Dealing with Incompatibilities During a Knowledge Bases Fusion Process

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, 63178 Aubière, France
{jean-pierre.chanet,catherine.roussey}@irstea.fr
[2] IRIT, UMR 5505, Université Toulouse – Jean Jaurès,
5 allées Antonio Machado, 31058 Toulouse Cedex, France
{fabien.amarger,ollivier.haemmerle,nathalie.hernandez}@univ-tlse2.fr

Abstract. More and more data sets are published on the linked open data. Reusing these data is a challenging task as for a given domain, several data sets built for specific usage may exist. In this article we present an approach for existing knowledge bases fusion by taking into account incompatibilities that may appear in their representations. Equivalence mappings established by an alignment tool are considered in order to generate a subset of compatible candidates. The approach has been evaluated by domain experts on datasets dealing with agriculture.

Keywords: Knowledge acquisition · Knowledge base fusion · Incompatibilities

1 Introduction

We propose in this paper a new fusion process of several Knowledge Bases (KBs) in order to extract consensual knowledge. We made the hypothesis that the final KB has a better quality because it will contain more reliable knowledge than the source KBs. Our fusion process starts by aligning KBs by means of an existing alignment tool. Then we generate candidates which are sets of ontological elements from different aligned KBs. For each candidate, we evaluate its trust score. Previous work [1] propose several trust measures, which evaluate consensus found in existing KBs. Merging different KBs about the same domain may generate errors since the KBs can propose different viewpoints on a same domain. In this paper we focus on one kind of errors coming from complex mappings between three or more ontological elements. When two KB are aligned, if two elements are mapped, then it means that the two elements are judged equivalent with a certain degree. When one element of the KB_A is aligned with N elements of KB_B, we consider that $N-1$ mappings are wrong and we propose a method to select the reliable mapping from the N ones starting with the element of KB_A. From a complex mapping, the set candidates that will result will be considered as incompatible. We will therefore seek to discover the maximum of candidates subsets (extension) such that all candidates of an extension are compatible.

This paper is organized as follows: (1) generic presentation of our fusion process, (2) generation of set of compatible candidates called an extension and (3) evaluation with a use case on wheat taxonomy generation.

2 Overview of Our Fusion Process

Our fusion process is composed of four activities as shown in Fig. 1. It starts from several knowledge bases called *Source Knowledge Bases* (SKB). The *Mapping Generation*, based on an alignment tool, computes mappings between ontological elements belonging to distinct sources. The *Candidate Generation* builds candidates which are sets of ontological elements coming from different sources, considered as similar. The *Trust Computation* computes the trust scores of the candidates with respect to consensus and the reliability degree of the mappings. The *Discovery of the Optimal Extension* allows the generation of an extension representing the maximal subset of compatible candidates validated by an expert (optimal extension). The first three activities have been already presented in [1,2]. In this article, we focus on the discovery of the optimal extension.

Knowledge Base. We consider that a knowledge base KB is an oriented labelled multigraph. $V_{kb} = C \cup PropO \cup PropDT \cup I \cup L$ is the finite set of vertices belonging to the knowledge base kb dispatched into disjoint subsets such that C is the set of classes, $PropO$ is the set of object properties, $PropDT$ is the set of datatype properties, I is the set of individuals, L is the set of vertices representing literals, including labels.

Ontological Element. An ontological element oe is a vertex of a knowledge base which can be mapped by an automatic alignment tool. As far as we know, alignment tools can only map individuals or classes, but not yet properties or RDF triples. Thus we limit ontological elements to classes or individuals in our fusion process: $oe \in C \cup I$.

We also define a function $nature(oe)$ which returns the type of the ontological element, which can be "class" if $oe \in C$, "individual" if $oe \in I$, "null" otherwise.

