SemMF: A Framework for Calculating Semantic Similarity of Objects Represented as RDF Graphs

Radoslaw Oldakowski[1], **Christian Bizer**[2]
Freie Universität Berlin, Institut für Produktion, Wirtschaftsinformatik und OR
Garystr. 21, D-14195 Berlin, Germany
[1]cax@zedat.fu-berlin.de, [2]bizer@wiwiss.fu-berlin.de

Abstract

In this paper we introduce an easy-to-use flexible framework for computing semantic similarity between objects represented as arbitrary RDF Graphs, based on similarities of specified object properties. We outline the general architecture of the framework and describe three kinds of concept matching techniques implemented: the string matcher, the numeric matcher, and the taxonomic matcher exploiting a concept hierarchy. In a use case example we utilize the framework to match job postings with applicants' profiles.

1 Introduction

Semantic similarity measures play an important role in information retrieval by providing means to improve recall and precision. They are used in various application domains ranging from product comparison to job recruitment.

Whereas traditional similarity measures treat objects as sets of concepts drawn from a flat domain and calculate their similarity based on set intersection (shared concepts) more sophisticated approaches enhance these object models by adding a taxonomy describing the relationships among concepts leading to improved measures of similarity [3].

In this paper we introduce an easy-to-use flexible framework for computing semantic similarity between objects represented as arbitrary RDF graphs, based on similarities of specified object properties. The user is given the choice to apply either hierarchical or non-hierarchical concept matching techniques for different kinds of object properties.

2 Framework Architecture

The Matching Engine takes as input a query object and a collection of resource objects to be matched against the query object (see Figure 1). Both are represented in RDF and may utilize different schema vocabularies. If they use concepts from a common taxonomy, an RDFS or OWL representation of this taxonomy has to be provided.

2.1 Matching Description

In most cases not every object property is relevant for the similarity computation. For example, an object representing a certain product may contain manufacturer's phone number which may be irrelevant for comparing product's characteristics with customer's preferences (query object). Thus, each relevant property in the query RDF graph must be explicitly specified and mapped to the semantically corresponding property (i.e. holding the same kind of information, e.g. price information) in a resource RDF graph. Each mapping is assigned a concept matcher implementing a certain matching technique (see Section 2.2 for matcher examples).

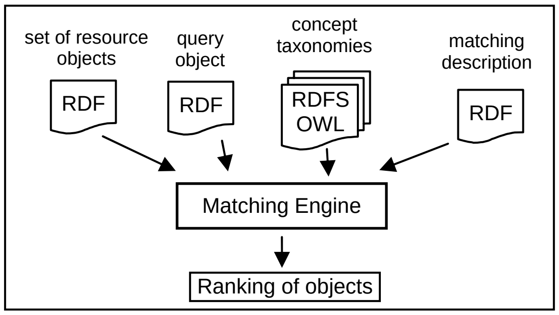


Figure 1. Matching Process

In the matching description the importance of each property can be specified by assigning it a certain *weight*. Moreover, properties to be matched can be grouped into thematic clusters, for example all properties describing the technical specification of a product. The property clustering enables to sort the matching result by cluster similarities. The matching description is represented in RDF using schema vocabulary provided with the framework.

2.2 Matchers

Owing to its flexibility by being able to work with different kinds of property matchers implementing a simple interface the framework can be utilized in various use case scenarios. In this paper we present three example implementations of different property matching techniques operating on either concepts from a taxonomy or RDF Literals. Each matcher takes as input an RDF Node from the query graph and a semantically corresponding node from the resource graph. The value returned is their similarity score. There are a number of parameters for each matcher for customizing it according to a given use case.

Taxonomic Matcher

The taxonomic matcher computes the similarity between two concepts c_1 and c_2 based on the distance $d_c(c_1, c_2)$ between them, which reflects their respective position in the concept hierarchy [4]. The matcher is able to handle multiple inheritance of concepts at the leaf level of a taxonomy.

