

An Ontology-Based Method for Duplicate Detection in Web Data Tables

Patrice Buche, Juliette Dibie-Barthelemy, Rania Khefifi, Fatiha Saïs

▶ **To cite this version:**

Patrice Buche, Juliette Dibie-Barthelemy, Rania Khefifi, Fatiha Saïs. An Ontology-Based Method for Duplicate Detection in Web Data Tables. DEXA 2011 - 22nd International Conference on Database and Expert Systems Applications, Aug 2011, Toulouse, France. pp.511-525, 10.1007/978-3-642-23088-2_38 . lirmm-00611944

HAL Id: lirmm-00611944

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00611944

Submitted on 3 Jun 2020

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.

An Ontology-based method for Duplicate Detection in Web Data Tables

Patrice Buche(1), Juliette Dibie-Barthélemy(2),
Rania Khefifi(3), and Fatiha Saïs(3)

(1) INRA - UMR IATE, 2, place Pierre Viala, F-34060 Montpellier Cedex 2, France
LIRMM, CNRS-UM2, F-34392 Montpellier, France
Patrice.Buche@supagro.inra.fr

(2) INRA - Mét@risk & AgroParisTech, 16 rue Claude Bernard, F-75231 Paris Cedex 5, France
Juliette.Dibie@agroparistech.fr

(3) LRI (CNRS & Paris-Sud XI University)/INRIA Saclay,
4 rue Jacques Monod, bât. G, F-91893 Orsay Cedex, France
{Rania.Khefifi, Fatiha.Sais}@lri.fr

Abstract. We present, in this paper, a duplicate detection method in semantically annotated Web data tables, driven by a domain Termino-Ontological Resource (TOR). Our method relies on the fuzzy semantic annotations automatically associated with the Web data tables. A fuzzy semantic annotation is automatically associated with each row of a Web data table. It corresponds to the instantiation of a composed concept of the domain TOR, which represents the semantic n-ary relationship that exists between the columns of the Web data table. A fuzzy semantic annotation contains fuzzy values expressed as fuzzy sets. We propose an automatic duplicate detection method which consists in detecting the pairs of duplicate fuzzy semantic annotations and relies on (i) knowledge declared in the TOR and on (ii) similarity measures between fuzzy sets. Two new similarity measures are defined to compare both, the symbolic fuzzy values and the numerical fuzzy values. Our method has been tested on a real application in the domain of chemical risk in food.

1 Introduction

Today's Web is not only a set of semi-structured documents interconnected via hyper-links. A huge amount of scientific and technical documents, available on the Web or on the hidden Web (digital libraries, ...), include structured data represented in 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 building a data warehouse dedicated to a given application domain. They can be used to enrich local data sources or to compare local data with external ones. In order to integrate data, a preliminary step consists in harmonizing the vocabulary of the external data with the vocabulary of the local data, which is represented by a

domain ontology. Therefore, external and local data can be indexed and queried using the same vocabulary. In [1], Hignette and al. have developed an automatic and ontology-based method for semantic annotation of Web data tables. The obtained annotations are expressed thanks to the domain ontology and are fuzzy (see [2], for more details on fuzzy sets). Fuzzy annotations may have two different semantics: they represent either data imprecision or similarities between terms of data tables and terms of the ontology.

The semantic annotation allows the integration of Web external data with local ones, solving the vocabulary heterogeneity problem, but it does not prevent the integration of duplicate data into the data warehouse. The presence of duplicates in the data warehouse impacts the data quality and therefore the results of their exploitation (for instance, data analysis and decision aid). We propose in this paper to study the duplicate detection problem in Web data tables, using the fuzzy semantic annotations associated with the data tables thanks to a domain ontology. We propose an automatic method of duplicate detection which relies on (i) knowledge declared in the domain ontology, as it is done in [3], and on (ii) similarity measures between fuzzy sets.

The result of this work has been integrated in the @Web system which was previously developed (see [1]). @Web system is based on the semantic Web framework[1] and language recommendations (XML, RDF, OWL), which allow an XML/RDF data warehouse to be supplemented with Web data tables, as presented in Figure 1. @Web system relies on a domain Termino-Ontological Resource (TOR) manually built by domain experts.

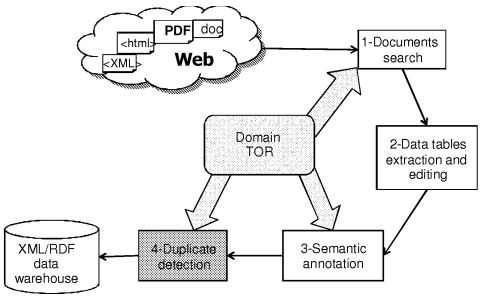


Fig. 1. Main steps of the ONDINE system.

