HAL
open science

Flexible SPARQL querying of Web data tables driven by an ontology

Patrice Buche, Juliette Dibie-Barthelemy, Hajer Chebil

To cite this version:

Patrice Buche, Juliette Dibie-Barthelemy, Hajer Chebil. Flexible SPARQL querying of Web data tables driven by an ontology. 8. International Conference on Flexible Querying and Answering Systems, Oct 2009, Roskilde, Denmark. 676 p. hal-01123231

HAL Id: hal-01123231

https://hal.science/hal-01123231v1

Submitted on 7 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Flexible SPARQL querying of Web data tables driven by an ontology

Patrice Buche and Juliette Dibie-Barthélemy and Hajer Chebil

UMR AgroParisTech/INRA MIA - INRA Unité Mét@risk
AgroParisTech, 16 rue Claude Bernard, F-75231 Paris Cedex 5, France
{buche, dibie}@agroparistech.fr

Abstract. This paper concerns the design of a workflow which permits to feed and query a data warehouse opened on the Web, driven by a domain ontology. This data warehouse has been built to enrich local data sources and is composed of data tables extracted from Web documents. We recall the main steps of our semi-automatic method to annotate Web data tables driven by a domain ontology. The output of this method is an XML/RDF data warehouse composed of XML documents representing Web data tables with their fuzzy RDF annotations. We then present how to query simultaneously the local data sources and the XML/RDF data warehouse, using the domain ontology, through a flexible querying language. This language allows preferences to be expressed in selection criteria using fuzzy sets. We study more precisely how to retrieve approximate answers extracted from the Web data tables by comparing preferences expressed as fuzzy sets with fuzzy annotations using SPARQL.

1 Introduction

Today's Web is not only a set of semi-structured documents interconnected via hyper-links. A huge amount of technical and scientific documents, available on the Web or the hidden Web (digital libraries, ...), include data tables. Those data tables can be seen as small relational databases even if they lack the explicit meta data associated with a database. They represent a very interesting potential external source for feeding the data warehouse of a company dedicated to a given domain of application. They can be used to enrich local data sources or to compare local data with external ones. To reach this aim, a preliminary step consists in harmonizing external data with local ones. It means that external data must be expressed with the same vocabulary as the one used to index the local data. We have designed and created a software called @WEB (Annotating Tables from the WEB), using the semantic web framework, which implements the entire management workflow, presented in Figure 1, to complement existing local data sources with Web data tables. This workflow relies on a domain ontology extracted from the local data sources and can be divided into the following two main steps: (1) feeding an XML/RDF data warehouse with data tables which have been extracted from documents retrieved from the Web and annotated according to the domain ontology (tasks 1 to 3 in Figure

1); (2) querying simultaneously the local data sources and the XML/RDF data warehouse using the domain ontology in order to retrieve approximate answers in an homogeneous way (task 4 in Figure 1).

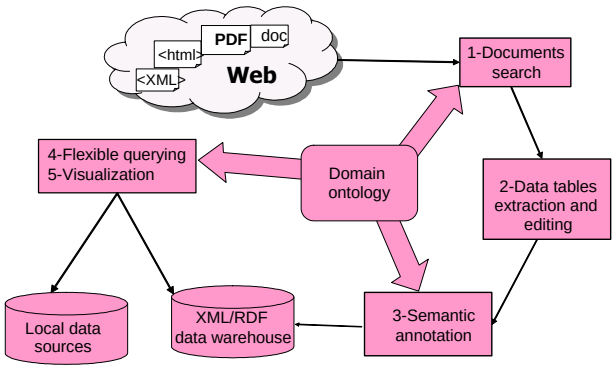


Fig. 1. Main steps of the @WEB workflow.

The first step of the @WEB software generates fuzzy annotations, which are represented in a fuzzy extension of RDF, of Web data tables. These fuzzy RDF annotations consist in: (i) the recognition and the representation of imprecise data appearing in the cells of the Web data table; (ii) an explicit representation of the semantic distance between the Web data tables and the ontology. The second step of the @WEB software allows the fuzzy RDF annotations to be queried using SPARQL which is recommended by the W3C to query RDF data sources (see http://www.w3.org/TR/rdf-sparql-query/). The main originalities of our flexible querying system are: (i) to retrieve not only exact answers compared to the selection criteria but also semantically close answers thanks to the use of hierarchical fuzzy sets (see [1]); (ii) to compare the selection criteria expressed as fuzzy sets representing preferences with the fuzzy annotations of Web data tables representing either imprecise data or the semantic distance between Web data tables and the ontology.

