Job Offer Management: How Improve the Ranking of Candidates

Rémy Kessler[1], Nicolas Béchet[2], Juan-Manuel Torres-Moreno[1], Mathieu Roche[2], and Marc El-Bèze[1]

[1] LIA / Université d'Avignon, 339 chemin des Meinajariès, 84911 Avignon
[2] LIRMM - UMR 5506, CNRS / Université Montpellier 2 - France
{remy.kessler, juan-manuel.torres, marc.elbeze}@univ-avignon.fr
{nicolas.bechet,mathieu.roche}@lirmm.fr

Abstract. The market of online job search sites grows exponentially. This implies volumes of information (mostly in the form of free text) become manually impossible to process. An analysis and assisted categorization seems relevant to address this issue. We present E-Gen, a system which aims to perform assisted analysis and categorization of job offers and of the responses of candidates. This paper presents several strategies based on vectorial and probabilistic models to solve the problem of profiling applications according to a specific job offer. Our objective is a system capable of reproducing the judgement of the recruitment consultant. We have evaluated a range of measures of similarity to rank candidatures by using ROC curves. Relevance feedback approach allows to surpass our previous results on this task, difficult, diverse and higly subjective.

1 Introduction

The exponential growth of Internet allowed the development of a market for online job-search [1, 2]. Over last few year it is in a significant expansion (August 2003: 177 000 job offers, May 2008: 500 000 job offers)[3]. The Internet has become essential in this process because it allows a better flow of information, either through job search sites or by e-mail exchanges. The answers of candidates confer a lot of information that cannot be managed efficiently by companies [3]. Even though a browser has become a universal and easy tool for the users, frequent need to enter data into Web forms from paper sources, "copy and paste" data between different applications, is symptomatic of the problems of data integration. Therefore it is essential to process this information by an automatic or assisted way. We developed the E-Gen system to resolve this problem.

It is composed of three main modules:

1. The first module extracting the information from a corpus of e-mails of job offers from Aktor's database[4].

[3] www.keljob.com
[4] Aktor Intéractive (www.aktor.fr)

2. The second module analysing the candidate answers (splitting e-mails into Cover Letter (CL) and Curriculum Vitae (CV)).
3. The third module analysing and computing a relevance ranking of the candidate answers.

Our previous works present the first module [4] the identification of different parts of a job offer and the extraction of relevant information (contract, salary, localization etc.). The second module analyses the content of a candidate's e-mail, using a combination of rules and machine learning methods (Support Vector Machines, SVM). Furthermore, it separates the distinct parts of CV and CL with an Precision of 0.98 and a Recall 0.96 [5]. Reading a large number of candidate answers for a job is a very time consuming task for a recruiting consultant. In order to facilitate this task, we propose a system capable of providing an initial evaluation of candidate answers according to various criteria. In this paper, we present the last module of E-Gen. Some related works are briefly discussed in section 2. Section 3 shows a general system overview. In section 4, we describe the pre-processing task and strategy used to rank the candidate answers. In section 5, we present statistics about the textual corpus, experimental protocol and results.

2 Related Work

Many approaches have been proposed in literature to reduce the costly and tedious task of managing the Human Resources. Candidate answers to a job-offers are particular and ad hoc documents, it allows to develop semantic approaches to analyse than. [6] proposes an indexing method based on the *BONOM* system [7]. Their method consists of using distributional attributes of documents to locate each part to finally index the document. A semantic-based method to select candidate answers and to discuss the economical impacts in the German government was proposed by [8]. Limitations of actual systems of automatic selection of candidate answers are presented in [2]. They propose a system based on collaborative filters (*ACF*) to automatically select profiles of candidate answers in the *JobFinder* Website. [9] discuss the relevance of a common ontology (*HR ontology*) to working efficiently with this kind of documents. [3] describes an ability model and a management tool used for the candidate-answers selection. Using the same model, [10] outline an *HR-XML* based prototype dedicated to the job search task. The prototype selects and favors relevant information (paycheck, topic, abilities, etc.) from many job-service Websites, such as *Jobs.net, aftercollege.com, Directjobs.com* etc.
The study of the more relevant document – the CV – to use it automatically has been a subject of many researches. [11] proposes a data mining approach. Their aim is to build automates which recognize CV topologies and candidate/job-offers profiles. A first step differentiates the CV of executive employed from other CV employed. They make a specific term extraction to obtain a categorization with the C4.5 decision tree algorithm [12]. This method focuses on the specificity of selected terms or concepts, as education level or relevant abilities, to

