Semantic Matchmaking for Job Recruitment: An Ontology-Based Hybrid Approach

Maryam Fazel-Zarandi[1], Mark S. Fox[2]

[1] Department of Computer Science, University of Toronto, Canada
[2] Department of Mechanical and Industrial Engineering, University of Toronto, Canada
mfazel@cs.toronto.edu, msf@eil.utoronto.ca

Abstract. Our aim in this work is to propose an ontology-based hybrid approach to effectively match job seekers and job advertisements. The approach uses a deductive model to determine the kind of match between a job seeker and an advertisement, and applies a similarity-based approach to rank applicants.

Keywords: Semantic Matching, Recruitment, Skill Ontology

1 Introduction

In human resource management it is often necessary to locate and match individuals and positions. Examples of such tasks include human resource recruiting, selecting individuals for teams based on different skills and qualifications, and finding the right expert to acquire information or to learn from within an organization. Currently, for human resource recruiting, the Internet is being mainly used to place online job advertisements, to perform resume search, and to acquire information about skills and competencies of individuals [4]. In order to augment and assist this process, the study and development of totally or partially automated techniques and tools have received the attention of both researchers and organizations.

To effectively locate and match individuals and positions, within or from outside an organization, it is important to use semantic technology [3][10]. The use of semantic descriptions of job offers and applicant profiles allows for qualitative and quantitative reasoning about matchings between available and required skills and competencies which is needed to improve the process of deciding who to hire and assigning individuals to tasks and teams [1]. Furthermore, semantic descriptions of applicant profiles within an organization help improve the management of individual skills and competencies of available human resources, and provide a global view of the skills available at the organizational level.

In this paper, different kinds of matchmaking strategies are combined to improve the recruitment process. We formally define job seekers and job advertisements using a skill ontology, use a deductive model to determine the kind of match between a job seeker and an advertisement, and finally use a similarity based approach to rank applicants. Thus, a description logic-based classification is performed to determine the type of match between a job seeker and an advertisement, and then ranking of applicants with partial match is done based on their similarity degree.

Related to this work are [2] and [10] which present a scenario for supporting the recruitment process with semantic web technologies for the German Government. Their approach uses [12]'s similarity measure to evaluate the degree of match between job offers and applicants. Our approach is different from theirs in that in addition to using different similarity measures, it uses logic deductive facilities which present high precision and recall. Also, desired skills and competencies are not considered in their work.

The organization of this paper is as follows: Section 2 presents the proposed ontological framework for modeling job seekers and job descriptions. Section 3 describes the matchmaking model, while Section 4 presents the ranking algorithm. Finally, Section 5 concludes the paper with a discussion of contributions made and areas of future work.

2 Ontological Framework

In human resource recruiting, two perspectives are distinguished:

- A job seeker creates an application by specifying her skills, level of competence and some sort of proof for each skill using the atomic concepts defined in an OWL-DL ontology. In other words, a job seeker is considered to be equivalent to a set of skill statements. Skills are semantically organized in a skill ontology *SkillOnt*[1]. A proof for a skill can either be a degree and/or previous work experience.
- A job advertisement is a set of requirements in terms of job related descriptions and constraints on skills, competency levels, and/or proofs. The requirements can either be must-have constraints or desired (nice-to-have) skills or degrees. In case of desires, a numeric value is also assigned indicating the importance of having that skill or degree according to the recruiter.

Description logics (DL) are used to formally represent applications and job advertisements. As mentioned, a job seeker is a person having a set of skill statements:

$$\texttt{JobSeeker} \equiv \texttt{Person} \sqcap \exists_{\geq 1}\texttt{hasSkillStatement.SkillStatement}$$

where, the concept *SkillStatement* is represented as:

$$\begin{aligned}\texttt{SkillStatement} \equiv\ & \exists_{=1}\texttt{hasSkill.Skill} \sqcap \\ & \exists_{=1}\texttt{hasCompetencyLevel} \sqcap \\ & \exists\texttt{hasProof.Proof}\end{aligned}$$

[1] We consider technical, social, and organizational skills in a specific domain of interest.

where, *hasCompetencyLevel* can take a value in the range {1,5}. A proof can either be a degree and/or previous work experience:

$$
\begin{aligned}
\texttt{Degree} \equiv\ & \texttt{Proof} \sqcap \exists_{=1}\texttt{hasTitle.Title} \sqcap \\
& \exists\texttt{hasField.StudyField} \sqcap \\
& \exists_{=1}\texttt{from.Insitution} \sqcap \\
& \exists_{=1}\texttt{startDate.Date} \sqcap \\
& \exists_{=1}\texttt{endDate.Date}
\end{aligned}
$$

$$
\begin{aligned}
\texttt{WorkExperience} \equiv\ & \texttt{Proof} \sqcap \exists_{=1}\texttt{hasPosition.Title} \sqcap \\
& \exists_{=1}\texttt{hasOrganization.Company} \sqcap \\
& \exists_{=1}\texttt{startDate.Date} \sqcap \\
& \exists_{=1}\texttt{endDate.Date} \sqcap \\
& \exists_{=1}\texttt{hasDuration.Date} \sqcap \\
& \exists\texttt{hasFunction.JobFunction} \sqcap \\
& \exists\texttt{hasSupervisor.Person}
\end{aligned}
$$

