# A Content-Based Recommendation Approach Using Semantic User Profile in E-recruitment

Oualid Chenni[1], Yanis Bouda[1], Hamid Benachour[2], and Chahnez Zakaria[3](✉)

[1] Ecole Nationale Supérieure d'Informatique,
BP 68M, 16309 El Harrach, Algiers, Algeria
{ao_chenni,ay_bouda}@esi.dz
[2] Laboratoire de Recherche En Intelligence Articifielle (LRIA),
USTHB, El Alia BP 32, Bab Ezzouar, Algiers, Algeria
hamid.benachour@gmail.com
[3] Laboratoire de Méthodes de Conception des Systèmes (LMCS),
Ecole Nationale Suprieure d'Informatique, 16309 El Harrach, Algiers, Algeria
c_zakaria@esi.dz

**Abstract.** In this paper, we propose a content-based recommendation approach in the domain of e-recruitment to recommend users with job offers that suit the most their profile and learned preferences. In order to present the best offers, we construct a semantic vocabulary of the domain from the job offers corpus and initialize a profile for each user based on his Curriculum Vitae. Our method is enriching the user profiles using triggers and statistical methods following his actions regarding the job offers. The approach we propose presents to the users job offers that are the closest to their learned needs and interests which also can be updated based on his daily actions regarding these offers.

**Keywords:** Profiling · Recommendation · User profile · Semantic vocabulary · Triggers · E-recuitement

## 1 Introduction

The recent growth of the Online Web Recruitment Market has made traditional recruitment methods all but obsolete. In a world where companies are in a constant competition to hire the best profiles and increase incomes while decreasing the risks. According to the Harvard business school, the cost of a bad hire is three to five times an employee's annualized compensation. In specialist functions it reaches 10 times an annual salary[1]. Aware of these challenges, companies today invest massively in the best e-recruitment technologies and platforms. We have a plethora of online communities involving billions of people, and businesses use them to get opinions, generate consumer insights...etc., They use the web

[1] http://www.eremedia.com/ere/recruitment-5-0-the-future-of-recruiting-the-final-chapter/.


A.-H. Dediu et al. (Eds.): TPNC 2015, LNCS 9477, pp. 23–32, 2015.
DOI: 10.1007/978-3-319-26841-5_2

to scan and watch social trends and needs. We have an explosion of data, trillions of information about customers, job seekers, employees...etc. From which we can learn everything about people and their habits. The analysis of these sets of data is the main point of the competition in recruiting. Recruiters today want to receive the "ideal" shortlist of candidates, after analysing and weighting job application based on data patterns in the cloud which regroup: skill sets, experiences, behavioural patterns....etc.

The same objective is pursued by the candidates who want to receive the job offers that correspond the best to their needs and deep interests. This is the challenge that e-recruitment faces today, it's all about personalization and reduction of mis-hire and gaining the loyalty of the users. *Emploitic.com*[2] as the leader of e-recruitment in Algeria wants to build a specific recommendation system that will be integrated in their platform. In facts, their platform users, receive offers that don't correspond all the time to their interest because it is purely based on a key-words research, and recruiters suffer from the same problem, as they receive hundreds and thousands of job applications which generally do not fit the offer, due to some aspects as the difficulty for some users to understand the job description. So, to meet the needs of the users, we proposed a solution that personalizes the results which are given by the search engine and also by the suggestion and recommendation systems.

Recommendation systems are a specific form of information filtering which aims to present information items that might be of interest to the user. In general, a recommendation system allows you to compare a user profile to other certain reference characteristics, and tries to predict the opinion of a user [1]. [12] explain the recommendation systems as systems that collect opinions of a user's community, about items (job offers, TV programs...etc.) in order to use these opinions and likes to recommend interesting items to other users of this community.

In this paper, we describe the approach that we built to answer the needs of the users. We made a solution that creates a personalized profile for each user of the platform then enriches it through the use of the job offers corpus and the user's actions monitoring. The body of this paper is organized as follows: Section 2 discusses the related work. Section 3 presents the approach built. Section 4 presents the evaluation of the results obtained with the system. Finally, Section 5 is about our perspectives and a conclusion to our work.

