OWL-CM : OWL Combining Matcher based on Belief Functions Theory

Boutheina Ben Yaghlane[1] and Najoua Laamari[2]

[1] LARODEC, Université de Tunis,
IHEC Carthage Présidence 2016 Tunisia
boutheina.yaghlane@ihec.rnu.tn
[2] LARODEC, Université de Tunis,
ISG Tunis Tunisia
laamarinajoua@yahoo.fr

Abstract. In this paper we propose a new tool called OWL-CM (OWL Combining Matcher) that deals with uncertainty inherent to ontology mapping process. On the one hand, OWL-CM uses the technique of similarity metrics to assess the equivalence between ontology entities and on the other hand, it incorporates belief functions theory into the mapping process in order to improve the effectiveness of the results computed by different matchers and to provide a generic framework for combining them. Our experiments which are carried out with the benchmark of Ontology Alignment Evaluation Initiative 2007 demonstrate good results.

1 Presentation of the system

1.1 State, purpose, general statement

Semantic heterogeneity has been identified as one of the most important issue in information integration [5]. This research problem is due to semantic mismatches between models. Ontologies which provide a vocabulary for representing knowledge about a domain are frequently subjected to integration.

Ontology mapping is a fundamental operation towards resolving the semantic heterogeneity. It determines mappings between ontologies. These mappings catch semantic equivalence between ontologies. Experts try to establish mappings manually. However, manual reconciliation of semantics tends to be tedious, time consuming, error prone, expensive and therefore inefficient in dynamic environments, and what's more the introduction of the Semantic Web vision has underscored the need to make the ontology mapping process automatic.

Recently, a number of studies that are carried out towards automatic ontology mapping draw attention to the difficulty to make the operation fully automatic because of the cognitive complexity of the human. Thus, since the (semi-) automatic ontology mapping carries a degree of uncertainty, there is no guarantee that the outputted mapping of existing ontology mapping techniques is the exact one.

In this context, we propose a new tool called OWL-CM (OWL Combining Matcher) with the aim to show how handling uncertainty in ontology mapping process can improve effectiveness of the output.

1.2 Specific techniques used

On the one hand OWL-CM uses the Dempster-Shafer theory of evidence [11] to deal with uncertainty inherent to the mapping process, especially when interpreting and combining the results returned by different matchers. On the other hand it uses the technique of similarity measures in order to assess the correspondence between ontology entities. For the OWL-CM tool contest we have proposed an architecture (see figure 1) that contains four components. The *transformer* takes as input two ontologies (O_1 and O_2) and constructs for each one a database (DB_1 and DB_2). The database schema meets a standard schema that we designed based on some axioms of RDF(S) and OWL languages. The *filters* decide on result mappings. Whereas *simple matchers* and *complex matchers* assess the equivalence between entities.

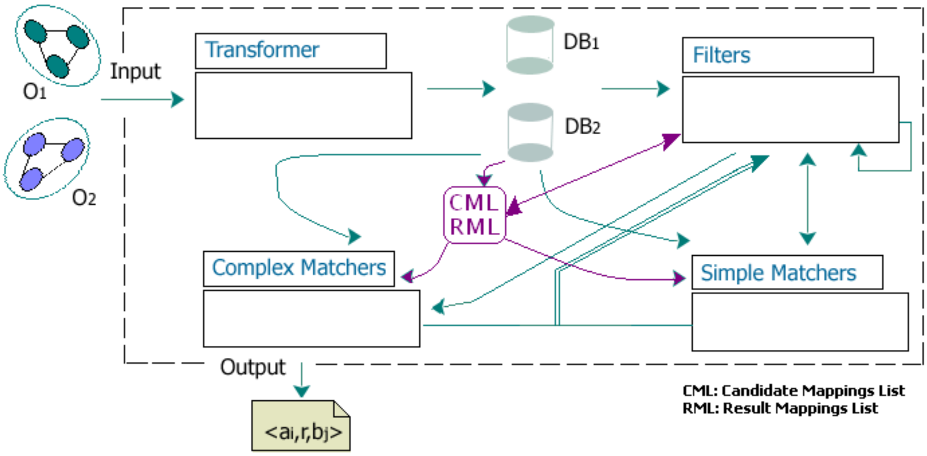

Fig. 1. OWL-CM Architecture.