We will present in this paper how @Web system can be extended with a new duplicate detection step using the fuzzy semantic annotations associated with the Web data tables. We suppose that the Web data tables have been previously automatically annotated thanks to the annotation method described in [1]. In section 2, we briefly present the domain TOR, we recall the semantic annotation method of @Web system (see [1]), and we recall the reference reconciliation

[1] http://www.w3.org/standards/semanticweb/

method (N2R) of [3] on which relies our work. In section 3, we present our duplicate detection method in Web data tables driven by a domain TOR. In section 4, we present some experiment results obtained on real data of chemical risk in food domain. We conclude and present some future work in section 5.

2 Preliminaries

In subsection 2.1, we present the domain TOR. In subsection 2.2, we recall the semantic annotation method of the @Web system presented in [1]. Finally, in subsection 2.3, we recall the numerical reference reconciliation method N2R of [3] on which relies our work.

2.1 The domain Termino-Ontological Resource

A Termino-Ontological Resource (TOR) [4, 5] is composed of a conceptual component and a terminological component. The conceptual component represents the ontology of the TOR. It is composed of two main parts: a generic part, commonly called *core ontology*, which contains the structuring concepts of the data table integration task, and a specific part, commonly called *domain ontology*, which contains the concepts that are specific to the domain of interest. The core ontology is composed of three kinds of *generic concepts*:

1. *simple concepts* which contain the symbolic concepts and the numerical concepts. Symbolic concepts are hierarchically organized by the "is-a" relationship. A numerical concept is described by a set of units, which are sub concepts of the *unit concept*, and eventually a numerical interval;
2. *unit concepts* which contain the units used to characterize the numerical concepts;
3. *composed concepts* which allow n-ary relationships to be represented between simple concepts. A composed concept is described by a signature, which is defined by a domain and a range. The domain contains one or several simple concepts, called *access concepts*, while the range contains only one simple concept, called *result concept*. A composed concept is denoted by $CC(Aa_1, Aa_2, \ldots, Aa_n, Ar)$ where CC is the name of the composed concept and $(Aa_1, Aa_2, \ldots, Aa_n, Ar)$ represents its signature: $Aa_1, Aa_2, \ldots, Aa_n$ are the access concepts of CC and Ar its result concept. The simple concepts which belong to the signature of a composed concept can be declared as important or simply optional using FOL Horn rules.

The concepts belonging to the domain ontology, called *specific concepts*, appear in the domain TOR as sub concepts of the generic concepts.

In the domain TOR, all concepts are represented by OWL classes. The Horn rules are expressed using SWRL rules (Semantic Web Rule Language) recommended by the W3C[2]. The disjunction constraints, which can be declared between simple concepts and/or composed concepts, are expressed using OWL

[2] http://www.w3.org/Submission/SWRL/

constructor *owl:disjointWith*. Figure 2 gives an example of the conceptual component of a TOR in the domain of chemical risk in food. The concepts belonging to the core ontology are represented in bold.

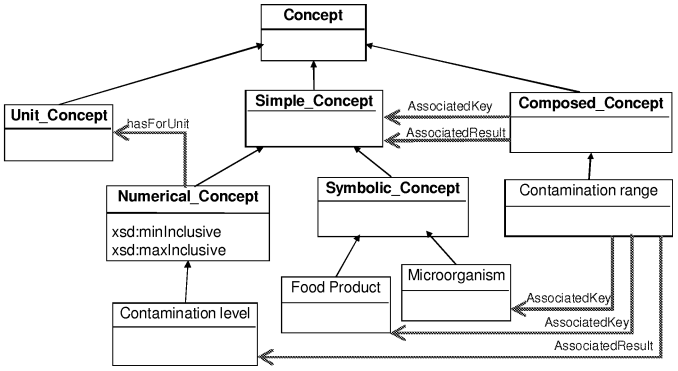


Fig. 2. Conceptual component of a TOR in the domain of chemical risk in food.

The terminological component is the terminology of the TOR: it contains the term set of the domain of interest. A term is defined as a sequence of words, in a language, and has a label.

2.2 Semantic annotation of Web data tables driven by a domain TOR

A data table is composed of columns, themselves composed of cells. The cells of a data table may contain terms or numerical values often followed by a measure unit. The semantic annotation of a Web data table consists in annotating cells content, in order to identify the symbolic or numerical concepts represented by its columns and finally the semantic n-ary relationships between its columns.

