Knowledge Engineering Method based on Consensual Knowledge and Trust Computation : the MUSKA System

Fabien Amarger[2], Jean-Pierre Chanet[1], Ollivier Haemmerlé[2], Nathalie Hernandez[2], and Catherine Roussey[1]

[1] Irstea, UR TSCF Technologies et systèmes d'information pour les agrosystèmes, 9 avenue Blaise Pascal - CS 20085 Aubiére, F-63178, France
{firstname.lastname}@irstea.fr
[2] IRIT, UMR 5505, UT2J, Département de Mathématiques-Informatique, 5 allées Antonio Machado,
F-31058 Toulouse Cedex, France firstname.lastname@univ-tlse2.fr

Abstract. We propose a method for building a knowledge base addressing specific issues such as covering an end-users' need and reuse non-ontological sources such as thesauri or classifications. After designing an ontology module representing the knowledge needed by end-users, we enrich and populate it automatically with knowledge extracted from existing sources. The originality of our proposition is to propose ontological object candidates from existing sources according to both their relatedness to the ontological module and to computed measures of trust. This paper describes the trust measures we propose which are obtained by analysing the consensus found in existing sources. Thus we consider that knowledge are more reliable if it has been extracted from several sources. The use of these measures has been evaluated on a real case study with experts from the agriculture domain.

Keywords: Ontology Development; Trust; Non-Ontological Sources; Ontology Design Pattern; Ontology Merging

1 Introduction

In many fields, domain specific information is distributed on the Web as structured data (such as databases or thesauri) gathered for a specific usage. End-users are often lost when facing this amount of data as they have to look for available sources, analyse their quality, retrieve specific information from each of them and compare them. Alongside, the Linked Open Data (LOD) initiative aims at linking and facilitating querying on available data. Approaches such as [13, 14] have been proposed to formalise existing sources, define vocabularies to describe them and publish them on the LOD. However, approaches are still needed in order to help end-users collect and access knowledge to achieve a specific task in specialised domains.

We proposed a method for building a Knowledge Base (KB) covering end-users' needs and extracted from non-ontological sources such as thesauri or classifications. The main idea is to design an ontology module representing the knowledge needed by end-users and to populate it and enrich it automatically with data extracted from existing sources. The originality of our proposition is to propose ontological object candidates from existing sources according to both their relatedness to the ontological module and to their trust score. This trust is obtained by analysing the consensus found in existing sources. Thus we consider that knowledge are more reliable if it has been extracted from several sources. A first experiment carried out with agronomic experts has validated the relevancy of our approach. This experiment has extracted instances and properties between instances using an ontological module [1]. As non-ontological sources often contain rich lexical data we propose to improve our method and add labels to the previous ontological objects. A deep analysis of existing sources has shown that we can improve our method and generate new classes, which specialised the ontological module classes and extract $rdf : type$ property between instances and classes. This paper is an extension of our previous work and presents measures to compute trust for each new type of ontological objects that are extracted by our method. More information about the extraction method are available in french language using a different formalisation [2, 3]. These measures are evaluated on a real case study in agriculture and we believe they can be applied in any knowledge base merging process.

The paper is organised as follows. First we give an overview on the MUSKCA system in order to explain how ontological object candidates are extracted from non-ontological sources. In section 3, we present our proposal for computing consensual trust measures. Finally we describe and analyse the experiments.

2 Overview on the MUSKCA approach

Our method is composed of three processes detailed in [1]

1. **Source analysing**: During this process, the domain expert and the ontologist work together to select the most appropriate sources to build their KB. They inspect each source to evaluate its coverage and to have a broad idea if the source can be transformed to a KB or not.
2. **Source Transformation**: This process transforms each source into a KB in OWL format. It is based on Neon methods and consists in using transformation patterns for enriching and populating an ontological module defining a users' information need. The module is composed of $owl : Classes$ and defines the set of $owl : Properties$ that may exist between them. It is designed using Ontology Design Patterns and vocabularies already published on the LOD. An example of one of our modules is AgronomicTaxon [12]. At the end of this stage, the automatically generated KBs are composed of specialisation and instantiation of the classes and properties defined in the ontological module. Figure 1 shows a sample of the automatically generated KB for the AgronomicTaxon module using Agrovoc.

