2
Background

2.1.
RDF/OWL

According to Klyne et al. (2004), a *resource* is anything that has an identity, be it a retrievable digital entity (such as an electronic document, an image, or a service), a physical entity (such as a book) or a collection of other resources. A *Uniform Resource Identifier* (URI) is a character string that identifies an abstract or physical resource on the Web. A *URI reference* (URIref) denotes the common usage of a URI, with an optional *fragment identifier* attached to it and preceded by the character "#".

An *RDF statement* (or simply a *statement*) is a triple *(S,P,O)*, where

- *S* is a URIref, called the *subject* of the statement
- *P* is a URIref, called the *property* (also called the *predicate*) of the statement, that denotes a binary relationship
- *O* is either a URIref or a literal, called the *object* of the statement; if *O* is a literal, then *O* is also called the *value* of the property *P*

RDF offers enormous flexibility but, apart from the `rdf:type` property, which has a predefined semantics, it provides no means for defining application-specific classes and properties. Instead, such classes and properties, and hierarchies thereof, are described using extensions to RDF provided by the *RDF Vocabulary Description Language 1.0: RDF Schema* – RDF-S (Brickley et al. 2004).

In RDF Schema, a *class* is any resource having an `rdf:type` property whose value is the qualified name `rdfs:Class` of the RDF Schema vocabulary.

A class *C* is defined as a *subclass* of a class *D* by using the predefined `rdfs:subClassOf` property to relate the two classes. The `rdfs:subClassOf` property is transitive in RDF Schema.

A *property* is any instance of the class `rdfs:Property`. The

`rdfs:domain` property is used to indicate that a particular property applies to a designated class, and the `rdfs:range` property is used to indicate that the values of a particular property are instances of a designated class or, alternatively, are instances (i.e., literals) of an XML Schema datatype.

The specialization relationship between two properties is described using the predefined `rdfs:subPropertyOf`. An RDF property may have zero or more subproperties; all RDF Schema `rdfs:range` and `rdfs:domain` properties that apply to an RDF property also apply to each of its subproperties.

An *instance* of a class *C* is a resource *I* having an `rdf:type` property whose value is *C*, which is indicated by the RDF statement `(I,rdf:type,C)`. A resource may be an instance of more than one class. To define that an instance *I* of a class has a property *P* with value *V*, we simply define an RDF statement *(I, P, V)*.

OWL can be viewed as an extension of RDF. Each OWL description is also an RDF description. OWL provides extra vocabulary for relationship, cardinality and other complex schema definitions. OWL has three sublanguages – OWL Lite, OWL DL and OWL Full – which are increasingly expressive. OWL Lite, for example, only permits cardinality values of 0 or 1, while in OWL Full there is no restriction for cardinalities.

In OWL (Bechhofer et al. 2004), a *class* is any resource having an `rdf:type` property whose value is the qualified name `owl:Class` of the OWL vocabulary, which is itself a subclass of the `rdfs:Class`.

OWL distinguishes between two main categories of properties: object properties, which link individuals to individuals, and datatype properties, which link individuals to data values. The first category defines the relationships between classes. Both are subproperties of `rdfs:Property`.

A special property, `owl:sameAs`, states that two resources represent the same individual, e.g., the RDF triple `(uri1,owl:sameAs,uri2)` means that `uri1` and `uri2` represent the same individual (or instance) in the database. The property `owl:equivalentClass` and `owl:equivalentProperty` are analogous to `owl:sameAs` property, but relates two classes and two properties, respectively.

We introduce an *OWL database schema* as a set *R* of triples in the OWL

vocabulary.

An RDF triple is *of an extension of R* iff it defines an instance of a class of *R* or the value of a property defined in *R*.

An *observed extension* for *R* is a subset o_R of RDF triples of *R*. The *set of observed values* of a property *P* of *R* in o_R is defined as

$$o_R[P] = \{ V / (S,P,V) \in o_R \}$$

Likewise, the *set of observed instances* of a class *C* of *R* in o_R is defined as

$$o_R[C] = \{ S / (S,\texttt{rdf:type},C) \in o_R \}$$

2.2. Similarity models

Similarity is a concept frequently used in many different applications. Various similarity functions have been proposed in the literature, such as information content (Resnik 1995), information theory (Brauner et al. 2008, Hindle 1990, Lin 1998), vector model (Frakes and Baeza-Yates 1992), distance measurements (Lee 1993) and the contrast model (Tversky and Gati 1978).