The corresponding algorithm that we have implemented follows four steps (see figure 2). The first step called pre-mapping is mainly devoted to convert each one of the input ontologies O_1 and O_2 into a database (DB_1 and DB_2). The following three ones allow performing sequentially the iteration about concepts mapping, followed by the iteration about object properties mapping, and ended by the iteration about datatype properties mapping. Each iteration is based on some methods belonging to four categories of tasks namely initialization, screening, handling uncertainty, and ending. The algorithm requires as input two ontologies to be mapped and two databases that have to be declared as

ODBC data source systems. It outputs three lists of result mappings which are produced sequentially, each one is returned close of the corresponding iteration of mapping. The total result is returned in the form of a file.

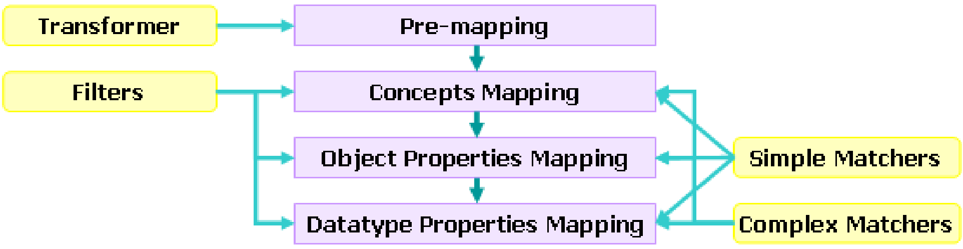

Fig. 2. OWL-CM Algorithm.

1.2.1 Preliminary concepts

The following list draws up some of the preliminaries that are used by our approach.

1. **Candidate Mapping**: We define a *candidate mapping* as a pair of entities $({e_i}^1, {e_j}^2)$ that is not yet in map.

2. **Result Mapping**: We define a *result mapping* as a pair of entities that had been related, $\langle {e_i}^1, \equiv, {e_j}^2 \rangle$ denotes that entity ${e_i}^1$ is equivalent to entity ${e_j}^2$, whereas $\langle {e_i}^1, \perp, {e_j}^2 \rangle$ denotes that the two entities are not equivalent.

3. **Similarity measure**: The *similarity measure*, *sim*, is a function defined in [3] based on the vocabularies ε_1 of the ontology O_1 and ε_2 of the ontology O_2 as follows:

$$sim\colon \varepsilon \times \varepsilon \times O \times O \rightarrow [0..1]$$

   - *sim(a, b) = 1 ⇔ a = b: two objects are assumed to be identical.*
   - *sim(a, b) = 0 ⇔ a ≠ b: two objects are assumed to be different and have no common characteristics.*
   - *sim(a, a) = 1: similarity is reflexive.*
   - *sim(a, b) = sim(b, a): similarity is symmetric.*
   - *Similarity and distance are inverse to each other.*

   A similarity measure function assesses the semantic correspondence between two entities based on some features. In table 1, we draw up the list of similarity measures employed depending on the type of entities to be mapped. Furthermore, we distinguish between two types of similarity: the *syntactic one* assessed by the measures that evaluate distance between strings (e.g., String similarity and String equality) and the other measures dedicated to assess *semantic similarity* (e.g., String synonymy, Explicit equality and Set similarity).

4. **SEE (Semantic Equivalent Entity)**: Depending on the type of entities, we formally define the *semantic equivalence* between two entities as follows:
   Definition (SEE) .
   An entity ${e_j}^2$ is semantically equivalent to an entity ${e_i}^1$ such that $({e_i}^1, {e_j}^2) \in \{C^1 \times C^2\}$, i.e., $\langle {e_i}^1, \equiv, {e_j}^2 \rangle$, if at least one of the following conditions is true:
   $$\left| \begin{array}{l} sim_{expeql}({e_i}^1, {e_j}^2) = 1, \text{ or} \\ \forall\ sim_k, \text{ with } k \neq expeql, sim_k({e_i}^1, {e_j}^2) = 1 \end{array} \right.$$

   An entity ${e_j}^2$ is semantically equivalent to an entity ${e_i}^1$ such that $({e_i}^1, {e_j}^2) \in \{{R_c}^1 \times {R_c}^2 \cup {R_d}^1 \times {R_d}^2\}$, i.e., $\langle {e_i}^1, \equiv, {e_j}^2 \rangle$, if:
   $$\left| \forall\ sim_k, sim_k({e_i}^1, {e_j}^2) = 1 \right.$$