3. **KB Merging**: This process builds the final KB based on all KBs extracted from sources. As far as we know, this process is not proposed in any ontology engineering method. Usually an ontology engineering method uses several sources separately in order to enrich the KB in an incremental way. Here, the merging process uses several KBs at the same time in order to extract consensual knowledge.

This paper focuses on the measures used during the last step in order to merge ontological objects extracted from the different KBs. Ontological objects can be of several kinds :

- subclasses of classes of the ontological module (for example *Subkingdom* in Figure 1)
- instances of classes or subclasses of the ontological module (*Plantae* in Figure 1)
- labels for these classes and instances (Not represented on Figure 1 for clarity)
- rdf:type property linking instances to classes (the property between *Plantae* and $Neon : kingdom$ in Figure 1)
- Any property defined in the ontological module that links instances (the relation $hasHigherRank$ between *Embryophyta* and *Plantae* in Figure 1

Extracting ontological objects from various sources requires considering the trust that can be given to these objects. Several definitions of the notion of trust in the fields of computer science and semantic web are presented in [4]. The one which corresponds the most to our purpose is:

"Trust of a party A to a party B for a service X is the measurable belief of A in that B behaves dependable for a specified period within a specified context (in relation to service X)."

Consider A as the user who wants to create a knowledge base, B as a source and X the extraction process, for a specific period and context.

In Information Retrieval and Information Extraction, a common assumption is that the frequency of a word or a phrase increases the trust that can be given to this piece of information [7]. On the web, the reputation of pages also takes into consideration by computing trust using the hypertext links they contains, as done in the well known PageRank algorithm. Authors of [6] claim that in Semantic Web there is more to trust than reputation, putting notably forwards that the context in which a statement occurs has to be considered. This position has motivated the fact that we consider building a KB corresponding to a specific users' need represented by an ontological module. [9] highlights the fact that measures are needed in order to assign trust to specific pieces of a source, considering that when evaluating trust the information contained in the source can not be considered as a whole. As many propositions have been made in order to measure the trust to give to the sources themselves [5], this paper focuses on how trust can be evaluated for elements of a KB candidate for reuse in a specific context.