In this paper, we focus on the flexible querying step of the @WEB software. In section 2, we recall the first step of the @WEB software by focusing on the semantic annotation method (see [2]) which allows Web data tables to be annotated thanks to a domain ontology. In section 3, we propose a mediator approach to perform flexible querying of the annotated Web tables using SPARQL. We provide some experimental results in 4 and we compare our approach with the state of the art in 5. We conclude and present the perspectives of this work in section 6.

2 Annotation of Web data tables driven by a domain ontology

In order to enrich local data sources with data extracted from the Web, we have designed a semi-automatic acquisition tool, called @WEB, driven by a domain ontology. We first recall the OWL representation of the domain ontology. Secondly, we recall the first step of the @WEB system which concerns the extraction of data tables from the Web and their semantic annotation according to a domain ontology.

2.1 OWL representation of the domain ontology

The OWL representation of the domain ontology used by the @WEB system is divided into two parts. First, the definition of the structure of the ontology is domain independent. It is composed of symbolic types, numeric types and relations between these types. Second, the definition of the content of the ontology is domain dependent. It is composed of the definition of the actual types and relations of the domain. This second part of the ontology has been extracted from the explicit meta-data and data of the local relational database which is enriched by the @WEB system. Examples provided in this paper concern an application to microbial risk in foods. Let us detail the symbolic types, the numeric types and the relations of the ontology.

Symbolic types are described by a type name, a list of synonyms for the type name and a taxonomy of possible values. Our ontology on food microbiology contains 3 symbolic types. For example, the symbolic type *Microorganism* is associated with a taxonomy of more than 150 microorganisms in which Clostridium botulinum and Staphylococcus Spp. are kind of Gram+ and Salmonella is a kind of Gram-. Each symbolic type is represented by an OWL class, subclass of the generic class SymbolicAttribute. The taxonomy of values of a symbolic type is viewed as a hierarchy of subclasses: the symbolic type is associated with the root of its hierarchy via the property *HasForTaxonomy*.

Numeric types are described by a type name, a list of synonyms for the type name and the set of units in which the type can be expressed and eventually a numeric range. Our ontology on food microbiology contains 18 numeric types. For example, the numeric type Aw[1] has no unit and is restricted to the range $[0, 1]$. Each numeric type is represented by an OWL class, subclass of the generic class NumericalAttribute. The optional numeric range of a numeric type is associated with the numeric type via the properties *HasForMinValue* and *HasForMaxValue*. The set of units, if there exists, is associated with a numeric type via the property *AssociatedUnitList*.

Relations are described by the name of the relation and its signature. The signature of a relation is divided into one result type (the range of the relation)

[1] Aw is the water activity and corresponds to an index of the water which is available in the food to be used by microorganisms.

and several access types (the domain of the relation). Our ontology on food microbiology contains 16 relations. For example, the relation *GrowthParameterAw* represents the growth limits of a microorganism concerning water activity of any food product. This relation has for domain the symbolic type *Microorganism* and for range the numeric type *Aw*. The relations in the ontology are n-ary. As advised by [3], each relation is represented by an OWL class, subclass of the generic class Relation, which is associated with the types of its signature via the properties *AssociatedKey* (for the access types) and *AssociatedResult* (for the result type).

The names of types and relations, as well as the possible values of a symbolic type defined in its taxonomy, are called terms. These terms will be used to annotate data tables extracted from the Web. We have separated in the OWL representation of the ontology the concepts (i.e. the types, the relations and the values of the symbolic types taxonomies) from their actual terms with their words. Each concept of the ontology is linked to its corresponding term via the property *AssociatedTerm*.

2.2 Annotation of Web data tables

The @WEB system relies on five tasks as described in Figure 1. We briefly present here its three first tasks concerning the feeding of the XML/RDF data warehouse with Web data tables. The first task consists in retrieving relevant Web documents for the application domain, in html or in pdf, using key-words, which have been extracted from the domain ontology, to define queries executed by different crawlers. In the second task, data tables are extracted from the Web documents and are semi-automatically translated into a generic XML format. The Web data tables are then represented in a classical and generic way – a table is a set of lines, each line being a set of cells. In the third task, the Web data tables are semantically annotated according to the domain ontology. The semantic annotation process of a Web data table consists in identifying which semantic relations of the domain ontology are represented in the data table (see [2] for more details). This process generates RDF descriptions which represent the semantic relations of the ontology recognized in each row of the Web data table.