Table 1. Features and Measures for Similarity

Entities to be compared	No.	Feature (f)	Similarity measure
Concepts: C	1	(label, C1)	$sim_{strsim}(C1, C2)$
	2	(sound (ID), C1)	$sim_{streql}(C1, C2)$
	3	(label, C1)	$sim_{strsyn}(C1, C2)$
	4	(C1,equalTo, C2) relation	$sim_{expeql}(C1, C2)$
	5	(C1,inequalTo, C2) relation	$sim_{expineq}(C1, C2)$
	6	all (direct-sub-concepts, S1)	$sim_{setsim}(S1, S2)$
Relations: R_c	7	(sound (ID), R1)	$sim_{streql}(R1, R2)$
	8	(domain, R1)∧(range, R1)	$sim_{objeql}(R1, R2)$
	9	(domain, R1)∧(range, R1)	$sim_{objineq}(R1, R2)$
	10	all (direct-sub-properties, S1)	$sim_{setsim}(S1, S2)$
Relations: R_d	11	(sound (ID), R1)	$sim_{streql}(R1, R2)$
	12	(domain, R1)∧(range, R1)	$sim_{objeql}(R1, R2) \wedge sim_{streql}(R1, R2)$
	13	(domain, R1)	$sim_{objineq}(R1, R2)$
	14	all (direct-sub-properties, S1)	$sim_{setsim}(S1, S2)$

5. **USEE (Uncertain Semantic Equivalent Entity)**: We extend the definition of SEE to USEE in order to be used throughout the process of handling uncertainty when performing and combining matchers.
   Definition (USEE) . An entity that we said to be uncertain and semantically equivalent to an ontological entity $e \in O_1$ is a pair (Θ, m), where:
   $$\left| \begin{array}{l} \Theta = E,\ E \in \{C^2, {R_c}^2, {R_d}^2\} \\ \text{m is a belief mass function (See Section 1.2.2).} \end{array} \right.$$

1.2.2 Handling uncertainty

The Dempster-Shafer theory of evidence [11] presents some advantages that encourage us to choose among other theories. In particular, it can be used for the problems where the existing information is very fragmented, and so the information can not be modelled with a probabilistic formalism without making arbitrary hypotheses. It is also considered as a flexible modelling tool making it possible to handle different forms of uncertainty, mainly the ignorance. Moreover, this theory provides a method for combining the effect of different beliefs to establish a new global belief by using Dempster's rule of combination.

The belief mass function *m(.)* is the basic concept of this theory ([11], [12]). It assigns some belief mass in the interval [0,1] to each element of the power set 2^{Θ} of the frame of discernment Θ. The total mass distributed is 1 and the *closed world hypothesis* (i.e. m($\emptyset$) = 0) is generally supported. In our work, $\Theta \in \{C^2, {R_c}^2, {R_d}^2\}$. The letter Φ in table 2 is the set of all candidate mappings.

Table 2. Frame of Discernment and Candidate Mappings Set.

	${e_1}^2$	$\dots$	${e_m}^2$	$\Rightarrow \Theta$
${e_1}^1$	$({e_1}^1, {e_1}^2)$	$\dots$	$({e_1}^1, {e_m}^2)$	
$\dots$	$\dots$	$\dots$	$\dots$	Φ
${e_n}^1$	$({e_n}^1, {e_1}^2)$	$\dots$	$({e_n}^1, {e_m}^2)$	

In order to discover USEEs, we use n functions called matchers $(matcher_k)^3$. A matcher compared to a "witness" that brings evidence in favor or against an advanced hypothesis. Matchers produce USEEs in order to support uncertainty. Some matchers are reliable than others. This is reflected in the confidence that is assigned to each matcher. The confidence is expressed like the mass that is distributed to Θ. For instance, if $matcher_1$ has a confidence of .6, then the masses assigned to the subsets should be normalized to sum .6, and .4 should be always affected to Θ.

We use Dempster's rule of combination to aggregate the produced USEEs. Figure 3 illustrates the architecture that we propose to discover USEEs. In addition, this theory makes it possible to express total ignorance. For instance, if the set that contains the entities having the same sound as the entity in question is empty, then the matcher $matcher_2$ will return a belief mass function $m(\Theta) = 1$.

1.3 Adaptations made for the evaluation

Our mapping algorithm has been recently conceived so to speak that our tool OWL-CM is in an alpha version and we evaluate it for the first time.

