own desired set of skills, which are semantically matched with the skill-set reported by each user at their LinkedIn profile. Specifically, the junior position requires programming skills in C++ or Java development languages, while the senior position requires a 5-year experience in J2EE technologies. The use of different requirements per position is expected to test the ability of our system to match candidates' profiles with the appropriate job position.

In our experiments, we assume that each applicant in the corpus has applied for all three available job positions. For each job position, applicants were ranked according to their suitability for the job position both by the system (automated ranking) and by an expert recruiter. Human recruiters had access to the same information as the system, i.e. the candidate's blog and LinkedIn profile. It must be noted though that despite the fact that the selection criteria are known to the system, the recruiter's interpretation of the data and the exact decision-making process is unknown and must be learned.

Table 1. Correlation coefficients for applicants' relevance scores vs. different machine learning models.

Correlation coefficient	**LR**		**M5' Tree**		**REP Tree**		**SVR, poly**		**SVR, PUK**	
	TE	**RE**	**TE**	**RE**	**TE**	**RE**	**TE**	**RE**	**TE**	**RE**
Sales engineer	0.74	0.74	0.81	0.81	0.81	0.81	0.61	0.61	0.81	0.81
Junior programmer	0.79	0.81	0.85	0.85	0.84	0.86	0.81	0.81	0.84	0.86
Senior programmer	0.64	0.73	0.63	0.71	0.68	0.80	0.62	0.68	0.73	0.82

In our first experiment, we use Weka to evaluate the learning-to-rank models. Specifically, we test the correlation of the scores output from the system (i.e. model predictions) with the actual scores assigned by the recruiters, using the Pearson's correlation coefficient metric. Table 1 shows the correlation coefficients for 4 different machine learning models, namely: *Linear Regression* (LR), *M5'* model tree (M5'), *REP Tree* decision tree (REP), and *Support Vector Regression* (SVR) with two non-linear kernels (i.e. polynomial kernel and PUK universal kernel). For each machine learning model we show the results derived using the Total Experience for a candidate (TE) and those that derived using only the Relevant Experience (RE).

As it can be seen, the Tree models and the SVR model with a PUK kernel produce the best results. On the other hand Linear Regression performs poorly, suggesting that the selection criteria are not linearly separable. It must be noted here that all values are averages, obtained with the 10-fold cross validation technique. For the sales position, the recruiter's judgment is dominated by the highly subjective extraversion score, thus increasing the uncertainty of the overall relevance score. Still, the system was able to achieve a correlation coefficient of up to 0.81, depending on the regression model used. On the other hand, the selection of junior programmer candidates is based on more objective criteria such as loyalty and education, thus resulting in a slightly higher correlation coefficient, up to 0.86. Finally, the senior programmer's position exhibited high consistency, with a Pearson's correlation of up to 0.82.

Concerning the first job position (i.e. sales engineer), there was no difference in the results of the two approaches as the relevant experience has no effect on the score calculations. For this position the candidate may have prior experience in any domain or industry (ranging from programmer to salesman) and thus the derived model exactly matches the model based on a candidate's total experience. In the case of the second job position, where only the relevant experience is taken into account, there is a slight difference in the consistency of the two approaches due to the small effect of the experience criterion to the overall score. In the last job position, where the weight of the experience criterion is increased, the difference in the correlation coefficient is clearly observed. More specifically, the values of the correlation coefficient are significantly improved (reaching up to 0.82 in the case of Support Vector Regression with PUK kernel) resulting in consistency values quite comparative to the other two job positions.

4 Conclusions

In this work we present a novel approach for evaluating job applicants in online recruitment systems, using machine learning algorithms to solve the candidate ranking problem and performing semantic matching techniques. The proposed scheme relies on objective criteria extracted from the applicants' LinkedIn profile and subjective criteria extracted from their social presence, to estimate applicants' relevance scores and infer their personality traits. Candidates that do not possess the required skills are filtered out of the selection process and for those remaining the relevant job experience is calculated using semantic matching techniques that allow significantly improved results. The implemented system was employed in a large-scale recruitment scenario, which included three different offered positions and 100 job applicants. The application of the approach in the real-world setting revealed that it is effective in calculating the applicants' suitability for a given job and ranking them accordingly.

