Recommendation of Job Offers Using Random Forests and Support Vector Machines

Jorge Martinez-Gil
Software Competence Center Hagenberg GmbH
Hagenberg, Austria
jorge.martinez-gil@scch.at

Bernhard Freudenthaler
Software Competence Center Hagenberg GmbH
Hagenberg, Austria
bernhard.freudenthaler@scch.at

Thomas Natschläger
Software Competence Center Hagenberg GmbH
Hagenberg, Austria
thomas.natschlaeger@scch.at

ABSTRACT

The challenge of automatically recommending job offers to appropriate job seekers is a topic that has attracted many research effort during the last times. However, it is generally assumed that there is a need of more user-friendly filtering methods so that the automated recommendation systems might be more widely used. We present here our research on two methods from the data analytics field being able to disseminate job offers to the right person at the right time, which are based on Random Forest and Support Vector Machines respectively. Both methods are used here to identify the actual attributes in which users are set when they are attracted to a job offer. Preliminary results in the context of automatic job recommendation suggest that these two methods seem to be promising.

KEYWORDS

e-recruitment, data analytics, random forests, support vector machines

1 INTRODUCTION

Today, the job market is becoming more and more dynamic. In fact, this is one of the major reasons for an increasing demand for better methods for publishing or finding interesting jobs offers. Moreover, this interest is bidirectional [13], what means that it stems not only from Human Resources (HR) departments in companies, intermediaries or manufacturers of recruiting software, but also from job seekers looking for facing new professional challenges. This means that, as a first step, it is assumed that a preliminary reduction of the most promising applicants and job offers can lead to considerable improvements and savings (in terms of money, time and effort) for both parties [10]. In this context, job portals and online recruitment platforms have been traditionally designed in order to help job providers and job seekers to easily find suitable candidates and job offers respectively.

At present, many job portals and web-based recruitment systems offer their services around the world. However, there is a great corpus of literature suggesting that the functionality of the existing portals could be improved [3, 7, 14–16, 19, 23]. As a general case, only references to online job advertisements are managed, which are then classified using a simple textual description or core attributes. This means that there are serious obstacles for a satisfactory support, at least, in the side of job seekers who are forced to browse through the list of available job offers to find what better fits their needs and interests.

In order to allow job seekers to efficiently find what they are looking for, the research community has been working in a kind of information filtering mechanism (a.k.a. job recommender system [2, 6, 20]) aiming to predict the potential interest of job seekers on given job offers. More specifically, job recommender systems aim automatically suggesting job openings in such a way that as many offers as possible are offered to the right candidates at the right moment.

To appropriate face these problems, a number of alternatives have been already explored: whether data concerning the offer should be provided in a structured or unstructured way [7], which communication channels are the most appropriate in a given context [4, 5], how knowledge extraction over the job descriptions should be performed [22], and so on. However, it is widely assumed that more accurate and user-friendly filtering methods need to be developed in order to reach a wider audience for these kind of software products [18].

Our research work proposes to make this process much more smooth and comfortable for the users looking for accurate job recommendations. In fact, our methods aim to automatically identify the criteria on what potential candidates evaluate the acceptance of a given job offer. Additionally, our research aim to improve the perceived quality of recommendations as feedback is received from users. Therefore, in view of the aforementioned issues, we propose here a novel approach for the accurate recommendation of job offers using two well-known methods from the data analytics field that can have great performance in this context. In fact, the major contributions of this ongoing work can be summarized as follows:

- We propose a novel mechanism to automatically recommend job offers based on Random Forests in an accurate way.
- We propose an alternative mechanism to automatically recommend job offers based on the computation of Support Vector Machines.
- We perform an empirical evaluation of our two proposed methods with real data concerning recruitment from one of our partners.

The remainder of this work is organized in the following way: Section 2 reports the state-of-the-art on existing methods and tools for the automatic recommendation of job offers. Section 3 presents the problem that we are addressing within the frame of this work. Section 4 described our two methods to face that problem, these two methods are based on Random Forests and Support Vector Machines respectively. Section 5 reports the empirical evaluation of our methods. Section 6 outlines the analysis of the results that we have achieved from our empirical evaluation. Finally, we remark the conclusions and the future lines of research.