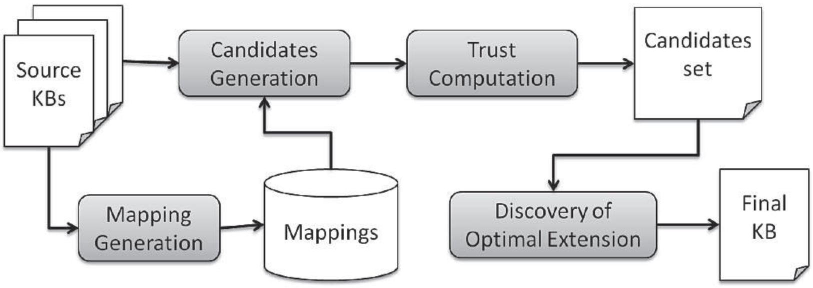

Fig. 1. Knowledge bases fusion process

Our work can be extended by considering as ontological elements other types of vertices. In that way, it could be possible to generate candidates in order to extract object properties or datatype properties.

Mappings. Assume SKB_i and SKB_j be two source knowledge bases used as input of our fusion process. We define a mapping m as an edge between a pair of vertices $\{oe_i, oe_j\}$, with $oe_i \in V_{skbi}, oe_j \in V_{skbj}$. A mapping represents an equivalence between two ontological elements, exhibited by means of the alignment tool. The mappings are defined in the following way:

- $V_{skbi} \neq V_{skbj}$: a mapping is always established between two vertices belonging to distincts SKB. $\nexists m = \{oe_i, oe_k\}$ such that $oe_i \in V_{skbi}$ and $oe_k \in V_{skbi}$.
- $nature(oe_i) = nature(oe_j) \neq$ "null". A mapping is always established between two ontological elements of same nature.
- $valueE : (C \cup I) \times (C \cup I) \rightarrow]0, 1]$ is a mapping which associates a unique reliability degree between 0 and 1 such that $valueE(oe_i, oe_j) = valueE(oe_j, oe_i)$ with each edge defined as a mapping.

In our work, we use the alignment tool LogMap[1] [6] since that system obtained good results during the evaluation OAEI 2014 [4] and, moreover, it allows to map individuals as well as classes.

The alignment systems allow to obtain mappings of type $1 : n$. In order to process these mappings, we made the hypothesis that such a mapping means that one element of a source is in relation of equivalence with one of the n elements of the other source, but the alignment tool could not make a choice and proposed a set of possible elements.

3 Generation of Compatible Candidates Set

On the basis of the hypothesis previously introduced, we first present the activity followed to generate candidates. Then, we explain how incompatibilities between candidates are identified. Finally we generate an extension which is a subset of compatible generated candidates.

3.1 Candidate Definition

A candidate *cand* represents an element that may belong to the final knowledge base. This candidate is the set of ontological elements extracted from the different SKBs and considered as equivalent by the alignment tool. We consider in the following that N SKBs have been aligned. A candidate $cand = (V_{cand}, E_{cand})$ is a non-oriented graph for which the vertices are ontological elements from the distinct SKBs and the edges are the mappings established by the alignment tool. We define the component of a candidate such as:

[1] http://www.cs.ox.ac.uk/isg/projects/LogMap/.

- $\forall v \in V_{cand}$ with $v \in V_{skbi}$ $\nexists v' \in V_{cand}$ such as $v' \in V_{skbi}$ et $v \neq v'$. All the vertices of a candidate belong to a different SKB. Therefore $|V_{cand}| \leq N$.
- E_{cand}: the edges of *cand* are mappings.
- A candidate is a connected graph. $\forall v_1, v_2 \in V_{cand}$, there exists necessarily a chain $\{e_1, ..., e_k\}$ with $e_i \in E_{cand}$ linking v_1 to v_2. Therefore, all the vertices of *cand* are linked to at least one other vertex of *cand* by a mapping, which implies that all the vertices of *cand* are of the same nature $\forall v_1, v_2 \in V_{cand}$ $nature(v_1) = nature(v_2)$.

Figure 2 presents two candidates for which elements are of the nature "individual". The two candidates – "Triticum" and "Triticum Durum" – represent ontological elements that can potentially belong to the final knowledge base. Edges drawn with dashes represent mappings with their reliability degrees.