## 2 Related Work

Basically, recommendation systems are divided into 4 categories: The collaborative filtering approach which predicts the interest of the user to an item by using a database of a group of other users preferences. This approach is itself divided into two subcategories: The Memory-based collaborative filtering which predicts the interests of the user by assigning him first to a group of similar users,through similarity or correlation measures, then it uses the weighted-notations of the

[2] http://www.emploitic.com.

same-group users regarding the items [4]; Model-based collaborative filtering on its side uses the predicted values of a user's notation regarding an item, based on the knowledge that the system has about the user. For example: using previous notations for other items [4], this model uses several different models as the cluster model or the Bayesian networks.

The second approach is the content-based filtering, which focuses on the content similarity between an item and the other items that the user has previously liked [2]. Systems that are based on this approach have a two-steps process: the user's profiling and the items representation. On one hand, they build the profile through the extraction, gathering and representation of its characteristics automatically through the monitoring of his actions regarding the items that are of interest to him. On the other hand, the items representation is made through structured data [2]. The third one is the knowledge based approach which suggests to the user, items based on the inference of his needs and preferences through the construction of a strong knowledge about the field [5], e.g.: e-recruitment. Finally, the Hybrid approach is the combination of the three previous approaches by using different technologies, we can find among them the weighted hybrid approach or meta-level hybrid systems....etc., [6].

Of course, as e-recruitment becomes more and more strategic and important, several approaches have been used in e-recruitment platforms to build personalized recommendation systems that provide better satisfaction to the user. Thus, within the sphere of job recommendation systems, we can find a lot of work that has been done using the different approaches listed above.

The common point between all the existing systems is the profile construction, an entity that represents the basis of every recommendation system. Some systems use only personal information like abilities or academic and professional experiences [9,15]. Other systems go even further and scan the users actions on the platform e.g.: safeguard, application...etc, to detect his needs and interests and save them in a re-usable way [7,10].

The collocation words to enrich the vocabulary, are used in many domains. In the translation, it is exploited to build multilingual dictionaries [8]. In emotions recognition, it is used to capture vocabularies of emotions [16]. Few works that use the collocation words in the recommendation systems. [9] use it to enrich the user profiles and update them regarding his actions and history.

## 3 Building a Recommendation Approach

This work is motivated by a willing to build a solution that would meet the needs of the e-recruitment platform users, and the necessity to provide them with a recommendation approach that analyses and explores their interests, infer their needs and present them opportunities that suit the best their learned preferences.

Our recommendation solution uses a content-based approach and is structured as a two-parts process. It creates first a personalized profile for each user then enrich it through the use of the job offers corpus and the user's actions monitoring.