Food	*Contaminant*	*Max Value (µg/kg)*	*Contamination Level (µg/kg)*
Breakfast cereals	Ochratoxin A	6	<0.2
Baby food	Patulin	58	6.3

Table 1. Example of a Web data table

Example 1 *Table 1 presents an example of a Web data table in which the composed concept* Contamination Range *was identified. The first line of the Web data table indicates that* Breakfast cereals *is contaminated by the* Ochratoxin A *at a* contamination level *smaller than 0.2 µg/kg.*

Several composed concepts of a domain TOR can be recognized to annotate a Web data table. The semantic annotation of a Web data table consists in instantiating each recognized composed concept for each row of the table. A composed concept instantiation associated with a row of a Web data table include values expressed as fuzzy sets [2]. In a fuzzy set $\mathcal{A}$ defined on a domain $\mathcal{X}$, each element $x \in \mathcal{X}$ can belong partially to the fuzzy set with a membership degree, denoted $\mu_{\mathcal{A}}(x)$, between 0 (element which is not part of the fuzzy set) and 1 (element which is completely part of the fuzzy set). The definition domain $\mathcal{X}$ can be continuous or discrete. The support $S(\mathcal{A})$ and the kernel $K(\mathcal{A})$ of the fuzzy set $\mathcal{A}$ are the sets: $S(A) = \{x \in \mathcal{A} | \mu_{\mathcal{A}}(x) > 0\}$ and $K(\mathcal{A}) = \{x \in \mathcal{A} | \mu_{\mathcal{A}}(x) = 1\}$. The fuzzy values, found in the composed concept instantiations, may express two of the three classical semantics of fuzzy sets [6]: similarity or imprecision. A discrete fuzzy set with a semantics of similarity is associated with each cell belonging to a column recognized as symbolic. It represents the ordered list of the most similar terms of the domain TOR associated with the original term present in the cell. A continuous fuzzy set with a semantics of imprecision may be associated with cells belonging to columns recognized as numerical ones. It represents an ordered disjunction of exclusive possible values.

Definition 1 A **discrete fuzzy set** $\mathcal{A}$, denoted DFS, is a fuzzy set associated with a symbolic concept of the domain TOR. Its definition domain is the set of terms of the domain TOR. We denote by $\{x_1/y_1, \ldots, x_n/y_n\}$ the fact that element x_k has membership degree y_k.

Definition 2 A **continuous fuzzy set** $\mathcal{A}$, denoted CFS, is a trapezoidal fuzzy set associated with a numerical concept of the domain TOR. A trapezoidal fuzzy set $\mathcal{A}$ is defined by its four (ordered) characteristic points $[a, b, c, d]$ which correspond to its support $[a, d]$ and its kernel $[b, c]$ (see Figure 3). Its definition domain is the interval of possible values for the numerical concept.

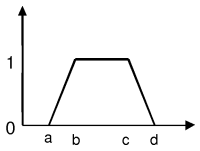


Fig. 3. A trapezoidal continuous fuzzy set

Example 2 *The discrete fuzzy set associated with the term* "Breakfast cereal" *of the first row of Table 1 is:* { *breakfast cereal sweet/0.602, breakfast cake/0.5, cereal bar chocolat/0.408, cereal bar/0.5, cereal bar low calorie/0.354* }. *The continuous fuzzy set associated with the numerical value* "< 0.2" *of the first row of Table 1 is: [0, 0, 0.2, 0.2].*

2.3 Reference Reconciliation method (N2R)

To develop a duplicate detection method we have chosen to rely on reference reconciliation methods which are automatic and ontology based, in order to benefit from the knowledge which is declared in the domain TOR. N2R method, developed by Saïs and al [3], is a method which has two main distinguishing characteristics. Firstly, it is fully unsupervised, i.e., it does not require any training phase from manually labeled data to set up coefficients or parameters. Secondly, two functions modeling the influence between similarities of references take into account the constraints associated with the functional properties declared in the OWL ontology in a declarative way. Furthermore, ontology and data knowledge (disjunctions and Unique Name Assumption) are exploited by N2R in a filtering step to reduce the number of reference pairs which are considered in the similarity computation step.

The duplicate detection method, present in the following, relies on N2R method in the sense that it exploits knowledge declared in the domain TOR to both, (i) filter the pairs of data to be compared, thanks to disjunctions declared in the domain TOR, and (ii) express the influence degrees existing between the different similarities, thanks to the declaration of concept importance.

3 Duplicate Detection method

We present in this section our duplicate detection method. Our method takes as input two Web data tables which were previously automatically semantically annotated thanks to a domain TOR using the method of [1]. Since each data table is annotated by a set of composed concept instances, our method consists in detecting the pairs of duplicate composed concept instances by comparing them two by two. We first present the definitions of simple concept instances and composed concept instances. Since the composed concept instances contain fuzzy values, we then propose two new similarity measures to compare, on the one hand, the discrete fuzzy sets and, on the other hand, the continuous fuzzy sets. We finally present the algorithms of our method and an illustrative example.

3.1 Definitions of simple and composed concept instances

The input of our method is a set of composed concept instances associated with each Web data table to be compared. A composed concept instance is composed of the instances of the simple concepts which belong to its signature.