3.2 Candidate Generation

From the set of sources and their alignments (such as presented in the background of Fig. 2) is extracted a multi-partite non-oriented graph of which the vertices are ontological elements of the different sources and edges are the mappings. All the connected components of this graph are computed. We consider that the minimal condition for identifying a consensus is if the candidate contains at least two ontological elements belonging to two distinct sources.

3.3 Incompatibilities

As explained previously, our process looks for connected components. This algorithm does not exclude the fact that an ontological element belongs to several candidates. This results from the fact that alignment tools identify $1 : n$ mappings, which means that an ontological element is mapped to 1 or several elements of the second knowledge base. We consider that these mappings are

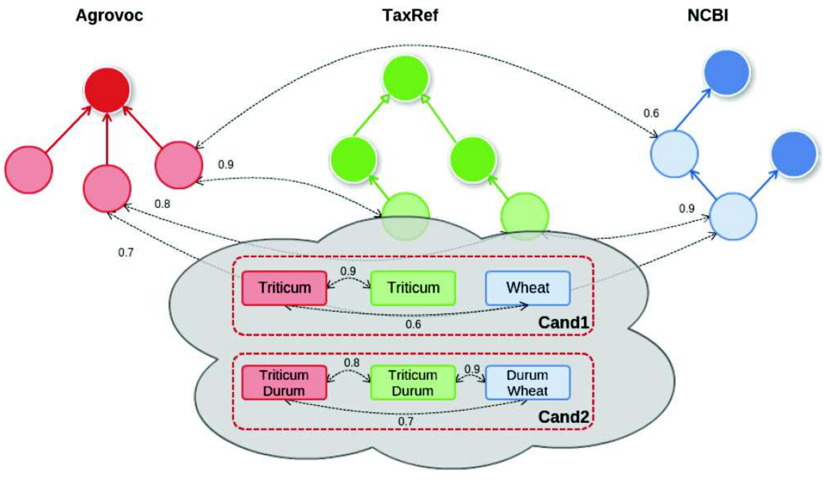

Fig. 2. Example of two candidates extracted from three sources

equivalence relations that the alignment tool wrongly established apart from one. We then define an incompatibility when several candidates share a common ontological element as shown in Fig. 3. We define an incompatibility as the couple of candidates that satisfy the following constraint: $Inc = \{Cand_1, Cand_2\}$ such that $\exists oe_{com}, oe_{com} \in V_{Cand_1}, oe_{com} \in V_{Cand_2}$.

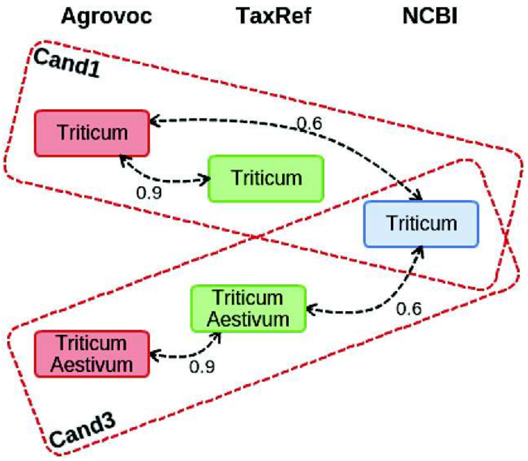

Fig. 3. Example of incompatibility between two candidates

Then we can define the non-oriented graph that represents the incompatibilities $G_{inc} = \{V_{inc}, E_{inc}\}$ such as the graph vertices V_{inc} are candidates and the graph edges E_{inc} represent an incompatibility between the connected vertices.

3.4 Extension Generation

By considering the incompatibility graph, we generate its complementary graph (the compatibility graph) that links only candidates that are not incompatible. Then we aim at generating the maximum cliques of the compatibility graph to obtain the extensions. Each maximum clique is a subset of candidates, each compatible. This corresponds to a classic graph theory problem: MCE (Maximum Clique Enumeration). To solve this MCE problem, several algorithms exist. The most used is Bron Kerbosch [3] for which many enrichment have been proposed such as Tomita [9] or more recently the algorithm of Eppstein et Strash [5]. Because of the NP-hardness of the problem, we do not aim at generating all the possible maximum cliques: our goal is to define a way of obtaining the optimal extension. To do so, we use the CSP solver GLPK (https://www.gnu.org/software/glpk/), which implements the Branch and Bound algorithm by maximising an objective function. The GLPK model we use tries to maximize the number of elements in *ext*, thus maximizing the number of candidates in the extension.

