Artif Intell Rev (2014) 42:515–528
DOI 10.1007/s10462-013-9414-y

On-line consistent ranking on e-recruitment: seeking the truth behind a well-formed CV

Evanthia Faliagka · Lazaros Iliadis · Ioannis Karydis · Maria Rigou · Spyros Sioutas · Athanasios Tsakalidis · Giannis Tzimas

Published online: 11 July 2013

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

Author names appear in alphabetical order.

E. Faliagka · M. Rigou · A. Tsakalidis
Computer Engineering and Informatics Department, University of Patras, Patras, Greece
e-mail: faliagka@ceid.upatras.gr

M. Rigou
e-mail: rigou@ceid.upatras.gr

A. Tsakalidis
e-mail: tsak@cti.gr

L. Iliadis
Forestry and Management of the Environment Department, Democritus University of Thrace, Komotini, Greece
e-mail: liliadis@fmenr.duth.gr

I. Karydis (✉) · S. Sioutas
Department of Informatics, Ionian University, Corfu, Greece
e-mail: karydis@ionio.gr

S. Sioutas
e-mail: sioutas@ionio.gr

G. Tzimas
Department of Applied Informatics in Management and Economy, Technological Educational Institute of Messolonghi, Messolonghi, Greece
e-mail: tzimas@cti.gr

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 (Meo et al. 2007). 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. Ramar and Sivaram, in their work (Ramar and Sivaram 2010), study an unnamed industry, concluding that on average 1 out of 120 student applicants gets selected, while the ratio of recruited candidates that made it to the interview phase is approximately 1 out of 20. Accordingly, 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 (Pande 2011) 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. JobVite and Monster already include a degree of automation during applicants' profiles screening process. The automation offered therein is integrated with the traditional ATS functions and ranges from easy to implement and error prone keyword queries to more sophisticated semantic matching techniques, an approach first proposed in Mochol et al. (2007). The semantic matching techniques associate semantically equivalent concepts from the CV of users with the descriptions of jobs by means of a synonyms' dictionary. *E-Gen* system (Kessler et al. 2007) 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 (Bizer et al. 2005). 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 (Radevski and Trichet 2006) applies Semantic Web technologies in the field of HR Management, while *HR-XML* can partly support the "standardized" representation of competency profiles (Dorn et al. 2007). 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 (Kessler et al. 2009), semantic matching in job seeking and procurement tasks (Mochol et al. 2007), Analytic Hierarchy Process (Faliagka et al. 2011b, 2012b) and NLP technology used to automatically represent CVs in a standard modeling language (Amdouni and Ben abdessalem Karaa 2010). 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 (Mochol et al. 2006). 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 (Faliagka et al. 2011a), are taken into account in his evaluation. In contrast to previous versions of this work, semantic matching techniques are put into use in order to include applicants' work experience in fields that are relevant to the job position. Moreover, a detailed algorithm that determines if a candidate's past work experience is within the domain of expertise of the job position is also presented. 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.

The rest of the paper is organized as follows. Section 2 offers an extended view of the proposed novel approach for evaluating job applicants in online recruitment systems, including the architecture of the system (Sect. 2.1), the modules for semantic matching (Sect. 2.2), personality mining (Sect. 2.3) as well as a candidate ranking process (Sect. 2.4). The design decisions and system implementation of a prototype is discussed in Sect. 3. Then, experimental evaluation is presented in Sect. 4 and the paper is concluded in Sect. 5.

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.

2.1 Architecture

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, shown in Fig. 1, consists of the following components:

- *Semantic matching*: Calculates the semantic distance between candidate skills and prior experience, as extracted from the respective LinkedIn profile and job position requirements.
- *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.
- *Job application module*: Implements the input forms that allow the candidates to apply for a job position.