build a classifier. The method results are yet poor (an accuracy between 0.5-0.6 of correctly categorized CV). [13, 14] have made a terminology study of corpus composed by CV (of the Vedior Bis company (*http://www.vediorbis.com*)). Their approach allows to extract collocations from CV corpus based on syntactic patterns as Noun-Noun, Adjective-Noun, etc. Then these collocations are ranked by relevance to build a specialized ontology. In this paper, we present an approach to the candidatures ranking by using a combination of similarity measures and Relevance Feedback.

3 System overview

Nowadays technology proposes new ways of on-line employment market. We propose a system which answers as fast and judiciously as possible to this challenge. An e-mail-box receives messages containing the offer. Firstly, the job offer language is identified by using n-grams. Then, E-Gen parses the e-mail, splits the offer into segments, and retrieves relevant information (contract, salary, location, etc.). Subsequently a filtering and lemmatisation process is applied to text and it will represented in a vector space model (VSM). A categorization of text segments (Preambule, Skills, Contacts,...) is obtained by means of Support Vector Machines. This preliminary classification is afterwards transmitted to a "corrective" post-process which improves the quality of the solution (Task 1, described in [4]). During the publication of a job offer, Aktor generates an e-mail address for applying to the job. Each e-mail is redirected to a Human Resources software, (Gestmax[5]) to be read by a recruiting consultant. At this step, E-Gen analyses the candidate's answers to identify each part of the candidacy and extracts the text the from e-mail and attached files (by using wvWare[6] and pdftotext[7]). After a pre-processing task, we use a combination of rules and machine learning methods to separate each distinct part (CV or CL). The process (task 2) is described in [5]. Once CL and CV are identified, the system performs an automated profiling of this candidature by using measures of similarity and a small number of candidatures previously validated as relevant candidatures by a recruitment consultant (Task 3). The whole of the chain of E-Gen System is represented in figure 1.

4 Ranking of candidatures

4.1 Corpus pre-processing

A classical pre-processing is applied to Textual information (CV et CL). We remove information such as names of candidates, addresses, e-mails, names of cities. Accents are deleted and capital letters are normalised. In order to avoid the introduction of noise into the models[8], the following items are also deleted:

[5] http://www.gestmax.fr
[6] http://wvware.sourceforge.net
[7] http://www.bluem.net/downloads/pdftotext_en
[8] These pre-processing are not applied in the n-grams representation.

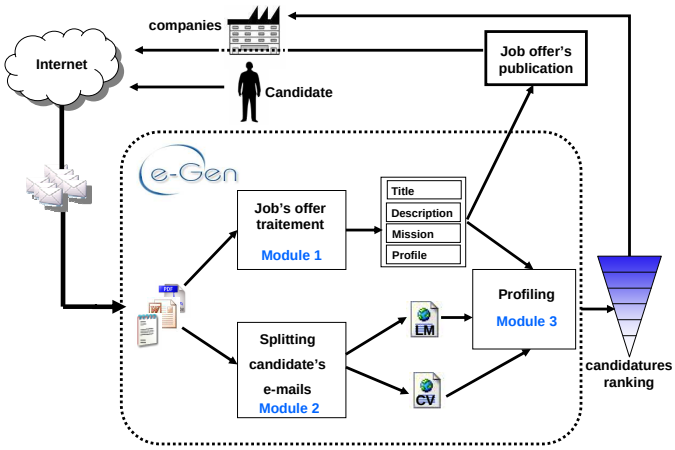

Fig. 1. System overview.

verbs and functional words (to be, to have, to need,...), common expressions with a stop words list[9](for example, that is, each of,...), numbers (in numeric and/or textual format), symbols such as "$","#", "*". Finally, lemmatisation[10] is performed to significantly reduce size of the lexicon. All these processes allow us to represent the collection of documents through the bag-of-words paradigm (a matrix of frequencies of terms (columns) for each candidate answer (rows)).

4.2 Comparison between candidatures and job offer using similarity measure

Each document is transformed into a vector with weights characterizing the frequency of terms *Tf* and *Tf-idf* [15].