In this thesis, we use and compare results of four similarity functions. The first one is based on the vector model. In text processing applications (Frakes and Baeza-Yates 1992), the documents and the queries are represented by vectors. The relevance of documents to queries is expressed as a measure of the similarity between the vectors, taken as the cosine of the angle between the vectors:

$$sim(\vec{A}, \vec{B}) = \frac{\vec{A} \bullet \vec{B}}{\left|\vec{A}\right|\left|\vec{B}\right|}$$

The dimensions of the vectors represent index terms of a document or a word of a query, and the corresponding coordinates are the weights (TF/IDF) of the term, e.g., a document of a corpus *C* with *n* index terms can be represented as the vector $\vec{A} = (w_{t_1}, w_{t_2}, ..., w_{t_N})$, where w_{t_i} is defined as follows:

$$w_{t_i} = \frac{f_{t_i}}{\max_{t \in A} f_t} \log \frac{N}{N_{t_i}}$$

where

- f_{t_i} = frequency of a term t_i in the document
- N = number of terms in C
- N_{ti} = number of documents of C that include term t_i

The second similarity function is based on Information Theory and was proposed in (Lin 1998). The similarity between two objects *A* and *B* is a function of the amount of information in the propositions of their *commonalities* and *descriptions*. The authors use the conclusion of (Cover and Thomas 1991), which says that the information contained in a statement is measured by the negative logarithm of the probability of the statement. The similarity is defined as follows:

$$sim(A,B) = \frac{\log(P(commonalities(A,B)))}{\log(P(description(A,B)))}$$

where *commonalities(x,y)* and *description(x,y)* are functions that return subsets of features of the objects *x* and *y*, and *P(x)* is the probability of the set of features *x*. For example, if *A* is an orange and *B* is an apple, the *commonalities* between *A* and *B* can be stated as the proposition *fruit(A) and fruit(B).* The predicate *fruit* is a possible common feature of both objects from a predefined set of features. If *A* and *B* belong to a set *S* of objects then

$$P(commonalities(A,B)) = P(fruit(A)).P(fruit(B))$$

i.e., the probability of commonalities is the probability of the occurrence of the predicate *fruit* for the two objects. The description of *A* and *B* is the probability of the union of all features of the two objects.

The third similarity function is presented in (Brauner et al. 2008), where the authors address the problem of matching the attributes of two relational schemes, $R[A_1,\ldots,A_m]$ and $S[B_1,\ldots,B_m]$. Given two relations σ_R and σ_S that follow the schemes *R* and *S*, the authors first propose to compute the $m \times n$ co-occurrence matrix $[m_{ij}]$ such that m_{ij} is the cardinality of $\sigma_R[A_i] \cap \sigma_S[B_j]$. The next step is to compute the *Estimated Mutual Information* matrix EMI defined as,

$$EMI_{rs} = \frac{m_{rs}}{M} \log\left(M \frac{m_{rs}}{\sum_j m_{rj} * \sum_i m_{is}} \right)$$

where $m_{ij} = |\, \sigma_R[A_i] \cap \sigma_S[B_j] \,|$, for $i \in [1,m]$ and $j \in [1,n]$, and $M = \sum_{i,j} m_{ij}$.

The authors then postulate that two attributes A_r and B_s *match* iff $EMI(A_r,B_s) \geq EMI(A_r,B_j)$, for all $j \in [1,n]$, with $j \neq s$, and $EMI(A_r,B_s) \geq EMI(A_i,B_s)$, for all $i \in [1,m]$,with $i \neq r$.

The last similarity function is based on the contrast model (Tversky and Gati 1978), which states that the similarity between *x* and *y* increases with the amount of features, measured by a given function *f*, which *x* and *y* have in common, and decreases with the amount of features which belong to just *x* or to just *y*. The notion of feature is used here with the same meaning as in the second similarity function presented before, i.e., they are predicates or characteristics of the objects. The contrast model has been evaluated and successfully used in many applications (Eidenberger 2006, Eidenberger and Breiteneder 2002, Tang et al. 2007). One possible reason for the success of the contrast model is that it is very close to the human perception of similarity.