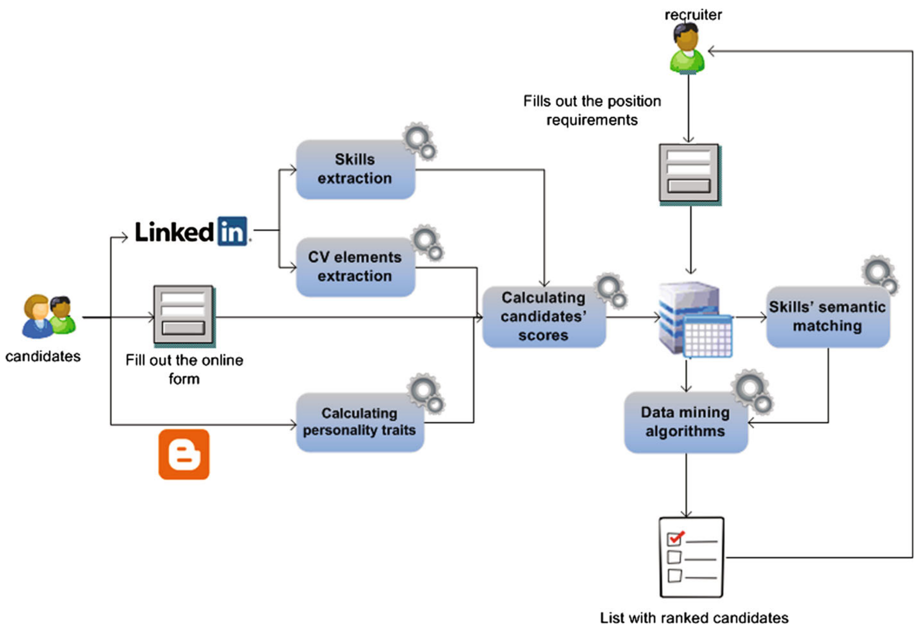

Fig. 1 System's architecture

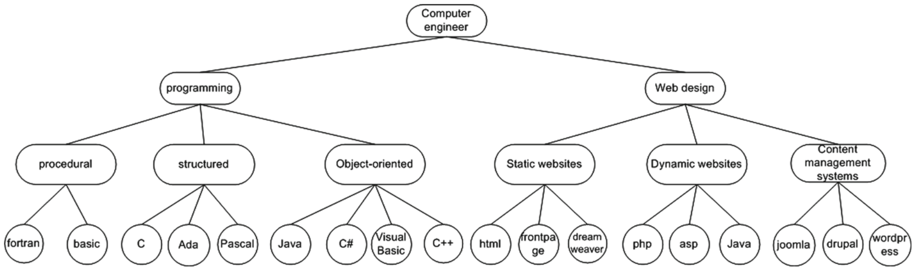

Fig. 2 Part of the implemented IT skills taxonomy

– *Applicant ranking 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.

2.2 Semantic matching

In the previous version of the system (Faliagka et al. 2012a) it was found that except from senior positions that required domain experience and specific qualifications, our system performed consistently with mean error of ±4 positions in the ranking. In order to compensate for the senior positions, an additional criterion was added that indicated whether or not the profile of the candidate was relevant to the job. This was a boolean criterion (yes/no) and was evaluated after keyword search of the qualifications required for the position and qualifications of the candidates. The results of this method offered improvement but again the senior position had reduced accuracy in relation to the other positions. 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 Sect. 4) 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 (Liu 2009). Semantic matching is a technique which combines annotations using controlled domain specific vocabularies with background knowledge about a certain application domain. In our case, the domain specific knowledge is created per domain by experts and is represented by a taxonomy of IT skills (Fig. 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 (i.e., `C`) has all of the characteristics of its parent term (i.e., `Structured`). The taxonomies used herein only contain broader and narrower relationships and their use alleviates the complexity of RDF triples solutions while, as shown in Sect. 4, achieving high performance.

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.

Fig. 3 LinkedIn skills example

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 (Fig. 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 there is no match then the candidate is excluded from the ranking process. In all aforementioned processes of the Semantic matching module, the semantic similarity is based on node distance metric.

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 is shown in Algorithm 1.