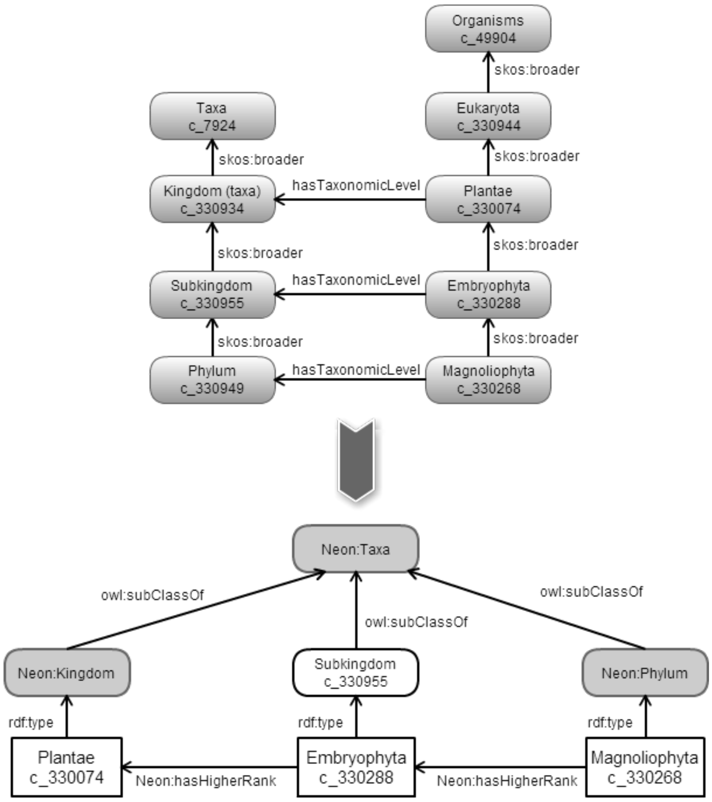

Fig. 1. Examples of ontological objects extracted from Agrovoc

3 Our Trust Computation Measures

To compute the trust we can give to an ontological object found in several sources, we propose to first reuse an alignment tool in order to identify redundant elements (sub-section 3.1). Then we generate potential candidates (sub-section 3.2). Finally we propose two ways of computing the trust for these candidates (sub-section 3.3 and 3.4).

3.1 Using mappings

In order to evaluated the trust of a specific object in a given KB, we propose to consider mapping that can be established from this object to others objects of different KBs. This idea follows the same principle as exploiting hyperlinks to compute trust for web pages. As mappings between objects of different KB have to be identify, we propose to use alignment techniques. Alignment is a large research area [8] and many methods have been proposed and implemented in tools. For the moment, mature tools only deal with homogeneous mappings identified between 2 knowledge bases [10]. We thus consider, in this paper, mappings between ontological objects of the same type (class, individual, property) belonging to two knowledge bases.

Let's consider two knowledge bases KB_1 and KB_2. Aligning KB_1 and KB_2 consists in computing all the mappings between objects of KB_1 and objects of KB_2 of the same type.

Let's define a mapping m as a triplet $< e_i, e_j, s_{ij} >$ such as:

- $e_i \in KB_i$: is an ontological object belonging to KB_i,
- $e_j \in KB_j$: is another ontological object belonging to KB_j ($KB_j \neq KB_i$),
- $type(e_i) = type(e_j)$: e_i and e_j are ontological objects of the same type (class, individual, object property, etc...),
- $s_{ij} = degree(e_i, e_j)$: where $degree$ is a function from $KB_i \times KB_j$ to $[0, 1]$ giving the similarity degree computed by an alignment tool for e_i and e_j.

Our goal is to explicitly identify the redundancy between sources, that is to say the ontological objects that correspond to each other in the different sources (at least 2, but possibly more). To do so, we consider that all the mappings between each possible pair of knowledge bases are generated.

3.2 Candidate Generation

According to the generated mapping between each possible pair of KB, we group similar objects in what we call a candidate. We identify five kinds of candidates :

Class Candidate (cc) We define a class candidate cc as a set of mappings associating pairs of classes belonging to different knowledge bases. The set of mappings identify similar classes in the different KBs. For n knowledge bases, a class candidate will be composed, at the most, of $n * (n-1)/2$ mappings.

Individual Candidate (ic) Individuals are instances of classes. In the same way as for class candidates, we identify an individual candidate ic as a set of mappings associating pairs of instances belonging to different knowledge bases.

Relation Candidate (rc) the property that links two individual candidates. We define a relation candidate rc as a pair of individual candidates, such as there exists the same property that links components of individual candidates of the different KB. For example in the Figure 2 the p property links $a1$ and $b1$ and also links $a3$ and $b3$.

Type Candidate (tc) $rdf : type$ properties that link an individual candidate to its type. Its type can either be a class candidate or a class that already exists in the module. Thus there exist two kinds of type candidate depending on the target of the $rdf : type$ property. We define a type candidate tc as an instance candidate that is linked to its type, such as there exists some $rdf : type$ properties that link a component of the individual candidate to a component of the class candidate or a class that is defined in the module.

Label Candidate (lc) A label candidate lc is associated to a class candidate cc or an individual candidate ic, called the *root* of lc. The root should have labels for at least two of its components. A new string mapping function is processed on the labels belonging to distinct components. We define a label candidate thanks to these new string mappings. A label candidate lc is a set of string mappings that links labels of distinct components of the root.

We define $dim(c)$ as the number of KBs involved in a candidate c. For simplicity dim of a label candidate lc will be equal to dim of its root.