The concept similarity is defined as: $sim_c(c_1, c_2) = 1 - d_c(c_1, c_2)$. Every concept in a taxonomy is assigned a milestone value. Since the distance between two given concepts in a hierarchy represents the path over the closest common parent (ccp), the distance is calculated as:

$$d_c(c_1, c_2) = d_c(c_1, ccp) + d_c(c_2, ccp)$$
$$\text{where } d_c(c, ccp) = milestone(ccp) - milestone(c)$$

The milestone values of concepts in a taxonomy are calculated with (as set the matching description) either:

- a *linear milestone calculator* using the formula:

$$milestone\ (n) = 1 - [l(n) / l(N)]$$

where *l(n)* is the depth of the node *n* in the hierarchy and *l(N)* represents the deepest hierarchy level.

- or an *exponential milestone calculator* using the formula from [4]:

$$milestone\ (n) = \frac{1/2}{k^{l(n)}}$$

where *k* is a factor larger than 1 indicating the rate at which the milestone values decrease along the hierarchy. It can be assigned different values depending on the hierarchy depth.

This formula implies two assumptions: the semantic differences between upper level concepts are bigger than those between lower level concepts (in other words: two general concepts are less similar than two specialized ones) and that the distance between 'brothers' is greater than the distance between 'parent' and 'child'.

Numeric Matcher

The numeric matcher is used to determine the similarity of two numeric values. A good application example for this matching technique is the comparison of product price some person is willing to pay (p_q) with the actual product price (p_r). For all $p_r > p_q$ the similarity shall decrease with increasing p_r. However, for all $p_r >$ *upper bound* specified in the matching description the similarity shall equal 0.

String Matcher

This matcher calculates the similarity of two given RDF Nodes based on their string serialization. Thus, in the case of RDF Literals not only their string value but also their language and datatype are compared.

2.3 Matching Process

Inside each thematic cluster the Engine calculates the similarity between each query property and the corresponding resource property. These similarities are multiplied by the indicated weights and summed up yielding the cluster similarity. All cluster similarities, in turn, multiplied by the specified cluster weights yield the object similarity.

However, if for a given query property value there is more than one semantically corresponding resource property value (e.g. a product may be available in different colors) the Engine chooses the one with the highest similarity.

For accessing and querying of resource and query graphs as well as the underlying taxonomies the Engine utilizes the Jena2 Semantic Web Framework [2].

2.4 Matching Result

The output of the Matching Engine is a ranking of objects by their similarity values. The Engine also provides a detailed description of the matching process (i.e. similarity values for all clusters and for each single property within a cluster together with the information about matchers used, property values, weights, etc.) which can be used to generate explanations for the calculated object similarity.

3 Use Case

The framework presented in this paper has been utilized as a foundation in a prototypical implementation of a semantic job portal [1]. It was used for matching job postings with applicants' profiles (and vice versa) represented in RDF. All three kinds of matching techniques were applied: the string matcher for Literal values (e.g. job duration [full time | part time]), numeric matcher for salary information, and the taxonomic matcher for concepts from different taxonomies like skills, industry, and occupational classifications.

4 Conclusion

Computation of semantic similarity of objects is an often occurring problem. In this paper we present an easy-to-use flexible framework for calculating semantic similarity between objects represented as arbitrary RDF Graphs, based on similarities of specified properties. The framework can be used in semantic web applications working with RDF graphs containing concepts drawn from both a flat and a hierarchical domain making it applicable in a wide range of different use cases.

SemMF is available under GNU Lesser General Public License (LGPL) from:
http://www.wiwiss.fu-berlin.de/suhl/radek/semmf/

References

[1] Ch. Bizer et al. The Impact of Semantic Web Technologies on Job Recruitment Processes. *7. Internationale Tagung Wirtschaftsinformatik (WI2005)*, Bamberg, Germany, February 2005.

[2] J. Carroll et al. Jena: Implementing the Semantic Web Recommendations. *The 14th International World Wide Web Conference (WWW2004)*, New York, May 2004.