2.3 Personality mining

It is highly common for many job positions to rely heavily on applicants' personality traits while such traits are, once again, commonly overlooked in existing e-recruitment systems. The usual method of asserting personality traits of the candidate is by interview, given that the candidate has successfully been admitted to the post-screening phase. Nevertheless, in cases that the personality traits are considered to be of critical importance to the position, a pre-process of collecting data concerning the candidate's personality would be valuable in the pre-screening phase. Current methodologies in carrying out this task mainly focus on human recruiters performing background checks on applicants and focusing especially on

Algorithm 1 Search to determine if the candidate's past work experience is within the domain of expertise of the job position

Require: $E < months, SC >$, the job position
Require: $SC(s_1, s_2, \dots, s_n)$, the skills of the candidate found in the title or description text of the position
Require: $SR(sr_1, sr_2, \dots, sr_n)$, the skills required that can be in any level of the hierarchy
Require: Stemp, intermediate set

```
for all job position E do
  for all elements of SR do
    if sr_i ∈ SC then
      Total_months+ = months
      go to next job position
    else if sr_i not ∈ SC and has leaves then
      Stemp ← SC leaves
      for all elements of Stemp do
        if Stemp_i ∈ SC then
          Total_months+ = months
          go to next job position
        else
          Stemp ← Stemp_i leaves
          go to next element of Stemp
        end if
      end for
    end if
  end for
end for
```

their web presence. Accordingly, it becomes obvious that an automated such processs taking advantage of data mining and text analysis techniques would be far more effective.

Nowadays, under the auspices of Web 2.0, large amounts of textual data exist for a large portion of the web users that have been indicated as reliable predictors of a user's personality. As mentioned before, the proposed system requires a link to each candidate's blog, as blogs have been shown to reflect important aspects of the personality of a blogger (Oberlander and Nowson 2006). Previous works have shown that by applying linguistic analysis to blog posts, the author's personality traits can be derived (Gill et al. 2009) as well as his mood and emotions (Mishne 2005). The text analysis in these works is performed with the 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.

LIWC tool (Pennebaker and Booth 2001) was developed by analyzing writing samples of several hundreds of university students, to correlate word use to personality traits. It uses a dictionary of word stems classified in certain psycholinguistic semantic and syntactic word categories. In Table 1 we can see an example of such word categories. LIWC analyzes written text samples by counting the relative frequencies of words that fall in each word category. Pennebaker and King have found significant correlations between these frequency counts and the author's personality traits (Pennebaker and King 1999) as measured by the Big-Five personality dimensions.

Among the Big-Five personality dimensions, extraversion has received the most research attention, as it has been shown that it is adequately reflected through language use in written speech and it is possible to be discriminated through text analysis (Mairesse et al. 2007). Extraversion is a crucial personality characteristic for candidate selection, especially in positions that interact with customers, while social skills are important for managing teams. What's more, it has been shown that charismatic speakers and people who dominate meetings are usually extroverts (Rienks 2005). Thus, in this work from the Big-Five personality

Table 1 Example of LIWC word categories

Feature	Example
Anger words	Hate, kill, pissed
Metaphysical issues	God, heaven, coffin
Physical state/function	Ache, breast, sleep
Inclusive words	With, and, include
Social processes	Talk, us, friend
Family members	Mom, brother, cousin
Past tense verbs	Walked, were, had
References to friends	Pal, buddy, coworker

dimensions we focus on extraversion due to its importance in candidate selection. Linguistic markers for extraversion are the use of many positive emotion words and social process words, but fewer negative emotion words (Pennebaker and Booth 2001). In this work, the extraversion score is estimated directly from LIWC scores (or frequencies), by summing the emotional positivity score and the social orientation score, also obtained from LIWC frequencies:

– Emotional positivity score was calculated as the difference between LIWC scores for positive emotion words and negative emotion words. Higher scores indicate higher emotional positivity.
– Social orientation score was obtained from LIWC as the frequency of social words (such as friend, buddy, coworker) and personal pronouns (the first person pronoun is excluded). High scores indicate a high degree of references to other people, and thus a high degree of sociability.

It must be noted here that extraversion score does not have a physical basis (i.e. we cannot state that a person is twice as extrovert because he has twice as high extraversion score) but rather quantifies the relative differences between individuals' degree of extraversion. For example, in Argamon and Pennebaker (2005) the authors label bloggers in the top third of the extraversion distribution as extroverts and the bottom third as introverts, while the rest of the sample is considered inconclusive. In this work we model extraversion via scalar values, rather than treating it as a classification problem (where each individual is marked as either introvert or extrovert).

