Application of Machine Learning Algorithms to an online Recruitment System

Evanthia Faliagka, Kostas Ramantas, Athanasios Tsakalidis
Computer Engineering and Informatics Department
University of Patras
Patras, Greece
faliagka@ceid.upatras.gr, ramantas@ceid.upatras.gr, tsak@cti.gr

Giannis Tzimas
Department of Applied Informatics in Management & Economy, Faculty of Management and Economics
Technological Educational Institute of Messolonghi
Messolonghi, Greece
tzimas@cti.gr

Abstract— **In this work, we present a novel approach for evaluating job applicants in online recruitment systems, leveraging machine learning algorithms to solve the candidate ranking problem. 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 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.*

I. INTRODUCTION

The rapid development of modern Information and Communication Technologies (ICTs) in the past few years has resulted in an increasing number of people turning to the web for job seeking and career development. A lot of companies use online knowledge management systems to hire employees, exploiting the advantages of the World Wide Web. These are termed e-recruitment systems and automate the process of publishing positions and receiving CVs. E-recruitment systems have seen an explosive expansion in the past few years [1], allowing Human Resources (HR) agencies to target a very wide audience at a small cost. This situation might be 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 fit the position's specifications could lead to an increased efficiency. For example, SAT telecom reported 44% cost savings and a drop in the 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. CommOn framework [4] applies Semantic Web technologies in the field of Human Resources Management. 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 [5] and Analytic Hierarchy Process [6]. Another approach proposed in [7] uses NLP technology to automatically represent CVs in a standard modeling language. These methods, although useful, suffer from the discrepancies associated with inconsistent CV formats, structure and contextual information. What's more they are unable to evaluate some secondary characteristics associated with CVs, such as style and coherence, which are very important in CV evaluation.

In this work, we propose the application of supervised learning algorithms in automated e-recruitment systems, to solve the candidate ranking problem. What's more, we have implemented and tested an integrated company oriented e-recruitment system that automates the candidate pre-screening and ranking process. In the proposed system, the applicants' 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 [8], 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.

The rest of this work is organized as follows. In Section II, we present an overview of the proposed e-recruitment system. In Section III, a personality mining scheme is proposed, to extract the applicant's personality traits from textual data available for the candidate in the web. In

ISBN: 978-1-61208-200-4

Section IV the supervised learning algorithms used to rank the candidates are detailed, and, in Section V, we present a set of experimental results that showcase the effectiveness of our system in a real-world recruitment scenario. Finally, the proposed system was implemented in the form of a web application, whose design and prototype implementation is presented in Section VI.

II. 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 applicant's relevance scores, which reflect how well their profile fits the positions' specifications. In this section, we present an overview of the proposed system architecture and candidate ranking scheme.

A. 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 4 complementary selection criteria, namely: Education (in years of formal academic training), Work Experience, Loyalty (average number of years spent per job) and Extraversion. The system architecture, which is shown in Fig. 1, consists of the following components:

- *Job Application* module: It implements the input forms that allow the candidates to apply for a job position. The candidate is given the option to log into our system using his LinkedIn account credentials, which allows the system to automatically extract all objective selection criteria directly from the user's LinkedIn profile.
- *Personality mining* module: If the candidate's blog URL is provided, it applies linguistic analysis to his blog posts to derive features reflecting the author's personality.
- *Applicant Grading* module: It 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.

Each applicant's qualifications, as well as his relevance score, are stored in the system's database. At the end of the recruitment process, the top candidates are called to participate in the interview process. It must be noted here that during the job application process, the applicant is not required to manually enter information or participate in time-consuming personality tests. Thus, the user friendliness and the practicality of the system are maintained.