[3] The index k is the No. of the matcher in the table 1.

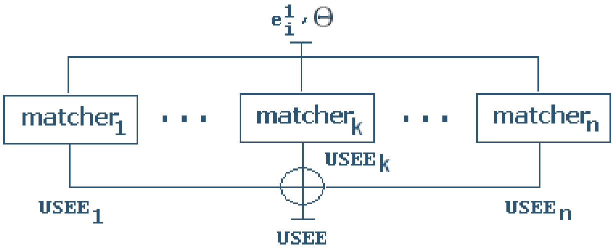

Fig. 3. Architecture for discovering USEEs.

2 Results

The tests have been carried out with the data of the benchmark of Ontology Alignment Evaluation Initiative 2007. Our experiments are restricted to the following metrics that evaluate the goodness of the algorithm output and which are derivatives of well-known metrics from the information retrieval domain [6]: *Precision*, *Recall*, and *FMeasure*. The mapping algorithm has been implemented in java and been updated so that it returns the results in the required format.

2.1 Tests 101-104

Our results (see result Figure 4) show that our mapping algorithm enabled us to achieve 100% precision and 100% recall in the tests 101, 103 and 104. The test 102 also shows the performance of the algorithm.

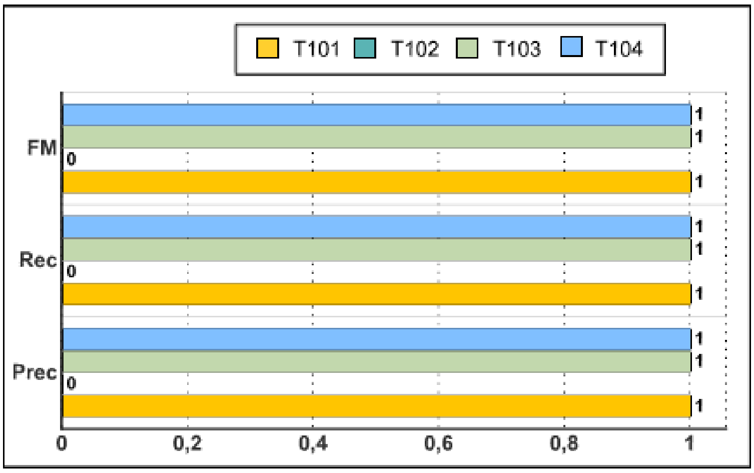

Fig. 4. Results of Tests 101-104.

2.2 Tests 201-204

The ontology 201 does not contain names and the ontology 202 contains neither names nor comments, so we will not consider the results of these tests. In fact, our algorithm considers concept and property IDs (identified by the "*rdf:ID*" tag) as well as their labels (extracted from "*rdfs:label*" tag), therefore the only information that can be used to create these result mappings in the test 201 is comments, but our algorithm does not use it. Although the performed tests are not worth considering, even though they reveal a higher precision (see result Figure 5).
Concerning the tests 203 and 204, our mapping algorithm creates the mapping with high precision (see result Figure 5). Recall values are also considerable.

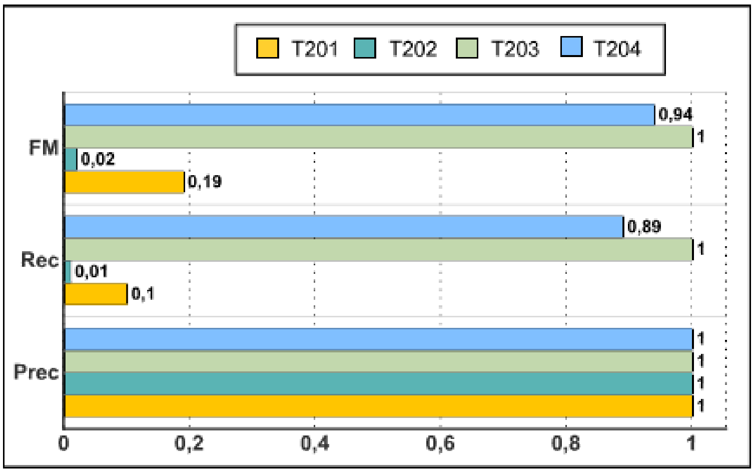

Fig. 5. Results of Tests 201-204.