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 system's and the recruiter's extraversion scores were initially expressed in a different rating scale. Thus a re-scaling of both scores was performed in the grading scale 0-5. The expert recruiter had access to the same blog content as the automated system. 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 Mairesse et al. (2007), 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:

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

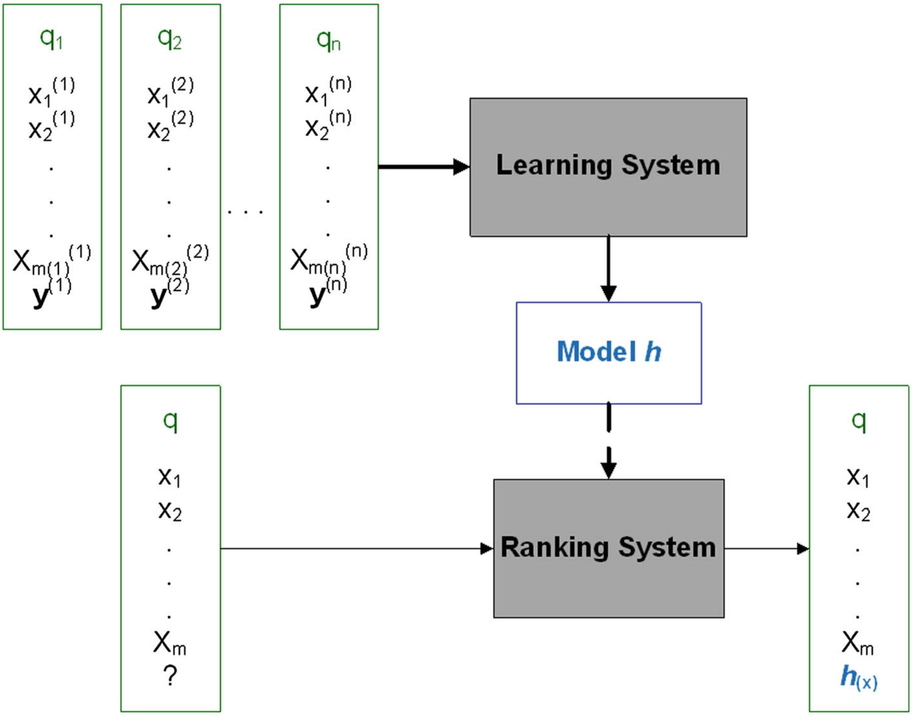

Fig. 4 The "learning to rank" process

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 Fig. 4 the typical "learning to rank" process is shown. The training set used 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 feature vector x_i consists of a set of m attributes $a_1, \ldots, a_m$ that correspond to the candidate's selection criteria. 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

Fig. 5 Job application process

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 (Eq. 2):

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

3 Prototype implementation

The proposed e-recruitment system was fully implemented as a web application, in the Microsoft .Net development environment. In this section we will present the main application screens and discuss our design decisions and system implementation. The system is divided in the recruiter's side and the user's side.

3.1 Job application process (user's side)

Job applicants are given the option to authenticate using their LinkedIn account credentials (see Fig. 5) 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. Users are authorized with LinkedIn API, which uses OAuth as its authentication protocol. After successful user authentication, an OAuth token is returned to our system which allows retrieving information from the candidate's private LinkedIn profile. It must be noted here that the system does not have direct access to the candidate's account credentials, which could be regarded as a security risk. Users without a LinkedIn profile are given the option to enter the required information manually.

As part of the job application process, the candidate is asked to fill-in the feed URI of his personal blog. This allows our system to syndicate the blog content and calculate the extraversion score with the personality mining technique presented. Blog posts are input to the TreeTagger tool (Schmid 1995) for lexical analysis and lemmatization. Then, using the LIWC dictionary which is distributed as part of the LIWC tool, 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.