When a new job advertisement is provided, a new concept representing this advertisement is added in a OWL-DL job ontology. This concept is represented using the DL formalism as the conjunction of:

- A concept in the form $\exists\texttt{hasDescription.JobDescription}$, where

$$
\begin{aligned}
\texttt{JobDescription} \equiv\ & \exists_{=1}\texttt{hasTitle.Title} \sqcap \\
& \exists_{=1}\texttt{forCompany.Company} \sqcap \\
& \exists\texttt{hasIndustry.Industry} \sqcap \\
& \exists\texttt{hasFunction.JobFunction} \sqcap \\
& \exists_{=1}\texttt{hasType.(PartTime} \sqcup \texttt{FullTime)}
\end{aligned}
$$

- One or more concepts in the form $\exists\texttt{hasRequirement.Requirement}$, where Requirement can either be a degree or skill requirement.

$$
\begin{aligned}
\texttt{DegreeRequirement} \equiv\ & \texttt{Requirement} \sqcap \\
& \exists_{=1}\texttt{requiresDegree.Title} \sqcap \\
& \exists_{=1}\texttt{requiresField.StudyField} \\
\texttt{SkillRequirement} \equiv\ & \texttt{Requirement} \sqcap \exists_{=1}\texttt{requiresSkill.Skill} \sqcap \\
& \exists_{=1}\texttt{requiresCompetencyLevel} \sqcap \\
& \exists_{\geq 0}\texttt{requiresExperience} \sqcap \\
& \exists_{\geq 0}\texttt{requiresDegree}
\end{aligned}
$$

- Zero or more concepts in the form $\exists\texttt{hasDesire.Desire}$, where

$$\mathtt{Desire} \qquad \equiv \mathtt{Requirement} \sqcap \exists_{=1}\mathtt{hasDesireLevel}$$

where, *hasDesireLevel* can take a value in the range {1, 10}.

Job advertisements are further categorized based on industry, function, and position title. This is particularly useful for searching for jobs in terms of job related descriptions. These expressions can be represented in OWL-DL, corresponding to the *SHOIN(D)* family of description logics.

3 Matching

When searching for jobs (or applicants), a job seeker (or recruiter) can ask for all job advertisements (or applications) that match her application. In addition, a job seeker can also express her requirements in terms of desired job related descriptions. For matching job seekers to job descriptions, only must-have requirements are considered. Desires are used later for ranking.

Let ***D*** be a job advertisement with a set of requirements {$d\_req_D^i$, $s\_req_D^k$}, where $d\_req_D^i$ is the *i*-th degree requirement, and $s\_req_D^k$ is the *k*-th skill requirement of ***D***. A *Qualified* match denotes that a job seeker satisfies all the requirements of ***D***. In order to determine a qualified match, we create a new concept *C* as a conjunction of the following terms:

- For each $d\_req_D^i$, requiring degree d_i in field f_i,

$$\begin{aligned}\mathit{term}_i = {} & \exists\mathtt{hasSkillStatement.(}\exists\mathtt{hasProof.(Degree} \sqcap \\ & \exists\mathtt{hasTitle.}d_i \sqcap \exists\mathtt{hasField.}f_i\mathtt{))}\end{aligned}$$

- For each $s\_req_D^k$, requiring skill s_k with competency level l_k,

$$\begin{aligned}\mathit{term}_k = {} & \exists\mathtt{hasSkillStatement.(}\exists\mathtt{hasSkill.}s_k \sqcap \\ & \exists\mathtt{hasCompetencyLevel.}l_k \sqcap \\ & \mathtt{extra)}\end{aligned}$$

where, if $s\_req_D^k$ requires a minimum amount of experience ex_k, then

$$\mathit{extra} = \exists\mathtt{hasProof.(WorkExperience} \sqcap \exists\mathtt{hasDuration.}ex_k\mathtt{)}$$

if $s\_req_D^k$ requires a certain degree d_k in field f_k, then

$$\mathit{extra} = \exists\mathtt{hasProof.(Degree} \sqcap \exists\mathtt{hasTitle.}d_k \sqcap \exists\mathtt{hasField.}f_k\mathtt{)}$$

otherwise, *extra* is $\top$.