2.3 Tests 205-210

Before starting the commentary, we note that ontologies 205, 206, 207, 209 and 210 contain doubloons in rdf-ID feature (e.g., there are two datatype properties with the rdf-ID "issue" in the ontologies 205 and 209). However, our algorithm does not allow this as it considers the rdf-ID to be the attribute that identifies the entity in the database during the pre-mapping step. So, in order to don't miss these tests, our algorithm only mapped the parts of ontologies that it was able to convert.
Since our algorithm does not make use of comments in the mapping, we group the tests according to alterations relating to names. Thus we distinguish three behaviors of the algorithm (see Figure 6):

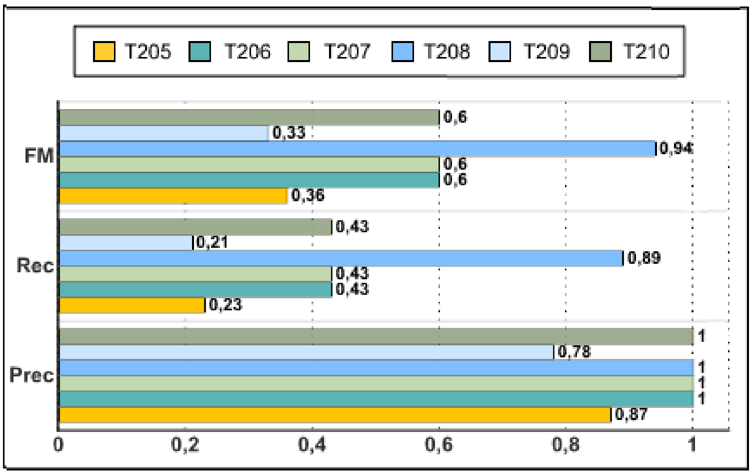

Fig. 6. Results of Tests 205-210.

- Both ontologies 205 and 209 were mapped with good precision but the recall scale is ever such low. Concerning the test case 205 we explain the weakness of the recall by the fact that the searching for Wordnet synonyms, which is the function of some matcher, is made based on full labels. The percentages of precision and recall of the second test case are a bit lower than the ones in the first test case. This note goes to show that the matchers, which deal with labels, have a part in the success of mapping.
- The algorithm generated quite good mappings for the ontologies 206, 207, and 210 with extreme precision and quite satisfactory recall. The results depicted in Figure 6 show that the precision and recall are the same for the three tests which can be explained by some reasons. On the one hand, the fact of keeping or suppressing comments does not have effect on the produced mappings at all as the algorithm doesn't make use of this information. On the other hand, since the labels are translated to French, so the matchers, which deal with labels, are faced with a situation of total ignorance. We conclude that the difference in language between ontologies affects the mapping.
- The test case 208 is similar to the test case 204 where the name of each entity is replaced by another one with different conventions.

2.4 Tests 221-247

Different categories of alteration have been carried out in each of these test cases. The precision and recall percentages of ontology mapping during these tests (see results in Figure 7) are equal or close to 100%. This result confirms that our algorithm takes both syntactic and semantic similarity into account.