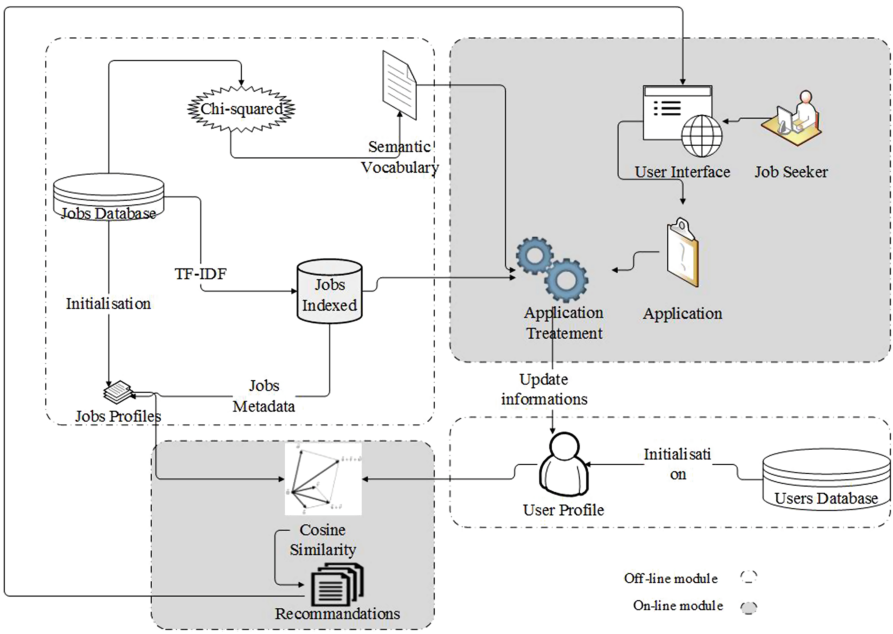


**Fig. 1.** Architecture of the jobs recommender.

The second part, is the recommendation module, which uses similarity metrics between the users updated profile and the different job offers available. Our solution is composed of two (02) modules:

**Offline Module:** Once the jobs indexed and the job profiles instantiated from jobs corpus, we initialize the user profiles from users database and establish the semantic vocabulary by searching relationships between words to enrich the profiles.

**Online Module:** When a user applies to a job offer in the user interface this module manages the enrichment of his profile and the recommendation of other job offers that match his updated profile (see Fig. 1).

### 3.1 First Challenge: Constructing the Semantic Vocabulary

**Words Listing.** We pull from the jobs corpus all useful unigrams, and for more efficiency we decided to consider the compound words, by using bigrams and trigrams. Then, we extract all correct bigrams $B$ and trigrams $T$. To avoid redundancy, for each bigram $b$ we take a subset $T'$ of all trigrams $t$ in which $b$ is included, and tested if:

$$\sum Number\ of\ occurrences\ of\ \boldsymbol{t} > \alpha \times Number\ of\ occurrences\ of\ \boldsymbol{b}$$

**Table 1.** Bigram/Trigram detection

| Bigram "b" | Occurrences | Trigrams "t" | Occurrences |
|---|---|---|---|
| Engineering Engineer | 804 | Electrical Engineering Engineer | 309 |
| | | Mechanical Engineering Engineer | 204 |
| | | Industrial Engineering Engineer | 136 |
| | | Energetic Engineering Engineer | 89 |
| | | **Sum** | 738 |

where:

$0 < \alpha < 1$: is a value to define.

$t \in T'$: is a trigram that belongs to the subset.

The Table 1 is an example that illustrate the idea.

If we define $\alpha = 0,8$, we will find that: $804 \times 0,8 \simeq 643$ is the threshold value, and $738 > 643$ so we will assume that bigram $b$ has no reason to exist and should be a trigram $t$.

**Collocations.** Once the list of unigrams, bigrams and trigrams is created, we build the semantic vocabulary. It is used to enrich the user profile. The idea is to find interesting relationships among words, using the triggers concept.

The triggers focus on words that often appear together. A word will probably trigger another if we can predict the second one when the first one occurs. Finding the collocation words is based on text windows or discourse units, in our case the text window is the job offer. The triggers are determined by calculating for each word(unigram, bigram and trigram) its *Chi-square measure* with each word(unigram, bigram and trigram) in the corpus. Then, only words with a high chi-square are kept used as triggered words.

$Chi-Square \mathcal{X}^2$: Pearson's chi-squared independence test is used in a text corpus to compute collocations which are, couples of words that occur together more than they should at random. It requires: a random sample and observations must be independent of each other.

The null hypothesis refers to a default position which corresponds to an absence of relationship between two words. Rejecting this hypothesis states that there is a relationship between these two words [17]:

$$\mathcal{X}^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

where:

$O_i$: the number of observations of type i.

$E_i = N \times p_i$: the expected (theoretical) frequency of type $i$, asserted by the null hypothesis that the fraction of type i in the population is $p_i$

$N$: total number of observations

$n$: the number of cells in the contingency table.

To assess the significance of the calculated value of $\mathcal{X}^2$, we refer to the standard chi-square table, with only one degree of freedom, which gives us a threshold value of 3.841 to compare with.