All instances of *C* are qualified matches for ***D***.

In real world situations, however, it rarely happens that applications match all the requirements specified in a job advertisement. For this, in addition to the qualified match, we define two types of *Under-Qualified* matches. In the first case, call it *Under-Qualified-Type-1*, an application is considered to be under-qualified for the job advertisement ***D*** if and only if[2]: 1) the competency level is less than the required competency level for a specific skill in ***D***; or 2) in case of required experience, the number of years is not satisfied. Note that when determining a match we are assuming that the skill specified in a requirement exists in the application. In other words, the cases in which one or more of the skill requirements are not satisfied at all are not considered. To determine such a match, the same approach for determining a qualified match is done with the unsatisfied constraints replaced by variables.

The *Under-Qualified-Type-2* match takes into account the fact that it is not always the case that all the required skills are present in an application. If considering all the requirements specified in a job advertisement results in finding no matching applications (qualified or under-qualified), it would be possible to iterate through all the requirements that are not satisfied, replace a skill at a time with its parent (which is a more general skill) and perform the search again until a matching application is found.

4 Ranking

In order to rank the applications matched to a job description, we need to consider three scenarios. The first scenario involves finding the most suitable applications in the set of all qualified applications (those that satisfy the qualified match criteria) for a job advertisement ***D***. In this scenario, desired skills or degrees are used to evaluate the match degree. The second scenario involves ranking the set of applications satisfying the under-qualified matching criteria. In this case some sort of a similarity measure needs to be taken into consideration. The third scenario involves the cases in which one or more of the skill requirements are not satisfied at all.

Considering the first scenario, we take into account the *desire level* values, $u(ds_i)$, assigned to each desire by the recruiter and normalize them to 1 (i.e., $\Sigma\, u(ds_i) = 1$). We can write the global match degree as the sum of the desire levels of the satisfied desired skills or degrees:

$$Sim_j = \sum x_{ji} \times u(ds_i)$$

where, x_{ji} is the Boolean variable indicating whether or not desire i is satisfied for each A_j in the set of all qualified applications. To calculate x_{ji}, for each desire a term similar to $term_i$ or $term_k$ is created and then instance checking is done to see if A_j is an instance of this term.

[2] For now we consider all degree requirements to be hard constraints.

To rank applications in the second scenario, we define two dissimilarity measures: one based on the competency levels, and another one based on the required experience.

$$CDissim_j = \sum (l_D^k > l_j^k)[(l_D^k - l_j^k)]^{\sqrt{(l_D^k - l_j^k)}}$$

where, l_D^k is the required competency level of skill requirement *k* of ***D***, and l_j^i is the competency level of application A_j for the matching skill. The term in the power exists so that if the difference in the competency levels is greater, then the two are more dissimilar. For example, if an application has a difference of competency level 4 in only one skill, and another application has a difference of competency level 2 in two skills, then the first application is more dissimilar to the job advertisement than the second one.

Similarly, we define the dissimilarity measure based on the required experience:

$$EXDissim_j = \sum (ex_D^k > ex_j^k)[(ex_D^k - ex_j^k)]^{\sqrt{(ex_D^k - ex_j^k)}}$$

where, ex_D^k is the required experience of skill requirement *k* of ***D***, and ex_j^i is the experience application A_j has for the matching skill.

Having these two measures, it is possible to rank the applications based $num_c + num_{ex}$ (or maybe a weighted sum in the case where one criteria is more important than the other) and choose the ones that are less dissimilar. If the job advertisement also includes some desires, then it is possible to use the technique used in the first scenario to further rank the applications that have equal dissimilarity values.

The ranking of applications in the third scenario is somewhat different from the previous two scenarios. For this we need to consider a node-based semantic similarity measure [12][11][6]. One such measure is the similarity measure introduced in [12] which is based on two assumptions: 1) distance between siblings is greater than the distance between parent and child, and 2) semantic distance between upper level sibling concepts is greater than between sibling concepts on lower levels:

$$SemSim(a,b) = 1 - d_c(a,b)$$

where, $d_c(a,b)$ is the distance between the two concepts *a* and *b*:

$$d_c(a,b) = d_c(a,cpp) + d_c(b,cpp)$$
$$d_c(x,cpp) = milestone(cpp) + milestone(x)$$

where, *cpp* is the closest comment parent of *a* and *b*, and *milestone* is the value assigned to each concept in the skill ontology *SkillOnt* and is calculated using the following formula:

$$milestone(x) = 1 \big/ 2k^{l(x)}$$

where, *k* is the rate the milestone decreases, and *l*(*x*) is the hierarchy level of concept *x*. Using such a similarity measure, then, it is possible to evaluate the match degree based on the similarity between the skill that was required and the skill that the applicant actually has.