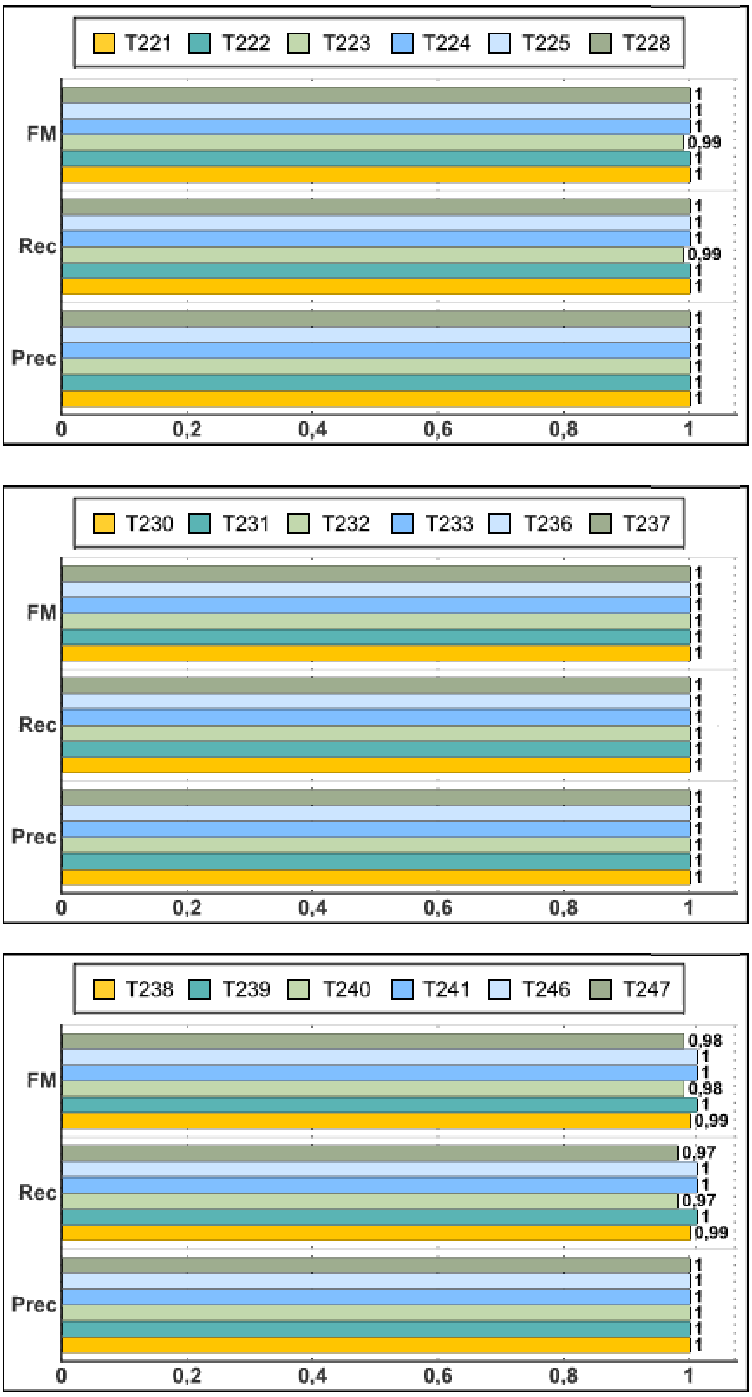

Fig. 7. Results of Tests 221-247.

2.5 Tests 248-266

As names of entities from reference ontology have been replaced by random strings and as our algorithm considers only concept and property IDs as identified by the *"rdf:ID"* tag, the ontologies of these tests were not mapped at all by our algorithm. What's more, the only information that the algorithm can make use of it to create mappings, except in tests 248, 253, 254 and 262, is the specialization hierarchies of classes and properties, which are described, respectively, through the tag "*rdfs:subClassOf*" and the tag "*rdfs:subPropertyOf*". However, since the matchers that make use of this information are complex, therefore these tests have not produced any result mapping.

2.6 Tests 301-304

Before starting the analysis of results, we note that we reduced the three collections of result mappings (col-301: from 61 to 39, col-302: from 48 to 26, col-304: from 76 to 74). This is due to, among other raisons, the fact that the three collections contain some concepts and properties that are matched with the "<" relation while our algorithm only uses the "=" relation.
The result mappings produced by the algorithm are with high precision (see Figure 8). The recall is high for the test 302 and relatively good for the test 304, but the ontology 301 was mapped with weak recall. More in detail, the weakness in the recall of the test 301 is in the mapping of datatype properties. This is due to some reasons that affect the execution of some matchers, such as the difference in the hierarchies between the ontologies in the test 301. Concerning the ontology 303, it was not mapped at all by the algorithm. In fact local entities of this ontology are identified by "*rdf:about*" tag while our algorithm makes use of the tag "*rdf:ID*" to identify local entities and makes use of the tag "*rdf:about*" only when identifying external entities.

3 General comments

3.1 Comments on the results

Since the main goal of this work is to strengthen the precision of the ontology mapping with developing an approach that deals with uncertainty inherent to the mapping process, the means of the three metrics are encouraging (see appendix raw results).