Organism	aw minimum	aw optimum	aw maximum
Clostridium	0.943	0.95-0.96	0.97
Staphylococcus	0.88	0.98	0.99
Salmonella	0.94	0.99	0.991

Table 1: Cardinal values

Fig. 2. Example of a Web data table

Example 1 *Figure 2 presents an example of a Web data table in which the semantic relation GrowthParameterAw has been identified. The first line of the Web data table indicates that Clostridium has a growing range between 0.943 and 0.97 which is optimal in the range [0.95, 0.96].*

Some of the RDF descriptions associated with Web data tables by our semantic annotation process include values expressed as fuzzy sets (see [4]).

Definition 1 A **fuzzy set** f on a definition domain $Dom(f)$ is defined by a membership function μ_f from $Dom(f)$ to $[0, 1]$ that associates the degree to which x belongs to f with each element x of $Dom(f)$. We call kernel (resp. support) of the fuzzy set, the set of elements x with $\mu_f(x) = 1$ (resp. $\mu_f(x) \neq 0$).

We distinguish two kinds of fuzzy sets: (i) discrete fuzzy sets and (ii) continuous fuzzy sets.

Definition 2 A **discrete fuzzy set** f, denoted by DFS, is a fuzzy set associated with a relation or a symbolic type of the ontology. Its definition domain is the set of relations or the type hierarchy.

Definition 3 A **continuous fuzzy set** f, denoted by CFS, is a trapezoidal fuzzy set associated with a numeric type of the ontology. A trapezoidal fuzzy set is defined by its four characteristic points which correspond to min(support(f)), min(kernel(f)), max(kernel(f)) and max(support(f)). Its definition domain is the interval of possible values of the type.

The fuzzy values used to annotate Web data tables may express two of the three classical semantics of fuzzy sets (see [5]): similarity or imprecision.

Example 2 *Figure 3 presents a part of the RDF descriptions corresponding to the recognition of the relation GrowthParameterAw in the first row of the Web data table shown in figure 2. The first description expresses that the first row (having the URI uriRow1 in the XML document) is annotated by a discrete fuzzy set. This fuzzy set, typed by the OWL class DFS, has a semantic of similarity and indicates the list of closest relations of the ontology compared to the first row. Only the relation GrowthParameterAw belongs to this fuzzy set with the pertinence score of 1.0. This pertinence score expresses the degree of certainty associated with the relation recognition by the semantic annotation process. The domain of the relation, which is an instance of the symbolic type Microorganism, is annotated by a discrete fuzzy set. This fuzzy set, typed by the OWL class DFS, has a semantic of similarity and indicates the list of closest values of the ontology compared to the value Clostridium. Two values (Clostridium Perfringens and Clostridium Botulinum) belong to this fuzzy set with a membership degree of 0.5. The range of the relation, which is an instance of the numeric type aw, is annotated by a continuous fuzzy set. This fuzzy set, typed by the OWL class CFS, has a trapezoidal form and a semantic of imprecision. It indicates the possible growth limits ([0.943, 0.97]) and the possible optimal growth limits ([0.95, 0.96]) represented respectively as the support and the kernel of the fuzzy set.*

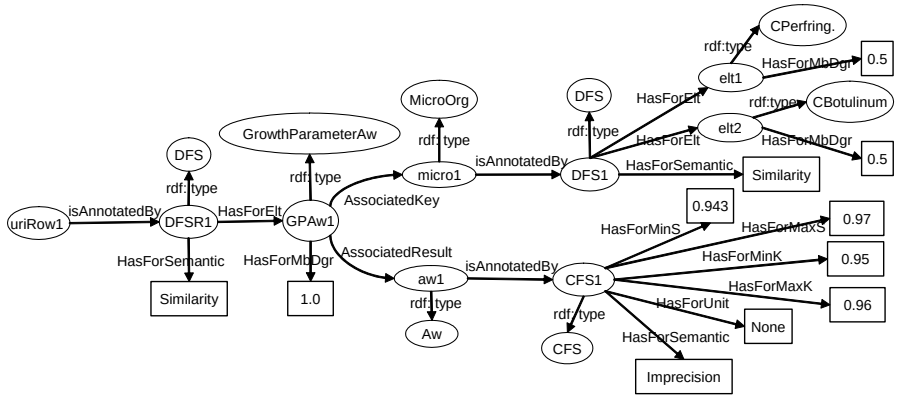


Fig. 3. Example of RDF annotations generated from the Web data table of figure 2

The output of the first step of the @WEB system is an XML/RDF data warehouse composed of a set of XML documents which represent Web data tables and their associated fuzzy RDF annotations.