Definition 3 *A* simple concept instance*, denoted* $inst_{c_i}$ *where* c_i *is a simple concept* ($c_i = SimpleConcept(inst_{c_i})$)*, can be represented by either:*

- *a discrete fuzzy set having a semantics of similarity which is composed of a set of terms* t_k *of the domain TOR with their membership degrees* d_k*:* $inst_{c_i} = (c_i, \{ t_1/d_1, \ldots, t_n/d_n \})$;

– *or a continuous trapezoid fuzzy set having a semantics of imprecision which is described by its support* $[sup_{min}, sup_{max}]$ *and its kernel* $[ker_{min}, ker_{max}]$*:* $inst_{c_i} = (c_i, [sup_{min}, ker_{min}, ker_{max}, sup_{max}])$.

We can therefore give the definition of a composed concept instance.

Definition 4 *A* composed concept instance*, denoted ICC, is a couple (id,* $descr_{id}$*) where id is the ID associated with the composed concept CC and* $descr_{id}$ *its description. The description of a composed concept is the set of the instances of the simple concepts which belong to its signature:* $descr_{id} = \{ (c_1, inst_{c_1}), \ldots, (c_n, inst_{c_n}) \}$*. We denoted id.inst the set of the simple concept instances:* $id.inst = \{inst_{c_1}, \ldots, inst_{c_n}\}$

Example 3 *The description of the composed concept instance associated with the first row of Table 1 is: { (Food Product, { breakfast cereal sweet/0.602, breakfast cake/0.5, cereal bar chocolat/0.408, cereal bar/0.5, cereal bar low calorie/0.354 }), (Contaminant, { Ochratoxin A/1 }), (Contamination level, [0, 0, 0.2, 0.2]) }.*

3.2 Two similarity measures to compare fuzzy sets

In this section, we propose two similarity measures to compare, on the one hand, discrete fuzzy sets, and on the other hand, continuous fuzzy sets. [7] proposed a classification of comparison measures between fuzzy objects into four categories: satisfiability, inclusion, resemblance and dissimilarity. In this paper, we are looking for a measure of resemblance, which is a measure of similitude between two fuzzy sets looking at the characteristics they have in common, without regarding one of them as a reference. In [7], this family of measures satisfies the two properties of reflexivity and symmetry, which can be easily checked for the two measures we propose in the following.

A similarity measure to compare discrete fuzzy sets There exist several similarity measures between sets of terms (see [8]). We can cite, in particular, the measure of Jaccard [9], the measure of Tversky [10] or the measure of SoftJaccard [11], which allow the comparison between sets of terms. The measure we have to choose must take into account the fact that the discrete fuzzy sets we want to compare are sets of terms associated with membership degrees (see definition 3). We therefore propose a new similarity measure, called *Sim*, which is inspired from the Jaccard measure. The Jaccard measure is defined as the intersection (number of common terms) divided by the union (total number of terms) of the two sets to compare. In our *Sim* measure, the number of common terms corresponds to the sum of the minimum degrees associated with the common terms of both fuzzy sets. The total number of terms corresponds to the sum of the maximum degrees associated with the terms of the fuzzy sets. These are the classical ways to represent the intersection and the union of two fuzzy sets. Let A and B be two discrete fuzzy sets, $deg_A(t)$ (respectively $deg_B(t)$) the membership

degree of the term t to the fuzzy set A (respectively B), the similarity measure *Sim* is defined as follows:

$$Sim(A,B) = \frac{\sum_{t \in A \cap B} \min(deg_A(t), deg_B(t))}{\sum_{t \in A \cup B} \max(deg_A(t), deg_B(t))} \quad (1)$$

A similarity measure to compare continuous fuzzy sets There exist several similarity measures between continuous fuzzy sets. We can cite, in particular, the measure of Hsieh and Chen [12], the measure of Chen [13] and the measure of Chen and Chen [14]. The measure we have to choose must take into account two constraints on the continuous fuzzy sets we want to compare. The first constraint to be considered is that continuous fuzzy sets are not necessarily normalized, i.e. their values are not necessarily included between 0 and 1. We can cite for instance the numerical concept pH whose values belong to [0, 14]. In the second constraint, redundancies between continuous fuzzy sets must be detected even if they represent values with a different precision scale. For instance, a table may contain the mean value of repeated experimental data whereas, in a redundant table, the value is expressed by a mean value and associated standard deviation. Since the measure of Chen [13] does not allow the comparison between not normalized continuous fuzzy sets and the measure of Hsieh and Chen [12] does not allow the comparison between continuous fuzzy sets of different precision scales, we propose to use the measure of simple center of gravity method (SCGM) of Chen and Chen [14]. This measure relies on a similarity measure between the center-of-gravity points of the fuzzy sets to compare. Let (a_1, a_2, a_3, a_4) be a continuous trapezoid fuzzy set, the coordinates x^* and y^* of the center-of-gravity points are computed by the SCGM method as follows:

$$If\ a_1 = a_4 \longrightarrow \begin{cases} y^* = 1/2 \\ x^* = a_1 \end{cases}, otherwise \longrightarrow \begin{cases} y^* = \frac{\frac{a_3 - a_2}{a_4 - a_1} + 2}{6} \\ x^* = \frac{y^*(a_3 + a_2) + (a_4 + a_1)(1 - y^*)}{2} \end{cases} \quad (2)$$