3.2 Discussions on the way to improve the proposed system

The results obtained with our OWL-CM tool turned out to be good, especially as this proposed version of the system is yet an alpha one which is still subject to improvements. In our future work, we will tend to investigate different horizons that we classify into three categories:

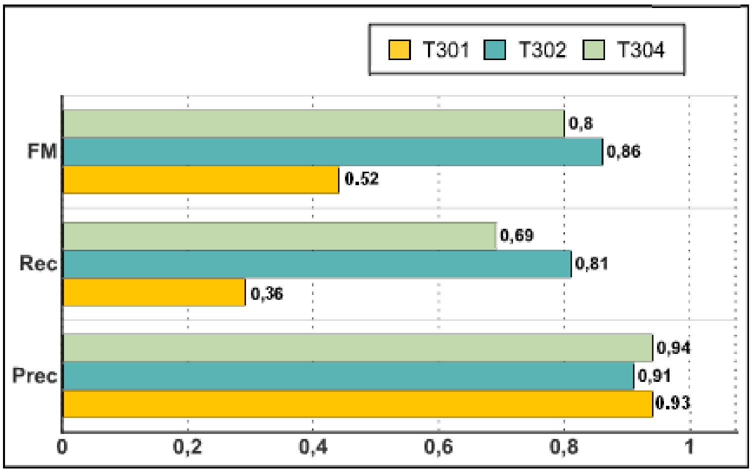

Fig. 8. Results of Tests 301-304.

1. **OWL-CM Improvements**: OWL-CM can be enhanced by different additive elements that were been revealed during the experimental study, i.e. when a full label does not have synonyms, search synonyms based on parts of the label.
2. **OWL-CM Tool Efficiency**: At this time we have exclusively worry about improving the effectiveness of the approach and left the efficiency to be investigated further.
3. **OWL-CM Tool Extensions**: OWL-CM can be extended so that it becomes, among others, able to map ontologies that differ in their language. We can use for example a translation tool.

3.3 Comments on the OAEI test cases

Concerning the benchmark, it is satisfactory since it was served as an experiment bed to assess both strong and weak points of our algorithm and gives an idea of the prospects for improving the algorithm effectiveness. But, it doesn't present some tests to interpret the use of some similarity measures that are based on the explicit assertions such as the following one:
Explicit Equality: it checks whether a logical assertion already forces two entities to be equal. In an OWL ontology, this assertion is expressed by using the axiom “owl:sameAs”. We refer to these assertions as “equalTo”.

$$\mathrm{sim}_{expeql}(a,b) := \begin{cases} 1 & \exists \text{ assertion (a,“equalTo”,b)}, \\ \\ 0 & \text{otherwise.} \end{cases}$$

4 Conclusions

Semantic ontology mapping is an immensely rich area of research. Recently, researchers have brought attention to the fact that the mapping process can be modelled as decision-making under uncertainty. So we have intended to apply handling uncertainty in ontology mapping. We have proposed a new framework called OWL-CM, which sets the foundations for the architecture of discovering mappings under uncertainty. We have designed an algorithm for ontology mapping, based on the guidelines already established, and implemented it. The results obtained with our algorithm turned out to be good. From the experimental study, different horizons have been revealed and can be investigated in our future work.

References

1. Besana P. (2006a), *A framework for combining ontology and schema matchers with Dempster-Shafer*, In Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, USA, pages 196-200.
2. Besana P. (2006b), *Dynamic ontology mapping: a first prototype*, SSP Talks, School of Informatics, University of Edinburgh.
3. Bisson G. (1995), *Why and How to Define a Similarity Measure for Object Based Representation Systems*, In Towards Very Large Knowledge Bases, IOS Press, Amsterdam, pages 236-246.
4. Castano, S., Ferrara, A., and Messa, G. (2006), *Results of the HMatch Ontology Matchmaker in OAEI 2006*, In Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, USA, pages 134-143.
5. Convent. (1986), *Unsolvable problems related to the view integration approach*, In Pro- ceedings of ICDT'86, volume 243 of LNCS, Rome, Italy, Springer, pages 141-156.
6. Do, H., Melnik, S., and Rahm, E. (2002), *Comparison of schema matching evaluations*, In Web, Web-Services, and Database Systems, Springer LNCS 2593, pages 221–237.
7. Ehrig, M. and Staab, S. (2004), *Qom - quick ontology mapping*, The Semantic Web Proceedings ISWC04, Springer-Verlag LNCS 3298, pages 289-303.
8. Hu, W., Cheng, G., Zheng, D., Zhong, X., and Qu, Y. (2006), *The Results of Falcon-AO in the OAEI 2006 Campaign*, In Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, USA, pages 124-133.
9. Nagy, M., Vargas-Vera, M., and Motta, E. (2006), *Dssim-ontology mapping with uncertainty*, In Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, USA, pages 115-123.
10. Niedbala, S.(2006), *OWL-CtxMatch*, In Proceedings of the International Workshop on Ontology Matching (OM-2006), Athens, Georgia, USA, pages 165-172.
11. Shafer G. (1976), *A mathematical theory of evidence*, Princeton University Press.
12. Smets P. (1988), *Belief functions*, In non-standard logics for automated reasoning, Academic Press, pages 253-286.