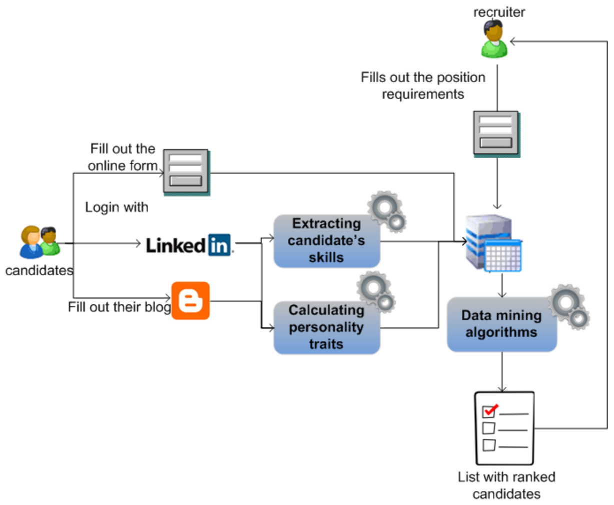

recruiter
Fills out the position requirements
Fill out the online form
Login with
Linked in
candidates
Fill out their blog
Extracting candidate's skills
Calculating personality traits
Data mining algorithms
List with ranked candidates

Figure 1. System Architecture

B. Candidate Ranking

The increasing number of submitted CVs may overwhelm HR departments, which typically perform manual evaluation of job applications. Automated candidate ranking systems, that have been proposed to speed-up the recruitment process typically require a model of the HR department's decision making process, as well as a careful parameterization by the department's expert recruiters. This is a complex and error-prone procedure, which must be repeated each time the selection criteria change. 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 recent years, learning to rank has become a hot research direction in information retrieval [9], but it can also be applied in many real-world ranking problems.

In Fig. 2, the typical "learning to rank" process is shown. 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 candidate's relevance score, denoted as y_i. Candidate's features can be assessed either on a numerical scale (e.g., years of work experience) or with a Boolean variable, which represents whether the candidate reports a certain skill or not in his LinkedIn profile. 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 candidate's 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. Many learning-to-rank algorithms can fit in the abovementioned process, and each one models the process of learning to rank in a different way.

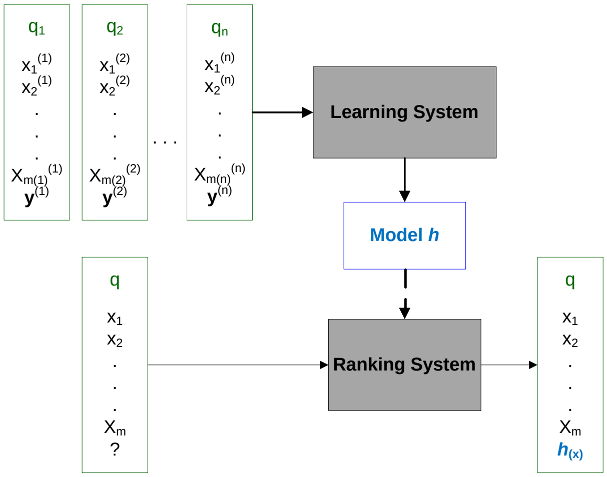


Figure 2. The "learning to rank" process

III. Personality Mining

The applicants' personality traits are critical for their selection in many job positions, but are usually overlooked in existing e-recruitment systems. Typically, candidates' personality is assessed during the interview stage, which is reserved to the candidates that passed the pre-screening phase. However, gathering some preliminary data for the candidate's personality in the pre-screening phase is considered valuable, and such information is often obtained through web searches. In the Web 2.0 era, there are large amounts of textual data for millions web users, that have been shown to be reliable predictors of user's personality. The proposed system automates the task of personality mining using text analysis, an approach proposed in [8].

Previous works have shown that by applying linguistic analysis to blog posts, the author's personality traits can be derived, [10] as well as his mood and emotions [11]. The text analysis in these works is performed with LIWC (Linguistic Inquiry and Word Count) system, which extracts linguistic features that act as markers of the author's personality. LIWC uses a dictionary of word stems classified in certain psycholinguistic semantic and syntactic word categories. It 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 [12] 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. It has been shown that extraversion is adequately reflected through language use in written speech and it is possible to be discriminated through text analysis. Specifically, the emotional positivity and social orientation of candidates, both directly extracted from LIWC frequencies, can act as predictors of extroversion trait [8].