In order to compute the similarity measure between two continuous trapezoid fuzzy sets A and B, denoted $Sim(A,B)$, we propose to use the distance between their center-of-gravity points as follows:

$$\begin{aligned} &Sim(A,B) = \frac{1}{1 + d(cent_A, cent_B)} \\ &where\ d(cent_A, cent_B) = \sqrt{(x_A^* - x_B^*)^2 + (y_A^* - y_B^*)^2} \end{aligned} \quad (3)$$

3.3 The Duplicate Detection Algorithm

We now detail our duplicate detection method between two Web data tables which were semantically annotated thanks to a domain TOR. Our method consists in detecting the pairs of duplicate composed concepts instances, which are

associated with the Web data tables. To do that, we propose to compute a similarity score between the descriptions of each pair of composed concept instances. This similarity score relies on the similarity measures presented in subsection 3.2 and on knowledge declared in the domain TOR. Algorithm 1 presents the main steps of our duplicate detection method.

Algorithm 1 Duplicate detection Algorithm

Input: – $Set_1(ICC)$: set of composed concept instances associated with the first Web data table T_1
– $Set_2(ICC)$: set of composed concept instances associated with the second Web data table T_2
– $Disj$: set of disjunction constraints between the concepts of the domain TOR
– Tax : hierarchical relationships between simple concepts in the domain TOR
– $ImportantSimpleConcepts$: set of the signatures of the composed concepts in the domain TOR with their important simple concepts
– T_{dup}: predefined threshold of the duplicate decision
Output: – set of duplicate pairs of composed concept instances
{1: *building of the set of pairs of comparable composed concept instances*}
$S \leftarrow comparableICCPairs(Set_1(ICC), Set_2(ICC), Disj)$
$DUP \leftarrow \emptyset$
{*2: computation of the similarity score*}
For Each $(icc_1, icc_2) \in S$ ***Do***
$score \leftarrow SimilarityScore((icc_1, icc_2), Disj, Tax, ImportantSimpleConcepts)$
{*3: duplicate decision*}
If $score > T_{dup}$ ***Then***
$DUP \leftarrow DUP \cup (icc_1, icc_2)$
EndIf
End Each
return DUP

Algorithm 1 requires three kinds of inputs. Let T_1 and T_2 be two Web data tables semantically annotated thanks to a domain TOR. The first input is the two sets of composed concept instances $Set_1(ICC)$ and $Set_2(ICC)$ which are respectively associated with the Web data tables T_1 and T_2. The second kind of input corresponds to the knowledge declared in the domain TOR: (1) the disjunctions between composed concepts and the disjunctions between simple concepts, which allows one to avoid some obvious comparisons between composed concept instances and between simple concept instances, (2) the hierarchical relationships between simple concepts represented by a taxonomy, (3) the importance of the simple concepts in the signatures of the composed concepts. The third kind of input is a predefined threshold used to determine if two composed concept instances are duplicate or not according to their similarity score. Algorithm 1 has for output the set of duplicate pairs of composed concept instances. The first step of Algorithm 1 consists in building the set of pairs of comparable composed concepts instances using the disjunction constraints defined in the domain TOR.

Two composed concept instances icc_1 and icc_2 are said *comparable* if the composed concepts cc_1 and cc_2 are not declared as disjoints in the TOR. A similarity score is then computed for each pair of comparable composed concept instances (step 2). The computation of this score is detailed in Algorithm 2 presented below. Finally, two composed concept instances are said redundant if the similarity score between their descriptions is greater than a given threshold (step 3).

Algorithm 2 gives details on the similarity score computation for one pair of comparable composed concept instances. This score is computed thanks to the similarity measures, presented in subsection 3.2, between each pair of comparable simple concept instances, which belong to the signatures of the composed concept instances. Two simple concept instances are said *comparable* if the corresponding simple concepts are not declared as disjoint in the domain TOR. In the first step of Algorithm 2, a similarity score is computed for each simple concept instance a, which belongs to the composed concept instance icc_1 ($a \in id_1.inst$), with each simple concept instance b, which belongs to the composed concept instance icc_2 ($b \in id_2.inst$). This score is a combination of:

1. a semantic similarity score $score_{sem}$ between the simple concepts associated with a and b, which relies on the notion of lowest common subsumer (LCS) in the hierarchy of simple concepts in the domain TOR;
2. an instance score $score_{inst}$ which is computed thanks to the similarity measures $Sim(a, b)$, presented in subsection 3.2, depending upon the simple concepts associated with a and b are symbolic or numerical.