Appendix: Raw results

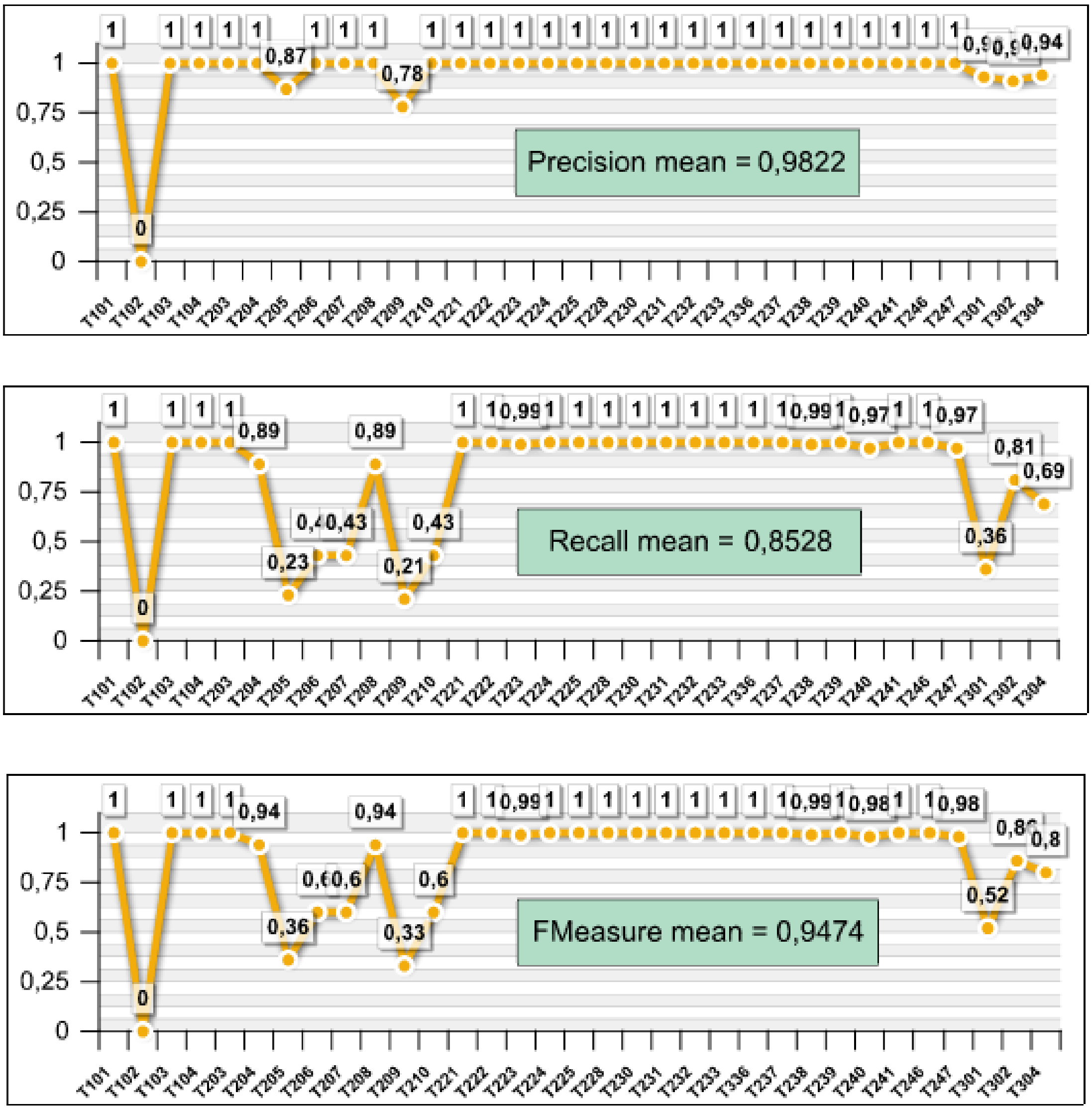

Fig. 9. Overview of Tests' Results.