In this work, 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 (see Section V for a detailed description of the 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 [13], due to its good 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:

$$E = S + 1.335 * P - 2.250 * N , \qquad (1)$$

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

IV. Learning to Rank Algorithms

In this work, we leverage machine learning techniques to solve the candidate ranking problem in e-recruitment systems. In the candidate ranking 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. As the relevance score is a continuous variable, the candidate ranking problem can be reduced to a regression problem where the candidate scoring function must be learned using supervised learning techniques. Then the system outputs the final ranked list by applying the learned function to sort the candidates. The score function h(x) derives the candidate's relevance degree y_i from the values of his feature vector x_i. In this work, the feature vector x_i consists of a set of m attributes {a_1, …, a_m} that correspond to the candidate's selection criteria. These can be either continuous variables (representing a candidate's feature assessed on numerical scale) or Boolean variables (declaring whether he has a desired skill or not). 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:

$$D = \left\{ (x_i, y_i) \mid x_i \in R^m, y_i \in R \right\}_{i=1}^{N} . \qquad (2)$$

In what follows, we present a set of representative "learning to rank" algorithms [9] that map the training set D of previous recruiting decisions to a regression model that serves as a predictor of future recruiting decisions.

1) Linear Regression

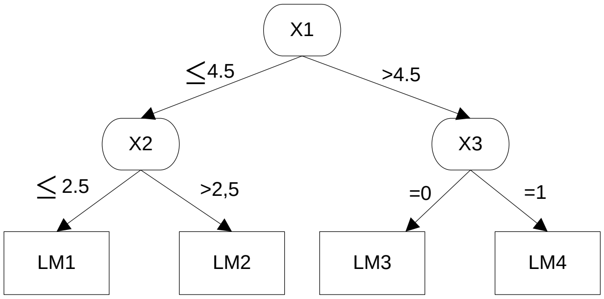


Figure 3. M5 Model tree

In linear regression, the relevance score y_i of the i^{th} candidate is predicted as a linear function of the selection criteria, which comprise the candidate's feature vector x_i plus noise *e* (regression error):

$$y_i = w^T x_i + e. \tag{3}$$

The linear regression algorithm finds the optimal parameter vector w that minimizes the regression error.

2) *Regression Tree*

When selection criteria interact in complex and non-linear ways, linear regression that constructs a linear prediction formula for all data space is not an appropriate model. Regression trees can be a viable alternative, as they recursively partition the predictor space using a divide and conquer approach. They have the same structure as propositional decision trees; internal nodes contain tests and leaves contain predictions for the class value (see Fig. 3). In our experiments, we use an M5' model tree and a REPTree regression tree.

3) *Support Vector Regression*

Support Vector Machines (SVMs) are a set of related methods for supervised learning, applicable to both classification and regression problems. The power of SVMs comes from the kernel representation, which allows a non-linear mapping of input space to a higher dimensional feature space. The objective of Support Vector Regression is to find a function *f* that minimizes the expected error – i.e., the integral of a certain loss function – according to the unknown probability distribution of the data. This minimizes the empirical risk that the estimated function differs from the original (yet unknown) one. Assuming N data points and a Kernel K, the support vectors and the support values of the solution define the following regression function:

$$f(x) = \sum_{i=1}^{N} a_i K(x, x_i) + b \mid b, a_i \in R. \tag{4}$$

V. 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.

A. *Data Collection*

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 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.

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 (because 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 matched with the skillset reported by each user at his 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 candidate's profiles with the appropriate job position.