5 Conclusion and Future Directions

This paper presents an approach to matching job seekers and job advertisements that combines a deductive matchmaking model based on description logics and a similarity based ranking model. Currently we are in the process of testing our approach with real data to compare the different matching and ranking criteria.

In addition to satisfying advertised job requirements, other factors such as job seekers' and recruiters' preferences, cultural fit, ability to adapt to the company's marketplace and ability to grow with the organization play an important part in selecting employees. Furthermore, when considering individuals for teams, complexities arise when considering the fitness between an individual and other team members [9]. It would be interesting to see how these complexities can be supported by automated techniques.

This approach can also be used to improve Skills Management Systems or Expertise Finding Systems within an organization. Currently the approach relies on self declarations of competences and experiences which can be inaccurate or insufficient. It would be interesting to use mechanisms to automatically discover up-to-date expertise information from secondary sources such as codes, documents, and forums. For this the domain ontology can be used to automatically annotate existing information resources and to perform automated reasoning to improve the detection and extraction of indicators of expertise [5]. Another useful ontology in this regard is the organization ontology [7] which formalizes the organizational structure and can be used to infer skills and expertise based on the roles that the agents play and the communications that occur among them. The knowledge provenance and trust ontologies presented in [8] are other examples of ontologies which can prove to be useful in this context. These ontologies can be used to formally define the semantics of information sources, information dependencies, relationships between information sources and experts, and trust relationships to improve expertise recognition and extraction.

References

1. Badica, C.; Popescu, E.; Frackowiak, G.; Ganzha, M.; Paprzycki, M.; Szymczak, M.; Park, M.W.: On Human Resource Adaptability in an Agent-Based Virtual Organization; Studies in Computational Intelligence, Vol. 134, pp. 111-120 (2008).

2. Bizer, C., Heese R., Mochol, M., Oldakowski, R., Tolksdorf, R., Eckstein, R.: The Impact of Semantic Web Technologies on Job Recruitment Processes; 7th International Conference Wirtschaftsinformatik (2005).
3. Colucci, S.; Di Noia, T.; Di Sciascio, E. ; Donini, F.M. ; Mongiello, M. ; Mottola, M.: A Formal Approach to Ontology-Based Semantic Match of Skills Descriptions, Journal of Universal Computer Science, Vol. 9, No. 12, pp. 1437-1454 (2003)
4. Dafoulas, G.A.; Nikolaou, A.N.; Turega, M.: E-Services in the Internet Job Market, in Proceedings of the 36th Hawaii International Conference on System Sciences, pp. 75 (2003)
5. Fazel-Zarandi, M.; Yu, E.: Ontology-Based Expertise Finding, in Proceedings of the 7th International Conference on Practical Aspects of Knowledge Management (PAKM 2008), Lecture Notes in Computer Science, Vol, 5345, pp. 232-243 (2008)
6. Fernandez, A.; Polleres, A.; Ossowski, S.: Towards Fine-grained Service Matchmaking by Using Concept Similarity, in Proceedings of the SMR^2 2007 Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web, South Korea (2007)
7. Fox, M.S., Barbuceanu, M., Gruninger, M., Lin, J.: An Organization Ontology for Enterprise Modelling. In: Prietula, M., Carley, K., Gasser, L. (eds.) Simulating Organizations: Computational Models of Institutions and Groups, pp. 131–152. AAAI/MIT Press, Menlo Park (1997)
8. Huang, J.: Knowledge Provenance: An Approach to Modeling and Maintaining the Evolution and Validity of Knowledge, PhD Thesis, Department of Mechanical and Industrial Engineering, University of Toronto, Canada (2008), https://tspace.library.utoronto.ca/bitstream/1807/11112/1/Huang_Jingwei_PhD_Thesis.2007-12.pdf
9. Malinowski, J.; Weitzel, T.; Keim T.: Decision Support for Team Staffing: An Automated Relational Recommendation Approach, Decision Support Systems, Vol. 42, pp 429-447 (2008)
10. Mochol, M.; Wache, H.; Nixon, L.: Improving the Accuracy of Job Search with Semantic Techniques. In Proceedings of the 10th International Conference on Business Information Systems (BIS'2007), Lecture Notes in Computer Science, Vol. 4439, pp. 301-313 (2007)
11. Schickel-Zuber, V.; Faltings, B.: OSS: A Semantic Similarity Function based on Hierarchical Ontologies. In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'2007), pp. 551-556 (2007)
12. Zhong, J.; Zhu, H.; Li, J.; Yu, Y.: Conceptual Graph Matching for Semantic Search. In Proceedings of the 10th International Conference on Conceptual Structures, pp. 92-105 (2002)