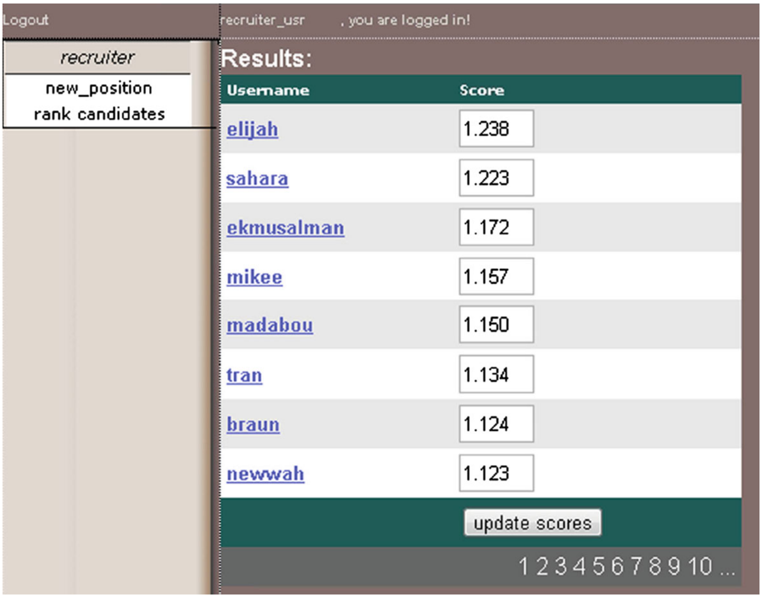

Fig. 6 Candidate ranking results

3.2 Recruitment process (recruiter's side)

After authenticating with their account credentials, recruiters have access to the recruitment module, which gives them 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 each applicant's relevance score and ranks them accordingly. This is achieved by calling the corresponding Weka classifier, via calls to the API provided by Weka. The recruiter can modify the candidate ranking, by assigning his own relevance scores to the candidates, as shown in Fig. 6. 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.

4 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 our recruitment scenario we compiled a corpus of 100 applicants with a LinkedIn account and a personal blog, since these are key requirements of the proposed system. 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, as well as a LinkedIn profile. Specifically, we used the Google profile search API to search for bloggers in the "Technology"

Table 2 Correlation coefficients for applicants' relevance scores versus 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

industry. The search results were manually inspected and only bloggers with a LinkedIn profile associated with their blogs were taken into account. What's more, blogs with no autobiographical content (e.g. technical blogs) were excluded from our study, as they carried no information regarding the author's personality. We also collected three representative technical positions announced by an unnamed IT company with different requirements. 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.

The 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.

In our first experiment, we use Weka[1] to evaluate the learning-to-rank models, based on the feature-selection process found in Hall (1999). 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 2 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

[1] Weka information interchange with .NET was based on ideas of Wikispaces (2013).

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.

5 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 data matching the selection criteria extracted from the applicants' LinkedIn profile and subjective data matching the selection 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 supporting the HR personnel in calculating the applicants' suitability for a given job and ranking them accordingly.

The experimental results showed that the algorithm generated models presented high accuracy except for the jobs that required special skills. Further on, an attempt was made to address this problem by adding a new feature to the system. Specifically we developed a taxonomy with the skills of a computer engineer organized into categories and subcategories and added the capability for semantic search skills. As far as seniority is concerned, the system now does not count all duration of service found in the profile of the candidate but only the time span of relevant work experience. Finally, the criterion showing if the profile of the candidate fits the position is calculated using semantic search skills.

In the future we plan to make some improvements to the system. Specifically, it is planned to extend the taxonomy, which is currently limited in skills related to information technology positions. The aim is to experiment with recruitment in other fields and to investigate whether the results are comparable. Another open direction is the additional mining of other metrics provided by LinkedIn (recommendations, contact number, etc.) and the evaluation of these metrics in correlation to the suitability of candidates.