B. *Experimental Results*

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

TABLE I. Correlation Coefficients For Applicants' Relevance Scores Vs Different Machine Learning Models

Correlation coefficient	LR	M5' Tree	REP Tree	SVR, poly	SVR, PUK
Sales engineer	0.74	0.81	0.81	0.61	0.81
Junior programmer	0.79	0.85	0.84	0.81	0.84

Senior programmer	0.64	0.63	0.68	0.62	0.73

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 [14] 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 I shows the correlation coefficients for 4 different machine learning models, namely: Linear Regression (LR), M5' model tree (M5'), REPTree decision tree (REP), and Support Vector Regression (SVR) with two non-linear kernels (i.e., polynomial kernel and PUK universal kernel). It can be seen that 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.

It can be seen in Table I that the consistency of the system's scores is highly dependent on the nature of the offered positions. 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 0.85. Finally, the senior programmer's position exhibited the lowest consistency, with a Pearson's correlation of up to 0.73. This can be attributed to the high complexity of building a regression model for a senior position, which typically requires domain-specific experience and specific qualifications.

In our second experiment, we evaluate the effectiveness of the personality mining scheme, presented in Section III. As mentioned earlier, our system exploits textual data from the candidate's blog to predict his extraversion score, as determined by an expert recruiter who had access to the same blog posts. The extraversion score is predicted by training a regression model to the extroversion scores assigned from the recruiter to each of the 100 candidates. In this experiment we use Weka to test the effectiveness of 4 different regression models, compiling a table (Table II) with the Pearson's correlation coefficients and relative errors between system's and recruiter's scores. It must be noted that regression models try to replicate the actual scalar values associated by the recruiter, which is a hard problem. Nevertheless, a significant correlation was found, with a Pearson's coefficient of up to 0.65.

TABLE II. CORRELATION COEFFICIENTS AND RELATIVE ERRORS FOR APPLICANT'S EXTRAVERSION SCORE VS MACHINE LEARNING MODELS

Correlation coefficient	LR	M5' Tree	REP Tree	SVR, poly	SVR, PUK
Pearson's Coefficient	0.63	0.63	0.65	0.28	0.65
Relative error	25.3%	25.3%	22.5%	57.4%	23.1%

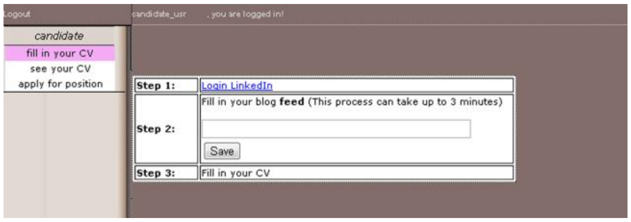
VI. PROTOTYPE IMPLEMENTATION

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.

A. *Job application process (user's side)*

Job applicants are given the option to authenticate using their LinkedIn account credentials (see Fig. 4) 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 [15] 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 in Section III. Blog posts are input to the TreeTagger tool [16] 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,



Step 1:	Login LinkedIn
Step 2:	Fill in your blog **feed** (This process can take up to 3 minutes)
Step 3:	Fill in your CV

Figure 4. Job application process

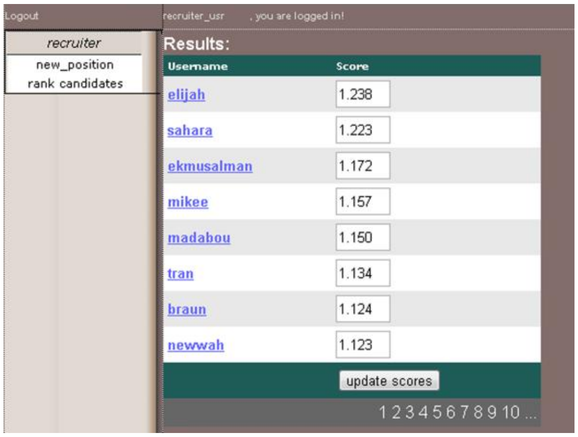
negative emotion and social words) and calculates the LIWC scores. Finally, the system estimates the applicant's extraversion score.

B. *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 applicants' relevance scores 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. 5. 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.