### 3.2 Second Challenge: Modelling the User Profile

**Profile Schema.** To modelize the user profile we considered his resume, skills, interests and activity fields. We have also modeled the job profile by its sector, title and the most significant terms in the job offer extracted with the $\mathcal{TF} \times \mathcal{IDF}$ *model* [14].

Considering that $V_j$ is the vector representing the job profile, and $V_u$ is the one representing the user profile, we have:

$Vector\_job =< Sector, Profession, Job\_Title, Meta-data\_job >$ annotated as $V_j =< S, P, T, M >$

$Vector\_user =< Sector, Profession, Cv\_Title, Skills, Meta - data\_user >$ annotated as $V_u =< S, P, T, S, M >$

Each component of these vectors is itself a vector that represents a collection of weighted words which are computed with the $\mathcal{TF} \times \mathcal{IDF}$.

Typically, the $\mathcal{TF} \times \mathcal{IDF}$ *weight* is composed by two terms:

$\mathcal{TF}$: Term Frequency, which measures how frequently a term occurs in a job offer. Since every job offer is different in length, it is possible that a term appears much more times in jobs with longer texts than those with shorter ones. To take into consideration this feature the term frequency is divided by the job offer text's length to normalize [16]:

$$\mathcal{TF}(t, d) = F(t, d)$$

where:

$t$: is a term

$d$: is a job offer

$F$: is the occurrence's frequency of $t$ in $d$

$\mathcal{IDF}$: Inverse Document Frequency, which measures how important a term is. While computing $\mathcal{TF}$, all terms are considered equally important. However it is known that some terms, like stop words, may appear more but have less importance.

Thus we need to weigh down the frequent terms while scaling up the rare ones, by computing the following [16]:

$$\mathcal{IDF}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

where:

$N$: is the total number of job offers in the corpus

$|\{d \in D : t \in d\}|$: is the number of job offers in which the term $t$ occurs.

**Initialisation.** To provide recommendations from the beginning, an initial profile can be used. This initial profile is automatically generated from the previously filled resume. Users are asked to fill out the resume, containing various demographic, professional data and other relevant information. The added-value of our initial profile, which uses the resume, is the fact that it allows us to avoid the cold-start. It generates recommendations from the very beginning and enables the recommender to suggest jobs according to the skills and Curriculum's content of the job seeker.

**Update.** Users are allowed to write applications for a specific job offer. The company, which has published the job, receives an email with the application and gains access to the user's resume. A user who applies to an offer clearly indicates an interest in this job offer. Meta-data of the user profile are updated as a consequence of his application to a job offer. For each word among the most significant ones in the job offer, we calculate its triggered terms using the semantic vocabulary. With these terms we constitue a new group of words, recompute their $\mathcal{TF} \times \mathcal{IDF}$ regarding the corpus, sort the results and take the best words to enrich the user profile. If the word is a new one, we add it, but if it is already included, we update its old value using the *Moving Average* [13]:

$$R_t = \alpha.W_t + (1 - \alpha).R_{t-1}$$

where:

$R_t$: is the word's weight after update

$R_{t-1}$: is word's weight before update

$W_t$: is the new weight to add pulled from $\mathcal{TF} \times \mathcal{IDF}$

$0 < \alpha < 1$: is a coefficient to define as: $\alpha = \frac{2}{N+1}$

$N$: is a a constant smoothing factor.

In our case, we take the average number of user visits per month which is $N = 7$, so we put $\alpha = 0,25$.

### 3.3 Third Challenge: Matching and Recommendations

The user and job profiles are,with all their dimensions, represented by vectors. After their construction, the cosine similarity, which is shown in the equation below, is used to compute the distance between the user profile and job profiles vectors [11]:

$$Cosine(V_u, V_j) = \frac{\sum_{i=1}^{n} V_{u_i} \times V_{j_i}}{\sqrt{\sum_{i=1}^{n} |V_{u_i}|^2 . \sum_{i=1}^{n} |V_{j_i}|^2}}$$

where:

$V_u$: is the user profile vector

$V_j$: is the job profile vector.