3 Flexible querying of Web data tables driven by a domain ontology

We present in this section the second step of the @Web system which proposes a unified flexible querying system, called MIEL++, of the local data sources and the XML/RDF data warehouse.The MIEL++ querying system relies on the domain ontology used to index the local data sources and to annotate the Web data tables. MIEL++ allows the end-user to retrieve the nearest data stored in both sources corresponding to his/her selection criteria: the ontology -more precisely the type hierarchies- is used in order to assess which data can be considered as near to the users selection criteria.

A query is asked to the MIEL++ system through a single graphical user interface (GUI), which relies on the domain ontology. The query is translated by each subsystem's wrapper into a query expressed in the query language of the subsystem: an SQL query in the relational subsystem (see [6] for more details about the SQL subsystem), a SPARQL query in the XML/RDF subsystem. Finally, the global answer to the query is the union of the local results of the two subsystems, which are ordered according to their relevance to the query selection criteria.

In this section, we present the XML/RDF subsystem which allows the end-user to query fuzzy RDF annotations of Web data tables, represented in XML documents, by means of SPARQL queries. We remind the notions of view and MIEL++ query (see [6] for more details). We then detail the translation of a

MIEL++ query into a SPARQL query. We finally present the construction of a MIEL++ answer in the XML/RDF subsystem.

3.1 MIEL++ query

A MIEL++ query is asked in a view which corresponds to a given relation of the ontology. A view is characterized by its set of queryable attributes and by its actual definition. Each queryable attribute corresponds to a type of the relation represented by the view. The concept of view must be understood with the meaning of the relational database model. It allows the complexity of the querying in a given subsystem to be hidden to the end-user.

A MIEL++ query is an instanciation of a given view by the end-user, by specifying, among the set of queryable attributes of the view, which are the selection attributes and their corresponding searched values, and which are the projection attributes. An important specificity of a MIEL++ query is that searched values may be expressed as fuzzy sets. A fuzzy set allows the end-user to represent his/her preferences which will be taken into account to retrieve not only exact answers (corresponding to values associated with the kernel of the fuzzy set) but also answers which are semantically close (corresponding to values associated with the support of the fuzzy set). Since the XML/RDF data warehouse contains fuzzy values generated by the annotation process, the query processing has to (1) take into account the pertinence score associated with the semantic relations identified in Web data tables and (2) compare a fuzzy set expressing querying preferences to a fuzzy set, generated by the annotation process, having a semantic of similarity or imprecision. For the first point, the end-user may specify a *threshold* which determines the minimum acceptable pertinence score to retrieve the data. The second point is studied in section 3.3.

Example 3 *Let us define a MIEL++ query Q expressed in the view GrowthParameterAw:* $Q = \{Microorganism, aw | (GrowthParameterAw(Microorganism, aw) \wedge$
$(Microorganism \approx MicroPreferences) \wedge (aw \approx awPreferences) \wedge (thresh \geq 0.5)\}$. *The discrete fuzzy set MicroPreferences, which is equal to {1.0/Gram+, 0.5/Gram-}, means that the end-user is firstly interested in microorganisms which are Gram+ and secondly Gram-. The continuous fuzzy set awPreferences, which is equal to [0.9, 0.94, 0.97, 0.99], means that the end-user is first interested in aw values in the interval [0.94, 0.97] which corresponds to the kernel of the fuzzy set. But he/she accepts to enlarge the querying till the interval [0.9, 0.99] which corresponds to the support of the fuzzy set. GrowthParameterAw relations having a pertinence score inferior to 0.5 are discarded.*

3.2 Translation of a MIEL++ query into a SPARQL query

In a MIEL++ query, the end-user can express preferences in his/her selection criteria as fuzzy sets. Since fuzzy sets are not supported in a standard SPARQL query, we propose to defuzzify the MIEL++ query before translating it into

SPARQL. We first present the defuzzification of a MIEL++ query, we then present the translation of the defuzzified MIEL++ query into a SPARQL query.