VII. CONCLUSIONS

In this paper, we have presented a novel approach for ranking job applicants in online recruitment systems. 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. Candidate ranking is based on machine learning algorithms that learn the scoring function based on training data provided by human recruiters. An integrated company oriented e-recruitment system was implemented based on the proposed scheme. Our system was employed in a large-scale recruitment scenario, which included three different offered positions and 100 job applicants. The application of our approach revealed that it is effective in identifying the job applicants' extraversion and ranking them accordingly.



Results:

Username	Score
elijah	1.238
sahara	1.223
ekmusalman	1.172
mikee	1.157
madabou	1.150
tran	1.134
braun	1.124
newwah	1.123

Figure 5. Candidate ranking results

REFERENCES

[1] P. De Meo, G. Quattrone, G. Terracina and D. Ursino, "An XML-Based Multiagent System for Supporting Online Recruitment Services," Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 37, July. 2007, pp. 464 – 480.

[2] S. Pande, "E-recruitment creates order out of chaos at SAT Telecom: System cuts costs and improves efficiency", Human Resource Management International Digest, Vol. 19, 2011 pp. 21–23.

[3] R. Kessler, J. Torres-Moreno and M. El-Beze, "E-Gen: automatic job offer processing system for human resources". Proc. of MICAI'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 985-995.

[4] V. Radevski and F. Trichet, "Ontology-Based Systems Dedicated to Human Resources Management: An Application in e-Recruitment," On the Move to Meaningful Internet Systems, vol. 4278, 2006, pp. 1068–1077.

[5] M. Mochol, H. Wache, and L. Nixon, "Improving the Accuracy of Job Search with Semantic Techniques", Business Information Systems, vol. 4439, 2007, pp. 301-313.

[6] E. Faliagka, K. Ramantas, A. Tsakalidis, M. Viennas, E. Kafeza and G. Tzimas, "An Integrated e-Recruitment System for CV Ranking based on AHP," Proc. of WEBIST 2011, May. 2011, pp. 147-150.

[7] S. Amdouni and W. Ben Abdessalem Karaa, "Web-based recruiting", Proc. Of International Conference on Computer Systems and Applications (AICCSA), 2010, pp. 1-7.

[8] E. Faliagka, L. Kozanidis, S. Stamou, A. Tsakalidis and G. Tzimas, "Personality Mining System for Automated Applicant Ranking in Online Recruitment Systems,"Proc. of ICWE 2011, Springer-Verlag, Berlin, Heidelberg, June. 2011, pp. 379-382.

[9] T. Liu, "Learning to Rank for Information Retrieval," Foundations and Trends in Information Retrieval, vol. 3, March 2009, pp. 225-331

[10] J.A. Gill, S. Nowson and J. Oberlander, "What are they blogging about? Personality, topic, and motivation in blogs", Proc. of AAAI ICWSM. 2009

[11] G. Mishne, "Experiments with mood classification in blog posts", Proc. of 1st Workshop on Stylistic Analysis Of Text For Information Access Style 2005. 2005

[12] J.W. Pennebaker and L. King, "Linguistic Styles: Language Use as an Individual Difference," Journal of Personality and Social Psychology, vol. 77, 1999, pp. 1296–1312.

[13] F. Mairesse, M.A. Walker, M.R. Mehl and R.K. Moore, "Using linguistic cues for the automatic recognition of personality in conversation and text," Journal of Artificial Intelligence Research, vol. 30, 2007, pp. 457-500.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. Witten, "The WEKA data mining software: an update," SIGKDD Explorer, News, vol. 11, 2009, pp. 10-18.

[15] E. Hammer-Lahav and D. Recordon, "The OAuth 1.0 Protocol", http://tools.ietf.org/html/draft-hammer-oauth-10, February 2010.

[16] H. Schmid, "Improvements In Part-of-Speech Tagging With an Application To German", Proc. of ACL SIGDAT, 1995, pp. 47-50.