We have established a strategy using measures of similarity, to rank all candidatures in relation to a job offer. We combined different similarity measures between the candidate answers (CV and LM) and the associated job offer. We also tested several similarity measures as defined in [16]: *cosine* (1), which calculates the angle between job offer and each candidate answer, Minkowski distances (2) ($p = 1$ for Manhattan, $p = 2$ for euclid). The last measure used is Okabis (3) [17]. Based on okapi [18] formula, this measure is often used in Information Retrieval.

$$sim_{\text{cosine}}(j,d) = \frac{\sum_{i=1}^{n} j_i \cdot d_i}{\sqrt{\sum_{i=1}^{n} j_i^2 \cdot \sum_{i=1}^{n} d_i^2}} \tag{1}$$

$$sim_{\text{Minkowski}}(j,d) = \frac{1}{1 + \left(\sum_{i=1}^{n} |j_i - d_i|^p\right)^{\frac{1}{p}}} \tag{2}$$

[9] http://sites.univ-provence.fr/~veronis/donnees/index.html

[10] Lemmatisation finds the root of verbs and transforms plural and/or feminine words to masculine singular form. So we conflate terms *sing, sang, sung, will sing* into sing.

$$\text{Okabis}(d,j) = \sum_{i \in d \cap j} \frac{\sum_{i=1}^{n} j_i \cdot d_i}{\sum_{i=1}^{n} j_i \cdot d_i + \frac{\sqrt{|d|}}{M_d}} \quad (3)$$

where j is a job offer, d is a candidate answer, i a term, j_i and d_i occurrence of i respectively in j and d, and M_d their average size.
Several other similarity measures (Overlap, Enertex, Needleman-Wunsch, Jaro-Winkler) have been tested but they are not retained in this study, because the results obtained are disapointing. All measures used and their combinations are described in [19].

4.3 Extraction of features

In the following sections, we describe a number of features that will be used to represent the documents. These features are based on grammatical information, n-grams of characters and semantic information.

Filtering and weighting of words according to their grammatical label To improve the performance of similarity measures (section 4.2), we performed an extraction of grammatical information in the corpus with TreeTagger[11] [20]. We found that CV are short documents (usually not exceeding one page) and syntactically poor: few subjects and verbs in sentences, sentences in summary form, many lists of nouns and adjectives, etc [13]. The words respecting specific grammatical labels can thus be more or less interesting. We propose to extract the following terms : **N** (Noun) **A**(adjective) **V**(Verb). These terms alone will be selected as the basis of the vector representation of documents. We tested different combinations and weights.

Character *n*-grams Mainly used in speech recognition, n-grams of characters have been used in text analysis [21]. Research shows the effectiveness of n-grams as a method of text representation [22, 23]. An n-gram is like a moving window over a text, where n is the number of character in the window. An n-gram is a sequence of n consecutive characters. The move is processed by steps, one step related to one character. Then the frequencies of n-grams found are computed. For example, the sentence "developer php mysql" is represented with tri-grams [dev, eve, vel, elo, lop, ope, per, er_, r_p, _ph, php, hp_, p_m, _my, mys, ysq, sql]. We represent the space in the n-grams by using the "_". This representation automatically captures the most stem of words, avoiding lexical root research. The second interest of this representation is their tolerance to spelling mistakes and typographical errors often found in CV and CL[12]. We tested different n-size windows (*3/4/5/6-grams*).

Semantic enrichment of the job offer Observation of terms with the most influence when computing the similarity measure, led us to consider enhancing the

[11] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/ : TreeTagger is a tool for annotating text with part-of-speech and lemma information.

[12] For example, a words system will have difficulty recognizing the word "Developper" misspelled (with two p).

content of the job offer with an ontology derived from the base ROME[13] from ANPE[14]. We enriched each job with skills and educational levels expected[15].

Relevance feedback We changed the system to incorporate a process of Relevance Feedback [24]. Relevance feedback is a classical method used particulary for manual query reformulation. For exemple, the user carefully checks the answer set resulting from an initial query, and then reformulates the query. Rocchio algorithm [25] and variations have found wide usage in information retrieval and related areas such as text categorisation [26]. Relevance Feedback has been proposed [27] to help the user to find a job with with server logs from the site JobFinder[16].