The smaller the angle is, the closer and more similar the job offer is to the user profile. We use this measure to compute a similarity score between the user profile and all the available offers, then we order the scores in a descending order.

### 3.4 Corpus

To achieve all of the steps stated above, we used a corpus of 49000 job offers, from which we removed approximatively 7000 offers written in English. We concentrated our work on the remaining 42140 offers written in French. We divided these offers in two parts. We used 42000 to search for the triggers and the remaining 140 to search for the experiments (see Table 2).

**Table 2.** Corpus details

| | |
|---|---|
| Corpus size | 83.238 |
| Vocabulary size | 5.032 |
| Number of Unigrams | 4.809 |
| Number of Bigrams | 169 |
| Number of Trigrams | 54 |

## 4 Evaluation

For the evaluation of our approach we proceed to a comparison between the actual recommendations service and ours. The primary goal of the experiments was to evaluate the performance of our recommendation approach, especially the contribution of the profile enrichment with triggers.

The experiment corpus is made up of 140 job offers, divided into two sets. The first one contains job offers distributed over four profiles (see Table 3). The second set has been integrated to the test corpus, in order to evaluate the real contribution of our method. It contains 48 job offers that have been randomly collected.

**Table 3.** Profiles

| Profession | Sector | Skills |
|---|---|---|
| Marketing Manager | Assistantship, secretarial | Interactions |
| HR Director | IT, Telecom, Internet | Legal Consulting |
| Networks/Systems Engineer | Education, Teaching | Cisco, Configuration |
| Communications Manager | Telecommunications, Networks | Communication |

In order to evaluate the experiment results, we used three standard metrics: *Recall*, *Precision* and *F-measure*. *Recall* is the ability of the system to return all relevant jobs, *Precision* is its ability to return only relevant jobs and *F-measure* characterizes the combined performance of *Recall* and *Precision*. Other performance metrics which are used in many fields, could measure the system's performance from its errors, namely the *False Acceptance*, where a job is wrongly accepted, and the *False Reject*, where a job is wrongly rejected. All those metrics are calculated as follows [3]:

$$Recall = \frac{Number\ of\ relevant\ jobs\ retrieved}{Number\ of\ jobs\ to\ retrieve}$$

$$Precision = \frac{Number\ of\ relevant\ jobs\ retrieved}{Number\ of\ jobs\ retrieved}$$

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision\ +\ Recall}$$

$$False\ Reject = \frac{Number\ of\ False\ Rejects}{Number\ of\ jobs\ to\ retrieve}$$

$$False\ Acceptance = \frac{Number\ of\ False\ Acceptances}{Number\ of\ jobs\ retrieved}$$

Table 4 summarizes job recommendation results obtained for the four profiles. These results show that the use of triggers have allowed to improve the performance of recommendations. Indeed, the average F-measure has increased from 53 % to 75 %.

Moreover, we obtained better Recall and Precision values, for all the profiles. For "Marketing Manager", all job offers are recommended (Recall = 100 %), thus no job offer is wrongly rejected (False Reject = 0 %).

**Table 4.** Performances of the job recommendation

| User Profile | System | Recall | Precision | F-measure | F. rejects | F. accepts. |
|---|---|---|---|---|---|---|
| Marketing Manager | Emploitic | 0,81 | 0,40 | 0,54 | 0,19 | 0,6 |
| | E-profiling | 1,00 | 0,57 | 0,72 | 0,00 | 0,43 |
| HR Director | Emploitic | 0,88 | 0,76 | 0,81 | 0,12 | 0,24 |
| | E-profiling | 0,96 | 0,77 | 0,86 | 0,04 | 0,23 |
| N/S Engineer | Emploitic | 0,58 | 0,26 | 0,36 | 0,42 | 0,74 |
| | E-profiling | 0,85 | 0,71 | 0,77 | 0,15 | 0,29 |
| Comm Manager | Emploitic | 0,65 | 0,32 | 0,43 | 0,35 | 0,68 |
| | E-profiling | 0,90 | 0,50 | 0,64 | 0,10 | 0,50 |
| Average | Emploitic | 0,73 | 0,43 | 0,53 | 0,27 | 0,57 |
| | E-profiling | 0,93 | 0,64 | 0,75 | 0,07 | 0,36 |