For each simple concept instance $a \in id_1.inst$, we keep the best similarity score with the simple concept instances $b \in id_2.inst$. We can therefore compute the similarity score of the pair of comparable composed concept instances (icc_1, icc_2) (step 2). This similarity score is computed thanks to the importance of the simple concepts in the signatures of the composed concepts associated with icc_1 and icc_2, defined in the domain TOR. It is a combination of (i) a similarity score f_{imp} for the instances of the simple concepts which are declared as important, computed as the product of the similarity scores of their pairs of instances and (ii) a similarity score f_{Nimp} for the instances of the simple concepts which are not declared as important, computed as the average value of the similarity scores of their pairs of instances.

3.4 An illustrative example of our duplicate detection method

To illustrate our method, let us consider Table 1 presented in subsection 2.2 and Table 2 presented below.

The identified composed concepts in Table 1 are the following:

- ContaminationRange (**Food**, **Contaminant**, year, **ContaminationLevel**)
- LodRelation (**Food**, **Contaminant**, year, **SamplesTotalNumber**, **lod**)

The identified composed concepts in Table 2 are the following:

- ContaminationRange (**Food**, **Contaminant**, year,**ContaminationLevel**)

Algorithm 2 Computation of the similarity score for a pair of comparable composed concept instances

Input: – (icc_1, icc_2) : pair of composed concept instances
– $Disj$: set of disjunction constraints between the concepts of the domain TOR
– Tax : hierarchical relationships between simple concepts in the domain TOR
– $ImportantSimpleConcepts$: set of the signatures of the composed concepts in the domain TOR with their important simple concepts
Output: similarity score of the pair of comparable composed concept instances (icc_1, icc_2)

$f_{imp} \leftarrow 1$
$Score_{Nimp} \leftarrow 0$
{*1: Computation of the similarity scores between each pair of comparable simple concept instances*}
For Each $a \in id_1.inst$ ***Do***
  $best \leftarrow \emptyset$
  For Each $b \in id_2.inst$ ***Do***
    $score_{max}(a,b) \leftarrow 0$
    If *ComparableSimpleConcepts(a, b, Disj)* ***Then***
      $score_{Inst}(a,b) = Sim(a,b)$
      If $SimpleConcept(a) \neq SimpleConcept(b)$ ***Then***
        $score_{Sem}(a,b) = \frac{1}{1+LCS(SimpleConcept(a),SimpleConcept(b),Tax)}$
        $score_{final}(a,b) = \frac{score_{Sem}(a,b)+score_{Inst}(a,b)}{2}$
      Else
        $score_{final}(a,b) = score_{Inst}(a,b)$
      EndIf
      If $score_{final}(a,b) > score_{max}(a,b)$ ***Then***
        $best \leftarrow b$
        $score_{max}(a,best) \leftarrow score_{final}(a,b)$
      EndIf
    EndIf
  End Each
  {*2: Computation of the similarity score of the pair of comparable composed concept instances* (icc_1, icc_2)}
  If *(Is_Important(a, ImportantSimpleConcepts) and Is_Important(best, ImportantSimpleConcepts))* ***Then***
    $f_{imp} = f_{imp} \times score_{max}(a,best)$
  Else
    $Score_{Nimp} = Score_{Nimp} + score_{max}(a,best)$
  EndIf
End Each
$f_{Nimp} = \frac{Score_{Nimp}}{\max(|id_1.inst|,|id_2.inst|)}$
$S = \max(f_{imp}, f_{Nimp}))$
return S

– MaxContaminationRange (**Food**, **Contaminant**, year, **MaxContamination-tionlevel**)

Food	*Contaminant*	*Year*	*Lod*	*Contamination Level*
Baby food	Patulin	2000	0.7	6.3
Apple juice	Patulin	1998	2	8.37
Breafast cereal	Ochratoxin A	2003	0.7	<0.2

Table 2. Example of a Web data table (T2)

The simple concepts in bold represent the important simple concepts of the signature of the composed concepts. We suppose that (i) the composed concepts are declared as pairwise disjoint in the TOR except the composed concepts *ContaminationRange* and *MaxContaminationRange* and (ii) the simple concepts are declared as pairwise disjoint in the TOR except the simple concepts *ContaminationLevel* and *MaxContaminationLevel*. In the following, the composed concept instances and the simple concept instances are denoted by the number of their table and the number of their row. For instance, ICC_{n_T,n_L} corresponds to the instance of the composed concept CC in Row n_L of Table n_T.

We first identify the pairs of comparable composed concept instances according to the disjunction constraints defined in the domain TOR. For simplicity reason, we only consider in the following the pair ($ContaminationRange_{1,3}$, $ContaminationRange_{2,1}$). The descriptions associated with the two composed concept instances are:
$descr_{1,3}$={ ($FoodProduct_{1,3}$, { "breakfast cereal sweet"/0.408, "cereal bar chocolat"/0.408, "cereal bar"/0.5, "cereal bar low calorie"/0.354 }), ($Contaminant_{1,3}$, { "Ochratoxin" A/1 }), ($year_{1,3}$, { [2003, 2003] }), ($ContaminationLevel_{1,3}$, [0, 0, 0.2, 0.2]) }.
$descr_{2,1}$={ ($FoodProduct_{2,1}$, { "breakfast cereal sweet"/0.602, "breakfast cake"/0.5, "cereal bar chocolat"/0.408, "cereal bar"/0.5, "cereal bar low calorie"/0.354 }), ($Contaminant_{2,1}$, {"Ochratoxin A"/1 }), ($Contamination\ level_{2,1}$, [0, 0, 0.2, 0.2]) }.