In our system, Relevance Feedback takes into account the recruiting consultant choice during a first evaluation of few CVs. Our goal is not a system capable of finding the best candidate, but a system capable of reproducing the judgement of the recruitment consultant. It is critical for recruiters not to miss a good candidate that they may have unfortunately rejected. The goal of this Relevance Feedback approach is to help them to avoid this kind of error. This approach exploits documents returned in response to a first request to improve the search results [28]. In this case, we randomly take few candidate answers (one to six in our experiments) amongst all relevant candidate answers. These are added to the job offer. So we use manual relevance feedback to reflect the user judgements in the resulting ranking. We increase the vector representation with the terms from the candidates considered relevant by a recruitment consultant. System will recompute similarity between the candidate's answer that we evaluate and job offer enriched with relevant candidates.

5 Experiments

We have selected a data subset from Aktor's database. This subset is called *Corpus Mission*. It contains a set of job offers with various thematics (jobs in accountancy, business enterprise, computer science, etc.) and their candidates. As described in [19],each document is segmented to keep relevant parts (we remove the description of the company for the job offer and the last third of CV and CL). Each candidate is tagged **relevant** or **irrelevant**. A **relevant** value corresponds to a potential candidate for a given job chosen by the recruiting consultant. A **irrelevant** value is associated to an unsuitable candidate for the job (this is a decision if the human ressources of the company). Our study was conducted on french job offers because the french market represents Aktor's main activity. Table 1 shows a few statistics about the *Corpus Mission*.

[13] Répertoire Opérationnel des Métiers et des Emplois , Operational List of Jobs and Skills

[14] Agence National Pour l'Emploi, National Agency for Employment http://www.anpe.fr/espacecandidat/romeligne/RliIndex.do

[15] Example: 32321/developer/**Bac+2** à **Bac+4** in **computing CFPA**, **BTS**, **DUT**;**development** and **maintenance** of **computing applications**, **functional analysis**, **engineering design**, **coding**, **development** and **documentation** of **programs** etc.

[16] JobFinder (jobfinder.com)

Number	Job's Title	Number of candidate answers	Number of **relevant**	**irrelevant**
34861	sales engineer	40	14	26
31702	accountant, Department suppliers	55	23	32
33633	sales engineer	65	18	47
34865	accountant assistant	67	10	57
34783	accountant assistant	108	9	99
33746	3 chefs	116	60	56
33553	Trade Commissioner	117	17	100
33725	urban sales consultant	118	43	75
31022	recruitment assistant	221	28	193
31274	accountant assistant junior	224	26	198
34119	sales assistant	257	10	247
31767	accountant assistant junior	437	51	386
Total		1917	323	1594

Table 1. Corpus statistics.

5.1 Experimental protocol

We want to measure the similarity between a job offer and its candidate's answers. *Corpus Mission* is composed of 12 job offers associated with at least 9 candidates identified as **relevant** for each one. These measures (section 4.2) rank the candidate answers by computing a similarity between a job offer and their associated candidate answers.

We use the ROC curves to evaluate the quality ranking obtained. ROC curves [29] come from the field of signal processing. They are used in medicine to evaluate the validity of diagnostic tests. In our case, ROC curves show the rate of irrelevant candidate answers on X-axis and the rate of relevant candidate answers on Y-axis. The *Area Under the Curve* (AUC) can be interpretated as the effectiveness of a measurement of interest. In the case of candidate answers ranking, a perfect ROC curve corresponds to obtain all relevant candidate answers at the beginning of the list and all irrelevant at the end. This situation corresponds to $AUC = 1$. The diagonal line corresponds to the performance of a random system, progress of the rate of relevant candidate being accompanied by an equivalent degradation of the rate of irrelevant candidate. This situation corresponds to $AUC = 0.5$. An effective measurement of interest to order candidate answers consists in obtaining the highest AUC value. This is strictly equivalent to minimizing the sum of the ranks of the relevant candidate's answers. ROC curves are resistant to imbalance (for example, an imbalance in number of positive and negative examples) [13]. For each job offer, we evaluated the quality of ranking obtained by this method. Candidate answers considered are only those composed of CV and CL.