Defuzzification of a MIEL++ query The defuzzification is not the same for a discrete fuzzy set and for a continuous fuzzy set.

When the fuzzy value of a selection criterion has a hierarchized symbolic definition domain, it is represented by a discrete fuzzy set defined on a subset of its definition domain. Such a fuzzy set defines degrees implicitly on the whole definition domain of the selection attribute. In order to take those implicit degrees into account, we propose to perform a closure of the discrete fuzzy set as defined in [1]. Intuitively, the closure propagates the degrees to more specific values of the hierarchy. Let us notice that the closure of a discrete fuzzy set is unnecessary if its definition domain is not hierarchized. The defuzzification of a discrete fuzzy set consists then in deleting the degrees associated with each of its elements.

The defuzzification of a continuous fuzzy set consists in only keeping the interval which corresponds to the support of the fuzzy set.

Example 4 *Let us consider the MIEL++ query Q of example 3. The closure of the discrete fuzzy set* $MicroPreferences$ *according to the type hierarchy of the symbolic type Microorganism is* {*1.0/Gram+, 0.5/Gram-, 1.0/Clostridium botulinum, 1.0/ Staphylococcus Spp., 0.5/Salmonella*} *and its defuzzification is* {*Gram+, Gram- , Clostridium botulinum, Staphylococcus Spp., Salmonella*}*. The defuzzification of the continuous fuzzy set* $awPreferences$ *is* $[0.9, 0.99]$.

Translation of a MIEL++ query into a SPARQL query The defuzzified MIEL++ query can now be translated into a SPARQL query composed of a CONSTRUCT clause and a WHERE clause. The CONSTRUCT clause allows the graph answers of the SPARQL query to be built according to the projection and selection attributes of the MIEL++ query. The WHERE clause contains, in its FILTER clause, the selection criteria and the threshold of the MIEL++ query.

The CONSTRUCT clause of the SPARQL query is automatically generated (i) from the definition of the relation represented by the view and associated with the MIEL++ query and (ii) from the sets of projection and selection attributes of the MIEL++ query and their associated RDF graph pattern representing the fuzzy annotation generated by the annotation process (see figure 3).

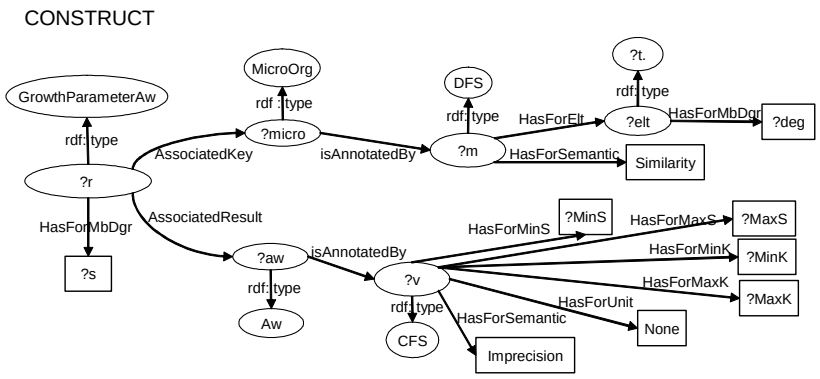
The WHERE clause of the SPARQL query contains the RDF graph already generated for its CONSTRUCT clause and three FILTER clauses which allows one to test the satisfaction of the threshold and the selection criteria of the MIEL++ query:

`Filter clause 1` The first filter clause tests the satisfaction of the threshold of the MIEL++ query: the pertinence score of the relation represented by a potential answer RDF graph must be greater than the threshold of the MIEL++ query.

`Filter clause 2` The second filter clause tests the satisfaction of each symbolic selection criterion of the MIEL++ query: at least one of the elements of the discrete fuzzy set present in a potential answer RDF graph must be equal to at least one of the elements of the defuzzified selection criterion.

`Filter clause 3` The third filter clause tests the satisfaction of each numeric selection criterion of the MIEL++ query: let $MinSP$ (resp. $MinSD$) and $MaxSP$ (resp. $MaxSD$) be respectively the lower and the upper bounds of the defuzzified selection criterion (resp. of the support of the imprecise datum present in a potential answer RDF graph). The condition $((MinSP < MinSD)$ and $(MinSD < MaxSP))$ or $((MinSP < MaxSD)$ and $(MaxSD < MaxSP))$ or $((MinSP < MinSD)$ and $(MaxSD < MaxSP))$ ensures that there is an overlap between the fuzzy set expressing preferences and the fuzzy set representing an imprecise datum and thus that the defuzzified selection criterion is satisfied.