Let's consider the example in Figure 2 with three knowledge bases KB_1, KB_2 and KB_3. KB_1 and KB_3 contain two individuals a_i, b_i linked by the same property p. The dash lines represent mappings between individuals. There are two individual candidates ic_1 and ic_2 and one relation candidate rc_1.

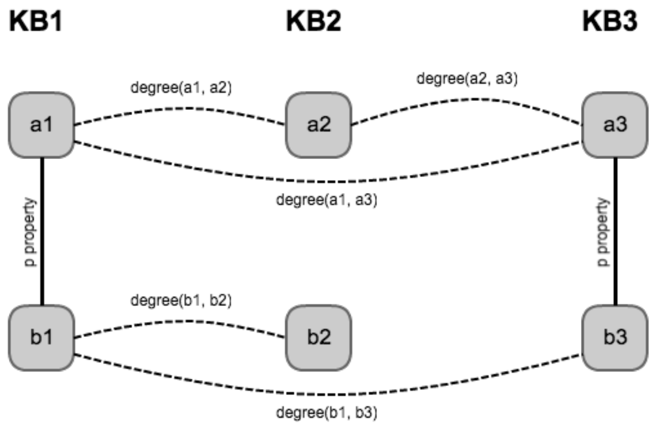

Fig. 2. Example of candidates

In this example, $dim(ic_1) = 3$, $dim(ic_2) = 3$ and $dim(rc_1) = 2$.

$$
\begin{aligned}
ic_1 &= \begin{cases} [< a_1, a_2, s_{12} >, < a_2, a_3, s_{23} >, < a_1, a_3, s_{13} >] \\ a_1 \in KB_1, a_2 \in KB_2, a_3 \in KB_3 \\ s_{12} = degree(a_1, a_2) \\ s_{23} = degree(a_2, a_3) \\ s_{13} = degree(a_1, a_3) \end{cases} \\
ic_2 &= \begin{cases} [< b_1, b_2, s_{12} >, < b_1, b_3, s_{23} >] \\ b_1 \in KB_1, b_2 \in KB_2, b_3 \in KB_3 \\ s_{12} = degree(b_1, b_2), s_{13} = degree(b_1, b_3) \end{cases} \\
rc_1 &= \begin{cases} [ic_1, ic_2] \\ [p(a_1, b_1), p(a_3, b_3)] \\ p(a_1, b_1) \in KB_1, p(a_3, b_3) \in KB_3 \end{cases}
\end{aligned}
\tag{1}
$$

Let's consider Figure 3 with three knowledge bases $KB1$, $KB2$ and $KB3$. In this figure there are two examples of label candidate having the same root (the shared individual candidate). The root contains three individuals $a1$, $a2$, $a3$. $a1$ has one label $la1$. $a2$ has two labels $la2.1$ and $la2.2$. $a3$ has also one label $la3$.