3.5 Finding the Optimal Extension

Thanks to our constraint model, we can obtain an extension among the possible ones. Our goal is to obtain the optimal extension that is the more likely to correspond to the expected knowledge base. To do so, we propose the expert to validate the generated extension. Then the process takes his/her opinion iteratively into account in order to add new constraints to the model and converge on an optimal solution for which all the candidates are correct.

During the validation step, all the candidates belonging to the current generated extension are presented to the expert. If the expert validates a candidate, a new constraint stating that the optimal extension must contain this candidate is added. For all non-validated candidates, the algorithm is run again in order to find a new extension. The extension is considered as optimal if all the candidates have been evaluated.

With this process, we can present to the expert a minimal number of candidates to evaluate. We deduce automatically that all incompatible candidates with a validated candidate must not belong to the optimal extension. The number of interactions with the expert is thus reduced. The time needed for the evaluation by the expert (number of interaction) is in the worst case equal to the number of generated candidates. When an incompatibility occurs, this number decreases.

4 Experiment About Wheat Taxonomy

Our evaluation use case is about the creation of a knowledge base on cereals. More information about this project are available in [1,8]. In this evaluation we only take care of the creation of a wheat taxonomy.

Our experts have chosen the following sources: (i) *Agrovoc*, the multilingual thesaurus managed by the FAO[2]. It contains over 40.000 terms, (ii) *TaxRef*, the french national taxonomic reference about living species managed by the national natural history museum[3]. It contains over 80.000 taxa, (iii) *NCBI Taxonomy*, the taxonomy created by the National Center for Biotechnology Information of United States[4]. It contains 1 million of taxa.

These sources have been transformed in knowledge bases using our SKOS transformation method [1], based on ontological module to drive the transformation process and define transformation patterns dedicated to each source. For our experiment, we use the ontological module called AgronomicTaxon[5] that merges several ontology design patterns [8]. Based on these three new knowledge bases, we can apply our activity of candidate generation. Then, we generate the graph of incompatibility. Table 1 presents data about this graph.

To evaluate our method, an expert evaluated candidates and we counted the number of interactions (validation or invalidation) that were required to

[2] http://aims.fao.org/standards/agrovoc/about.

[3] http://inpn.mnhn.fr/programme/referentiel-taxonomique-taxref.

[4] http://www.ncbi.nlm.nih.gov/.

[5] https://sites.google.com/site/agriontology/home/irstea/agronomictaxon.

Table 1. information about graph of incompatibility

Sources	$\|eos\|$	$\|cands\|$	$\|incomps\|$
Agrovoc	11	150	1555
TaxRef	19		
NCBI	130		

eos: ontological elements.

Table 2. Results

$\|ext_{max}\|$	$\|ext_{opti}\|$	Nb interactions	Ratio
25	23	62	0.41

ext_{max}: *nb of candidates in the largest extension,*
ext_{opti}: *nb of candidates in the optimal extension,*
$Ratio = \frac{nb\ interactions}{|cands|}$.

obtain the optimal extension. Table 2 presents these results. The size of the largest extension generated by our algorithm without manual evaluation was 25 candidates. The size of the extension manually evaluated was 23 candidates. Thus, 2 sets of incompatible candidates were not validated by the expert. None of the candidates pairwise incompatible were validated by the expert. That means that the $1 : n$ mappings between ontological elements were wrong. The expert needed 62 interactions to build the final extension. That means that he had to validate or invalidate 62 candidates among the 150 possible candidates. The ratio is 0.41. 41 % of candidates had to be observed by expert to build the optimal extension. Thus our method divided by 2 the number of candidates to build the optimal extension.