Example 5 *The defuzzified MIEL++ query of example 4 can be translated into the SPARQL query of Figure 4.*



WHERE

<idem Construct>

FILTER (xsd:float (?s) >= 0.5)

FILTER (?t = 'Gram+' || ?t = 'Gram-' || ?t = 'Clostridium_botulinum' || ?t = 'Staphylococcus_Spp.' || ?t = 'Salmonella')

FILTER ((xsd:float (?minS) >= 0.9 && xsd:float (?minS) <= 0.99) || (xsd:float (?maxS) >= 0.9 && xsd:float (?maxS) <= 0.99 || (xsd:float (?maxS) <= 0.99 && xsd:float (?minS) >= 0.9))

Fig. 4. SPARQL query associated with the MIEL++ query

3.3 Construction of the MIEL++ answer

An answer to a MIEL++ query must (1) satisfy the minimal acceptable pertinence score associated with the query; (2) satisfy all the selection criteria of the query and (3) associate a constant value with each projection attribute of the query. An answer to a MIEL++ query in the XML/RDF subsystem is computed in three steps. First, the corresponding SPARQL query is generated and

executed into the XML/RDF data warehouse. Then, the values associated with the selection attributes in each answer graph are extracted in order to measure how the answer graph satisfies the selection criteria. Finally, the values associated with the projection attributes in each answer graph are extracted to be retrieved to the user. Let us notice that the values extraction from an answer graph is performed through SPARQL queries which are defined for each selection and projection attributes of the MIEL++ query.

To measure the satisfaction of a selection criterion, we have to consider the two semantics -imprecision and similarity- associated with fuzzy values of the XML/RDF data warehouse. On the one hand, two classical measures ([7]) have been proposed to compare a fuzzy set representing preferences to a fuzzy set having a semantic of imprecision: a possibility degree of matching denoted Π and a necessity degree of matching denoted N. On the other hand, we propose to use the adequation degree as proposed in [8] to compare a fuzzy set representing preferences to a fuzzy set having a semantic of similarity.

Definition 4 Let $(a \approx v)$ be a selection attribute of the MIEL++ query Q, v' a value of the attribute a stored in the XML/RDF data warehouse, $sem_{v'}$ the semantic of v', μ_v and $\mu_{v'}$ being their respective membership functions defined on the domain Dom and cl the function which corresponds to the fuzzy set closure. The comparison result depends on the semantic of the fuzzy set: If $sem_{v'} = imprecision$, the comparison result is given by the **possibility degree of matching** between v and v' noted $\Pi(v, v') = sup_{x \in Dom}(min(\mu_v(x),\ \mu_{v'}(x))$ and the **necessity degree of matching** between v and v' noted $N(v, v') = inf_{x \in Dom}(max(\mu_v(x), 1 - \mu_{v'}(x))$. If $sem_{v'} = similarity$, the comparison result is given by the **adequation degree** between $cl(v)$ and $cl(v')$ noted $ad(cl(v),\ cl(v')) = sup_{x \in Dom}\ (min(\mu_{cl(v)}(x),\ \mu_{cl(v')}(x)))$.

The comparison results of fuzzy sets having the same semantic (similarity or imprecision) are aggregated using the min operator (which is classically used to interpret the conjunction). Therefore, an answer is a set of tuples composed of the pertinence score ps associated with the relation, three comparison scores associated with the selection criteria in the data warehouse: a global adequation score ad_g associated with the comparison results having a semantic of similarity and two global matching scores Π_g and N_g associated with the comparison results having a semantic of imprecision, and, the values associated with each projection attribute. Based on those scores, we propose to define a total order on the answers which gives greater importance to the most pertinent answers compared with the ontology. Thus, the answers are successively sorted according to firstly ps, then ad_g and thirdly a total order defined on N_g and Π_g, N_g being considered as of greater importance than Π_g.

