A Semantic-based Search Engine for Professional Knowledge

Simona Colucci
(Politecnico di Bari, Bari, Italy
s.colucci@poliba.it)

Tommaso Di Noia
(Politecnico di Bari, Bari, Italy
t.dinoia@poliba.it)

Eugenio Di Sciascio
(Politecnico di Bari, Bari, Italy
disciascio@poliba.it)

Francesco M. Donini,
(Università della Tuscia Viterbo, Italy
donini@unitus.it)

Azzurra Ragone
(Politecnico di Bari, Bari, Italy
a.ragone@poliba.it)

Michelantonio Trizio
(Politecnico di Bari, Bari, Italy
mikelantonio.trizio@gmail.com)

Abstract: The search for professional knowledge is affected by an endemic ambiguity in the definition of required and provided competences. Ontologies represent a mean for disambiguation, by providing a shared vocabulary for job market knowledge domain. In this paper we propose an ontology based search engine for curricula, which exploits the semantic annotation of available curricula to rank them with respect to a knowledge request and implements novel technologies for dynamic and interactive web applications development.
Key Words: Skill management, Semantic Search, Knowledge Representation
Category: I.2.4, K.6.1

1 Introduction

The main distinguishing element of knowledge with respect to any other kind of good is in its intangibility, which makes subjective and ambiguous all of transactions in job market. A step toward disambiguation consists in trying to share as

much as possible the vocabulary used to describe required and provided knowledge. The use of ontologies [4, 3], nowadays widely employed in knowledge representation, may realize such a sharing objective. The technologies developed for the Semantic Web initiative [2] make knowledge availability independent of the physical location of the knowledge provider. In the introduced scenario the deployed resources of interest contain curricula vitae: by annotating curricula vitae according to a given ontology, the disambiguation objective may be achieved. We present a semantic-based search engine for curricula vitae, exploiting Semantic Web technologies for curricula annotation according to the vocabulary provided by a shared ontology. The search process calls for reasoning tasks on formalized knowledge, which we solve by exploiting formalism and reasoning services provided by Description Logics(DL) [1].

2 The Search Engine

The decentralized architecture of the proposed search engine is sketched in Figure 1. The engine searches for a CV answering the knowledge request among the

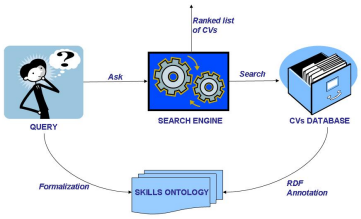

Figure 1: Search engine architecture

ones addressed by the database. The query, represented as an OWL-DL concept [7], is answered by a retrieval process exploiting RacerPro [5] to rank CVs w.r.t. the query and taking the semantics of each CV into account. CVs semantics is conveyed through their annotation w.r.t. the given ontology in a RDF file, whose URL is stored in the CVs database together with the URL of the web page containing the CV itself. The decentralized architecture allows such resources to be not necessarily resident on the same server as the search engine, given their accessibility trough URLs stored in the Database.
In order to sort CVs on the basis of their semantic similarity to required task, we propose here a ranking function, intuitively incrementing CV scores at each

occurrence of an owned skill relevant for the contextual query. More formally, consider the knowledge query Q formalized w.r.t. an ontology $\mathcal{T}$ as conjunction of concepts C_i, representing the required skills, that is to say $Q = \sqcap C_i$ with $i = 1, \ldots n$. Consider also the set $D = CV_j$ with $j = 1 \ldots m$, including CVs annotations according to $\mathcal{T}$. The scores computation mechanism is based on the algorithm detailed in the following:

Input : ontology $\mathcal{T}$, CV_j with $j = 1 \ldots m$ and $Q = \sqcap C_i$with $i = 1, \ldots n$ concepts in DL, satisfiable w.r.t. $\mathcal{T}$
Output: $V = (CV_j, score_j)$: score vector
$i = 0$;
foreach $i <= n$ **do**
  $j = 0$;
  foreach $j <= m$ **do**
    if $CV_j \sqsubseteq C_i$ **then**
      $score_j = score_j + 1$;
    end
    $j = j + 1$;
  end
  $i = i + 1$;
end
return $V = (CV_j, score_j)$;

For each element C_i of concept in the knowledge request Q, RacerPro is queried to compute CV scores as number of C_i concepts subsuming each curriculum CV_j. The search process returns curricula sorted by score and the list of URLs corresponding to such sorted list is retrieved by querying the DBMS.
The system provides a query GUI implementing a novel technology for Web 2.0 increasing the interactivity and the dynamism of web applications: Ajax [6]. The GUI also provides multilingual support, by exploiting information for translation stored in the database. The layout of the home page providing the GUI for the proposed search engine, shown in Figure 2, is divided in two main sections. The left one contains the menu through which the user performs an intensional

Figure 2: Main page layout

navigation of the knowledge domain by browsing the ontology, which is shown in a tree-like representation. Such a visualization is application independent and is built on the fly. The column on the right is aimed at containing the chosen search parameters, selected by using the Drag and Drop functionalities allowed in the GUI.

Consider for example the request for a task requiring *competences in engineering, ASP technology, and advanced knowledge about object oriented programming*. Consider also that *candidates who lived in United Kingdom(UK) for an unspecified period of time are preferred*. In order to formulate such a query the user has to drag concepts `Engineering`, `ASP`, `UK`, `OOP` from the taxonomy to the query panel. The selection of a concepts may cause the visualization of an option menu with the relations associated with the selected concept. In the introduced example the relations `livingIn` and `hasAdvancedKnowledge` have to be chosen for the concepts `UK` and `OOP`, respectively. The resulting conjunctive query is formalized as follows:
$Q = \texttt{Engineering} \sqcap \texttt{ASP} \sqcap \exists \texttt{livingIn.UK} \sqcap \exists \texttt{hasAdvancedKnowledge.OOP}$
Without loss of generality we show the application of the score computation algorithm to only one annotated curriculum, CV, belonging to a *Graduate in Engineering, living in UK*. The algorithm returns the vector $V = \{(CV, 2)\}$, because both the `Engineering` and the $\exists$`livingIn.UK` conjuncts of Q subsume CV. If several CVs are available, their scores are computed according to the proposed algorithm and the sorted list of URLs to the curricula web pages is finally shown in the results page.

3 Conclusions

We proposed a semantic based search engine for curricula supporting the process of retrieving professional knowledge. The search engine is provided with a GUI for query formalization exploiting the novel AJAX technology for dynamic and interactive web applications development.

References

1. F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2002.
2. Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 248(4):34–43, 2001.
3. T.R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies(IJHCS)*, 43(5/6):907–928, 1995.
4. N. Guarino. Formal ontology in information systems. 6-8 June 1998.
5. Volker Haarslev and Ralf Möller. RACER system description. *Lecture Notes in Computer Science*, 2083:701–706, 2001.
6. Tim O'Reilly. O'reilly – what is web 2.0.
7. OWL. www.w3.org/TR/owl-features/, 2004.