References

1. Meo, P.D., Quattrone, G., Terracina, G., Ursino, D.: An xml-based multiagent system for supporting online recruitment services. IEEE Transactions on Systems, Man, and Cybernetics, Part A **37**(4) (2007) 464–480

2. Pande, S.: E-recruitment creates order out of chaos at SAT telecom: System cuts costs and improves efficiency. Human Resource Management International Digest **19**(3) (2011) 21–23
3. Kessler, R., Torres-Moreno, J.M., El-Bèze, M.: E-gen: automatic job offer processing system for human resources. In: Proc. Mexican international conference on Advances in artificial intelligence. (2007) 985–995
4. Bizer, C., Heese, R., Mochol, M., Oldakowski, R., Tolksdorf, R., Eckstein, R.: The impact of semantic web technologies on job recruitment processes. In: Proc. Internationale Tagung Wirtschaftsinformatik. (2005)
5. Radevski, V., Trichet, F.: Ontology-based systems dedicated to human resources management: An application in e-recruitment. In: OTM Workshops (2). (2006) 1068–1077
6. Dorn, J., Naz, T., Pichlmair, M.: Ontology development for human resource management. In: Proc. International Conference on Knowledge Management. (2007) 109–120
7. Mochol, M., Wache, H., Nixon, L.: Improving the accuracy of job search with semantic techniques. In: Proc. International conference on Business information systems. (2007) 301–313
8. Faliagka, E., Ramantas, K., Tsakalidis, A.K., Viennas, M., Kafeza, E., Tzimas, G.: An integrated e-recruitment system for cv ranking based on ahp. In: WEBIST. (2011) 147–150
9. Faliagka, E., Tsakalidis, A., Tzimas, G.: An integrated e-recruitment system for automated personality mining and applicant ranking. Internet Research (2012)
10. Amdouni, S., Ben abdessalem Karaa, W.: Web-based recruiting. In: Proc. ACS/IEEE International Conference on Computer Systems and Applications. (2010) 1–7
11. Mochol, M., Paslaru, E., Simperl, B.: Practical guidelines for building semantic erecruitment applications. In: Proc. International Conference on Knowledge Management, Special Track: Advanced Semantic Technologies. (2006)
12. Faliagka, E., Kozanidis, L., Stamou, S., Tsakalidis, A., Tzimas, G.: A personality mining system for automated applicant ranking in online recruitment systems. In: Proc. International conference on Web engineering. (2011) 379–382
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. **11**(1) (2009) 10–18
14. Faliagka, E., Ramantas, K., Tsakalidis, A., Tzimas, G.: Application of machine learning algorithms to an online recruitment system. In: Proc. International Conference on Internet and Web Applications and Services. (2012)
15. Liu, T.Y.: Learning to rank for information retrieval. Found. Trends Inf. Retr. **3**(3) (2009) 225–331
16. Gill, A., Nowson, S., Oberlander, J.: What are they blogging about? personality, topic and motivation in blogs. (2009)
17. Mishne, G.: Experiments with mood classification in blog posts. In: Proc. Workshop on Stylistic Analysis Of Text For Information Access. (2005)
18. Pennebaker, J.W., King, L.A.: Linguistic styles: language use as an individual difference. Journal of personality and social psychology **77**(6) (1999) 1296–1312
19. Schmid, H.: Improvements in part-of-speech tagging with an application to german. In: Lexikon und Text. (1995) 47–50
20. Mairesse, F., Walker, M.A., Mehl, M.R., Moore, R.K.: Using linguistic cues for the automatic recognition of personality in conversation and text. Journal of Artificial Intelligence Research **30** (2007) 457–500