5.2 Results

Table 2 shows the best results obtained for each method. Each test is carried out 100 times with a random distribution of relevant candidatures for Relevance Feedback. Then we compute an average of AUC scores obtained (the curve shows the average for each size). The *TF* corresponds to the results obtained with the frequency of each term as unit. *TF-IDF* uses the product of terms frequency and inverse document frequency. *TF* and *TF-IDF* representations give globally similar results with AUC score at 0.64. Small size of corpus used can explain these results. Using combination and weighting

of grammatical classes representation (*Grammatical Labels*) gives also close results. *N-grams* results are obtained with 5-grams. With *AUC* score at 0.6, *n*-grams results are poor. We plan, in order to improve the *n*-grams results, to find and remove frequent and insignificant strings. *Job offer enriched* corresponds to the results obtained with semantic enrichment of job offer. With *AUC* score at 0.62, semantic expansion does not improve referent results. Additional information about job offer are not required and it seems degrade performance of the system but additional tests are necessary.

	N-grams	*Job offer enriched*	*TF*	*TF-IDF*	*Grammatical Labels*	*Relevance Feedback*
Job offer/CV and CL	0.60	0.62	0.64	0.64	0.64	**0.66**

Table 2. Comparison of *AUC* score for each method.

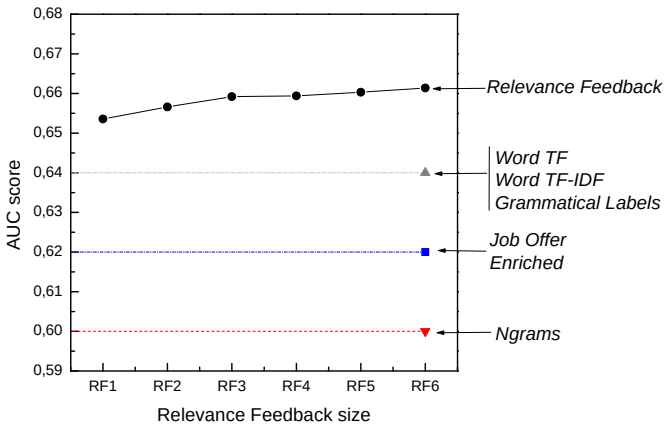

Fig. 2. Comparison of *AUC* score for each size of Relevance Feedback.

Figure 2 presents results obtained with different sizes of relevance feedback (RF1 corresponds to one candidature added to the job offer, RF2 two, etc.). We use actually *residual ranking* [30]: documents that are used for relevance feedback are removed from the collection before ranking with the reformulated query. We observe that Relevance Feedback allows to improve the results more significantly. RF1 gives an average *AUC* score at 0.65 and RF6 at 0.66. Currently, we study results for each mission, but they are quite disparate. For example, mission 33725 shows a good increase between each size of relevance feedback (*TF*: 0.595, RF1: 0.685, RF6:0.716) while for others the increase was less obvious (mission 33633 *TF*: 0.561, RF1: 0.555, RF6:0.579).The study of results shows that some missions has some empty candidate with label **relevant**. This leads the system to degrade performance when they are selected. Note that it is impossible

to experiment RFn with $n > 6$ because of the number of candidates too small for some job offers (see table 1).

6 Conclusion and future work

The processing of a job offer is a difficult and highly subjective task. The information we use in this kind of process is not well formated in natural language, but follows a conventional structure. In this paper, we present the third module of the E-Gen project, a system for processing of a job-offer. The system allows to assist an employer in a recruitment task. The third module we presented in this paper focuses on candidate-answers to job offers. We rank the candidate answers by using different similarity measures and different document representations in vector space model. We choose to evaluate the quality of our approaches by computing *Area Under the Curve*. *AUC* obtained with our relevance-feedback-based-approach shows an improvement of result. As future work, we plan to apply other treatments, such as finding discriminant features of irrelevant candidatures to use Rocchio algorithm [25], weighting the different segments of a mission, etc. to improve results. We also plan to take into account other parameters such as vocabulary used and spelling. Thus we will perform a better analysis of the cover letters. Actually, CL are not really used by an employer in a decision process. Finally we propose to measure the CV quality by building an evaluation in a Internet portal. Our aim with this evaluation is to present to a job-finder a list of relevant job-offers in agreement with this profile.

Acknowledgement

Autors thanks to Piotr Więcek.