2 BACKGROUND

For many years, information systems for human resources (a.k.a. Human Resources Management Systems or simply HRMs) have been mainly restricted to tracking applicant's data through the applicant's management systems [11]. However, through an increasing differentiation of labor and business worlds, the process of finding the right person for a job opening and vice versa is increasing its complexity. It is clear that upcoming social media channels in addition to an overwhelming number of job portals require new strategies and technologies for both recruiters and job seekers [9].

2.1 Uses Cases

Solutions for the automatic recommendation of job offers are currently of great interest for a number of organizations that wish to automatize their e-recruitment processes. Among the most important ones, we can mention HR departments, market intermediates, electronic job platforms and portals, or software manufacturers. We offer here a closer overlook to each of them.

2.1.1 HR departments. The Human Resources (HR) departments in companies have to daily face with problems of this kind. Currently, the HR departments of large companies receive lots of incoming e-mail applications. All the application documents have to be manually process, so that the relevant information extracted can be transferred into the internal recruiting systems. This process is very time consuming and spends a lot of resources (time, money, effort). For this reason, only the data from proper candidates should be transferred into the system.

2.1.2 Market intermediaries. HR Recruiters and headhunters usually receive the order of finding the most suitable candidate for a specific job description. The challenge is so complex that many companies are willing to pay big sums for successfully completing this task. Solutions for job recommendation can help to alleviate this problem, so that it can be performed much more efficiently and effectively.

2.1.3 Electronic job platforms and portals. The segment of electronic job platforms and/or portals is subject to a strong competition. To survive in this highly competitive market, these operators provide their customers continually new and additional services. With the envisaged research results in the field of automatic job recommendation, portal operators can increase their level of innovation and therefore generate additional competitive advantages for their customers.

2.1.4 Manufacturers of recruiting software. It is also necessary to mention the manufacturers of recruiting software, since this group is constantly striving to expand their software solution continuously with additional and innovative modules to increase customer satisfaction and generate additional revenue. For this reason, software manufacturers of recruiting solutions are potentially beneficiaries of results leading to a satisfactory job recommendation.

2.2 Existing Recommendation Engines

Existing job portals are mainly based on either the use of relational databases or well-known methods from the area of information retrieval (IR). A major difference between them is that relational systems are only able to work with job offers that are already stored in the databases, while IR-based approaches may allow global searches over the Web or social networks.

When using relational databases, job offers with descriptive attributes such as job title, location, company, required skills, etc. and the URL of the job advertisement are stored in relations, and access is provided by means of database queries in standard languages such as SQL [21]. Consequently, only those vacancies matching exactly the given search criteria can be found [17]. When using IR methods, the full text search is alternatively supported by keywords whereby standard search engines can be integrated. Both procedures can be used in a similar way when searching for offers. However, IR-based methods allow to exploit semantic similarity in keywords, but this is only supported to a limited extent by standard search engines. On the other hand, these approaches generate ordered lists of URLs, where users have a proven tendency to view only the highest ranking results.

For these reasons, and regardless of the way in which job offers are handled and processed, the task of recommending the right offer to the right user has been always an important task [12]. In this way, the research community is working to find ways to make this recommendation fully satisfactory to all parties involved in the process.

2.3 Existing Methods

Techniques for automatic recommendation of job offers are specifically designed to address the problem of information overload by giving priority to information delivery for individual users based on their learned preferences [1].

The most common to process this information nowadays consists of automatically processing the documents involved in the e-recruitment process. For each document, it is possible to extract a vector for each of its fields (which contain textual information) using the bag-of-words model and TF-IDF as weighting function. Then, some kind of methods for set comparison can shed results on the suitability of a given candidate for a specific job offer.