More precisely, let *C* be a set of features and let 2^C denote the power set of *C*. Let $f: 2^C \rightarrow \mathbb{R}^+$ be a *scale function* for *C*. A *contrast model* is a function $\tau: 2^C \times 2^C \rightarrow \mathbb{R}^+$ such that

(1) $$\tau(x, y) = \theta f(x \cap y) - \alpha f(x - y) - \beta f(y - x)$$

for any $x, y \in 2^C$, where $\theta, \alpha, \beta \in \mathbb{R}^+$ are the *parameters* of the contrast model. Note that this formula defines a class of models that depend on the choice of *f*, θ, α and β.

Now, let $|x|$ denote the cardinality of a set *x*. Using the cardinality as the scale function, we may successively rewrite (1) as:

(2) $$\tau_{\theta,\alpha,\beta}(x, y) = \theta \,|\, x \cap y \,| - \alpha \,|\, x - y \,| - \beta \,|\, y - x \,|$$

(3) $$\tau_{\theta,\alpha,\beta}(x, y) = \theta \,|\, x \cap y \,| - \alpha(|\, x \,| - |\, x \cap y \,|) - \beta(|\, y \,| - |\, x \cap y \,|)$$

(4) $\tau_{\theta,\alpha,\beta}(x,y) = (\theta+\alpha+\beta)\,|\,x\cap y\,| - \alpha\,|\,x\,| - \beta\,|\,y\,|$, for any $x, y \in 2^C$

In order to normalize the result, Equation (4) can be balanced with $|C|$. We then redefine the function $\tau_{\theta,\alpha,\beta}$ as

(5) $$\tau_{\theta,\alpha,\beta}(x,y) = \frac{(\theta+\alpha+\beta)\,|\,x\cap y\,| - \alpha\,|\,x\,| - \beta\,|\,y\,|}{|C|}$$

To simplify the notation, define $\overline{N}(x) = |\,x\,|/|C|$ and rewrite equation (5) as:

(6) $$\tau_{\theta,\alpha,\beta}(x,y) = (\theta+\alpha+\beta)\overline{N}(x\cap y) - (\alpha\overline{N}(x) + \beta\overline{N}(y))$$

The image of such function is contained in $\mathbb{R}^+$, which imposes serious restrictions on fixing a threshold to select similar properties. For this reason, it is convenient to rewrite the formula using *log(* $\overline{N}(x)$ *)*, instead of $\overline{N}(x)$.

(7) $$\tau_{\theta,\alpha,\beta}(x,y) = \log\left(\frac{\overline{N}(x\cap y)^{(\theta+\alpha+\beta)}}{\overline{N}(x)^{\alpha}\,\overline{N}(y)^{\beta}}\right)$$

Since $\overline{N}(x\cap y) \geq 0$, $\overline{N}(x) \geq 0$, $\overline{N}(y) \geq 0$, $\overline{N}(x\cap y) \leq \overline{N}(x)$, $\overline{N}(x\cap y) \leq \overline{N}(y)$, $\overline{N}(x\cap y) \leq 1$, then $\tau_{\theta,\alpha,\beta}(x,y)$ is always negative.

In order to limit the similarity values to the interval [0.0,1.0] equation (7) can be rewritten as following.

(8) $$\tau_{\theta,\alpha,\beta}(x,y) = \left(1 - \log\left(\frac{\overline{N}(x\cap y)^{(\theta+\alpha+\beta)}}{\overline{N}(x)^{\alpha}\,\overline{N}(y)^{\beta}}\right)\right)^{-1}$$

2.3. Summary and contributions

The assumptions that the database schemas to be matched are described in OWL and that the data obtained from the databases is available as sets of RDF triples facilitate the construction of matching techniques, since schema elements and data instances are similarly defined (as RDF triples). However, the techniques

introduced in the thesis can be directly applied to conceptual schemas described in other database models, such as the relational model. In conjunction, these assumptions permit us to concentrate on a strategy to unveil the semantics of the database schemas to be matched, without being distracted by syntactical peculiarities.

In fact, we see as a good practice to provide OWL descriptions of the export schemas of data sources providers. In conjunction with WSDL descriptions of the Web Services encapsulating the backend databases, this measure facilitates the interoperability of databases.

The techniques we describe in this thesis uniformly apply similarity functions to generate matchings and are grounded on the interpretation, traditionally accepted, that “terms have the same extension when true of the same things” (Quine 1968). In our context, two concepts match if they denote similar sets of objects. The techniques essentially differ on the nature of the sets to be compared and on the similarity functions adopted. For example and in a very intuitive way, two classes match if their sets of properties are similar, two terms from different thesauri match if the sets of instances they classify are similar, properties match if their sets of observed values are similar.