## 5 Discussion and Conclusion

In this paper, we have described an approach using semantic vocabulary with an indicator of interest to personalize information retrieval in an e-recruitment environment. Our approach consists of the integration of user profile in the recommendation process after catching implicit informations about him. The user profile is described using vectors of weighted words that reflect interests and preferences. This profile is constantly updated and exploited to compute the matching with job offers. The results obtained show that our approach has achieved a good performance, greatly increasing recall and precision. In the

perspectives, we propose to add other indicators of interest like: read-time of a job offer, and saving a job offer ..., We want also to use this profiling solution to optimize the search engine, and finally profile any simple visitor by creating a short term profile session.

## References


1. Adomavicius, G., Zhang, J.: Stability of recommendation algorithms. ACM Trans. Inf. Syst. (TOIS) **30**(4), 23 (2012)
2. Al-Otaibi, S.T., Ykhlef, M.: A survey of job recommender systems. Int. J. Phys. Sci. **7**(29), 5127–5142 (2012)
3. Baeza-Yates, R., Ribeiro-Neto, B., et al.: Modern Information Retrieval, vol. 463. ACM Press, New York (1999)
4. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 43–52, Morgan Kaufmann Publishers Inc (1998)
5. Burke, R.: Integrating knowledge-based and collaborative-filtering recommender systems. In: Proceedings of the Workshop on AI and Electronic Commerce, pp. 69–72 (1999)
6. Burke, R.: Hybrid web recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 377–408. Springer, Heidelberg (2007)
7. Hu, R., Pu, P.: Enhancing collaborative filtering systems with personality information. In: Proceedings of the fifth ACM Conference on Recommender Systems, pp. 197–204, ACM (2011)
8. Lavecchia, C., Smaili, K., Langlois, D., Haton, J.P.: Using inter-lingual triggers for machine translation. In: 8th Annual Conference of the International Speech Communication Association-INTERSPEECH 2007, pp. 2829–2832, ISCA (2007)
9. Lee, D.H., Brusilovsky, P.: Fighting information overflow with personalized comprehensive information access: a proactive job recommender. In: Third International Conference on Autonomic and Autonomous Systems 2007, ICAS07, pp. 21–21, IEEE (2007)
10. Rafter, R., Smyth, B.: Passive profiling from server logs in an online recruitment environment (2001)
11. Rahutomo, F., Kitasuka, T., Aritsugi, M.: Semantic cosine similarity (2012)
12. Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM **40**(3), 56–58 (1997)
13. Roberts, S.: Control chart tests based on geometric moving averages. Technometrics **1**(3), 239–250 (1959)
14. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York (1986)
15. Singh, A., Rose, C., Visweswariah, K., Chenthamarakshan, V., Kambhatla, N.: Prospect: a system for screening candidates for recruitment. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 659–668, ACM (2010)
16. Zakaria, C., Curé, O., Salzano, G., Smaïli, K.: Formalized conflicts detection based on the analysis of multiple emails: an approach combining statistics and ontologies. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2009, Part I. LNCS, vol. 5870, pp. 94–111. Springer, Heidelberg (2009)
17. Zibran, M.F.: Chi-squared test of independence. Department of Computer Science, University of Calgary, Alberta, Canada (2007). Accessed 12 Aug 2010

http://www.springer.com/978-3-319-26840-8

Theory and Practice of Natural Computing
Fourth International Conference, TPNC 2015, Mieres, Spain, December 15-16, 2015. Proceedings
Dediu, A.-H.; Magdalena, L.; Martín-Vide, C. (Eds.)
2015, XIV, 175 p. 53 illus. in color., Softcover
ISBN: 978-3-319-26840-8