In general, most of methods try to exploit solutions based on the Vector Space Model (VSM) to measure the similarity ratio between the original job offer and the application received. It is a solution easy to implement, with very low computational costs, and that traditional has achieved very good results in the context of job recommendation. However, new trends bet on the use of machine learning technology in order to overcome the traditional limitations concerning the incapability of going further beyond the syntactical representation of the documents.

3 PROBLEM STATEMENT

The problem that we address within the frame of this work is being able to automatically recommend job offers to the appropriate candidates. We are given past solved cases

$$(x_i, y_i), \quad x \in R^d, y \in \{-1, 1\}.$$

We want a classifier so that,

$$g(x) = \text{sign}(\phi(w) \cdot \phi(x) + b), \tag{1}$$

where

$$\phi(w) \cdot \phi(x) = K(w, x). \tag{2}$$

The key here is being able to evaluate the performance of the proposed method in relation to the past solved cases that are used to feed the algorithm in each iteration to readjust the internal parameters.

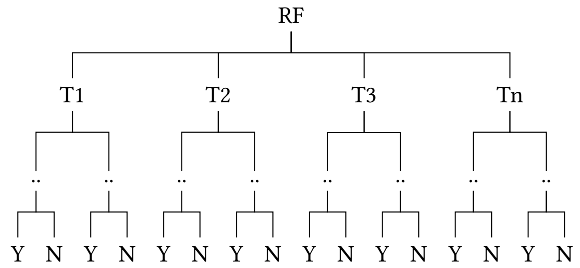

Figure 1: Example of Random Forest bagging N decision trees. Each decision tree gives a vote for a given class. Then, the random forest chooses the classification having the most votes.

4 METHODS

In order to improve the accuracy of the predictions, great research efforts have been made in the last few years concerning the definition of methods for combining a number of simple methods. These methods construct a set of hypotheses (a.k.a. ensemble), and combine the predictions of the ensemble in some way to classify new data. The precision obtained by this combination of hypotheses is usually better than the precision of each individual component. One of the most popular methods in this context are random forests.

On the other hand, algorithms based on n-dimensional geometry where given a set of past solved cases from the past are also gaining popularity. In this way, it is possible to label the classes and train the algorithm to build a geometric model that correctly classify a new sample. We give a deeper insight of these two methods below.

4.1 Random Forests

The first method that we envision in this research work is the Random Forest (RF). The rationale behind RF is to work with a given number of decision trees at the same time. Each tree gives a vote for a given class. This process is iterated by all trees. Then, the RF indicates the results having the most votes.

One of the advantages of RFs using is that, in most situations, this method is able to avoid overfitting of the training set, what it is not always possible by using other machine learning techniques. Figure 1 shows us an example of RF. Please note that, in order to work in a correct way, each decision tree has to been built following these steps:

(1) Be N the number of test cases, M is the number of variables in the classifier.
(2) The number of input variables to be used to determine the decision on a node is m; more m must be always smaller than M
(3) Select a training set for this tree and use the remaining test cases to calculate the error.
(4) For each node of the decision tree, randomly select m variables on which to base the decision. Calculate the best distribution of the training set from the variable m.

We think that the main advantages of using RF in this context can be summarized as follows:

- In general, RF has only one parameter to configure, the number of trees in the RF
- Unlike black-box models, the results obtained by RF are easier to interpret
- RF, in general, can be easily extended to support multiple classes
- RF are based on probabilistic principles

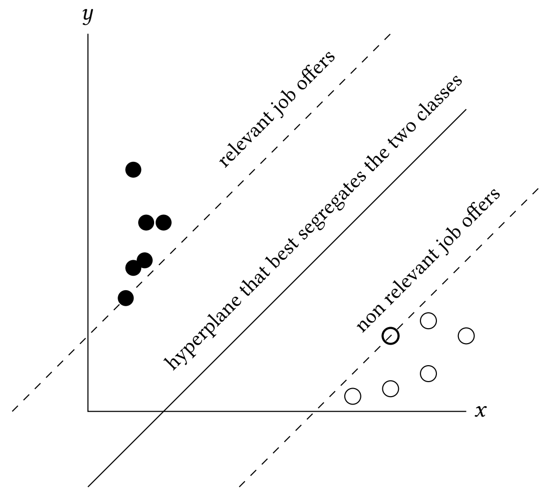

Figure 2: Example of 2-dimensional Support Vector Machine. The method consists of looking for the hyperplane that maximizes the separation between the two given classes