[3] P. Ganesan et al. Exploiting hierarchical domain structure to compute similarity. *ACM Transactions on Information Systems*, 21(1):63-94, 2003

[4] J. Zhong et al. Conceptual Graph Matching for Semantic Search. *The 2002 International Conference on Computational Science (ICCS2002),* The Netherlands, Amsterdam, April 2002.

Live Demonstration of SemMF

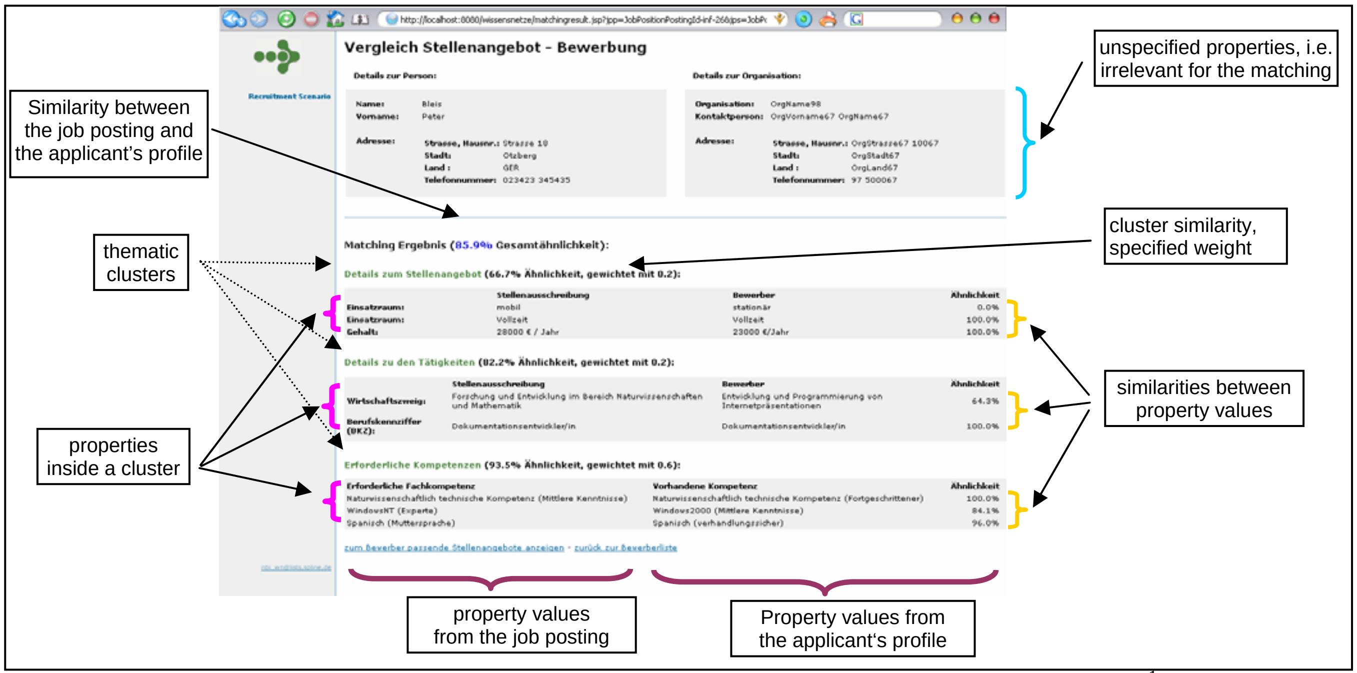
The authors are going to demonstrate the application of the Semantic Matching Framework (SemMF) in the online job recruitment domain. The framework presented in this paper has been utilized as a foundation in a prototypical implementation of a semantic job portal [1]. It is used for matching job postings with applicants' profiles (and vice versa) represented in RDF using both concepts from a flat domain as well as concepts from different taxonomies represented in RDFS.