Example 6 *The answer to the SPARQL query of Figure 4 compared with the Web data table presented in Figure 2 of which the first row is annotated in Figure 3 is given below:*
{ {$ps_r = 1$, $ad_g = 0.5$, $N_g = 1$, $\Pi_g = 1$, *Microorg=(0.5/Clostridium Perfringens+0.5/Clostridium Botulinum), aw=[0.943, 0.95, 0.96, 0.97]*},

{ $ps_r = 1$, $ad_g = 0.5$, $N_g = 0.5$, $\Pi_g = 0.68$, *Microorg=(0.5/Staphylococcus spp.+0.5/Staphylococcus aureus), aw=[0.88, 0.98, 0.98, 0.99]*},
{ $ps_r = 1$, $ad_g = 0.5$, $N_g = 0$, $\Pi_g = 0.965$, *Microorg=(1.0/Salmonella), aw=[0.94, 0.99, 0.99, 0.991]*} }.

4 Experimental results

As the quality of the querying depends mainly on the quality of the annotation process, we firstly present experimental results about the recognition of (i) relations of the ontology in Web tables, (ii) fuzzy annotations associated with numerical columns, (iii) fuzzy annotations associated with symbolic columns. Secondly, we present some preliminary experimental results about the querying of the RDF database.

The fuzzy annotation step presented in section 2, more precisely the recognition of relations in the Web tables, has been validated experimentally on three different applications: microbial risk in food, chemical risk in food and aeronautics. The number of relations defined in the associated ontologies are: 16 for microbial risk, 4 for chemical risk and 26 for aeronautics. Three corpora of Web tables associated with the three applications have been manually annotated using the relations of the corresponding ontologies. We compared the results generated automatically with our annotation process with the ones obtained manually. The precision (resp. recall) obtained are 80% (resp. 97%) for microbial risk, 93% (resp. 79%) for chemical risk and 98% (resp. 88%) for aeronautics.

The fuzzy annotations associated with numeric values, which may be imprecise, were manually checked on 119 relations instanciated in 60 tables of the microbial risk application. For 100 relations, all numeric values were correctly annotated. In the majority of the remaining errors (13 on 19), the numeric type $Temperature$ was not instanciated because its value was not in the table but in the paragraphs surrounding the table in the scientific documents.

Concerning fuzzy annotations associated with symbolic types, we have compared the fuzzy annotation of 185 instances of food products (microbial risk application) extracted from Web tables with the "best match" manually defined in the ontology. For 78% of the 185 terms from the Web, their "best match" is not null in the computed annotation. 46% of the Web terms had their "best match" in first position in the computed annotation, while 66% had their "best match" among the five best positions. This validates the approach of keeping a fuzzy set for instanciating the symbolic types, instead of keeping only the term in the taxonomy having the best term similarity with the Web term.

In preliminary tests performed on a RDF base, composed of more than 22000 RDF triples (312 graphs) associated with the microbial risk application, we have evaluated 5 queries (see Figure 5) covering at least 50% of the database entries. We obtain better results in the queries where the selection criterium concerns microorganisms than in the ones concerning food products. This is due to the fact that microorganism names are more standardized in Web tables than

food product names. Therefore, the quality of the fuzzy annotations associated with the microorganism symbolic type is better than for the food product type. Nevertheless, we obtain a precision of 100% for the two last queries concerning food product if we put a threshold of 0.7 on the term similarity degrees.

Queried relation	Selection criteria	Precision-recall	Number of answer graphs
Lag Time	Microorganism=L. Monocytogenes	100%-100%	47 graphs
Lag Time	Microorganism=P. Fluorescens	100%-100%	29 graphs
Growth kinetics	Microorganism=E. Coli	100%-100%	39 graphs
Lag Time	FoodProduct= Egg salad	50%-100%	24 graphs
Growth kinetics	FoodProduct= Salad	54%-100%	26 graphs