4.2 Support Vector Machines

Support Vector Machines (SVM) is a state-of-the-art classification method that separates data samples using the geometric notion of hyperplane. The concept behind SVM is very intuitive and easy to understand: If we have data samples that has been already classified, SVM can be used to generate multiple separation hyperplanes so that the data samples already classified can be divided into segments.

The idea is that each of these segments contain only one class. The SVM technique is generally useful and very accurate in scenarios involving some kind of classification. The reason is that SVM is designed to minimize the classification error and maximize the geometric margin.

From all the classifiers which are able to correctly classify the past samples, we are just interested in picking the closest to the hyperplane. Figure 2 shows us the rationale behind SVM with an example that represents a space of two dimensions. The aim here is to find the hyperplane that best segregates the class of relevant job offers from the class of non relevant job offers. When a new instance is added, then this hyperplane has to be recalculated in order to facilitate future classifications.

SVM has demonstrated a great performance in a number of scenarios involving some kind of classification of data samples in the past. We also think that SVM offers several advantages in the context of automatic recommendation of job offers. These advantages are the following:

- SVM has a regularization mechanism which allows avoiding over-fitting (a.k.a. geometric margin)
- SVM is defined by a optimization problem for which there is a number of existing efficient solutions
- SVM provides an approximation to a bound on the test error, which makes it very robust

SVM also has additional advantage that consists of using kernels, so that it is possible to add expert knowledge about the

Table 1: Average values and standard deviations for the numerical attributes of our data set

	Average	Std. Deviation
Workers	5069.5	9195.2
Inhabitants	361547.5	642882.9
Distance	36.3	37.4
Salary	52437.5	13717.4
Working time	38.8	3.5

problem. This aspect is out the scope of the present work, but it could be quite interesting to face it as part of our future work.

5 RESULTS

We report here the results from our experiments in the field of automatic job recommendation. We have worked with a data set of 40 job offers that have been evaluated on basis of templates or profiles. A template or profile is a pre-defined pattern that shows interest on job offers that follow certain conditions.

The sample set we are working with is not too large (mainly due to the cost of acquiring data in this context) but it can give us a good starting point to test the accuracy of these methods for solving the problem we are facing.

Before each execution, our complete data set is randomly divided in training set (80% of samples) and test set (20% of samples). The former is intended to train both RF and SVM, and the latter is intended to verify the accuracy of the method.

It is also important to mention that the attributes for each job offers are the following:

- Company name
- Position title
- City
- Distance to home
- Working hours
- Yearly salary before taxes
- Are your potentially interested (Y/N)? (to be predicted)

Table 1 shows us the average values for the attributes and its corresponding standard deviations (the amount of variation or dispersion of the values)

Moreover, the most repeated *Position Title* is programmer, although other occupations that appear in the data set are analyst, researcher, desk support or developer. The attribute to be predicted is dependent of the profile that we are analyzing. And in some cases it can be strongly unbalanced (what means that it will be an an overwhelming majority samples of one class) what makes the learning process even more difficult. However, this is how things work in real e-recruitment scenarios, where users click in either just a few or in many potential job offers, so we are facing here a realistic situation.

The results will show us the degree of accuracy that we have achieved in each case. In order to identify what is the best strategy in each of these cases, we propose a baseline method that it does not involve any kind of learning.

5.1 Baseline

In order to compare the results from our methods, we need to define a baseline method. Since we want to verify the advantages of using methods being able to analyze past solved cases, we are going to choose a baseline method with no learning capabilities. In this case, we are considering to calculate the average of the attributes for each of the offers that the potential candidate liked in the past. Then, we compare new offers with the 'average' one, and we decide if it is similar or not based on the number of similar attributes, i.e. attributes closer to the average.