The properties to be matched were grouped into 3 thematic clusters with following example *properties* inside each cluster:

- job details
  - o *job duration* [part time | full time]
  - o *travel desired* [true | false]
  - o *salary* [integer value]
- job specification
  - o *industry* [concepts from a taxonomy based on the *Classification of Industry Sectors (WZ2003)* – a German classification standard for economic activities]
  - o *occupation* [concepts from a taxonomy based on the *Occupation Codes (Berufskennziffer – BKZ)* – a German version of the SOC (*Standard Occupational Classification System*) which classifies workers into 5597 occupational categories according to occupational definitions]
- skill specification
  - o *skills* [concepts from a skill classification]

Using this prototypical implementation of a semantic job portal based on SemMF the authors want to show:

- an example of a Matching Description (Section 2.1) created for this use case, using schema vocabulary provided with the Framework
- what kinds of matchers (including parameters set) were used to match different properties
- how weights at property and cluster level were applied to indicate the importance of certain properties
- the generated ranked list of best matching candidates, and justify the usefulness of property clustering
- how the detailed matching information returned by the Engine was used to generate explanations (see Figure 2) for the similarities computed. Moreover, what interfaces are provided by the Engine to access this information.
- how the Matching Engine works in detail.



http://localhost:8080/wissensnetze/matchingresult.jsp?jpp=JobPositionPostingId-inf-26&jps=JobP

Recruitment Scenario

Vergleich Stellenangebot - Bewerbung

Details zur Person:

Name:	Bleis	
Vorname:	Peter	
Adresse:	Strasse, Hausnr.:	Strasse 10
	Stadt:	Otzberg
	Land :	GER
	Telefonnummer:	023423 345435

Details zur Organisation:

Organisation:	OrgName98	
Kontaktperson:	OrgVorname67 OrgName67	
Adresse:	Strasse, Hausnr.:	OrgStrasse67 10067
	Stadt:	OrgStadt67
	Land :	OrgLand67
	Telefonnummer:	97 500067

Matching Ergebnis (85.9% Gesamtähnlichkeit):

Details zum Stellenangebot (66.7% Ähnlichkeit, gewichtet mit 0.2):

	Stellenausschreibung	Bewerber	Ähnlichkeit
Einsatzraum:	mobil	stationär	0.0%
Einsatzraum:	Vollzeit	Vollzeit	100.0%
Gehalt:	28000 € / Jahr	23000 €/Jahr	100.0%

Details zu den Tätigkeiten (82.2% Ähnlichkeit, gewichtet mit 0.2):

	Stellenausschreibung	Bewerber	Ähnlichkeit
Wirtschaftszweig:	Forschung und Entwicklung im Bereich Naturwissenschaften und Mathematik	Entwicklung und Programmierung von Internetpräsentationen	64.3%
Berufskennziffer (BKZ):	Dokumentationsentwickler/in	Dokumentationsentwickler/in	100.0%

Erforderliche Kompetenzen (93.5% Ähnlichkeit, gewichtet mit 0.6):

Erforderliche Fachkompetenz	Vorhandene Kompetenz	Ähnlichkeit
Naturwissenschaftlich technische Kompetenz (Mittlere Kenntnisse)	Naturwissenschaftlich technische Kompetenz (Fortgeschrittener)	100.0%
WindowsNT (Experte)	Windows2000 (Mittlere Kenntnisse)	84.1%
Spanisch (Muttersprache)	Spanisch (verhandlungssicher)	96.0%

zum Bewerber passende Stellenangebote anzeigen - zurück zur Bewerberliste

Similarity between the job posting and the applicant's profile
thematic clusters
properties inside a cluster
unspecified properties, i.e. irrelevant for the matching
cluster similarity, specified weight
similarities between property values
property values from the job posting
Property values from the applicant's profile

Figure 2. A Detailed Explanation of the Calculated Similarity Generated by the Job Portal[1].

[1] At the conference the prototype will be presented in English.