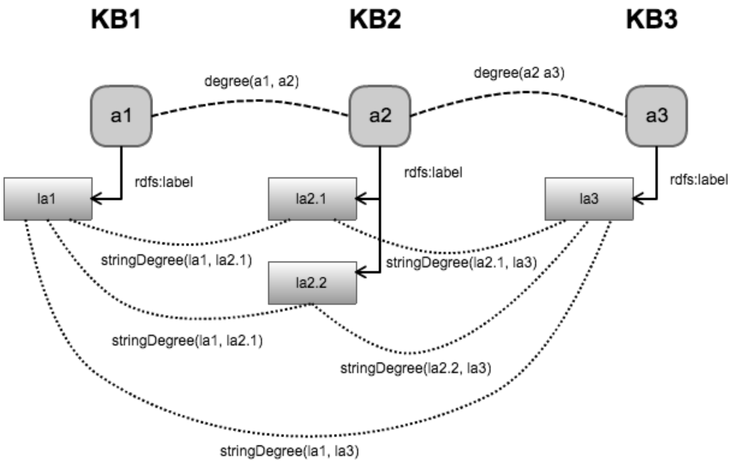

Fig. 3. Example of label candidates

$$
\begin{aligned}
root &= \begin{cases} [< a_1, a_2, s_{r1} >, < a_2, a_3, s_{r2} >, < a_1, a_3, s_{r3} >] \\ a_1 \in KB_1, a_2 \in KB_2, a_3 \in KB_3 \\ s_{r1} = degree(a_1, a_2) \\ s_{r2} = degree(a_2, a_3) \\ s_{r3} = degree(a_1, a_3) \end{cases} \\
lc_1 &= \begin{cases} [< la_1, la_{2.1}, s_{11} >, < la_{2.1}, la_3, s_{12} >] \\ la_1 \in KB_1, la_{2.1} \in KB_2, la_3 \in KB_3 \\ s_{11} = stringDegree(la_1, la_{2.1}), \\ s_{12} = stringDegree(la_{2.1}, la_3) \end{cases} \\
lc_2 &= \begin{cases} [< la_1, la_{2.2}, s_{21} >, < la_{2.2}, la_3, s_{22} >] \\ la_1 \in KB_1, la_{2.2} \in KB_2, la_3 \in KB_3 \\ s_{21} = stringDegree(la_1, la_{2.2}), \\ s_{22} = stringDegree(la_{2.2}, la_3) \end{cases}
\end{aligned} \tag{2}
$$

In the label candidates example 2 the $stringDegree(l1, l2)$ is the score of the similarity between the two strings.

Each candidate has a trust score to define how much we can trust this candidate. There are several way to compute this score. We define several trust functions that we will test in the experiments.

3.3 Simple Trust Function

A simple way to extract consensual ontological objects is to determine in how many KBs the candidate appears. We compute a ratio between the $dim(c)$ and the $nb_{sources}$ (the total number of sources used). We defined a function called $trust_{simple}$ to implement simple consensus:

$$trust_{simple}(c) = \frac{dim(c)}{nb_{Sources}} \tag{3}$$

3.4 Degree Trust Function

With the previous fonction, we consider that all the mappings proposed by the alignment tool are correct and can be trusted in the same way. With the following measures, we take into consideration both the number of sources in which the ontological object has been found and the number of mappings that have been established for each object. The intuition behind is to reuse works exploiting hypertext links thus considering that the more mappings can be established from the object to objects of other sources the more it can be trusted. The measures also take into consideration the degree assigned to each pair of ontological objects by the alignment tool thus considering that the more potential mappings (even if not sure) can be established the better is the candidate.

For the degree consensus implementation there is a different formula for each kind of candidate.

Individual candidate trust degree function This function is defined by the formula:

$$trust_{degree}(ic) = \frac{\sum_{i=1}^{dim(ic)} \sum_{j=i+1}^{dim(ic)} degree(a_i, a_j)}{\frac{nb_{Sources}(nb_{Sources}-1)}{2}} \tag{4}$$
$$such\ as (a_i, a_j) \in ic$$

This function sums all mapping degrees involved in the candidate. We normalised the result with the maximum number of individual mappings possible in an individual candidate (we have 3 KBs thus we can have at most 3 mappings in an individual candidate). Here, $nb_{sources}$ is the total number of KBs involved in the merging process.

Class candidate trust degree function This function is defined by the formula:

$$trust_{degree}(cc) = \frac{\sum_{i=1}^{dim(cc)} \sum_{j=i+1}^{dim(cc)} degree(c_i, c_j)}{\frac{nb_{Sources}(nb_{Sources}-1)}{2}} \tag{5}$$
$$such\ as (c_i, c_j) \in cc$$

This function is the same as the individual candidate trust function except that it considers the classes. It sums all mapping degree involved in the candidate and normalises as in formula 4.

Relation candidate trust degree function This trust function is defined by the formula:

$$trust_{degree}(rc) = \frac{dim(rc) + \frac{trust(ic1)+trust(ic2)}{2}}{nb_{Sources} + 1} \quad \text{such as } ic1 \in rc, ic2 \in rc \tag{6}$$

This formula takes into account the $dim(rc)$ and the average of the trust scores of individual candidates, components of the relation candidate. We do so to simulate a mapping degree between object properties as alignment tools do not match object properties. We normalise this result with the $nb_{Sources}$, which is the maximum value that $dim(rc)$ could be, plus 1, which is the maximum value that the average of the two ic trust scores could be.