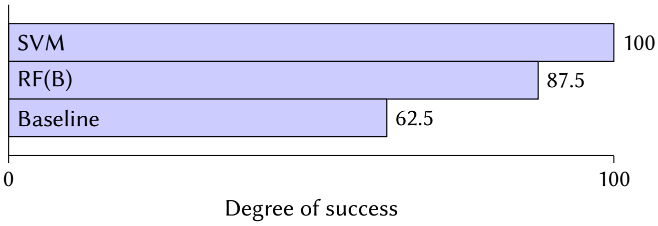

Figure 3: Results obtained for the experiment that generates a salary driven profile

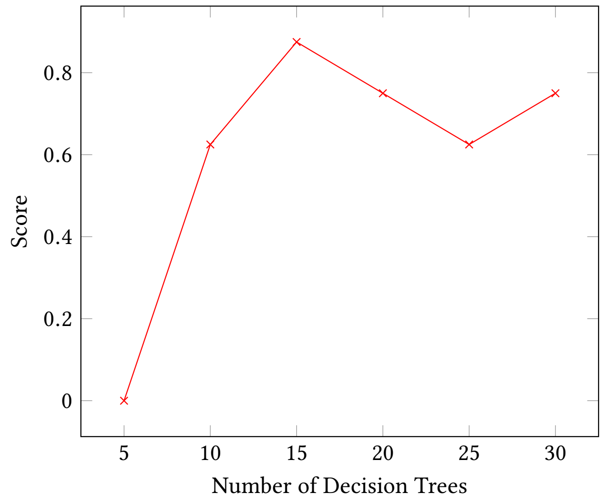

Figure 4: Evolution of the performance as more decision trees are considered in the case of a salary driven profile

5.2 Salary driven profile

The first case we are going to study is the profile of a person who is willing to be interested in job offers with very high salaries. Figure 3 shows us the results. Please note that for the RF, we pick the best result since this result can vary depending on the number of decision trees that our method is trying to bag, as we explain later.

It is very important to determine the number of decision trees that we are going to work with. To do that, we run several time the algorithm in order to determine what is the appropriate number of trees to be bagged.

From Figure 4 it is possible to see, the more decision trees we add the better get the results. However, at a certain point the benefit is lower than the cost (in terms of computing time) of including additional decision trees.

5.3 Distance driven profile

In this case, we are going to study the profile of a person who is willing to be interested in job offers for those companies that are located near its current location. Therefore, the template will have Yes in job offers with shorter distances and No in job offers for positions located further away. However, what in principle seems to be an easy scenario, it is not so easy to solve as we

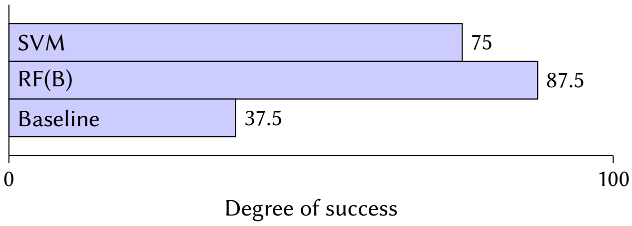

Figure 5: Results obtained for for the experiment that generates a distance driven profile

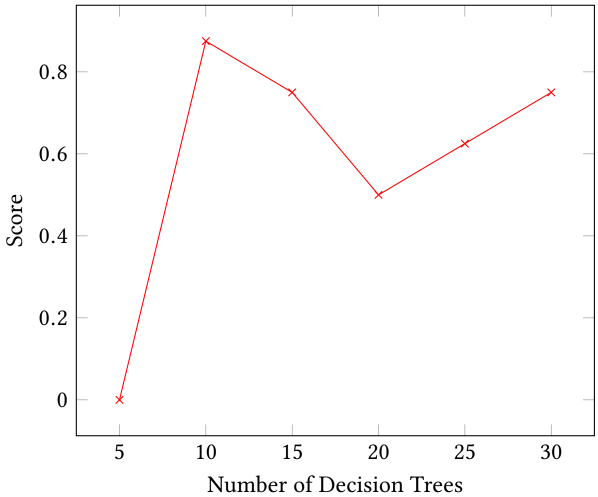

Figure 6: Evolution of the performance as more decision trees are considered in a case of distance driven profile