We can now compute the similarity scores between each pair of comparable simple concept instances:
$score_{Inst}(FoodProduct_{1,3}, FoodProduct_{2,1}) = \frac{0.408+0.408+0.5+0.354}{0.602+0.5+0.408+0.5+0.354} = 0.7$
$score_{Inst}(Contaminant_{1,3}, Contaminant_{2,1}) = 1$
$score_{Inst}(ContaminationLevel_{1,3}, ContaminationLevel_{2,1}) = 1.$

Finally, we compute the similarity score of the pair ($ContaminationRange_{1,3}$, $ContaminationRange_{2,1}$) thanks to the importance of the simple concepts in the signature of the composed concept *ContaminationRange*:
$f_{Imp} = score_{Inst}(FoodProduct_{1,3}, FoodProduct_{2,1}) \times score_{Inst}(Contaminant_{1,3}, Contaminant_{2,1}) \times score_{Inst}(ContaminationLevel_{1,3}, ContaminationLevel_{2,1}) = 0.7 \times 1 \times 1 = 0.7$
$f_{NImp} = 0$. Then, we obtain $S = max(f_{Imp}, f_{NImp}) = 0.7$

If we set the duplicate threshold T_{dup} at 0.5, the third row of Table 2 and the first row of Table 1 are therefore duplicates with the similarity score of 0.7.

4 Experimentation

To evaluate the efficiency of our method we have applied the duplicate detection algorithm on several real Web Tables in the chemical risk in food domain. We will first give details on the dataset and then discuss the obtained results.

Dataset description. The considered data set is composed of seven Web data tables which were annotated by @Web system using the TOR of the chemical risk in food domain. In Table 3, we give the Web data table list with the set of composed concepts which were identified within them.

Tables	**Identified composed concepts**
T1	Lod, MaxContamination, MeanContaminationLevel, MedianContamination
T2	MaxContamination, MeanContaminationLevel, MedianContamination
T3	MaxContamination, MeanContaminationLevel, SamplesPositives, SdContaminationLevel
T4	MeanContaminationLevel, SamplesPositives, RangeContamination
T5	ContaminationLevel
T6	ContaminationLevel
T7	MeanContaminationLevel, SamplesPositives, RangeContamination

Table 3. Data set description

The obtained results. We present here the results obtained by applying our algorithm on the combinations of the above seven tables. We only present the results obtained by the following comparisons: (T1, T2), (T1, T3), (T4, T7) and (T5, T6). These combinations have been made on data tables having the most common composed concepts.

(a)

	Recall	Precision	F-measure	Best T_{dup}
(T1, T2)	1	1	1	1
(T4, T7)	1	1	1	1
(T5, T6)	1	1	1	0.75
(T1, T3)	-	-	-	-

(b)

	Recall	Precision	F-measure
(T1, T2)	1	1	1
(T4, T7)	1	0.59	0.74
(T5, T6)	1	0.54	0.7
(T1, T3)	-	-	-

Table 4. Results in terms of recall, precision and F-Measure for the 4 combinations of tables: table (a) shows the best results for the 4 combinations and their corresponding threshold T_{dup} and table (b) shows the results for the 4 combinations where $T_{dup} = 0.7$

We have computed the recall, the precision and the F-measure by comparing the results obtained by our method with the gold-standard results given by a

domain expert. In Table 4 (a), we give the best results which are obtained for each pair of tables and their corresponding threshold T_{dup}. For the table pairs (T1, T2) and (T4, T7) we have obtained the maximum results, i.e. all the duplicate data have been detected by our method and all the detected duplicates are correct. These results are represented by a F-measure equals to 1 for a threshold equals to 1. For the table pair (T5, T6) we have obtained the maximum results where T_{dup} equals to 0.75. In Table 4 (b) we show the obtained results for the four combinations where T_{dup} is fixed at 0.7. We obtain the maximum results (F-measure equals to 1) for the tables T1 and T2. We obtain an F-measure of 0.74 and 0.7 for the table pairs (T4, T7) and (T5, T6) respectively. We note that comparisons between T1 and T3 correspond to the case of tables without duplicates. No duplicates have been detected by the method, which corresponds to the expected behavior. It is denoted by a dash in Table 4 (a) and (b).

5 Conclusion

In this paper we have presented our automatic and ontology-based approach of duplicate detection in Web data tables. The originalities of this work are threefold: (i) the declarative way of exploiting ontology knowledge in the duplicate detection process, (ii) the development and the use of suitable similarity measures between numerical and symbolic fuzzy sets; and (iii) the ability to handle heterogeneous and imprecise data at different levels of granularity.