Type candidate trust degree function There exist two trust functions for type candidate:

$$trust_{degree}(tc_1) = \frac{dim(tc_1) + \frac{trust(ic1)+trust(cc2)}{2}}{nb_{Sources} + 1} \quad \text{such as } ic1 \in tc_1, cc2 \in tc_1 \tag{7}$$

Formula 7 is dedicated to type candidate tc_1 that is composed of an individual candidate $ic1$ and a class candidate $cc2$. This formula takes into account the $dim(tc_1)$ and the average of trust scores of individual candidate and class candidate, components of the type candidate. We do so to simulate a mapping degree between properties. We normalise this result as in formula 6

$$trust_{degree}(tc_2) = \frac{dim(tc_2) + trust(ic)}{nb_{Sources} + 1} \quad \text{such as } ic \in tc_2 \tag{8}$$

Formula 8 is dedicated to type candidate tc_2 that is composed of an individual candidate ic and a class that already exists in the module.

Label candidate trust degree function This function is defined by the formula:

$$
\begin{aligned}
&trust_{degree}(lc) = \frac{trust_{degree}(root) + sum\_string\_degree(lc)}{2}\\
&sum\_string\_degre(lc) =\\
&\frac{\sum_{i=1}^{dim(root)} \sum_{j=i+1}^{dim(root)} stringDegree(label(a_i), label(a_j))}{\frac{nb_{Sources}(nb_{Sources}-1)}{2}} \qquad (9)\\
&\text{such as } a_i, a_j \in root\\
&label(a_i), label(a_j) \in lc
\end{aligned}
$$

This formula uses a label function called $label()$ that returns the label of an individual or a class, component of the root of lc. The $sum\_string\_degree()$ function sums the degrees of string mappings involved in the label candidate. We normalise the result with the maximum number of possible string mappings for candidate labels. The trust score of a label candidate sums the trust score of its root and the sum_string_degree value. We normalise the result, because its component takes its value between zero and one.

4 Experiments

Our approach has been implemented in a prototype called MUSKCA developed in Java. It is available on github at https://github.com/Murloc6/Muskca. After analysing the results of the OAEI challenge[3] and especially the ones of the instance matching task[4] [10] dealing with tools mapping all kinds of ontological objects, we chose to use LogMap [11] of which the source code is available online[5].

To experiment our work, we used MUSKCA on a real case study for which a knowledge base about plant classification is needed. This knowledge base will be used for analysing and annotating alert bulletins that inform farmers of pest attacks on crops. The specific information need is represented in the ontological module AgronomicTaxon described in [12]. Three well known sources : Agrovoc, Taxref, NCBI were considered in order to generate the knowledge base enriching and populating the AgronomicTaxon module. As our aim is to evaluate to what extent the trust measures help identifying relevant ontological objects from existing sources, we decided to compare the candidates identified thanks to the consensual trust score computation with the ontological objects manually selected from each source by the experts.

4.1 Gold standard

We asked three domain experts to analyse the three knowledge bases extracted automatically from the three sources. The experts had to determine for each

[3] Ontology Alignment Evaluation Initiative - http://oaei.ontologymatching.org/2013/
[4] http://www.instancematching.org/oaei/imei2013/results.html
[5] https://code.google.com/p/logmap-matcher/

source which ontological objects were relevant and if they are in the scope of the ontological module. An interface was implemented to collect the experts' opinion. Note that a real effort was made in order to present ontological objects in a way understandable for experts. Here are some questions asked:

- **Does *Magnoliophyta* belong to the domain?** We want to know if the instance Magnoliophyta is relevant and in the scope of the KB.
- **Does *angiosperm* designate *Magnoliophyta*?**
  We want to know if the labels associated to Magnoliophyta are correct as they are sometimes inexact (most of the time not synonyms) or not the right translation (if the source contains multilingual labels).