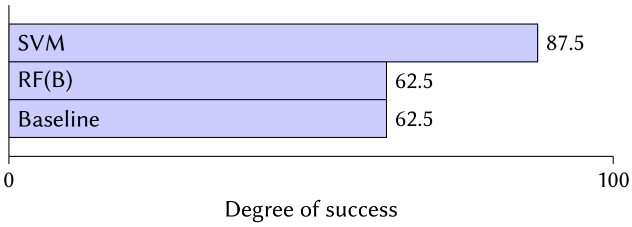

Figure 7: Results obtained for the experiment that generates a profile for a highly paid hour profile

can see in Figure 5. Reason is that the data set generated by the template is very unbalanced, what means that only a few offers a located in a surrounding area.

In Figure 6, we can see once again how the score improvement decreases as the number of decision trees increases, what means that a larger amount of trees is usually fine just to some extent.

5.4 Highly paid hour profile

In this experiment, the template is going to choose those job offers which offers the best hourly rate by the potential employer, i.e. the proportion between salary and work time seems to be more advantageous. This case is quite interesting because it might allow us understanding how our methods behave when the user looks for a complex aggregation of attributes. Figure 7 shows us the results for this experiment.

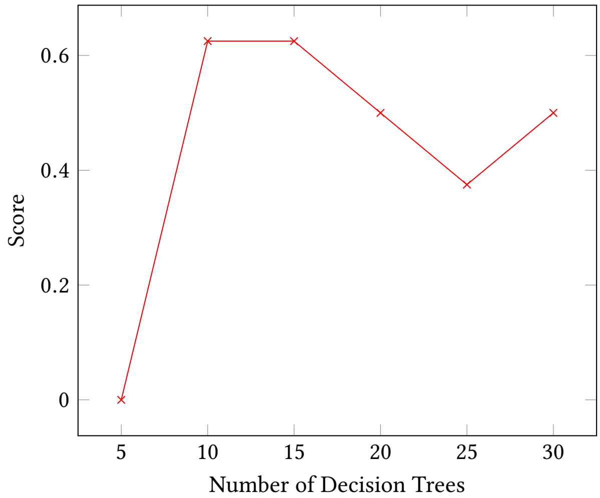

Figure 8: Evolution of the performance as more decision trees are considered for a highly paid hour profile

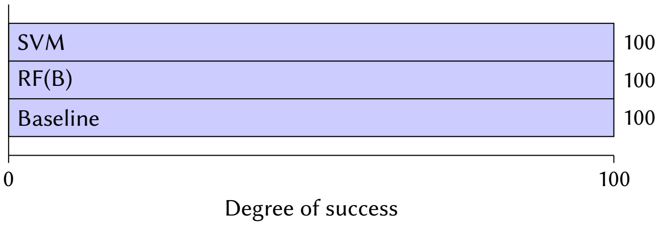

Figure 9: Results obtained for the experiment that generates a profile giving importance to big companies located in big cities

In Figure 8, we can see once again how, at some point, the improvement of the results decreases as the number of decision trees increases.

5.5 Big companies located in big cities profile

In this experiment, the template is going to choose those job offers which are offered by large companies located in big cities. This case is also interesting because it might allow us seeing how our methods deal with the fact that more than one attribute has an impact in the user's decision. Figure 9 shows us the results of the experiment. As it can be seen, it was not a difficult scenario for any of the methods considered.

For the case of RF, Figure 10 shows us the evolution of the score in relation to the number of decision trees. In this case, the RF remains stable during all the experiments.