Our proposal in this paper can be compared to approaches studying the reference reconciliation problem, i.e., detecting whether different data descriptions refer to the same real world entity (e.g. the same person, the same paper, the same protein). Different approaches have been proposed. [15, 16, 8] have developed supervised reference reconciliation methods which use supervised learning algorithm in order to help the duplicate detection. Those methods require a set of reference pairs labeled as reconciled or not reconciled. [17, 3] proposes a declarative approach which relies on expert knowledge expressed in an ontology and does need a learning phase. Since we have a domain TOR and we do not want to add a learning phase, we have proposed to extend the work of [3] in order to detect duplicates between data tables using their fuzzy semantic annotations. In a close domain to the references reconciliation, works have been done on data table fusion. [18, 19], in particular, study the data integration into the Cloud in order to help end-users to collaboratively manage their data. Our approach is complementary since it detects duplicates between data tables which were extracted from the Web, before storing them in a data warehouse.

The efficiency of our duplicate detection method has been evaluated and validated on real data in the chemical risk in food domain. As future work, we plan to test our method on bigger data sets, in order to show its scalability. We aim also to study how information on data provenance (e.g., document authors, source reputation, etc) can help to improve the distinction between duplicate data, similar data and distinct data. Finally, it will be interesting to extend the

proposed approach by studying how to deal with duplicate detection when data tables have been annotated thanks to different ontologies.

References

1. Hignette, G., Buche, P., Dibie-Barthélemy, J., Haemmerlé, O.: Fuzzy annotation of web data tables driven by a domain ontology. In: ESWC. Volume 5554 of Lecture Notes in Computer Science. (2009) 638–653
2. Zadeh, L.: Fuzzy sets. Information and control **8** (1965) 338–353
3. Saïs, F., Pernelle, N., Rousset, M.C.: Combining a logical and a numerical method for data reconciliation. J. Data Semantics **12** (2009) 66–94
4. Roche, C., Calberg-Challot, M., Damas, L., Rouard, P.: Ontoterminology - a new paradigm for terminology. In: International Conference on Knowledge Engineering and Ontology Developmenet, KEOD. (2009) 321–326
5. Reymonet, A., Thomas, J., Aussenac-Gilles, N.: Modelling ontological and terminological resources in OWL DL. In: OntoLex 2007 - Workshop at ISWC07 6th International Semantic Web Conference. (2007)
6. Dubois, D., Prade, H.: The three semantics of fuzzy sets. Fuzzy Sets and Systems **90** (1997) 141–150
7. Bouchon-Meunier, B., Rifqi, M., Bothorel, S.: Towards general measures of comparison of objects. Fuzzy sets and Systems **11** (1996) 143–153
8. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: KDD. (2003) 39–48
9. Jaccard, P.: ètude comparative de la distribution florale dans une portion des alpes et des jura. Bulletin de la société vaudoise des sciences naturelles **37** (1901) 547579
10. Tversky, A.: Features of similarity. Psychological Review **84** (1977) 327–352
11. Largeron, C., Kaddour, B., Fernandez, M.: Softjaccard: une mesure de similarité entre ensembles de chaînes de caractères pour l'unification d'entités nommées. Extaction et gestion des connaissances(EGC) (2009)
12. Hsieh, C.H., Chen, S.H.: Similarity of generalized fuzzy numbers with graded mean integration represntation. in Proc. 8th Int. Fyzzy System Association World Congr. **2** (1999) 551–555
13. Chen, S.M.: New methods for subjective mental workload assessment and fuzzy risk analysis. Cybernetics and Systems **27** (1996) 449–472
14. Chen, S.J., Chen, S.M.: Fuzzy risk analysis based on similarity measures of generalized fuzzy numbers. IEEE **11**(1) (2003) 45–56
15. Cohn, D.A., Atlas, L.E., Ladner, R.E.: Improving generalization with active learning. Machine Learning **15**(2) (1994) 201–221
16. Tejada, S., Knoblock, C.A., Minton, S.: Learning object identification rules for information integration. Inf. Syst. **26**(8) (2001) 607–633
17. Saïs, F., Pernelle, N., Rousset, M.C.: L2R: A logical method for reference reconciliation. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, British Columbia, Canada. (2007) 329–334
18. Gonzalez, H., Halevy, A.Y., Jensen, C.S., Langen, A., Madhavan, J., Shapley, R., Shen, W.: Google fusion tables: data management, integration and collaboration in the cloud. In: SoCC. (2010) 175–180
19. Gonzalez, H., Halevy, A.Y., Jensen, C.S., Langen, A., Madhavan, J., Shapley, R., Shen, W., Goldberg-Kidon, J.: Google fusion tables: web-centered data management and collaboration. In: SIGMOD Conference. (2010) 1061–1066