We thus obtained a list of ontological objects validated for each source. We then compared them with the candidates generated by the prototype. To do that we computed the precision, recall and f-measure for each kind of candidate.

4.2 Results

We ran two experiments:

1. The first experiment uses the $trust_{simple}$ function to compute the trust score of the candidates and a relatively permissive threshold for filtering candidates (fixed at 0,5 which means that ontological objects are found in at least half of the sources).
2. The second experiment uses the $trust_{degree}$ function to compute the trust score of the candidates and a threshold relatively permissive for filtering candidates (fixed at 0,6 in order to compare the results of these measures on the same amount of candidates as in the first experiment)

Tables 1 and 2 present the results of experiments on each kind of candidates.

$Candidatetype$	$Precision$	$Recall$	$F - Measure$
$Individual$	0.92	0.66	0.77
$Relation$	0.65	0.51	0.57
$Type$	0.70	0.43	0.54
$Label$	0.32	0.35	0.34
$Class$	1	0.38	0.55

Table 1. results of the first experiment: $trust_{simple}$ and threshold permissive

As we can see in Table 1, filtering candidates if they appear in at least half of the sources through the $trust_{simple}$ measure helps identifying relevant candidates. The relatively significant precision we obtain validate our intuition that redundant objects identified through mappings are a way of finding and ranking relevant candidates. Note that the results are less meaningful on labels. This can be explained by the fact that NCBI contains only few labels. Agrovoc contains many labels in several languages. It contains scientific labels and vernacular labels. TaxRef contains labels but they are limited due to the fact that the engineering process of this source implies a long verification phase. Recall results show that consensual trust is not the only approach that should be used in order to extract all the candidate from sources but at least those extracted are relevant.

Table 2 presents the results of the second experiment carried out with the $trust_{degree}$ measures exploiting the number of mappings and their degree. As we can see the precision increases. This is due to the fact that candidates that group 3 similar objects linked by 3 mappings are better ranked than candidates that group 3 similar objects linked by 2 mappings. Figure 4 shows two different candidates : for Triticum the 3 possible mappings have been found where as only 2 are identified for Triticum Dicoccoides. With the simple degree measure, Triticum Dicoccoides is considered as a valid candidate as it appears in the 3 sources but was rejected by the experts. We believe that the improvement would be more significant in an experiment involving more than 3 sources. Indeed in our experiment the number of mappings for an object vary from 2 to 3. With more than 3 sources, we hope that the number of mappings will vary to a larger extent.

$Candidatetype$	$Precision$	$Recall$	$F - Measure$
$Individual$	0.97	0.63	0.76
$Relation$	0.68	0.51	0.58
$Type$	0.77	0.39	0.52
$Label$	0.39	0.24	0.30
$Class$	1	0.13	0.23

Table 2. Results of the second experiment: $trust_{degree}$ and threshold permissive

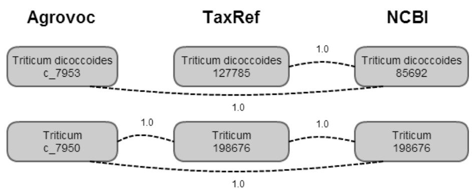

Fig. 4. Real example of an individual candidate with two alignments

During these experiments we compared two ways of computing a trust score on ontological objects based on their consensual degree. The comparison of the Table 1 and Table 2 shows that the use of $trust_{degree}$ ranks relevant candidates better. This happens because the consensual aspect has more impact in this formula than in $trust_{simple}$. This first experiment shows that the use of the consensus in the trust score computation increases the quality of the results.