6 DISCUSSION

From the results that we have achieved in our pool of experiments, it is possible to see that the most important advantages of our approach are:

- Both RF and SVM are quite accurate learning algorithms in the context of automatic job recommendation. For a sufficiently large data set, it is possible to build very accurate classifiers. Even for smaller samples like ours, results are better than those from methods with no learning capabilities.

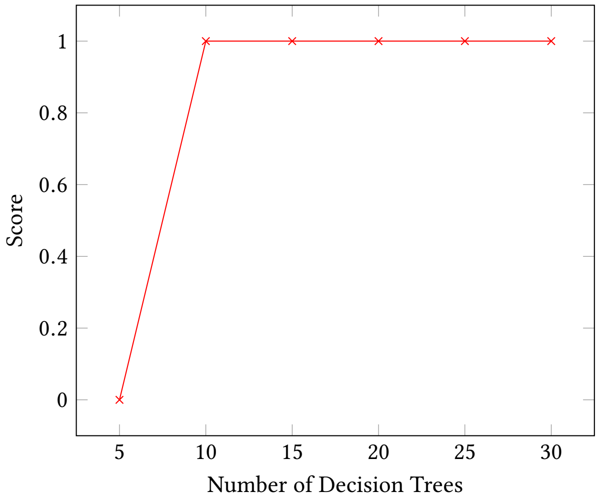

Figure 10: Evolution of the performance as more decision trees are considered for the profile Big companies located in big cities

- RF and SVM both can handle many variables without discarding any of them, what makes them good candidates to efficiently work at web scale, in large databases or with large instances.
- Last, but not least, RF is able to provide useful insights for understanding the interactions between the different variables. On the other hand, SVM operate in a less intuitive way, but in exchange, has had a better performance in most of cases.

However, an complete empirical evaluation over larger data sets should be performed in order to gain deeper insights on the advantages of these two methods. The reason is that, as we have seen, it is not always possible to obtain optimal results with small samples like ours.

7 CONCLUSIONS AND FUTURE WORK

In this work, we have presented our proposal for the automatic recommendation of job offers. Our goal here is being able to build methods being able to deliver appropriate job offers to those job seekers that could be potentially interested on them. To do that, we have based our research efforts on two well-known classification methods: random forests (RF) and support vector machines (SVM).

Our empirical evaluation shows us interesting facts. For example, RF are more likely to be interpreted although they do no present a particularly good performance in relation to SVM. On the other hand, SVM are more accurate, although they work with a model being much harder to interpret by human. What it is clear is, that in both cases, we have shown that these two methods are quite appropriate for accurately working in the context of automatic job recommendation.

As future work, we propose to design novel computational methods being able to process the textual description from the job offers. At that point, we were using just the quantitative information that is advertised. However, we think that the way an offer is written can help attracting potential candidates as well, maybe new methods for natural language processing using neural networks could help in this task. We also would like to explore the possibilities to work with expert knowledge via kernel mapping in the case of SVM as we mentioned earlier. Finally, it is also necessary to study how to integrate this technology with existing web information systems so that these two methods can be put into operation by the industry.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their useful suggestions to improve this work. The research reported in this paper has been supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH.