References

Amdouni S, Ben abdessalem Karaa W (2010) Web-based recruiting. In: Proceedings of ACS/IEEE international conference on computer systems and applications, pp 1–7
Argamon DSKMS, Pennebaker J (2005) Lexical predictors of personality type. In: Proceedings of joint annual meeting of the interface and The Classification Society of North America, pp 1–16
Bizer C, Heese R, Mochol M, Oldakowski R, Tolksdorf R, Eckstein R (2005) The impact of semantic web technologies on job recruitment processes. In: Proc. internationale Tagung Wirtschaftsinformatik, pp 1367–1383
Dorn J, Naz T, Pichlmair M (2007) Ontology development for human resource management. In: Proceedings of international conference on knowledge management, pp 109–120
Faliagka E, Kozanidis L, Stamou S, Tsakalidis A, Tzimas G (2011a) A personality mining system for automated applicant ranking in online recruitment systems. In: Proceedings of international conference on web engineering, pp 379–382
Faliagka E, Ramantas K, Tsakalidis AK, Viennas M, Kafeza E, Tzimas G (2011b) An integrated e-recruitment system for CV ranking based on AHP. In: WEBIST, pp 147–150
Faliagka E, Ramantas K, Tsakalidis A, Tzimas G (2012a) Application of machine learning algorithms to an online recruitment system. In: Proceedings of international conference on internet and web applications and services
Faliagka E, Tsakalidis A, Tzimas G (2012b) An integrated e-recruitment system for automated personality mining and applicant ranking. Internet research
Gill A, Nowson S, Oberlander J (2009) What are they blogging about? Personality, topic and motivation in blogs. In: Proceedings of international AAAI conference on weblogs and social media
Hall MA (1999) Correlation-based feature selection for machine learning. PhD thesis, Department of Computer Science, The University of Waikato
Kessler R, Torres-Moreno JM, El-Bèze M (2007) E-gen: automatic job offer processing system for human resources. In: Proceedings of Mexican international conference on advances in artificial intelligence, pp 985–995
Kessler R, Bechet N, Torres-Moreno JM, Roche M, El-Bèze M (2009) Job offer management: how improve the ranking of candidates. In: Foundations of intelligent systems, lecture notes in computer science, vol 5722, Springer, Berlin, Heidelberg, pp 431–441
Liu TY (2009) Learning to rank for information retrieval. Found Trends Inf Retr 3(3):225–331
Mairesse F, Walker MA, Mehl MR, Moore RK (2007) Using linguistic cues for the automatic recognition of personality in conversation and text. J Artif Intell Res 30:457–500
Meo PD, Quattrone G, Terracina G, Ursino D (2007) An xml-based multiagent system for supporting online recruitment services. IEEE Trans Syst Man Cybern Part A 37(4):464–480
Mishne G (2005) Experiments with mood classification in blog posts. In: Proceedings of workshop on stylistic analysis of text for information access
Mochol M, Paslaru E, Simperl B (2006) Practical guidelines for building semantic erecruitment applications. In: Proceedings of international conference on knowledge management, special track: advanced semantic technologies
Mochol M, Wache H, Nixon L (2007) Improving the accuracy of job search with semantic techniques. In: Proceedings of international conference on business information systems, pp 301–313
Oberlander J, Nowson S (2006) Whose thumb is it anyway? Classifying author personality from weblog text. In: Proceedings of Association for Computational Linguists
Pande S (2011) E-recruitment creates order out of chaos at SAT telecom: system cuts costs and improves efficiency. Hum Resour Manag Int Dig 19(3):21–23
Pennebaker JW, Booth RJ (2001) Linguistic inquiry and word count: Liwc 2001. Word J Int Linguist Assoc
Pennebaker JW, King LA (1999) Linguistic styles: language use as an individual difference. J Pers Soc Psychol 77(6):1296–1312
Radevski V, Trichet F (2006) Ontology-based systems dedicated to human resources management: an application in e-recruitment. In: OTM workshops (2), pp 1068–1077
Ramar K, Sivaram N (2010) Applicability of clustering and classification algorithms for recruitment data mining. Int J Comput Appl 4(5):23–28
Rienks HDR (2005) Dominance detection in meetings using easily obtainable features. In: Proceedings of machine learning for multimodal interaction, pp 76–86
Schmid H (1995) Improvements in part-of-speech tagging with an application to german. In: Lexikon und Text, pp 47–50
Wikispaces (2013) Use weka with the microsoft.net framework. http://weka.wikispaces.com/Use+WEKA+with+the+Microsoft+.NET+Framework