5 Conclusion and future works

Our work consists in building a knowledge base with several kinds of ontological objects (individual, relation instance, type relation, label and classes) extracted from non-ontological sources. In this paper, we proposed two ways to compute the consensual trust score for filtering the potential candidates extracted from the sources. The first formula, $trust_{simple}$ is the ratio between the number of sources in which the candidate appears and the total number of sources considered. The second formula, $trust_{degree}$ takes into account, for each similar object in the different sources, the degree of all the mappings given by an alignment system. This formula gives more weight to consensus and to the quality of the agreement between the sources. An experiment involving experts from the agriculture domain has shown that the use of consensus in the trust score computation increases the quality of the results. We are currently defining a new evaluation protocol in order to analyse more deeply our approach when more than 3 sources are considered. The low precision obtained for the label candidates can be explained by the lack of consensus for this kind of ontological object in the sources we considered. The solution to this problem could be to emphasise the potential strength of each source in our process. In the context of our experiment, before analysing in depth the sources, the experts believed that labels extracted from Agrovoc were going to be more relevant than labels from NCBI. Approaches such as [5] could be used to evaluate beforehand the strengths of the considered sources. In this paper we consider all candidates independently from one another. We are aware that some candidates can be contradictory. We are thinking of using the argumentation theory to solve this problem.

Acknowledgments. We want to thank specially the three experts who helped us validate our results by generating the gold standard: Franck Jabot and Vincent Soulignac from Irstea Clermont-Ferrand and Jacques Le Gouis from INRA Clermont-Ferrand.

References

1. F. Amarger, C. Roussey, J.P. Chanet, O. Haemmerlé, and N. Hernandez. Skos sources transformations for ontology engineering: Agronomical taxonomy use case. *MTSR*, 2014.
2. Fabien Amarger. *Vers un systme intelligent de capitalisation de connaissances pour l'agriculture durable : construction d'ontologies agricoles par transformation de sources existantes*. PhD thesis, Université de Toulouse 2 le Mirail, 2015.
3. Fabien Amarger, Jean-Pierre Chanet, Ollivier Haemmerlé, Nathalie Hernandez, and Catherine Roussey. Construction d'une ontologie par transformation de systèmes d'organisation des connaissances et évaluation de la confiance. *Ingénierie des Systèmes d'Information*, 20(3):37–61, 2015.
4. D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, pages 58–71, 2007.

5. Raju Balakrishnan and Subbarao Kambhampati. Sourcerank: Relevance and trust assessment for deep web sources based on inter-source agreement. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 227–236, New York, NY, USA, 2011. ACM.
6. Christian Bizer and Radoslaw Oldakowski. Using context- and content-based trust policies on the semantic web. In *Proceedings of the 13th international conference on World Wide Web - Alternate Track Papers & Posters, WWW 2004, New York, NY, USA, May 17-20, 2004*, pages 228–229, 2004.
7. Charles L. A. Clarke, Gordon V. Cormack, and Thomas R. Lynam. Exploiting redundancy in question answering. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, pages 358–365, 2001.
8. J. Euzenat and P. Shvaiko. *Ontology matching.* 2007.
9. Yolanda Gil and Donovan Artz. Towards content trust of web resources. In *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*, pages 565–574, 2006.
10. Bernardo Cuenca Grau, Zlatan Dragisic, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jiménez-Ruiz, Andreas Oskar Kempf, Patrick Lambrix, et al. Results of the ontology alignment evaluation initiative 2013. In *ISWC workshop on ontology matching (OM)*, pages 61–100, 2013.
11. Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Yujiao Zhou, and Ian Horrocks. Large-scale interactive ontology matching: Algorithms and implementation. In *European Conference on Artificial Intelligence*, pages 444–449, 2012.
12. C Roussey, J.P. Chanet, V. Cellier, and F. Amarger. Agronomic taxon. In *WOD*, page 5, 2013.
13. D. Soergel, B. Lauser, A. Liang, F. Fisseha, J. Keizer, and S. Katz. Reengineering thesauri for new applications: The AGROVOC example. *Journal of Digital Information*, pages 1–23, 2004.
14. B. Villazón-Terrazas, M. C. Suárez-Figueroa, and A. Gómez-Pérez. A pattern-based method for re-engineering non-ontological resources into ontologies. *Int. J. Semantic Web Inf. Syst.*, pages 27–63, 2010.