REFERENCES

[1] Fabian Abel, Andras A. Benczur, Daniel Kohlsdorf, Martha Larson, Robert Palovics: Proceedings of the 2016 Recommender Systems Challenge, RecSys Challenge 2016, Boston, Massachusetts, USA, September 15, 2016. ACM 2016.
[2] Daniel Bernardes, Mamadou Diaby, Raphael Fournier, Francoise Fogelman-Soulie, Emmanuel Viennet: A Social Formalism and Survey for Recommender Systems. SIGKDD Explorations 16(2): 20-37 (2014).
[3] Stefan Buschner, Rafael Schirru, Hanna Zieschang, Peter Junker: Providing recommendations for horizontal career change. I-KNOW 2014: 33:1-33:4
[4] Mamadou Diaby, Emmanuel Viennet, Tristan Launay: Toward the next generation of recruitment tools: an online social network-based job recommender system. ASONAM 2013: 821-828
[5] Mamadou Diaby, Emmanuel Viennet, Tristan Launay: Exploration of methodologies to improve job recommender systems on social networks. Social Netw. Analys. Mining 4(1): 227 (2014)
[6] Frank Faerber, Tim Weitzel, Tobias Keim: An Automated Recommendation Approach to Selection in Personnel Recruitment. AMCIS 2003: 302.
[7] Evanthia Faliagka, Lazaros S. Iliadis, Ioannis Karydis, Maria Rigou, Spyros Sioutas, Athanasios K. Tsakalidis, Giannis Tzimas: On-line consistent ranking on e-recruitment: seeking the truth behind a well-formed CV. Artif. Intell. Rev. 42(3): 515-528 (2014).
[8] Evanthia Faliagka, Athanasios K. Tsakalidis, Giannis Tzimas: An Integrated E-Recruitment System for Automated Personality Mining and Applicant Ranking. Internet Research 22(5): 551-568 (2012).
[9] Tobias Keim: Extending the Applicability of Recommender Systems: A Multilayer Framework for Matching Human Resources. HICSS 2007: 169.
[10] Stefan Lang, Sven Laumer, Christian Maier, Andreas Eckhardt: Drivers, challenges and consequences of E-recruiting: a literature review. CPR 2011: 26-35.
[11] Sven Laumer, Andreas Eckhardt: Help to find the needle in a haystack: integrating recommender systems in an IT supported staff recruitment system. CPR 2009: 7-12.
[12] Jochen Malinowski, Tim Weitzel, Tobias Keim: Decision support for team staffing: An automated relational recommendation approach. Decision Support Systems 45(3): 429-447 (2008).
[13] Jochen Malinowski, Tobias Keim, Oliver Wendt, Tim Weitzel: Matching People and Jobs: A Bilateral Recommendation Approach. HICSS 2006.
[14] Jorge Martinez Gil: An Overview of Knowledge Management Techniques for e-Recruitment. JIKM 13(2) (2014).
[15] Jorge Martinez Gil, Alejandra Lorena Paoletti, Klaus-Dieter Schewe: A Smart Approach for Matching, Learning and Querying Information from the Human Resources Domain. ADBIS (Short Papers and Workshops) 2016: 157-167.
[16] Alejandra Lorena Paoletti, Jorge Martinez Gil, Klaus-Dieter Schewe: Extending Knowledge-Based Profile Matching in the Human Resources Domain. DEXA (2) 2015: 21-35.
[17] Alejandra Lorena Paoletti, Jorge Martinez Gil, Klaus-Dieter Schewe: Top-k Matching Queries for Filter-Based Profile Matching in Knowledge Bases. DEXA (2) 2016: 295-302.
[18] Ioannis K. Paparrizos, Berkant Barla Cambazoglu, Aristides Gionis: Machine learned job recommendation. RecSys 2011: 325-328.
[19] Gabor Racz, Attila Sali, Klaus-Dieter Schewe: Semantic Matching Strategies for Job Recruitment: A Comparison of New and Known Approaches. FoIKS 2016: 149-168.
[20] Amit Singh, Rose Catherine, Karthik Visweswariah, Vijil Chenthamarakshan, Nandakishore Kambhatla: PROSPECT: a system for screening candidates for recruitment. CIKM 2010: 659-668.
[21] Eufemia Tinelli, Simona Colucci, Francesco M. Donini, Eugenio Di Sciascio, Silvia Giannini: Embedding semantics in human resources management automation via SQL. Appl. Intell. 46(4): 952-982 (2017).
[22] Eufemia Tinelli, Simona Colucci, Silvia Giannini, Eugenio Di Sciascio, Francesco M. Donini: Large Scale Skill Matching through Knowledge Compilation. ISMIS 2012: 192-201.
[23] Xing Yi, James Allan, W. Bruce Croft: Matching resumes and jobs based on relevance models. SIGIR 2007: 809-810.