References

1. Bizer, R.H., Rainer, E.: Impact of Semantic web on the job recruitment Process. International Conference Wirtschaftsinformatik (2005)
2. Rafter, R., Bradley, K., Smyt, B.: Automated Collaborative Filtering Applications for Online Recruitment Services. International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (2000) 363–368
3. Bourse, M., Leclère, M., Morin, E., Trichet, F.: Human resource management and semantic web technologies. In: ICTTA. (2004) 641–642
4. Kessler, R., Torres-Moreno, J.M., El-Bèze, M.: E-Gen: Automatic Job Offer Processing system for Human Ressources. MICAI 2007, Aguscalientes, Mexique, pp 985-995 (2007)
5. Kessler, R., Torres-Moreno, J.M., El-Bèze, M.: E-Gen: Profilage automatique de candidatures. TALN 2008, Avignon, France 370–379
6. Morin, E., Leclère, M., Trichet, F.: The semantic web in e-recruitment (2004). In: The First European Symposium of Semantic Web (ESWS'2004). (2004)
7. Cazalens, S., Lamarre, P.: An organization of internet agents based on a hierarchy of information domains. In: Proceedings MAAMAW. (2001)
8. Tolksdorf, R., Mocho, M., Heese, R., Oldakowski, R., Christian, B.: Semantic-Web-Technologien im Arbeitsvermittlungsprozess. International Conference Wirtschaftsinformatik (2006) 17–26

9. Mocho, M., Paslaru, E., Simperl, B.: Practical Guidelines for Building Semantic eRecruitment Applications. I-Know'06 Special track on Advanced Semantic Technologies (2006)
10. Dorn, J., Naz, T.: Meta-search in human resource management. In: in Proceedings of 4th International Conference on Knowledge Systems ICKS'07 Bangkok,Thailand. 105 – 110
11. Clech, J., Zighed, D.A.: Data mining et analyse des cv : une expérience et des perspectives. In: EGC'03. (2003) 189–200
12. Quilan, J.: C4.5: Programs for machine learning. In: Kaufmann, San Mateo, CA. (1993)
13. Roche, M., Kodratoff, Y.: Pruning Terminology Extracted from a Specialized Corpus for CV Ontology Acquisition. In: OTM'06, Montpellier, France. (2006) 1107–1116
14. Roche, M., Prince, V.: Evaluation et détermination de la pertinence pour des syntagmes candidats à la collocation . in JADT2008 (2008) 1009–1020
15. Salton, G., Mcgill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA (1986)
16. Bernstein, A., Kaufmann, E., Kiefer, C., Bürki, C.: Simpack: A generic java library for similarity measures in ontologies. Technical report, University of Zurich (2005)
17. Bellot, P., El-Bèze, M.: Classification et segmentation de textes par arbres de décision. In: TSI. Volume 20. Hermès (2001) 107–134
18. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M.: Okapi at trec-3. NIST Special Publication 500-225: TREC-3 (1994) 109–126
19. Kessler, R., Béchet, N., Roche, M., El-Bèze, M., Torres-Moreno, J.M.: Automatic profiling system for ranking candidates answers in human resources. In: OTM '08 in Monterrey, Mexico. (2008) 625–634
20. Schmid, G.: Treetagger - a language independent part-of-speech tagger. In Proceedings of EACL-SIGDAT 1995. Dublin, Ireland. (1994) 44–49
21. Damashek, M.: Gauging similarity with n-grams : Language-independent categorization of text. Science 1995 ; 267 (1995) 843–848
22. Mayfield, J., Mcnamee, P.: Indexing using both n-grams and words. NIST Special Publication (1998) 500–242
23. Hurault-Plantet, M., Jardino, M., Illlouz, G.: Modèles de langage n-grammes et segmentation thématique. Actes TALN & RECITAL, vol 2 (2005) 135–144
24. Spärck Jones, K.: Some thoughts on classification for retrieval. Journal of Documentation (1970) 89–101
25. Rocchio, J. In: Relevance Feedback in Information Retrieval. (1971) 313–323
26. Joachims, T.: A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In: ICML '97. (1997) 143–151
27. Smyth, B., Bradley, K.: Personalized Information Ordering: A Case-Study in Online Recruitment. Journal of Knowledge-Based Systems (2003) 269–275
28. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science (1990) 288–297
29. Ferri, C., Flach, P., Hernandez-Orallo, J.: Learning decision trees using the area under the ROC curve. In: Proceedings of ICML'02. (2002) 139–146
30. Billerbeck, B., Zobel, J.: Efficient query expansion with auxiliary data structures. Inf. Syst. (7) (2006) 573–584