Then we calculated the coverage and conciseness metrics defined in [7]. The redundancy metric is not relevant for our fusion process because we know that we have eliminated redundancy. This is the main advantage of our fusion process.

The coverage is the ratio between the number of the candidates in the extension (23) and the number of ontological elements extracted for each source. We obtained **Agrovoc:** 2.09, **TaxRef:** 1.12 and **NCBI:** 0.17.

The conciseness evaluates if the extension is the minimal representation of knowledge. This metric is useful to detect fusion process that merge data without aggregation. Conciseness calculates the ratio between the size of the extension (23) and the sum of elements extracted from sources (160). We obtained the value 0.14. Thus, we can conclude that the optimal extension is close to the minimum representation of knowledge.

During these evaluations, a phenomenon appeared that could significantly reduce the number of expert interactions. Indeed, several incompatible candidates are successively presented to the expert until he validates one. All these incompatible candidates come from the same complex mapping $1 : n$. We notice that the expert did not validate the candidates which contained elements that had less labels in common. Furthermore he validated the candidate that contains elements which had more labels in common. It would be interesting to present first the candidate that contains elements which have more labels in common. This idea could be generalized by taking into account not only the labels but the neighborhood. We can add these information in the objective function. Thus we would first present the candidates that contain elements which have the more neighbours (vertices or edges) in common.

5 Conclusion and Future Works

This paper presents a process to manage incoherencies between knowledge bases in a fusion process. The fusion process generates candidates that group similar elements extracted from distinct knowledge bases. We define what are incompatible candidates. These incompatibilities help to identify subsets of compatible candidates, which are called extensions. A method to find the optimal extension is proposed. Incompatibilities between candidates are also used during manual evaluation in order to limit the number of candidates that should be evaluated by experts. These methods have been validated on a real use case. The use case creates a knowledge base on plant taxonomy from multiple sources.

Our method of extension generation can be improved by taking into account the trust score of candidates. Our previous works present several functions to compute the trust score [1]. These scores can be used to optimize the objective function of our extension generation algorithm. Another interesting perspective should be to work on edge candidates. An edge candidate is composed of two candidates linked by edges having the same label (the same property). Our extension generation method should integrate neighboring candidates and edge candidates in order to faster the optimal extension discovery.

References

1. Amarger, F., Chanet, J.-P., Haemmerlé, O., Hernandez, N., Roussey, C.: SKOS sources transformations for ontology engineering: agronomical taxonomy use case. In: Closs, S., Studer, R., Garoufallou, E., Sicilia, M.-A. (eds.) MTSR 2014. CCIS, vol. 478, pp. 314–328. Springer, Heidelberg (2014)
2. Amarger, F., Chanet, J.-P., Haemmerlé, O., Hernandez, N., Roussey, C.: 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)
3. Bron, C., Kerbosch, J.: Algorithm 457: finding all cliques of an undirected graph. Commun. ACM **16**, 575–577 (1973)
4. Dragisic, Z., Eckert, K., Euzenat, J., Faria, D., Ferrara, A., Granada, R., Ivanova, V., Jiménez-Ruiz, E., Kempf, A.O., Lambrix, P., et al.: Results of the ontology alignment evaluation initiative 2014 (2014)
5. Eppstein, D., Strash, D.: Listing all maximal cliques in large sparse real-world graphs. In: Pardalos, P.M., Rebennack, S. (eds.) SEA 2011. LNCS, vol. 6630, pp. 364–375. Springer, Heidelberg (2011)
6. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: logic-based and scalable ontology matching. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 273–288. Springer, Heidelberg (2011)
7. Raunich, S., Rahm, E.: Towards a benchmark for ontology merging. In: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (eds.) OTM-WS 2012. LNCS, vol. 7567, pp. 124–133. Springer, Heidelberg (2012)

8. Roussey, C., Chanet, J.P., Cellier, V., Amarger, F.: Agronomic taxon. In: WOD, p. 5 (2013)
9. Tomita, E., Tanaka, A., Takahashi, H.: The worst-case time complexity for generating all maximal cliques and computational experiments. Theoret. Comput. Sci. **363**, 28–42 (2006)