Fig. 5. Evaluation of query results

5 Related works

Our proposal in this paper can be compared to papers studying flexible querying extending XPATH or SPARQL. Different approaches have been proposed. [9] proposes FUZZYXPATH, a fuzzy extension of XPATH to query XML documents. Extensions are of two kinds : (i) the deep-similar function permits a relaxed comparison in term of structure between the query tree and the data tree; (ii) the close and similar predicates extend the equality comparison to a similarity comparison between the content of a node and a given value expressed in the query. [10] proposes an extension of the SPARQL Optional clause (called Relax). This clause permits to compute a set of generalizations of the RDF triplets involved in the SPARQL query using especially declarations done in the RDF Schema. [11] also proposes the same kind of extension of the SPARQL query using a distance function applied to the classes and properties of the RDF Schema. The originality of our approach in flexible SPARQL querying is that we propose a complete and integrated solution which permits (1) to annotate Web data tables with the vocabulary defined in an OWL ontology, (2) to perform a flexible querying of the annotated tables using the same vocabulary and taking into account the fuzzy degrees generated by the annotation system according to their associated semantic. Our work did not use the fuzzy extension of SPARQL based on a fuzzy extension of DL-Lite proposed by [12] for two main reasons: (i) our OWL ontology requires a higher level of expressivity (OWL-DL) which is useful for consistency checking (by example, in order to express that the class NumericalAttribute is distinct from SymbolicAttribute); (ii) the SPARQL extension does not yet permit to make the distinction between fuzzy sets having a semantic of similarity and imprecision.

6 Conclusion and perspectives

We have presented in this paper a complete workflow, called @WEB, realized using the recommendations of the W3C, which permits thanks to a domain on-

tology expressed in OWL: (1) to annotate Web data tables with fuzzy RDF descriptions; (2) to perform a flexible SPARQL querying of the annotated Web data tables. To the best of our knowledge, @Web is the only software which permits simultaneously (1) to annotate accurately a Web data table with a domain ontology; (2) to perform approximate reasoning in the flexible querying step comparing preferences expressed by the end-user with fuzzy annotations. @WEB has been successfully tested on three different applications (microbial risk in food, chemical risk in food and aeronautics) which illustrate the generic potential of the proposal. In this paper, we have presented in detail the flexible querying system of @WEB. Thanks to the defuzzification pre-treatment proposed in this paper, the implementation of this SPARQL flexible querying system may be done easily reusing any implementation of a standard SPARQL engine (JENA for the current one). In the very next future, we want to explore two new ideas. The first one consists in enhancing the annotation process using machine learning techniques on the knowledge of the ontology but without manual training on a subset of the corpus. The second idea consists in studying the way the work of [12] could be extended in order to support the level of expressivity of our OWL ontology and the comparison between fuzzy values having different semantics.

References

1. Thomopoulos, R., Buche, P., Haemmerlé, O.: Fuzzy sets defined on a hierarchical domain. IEEE Transactions on Knowledge and Data Engineering **18**(10) (2006) 1397–1410
2. Hignette, G., Buche, P., Dibie-Barthélemy, J.: Fuzzy annotation of web data tables using a domain ontology. In: ESWC 2009. Volume 5554 of LNCS. (2009) 638–653
3. Noy, N., Rector, A., Hayes, P., Welty, C.: Defining n-ary relations on the semantic web. W3C working group note http://www.w3.org/TR/swbp-n-aryRelations.
4. Zadeh, L.: Fuzzy sets. Information and control **8** (1965) 338–353
5. Dubois, D., Prade, H.: The three semantics of fuzzy sets. Fuzzy Sets and Systems **90**(2) (1997) 141–150
6. Buche, P., Dervin, C., Haemmerlé, O., Thomopoulos, R.: Fuzzy querying of incomplete, imprecise, and heterogeneously structured data in the relational model using ontologies and rules. IEEE T. Fuzzy Systems **13**(3) (2005) 373–383
7. Dubois, D., Prade, H. In: Possibility theory- An approach to computerized processing of uncertainty. Plenum Press, New York (1988)
8. Baziz, M., Boughanem, M., Prade, H., Pasi, G. In: A fuzzy logic approach to information retrieval using a ontology-based representation of documents. in Fuzzy logic and the Semantic Web, Elsevier (2006) 363–377
9. Campi, A., Damiani, E., Guinea, S., Marrara, S., Pasi, G., Spoletini, P.: A fuzzy extension for the xpath query language. In: FQAS. LNCS 4027 (2006) 210–221
10. Hutardo, C.A., Poulovassilis, A., Wood, P.T.: A relaxed approach to rdf querying. In: ISWC. Volume 4273 of LNCS. (2006) 314–328
11. Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., Gandon, F.: Searching the semantic web: Approximate query processing based on ontologies. IEEE Intelligent Systems Journal **21**(1) (2006) 20–27
12. Pan, J.Z., Stamou, G.B., Stoilos, G., Taylor, S., Thomas, E.: Scalable querying services over fuzzy ontologies. In: WWW 2008. 575–584