Using Demster-Shafer for Combining Ontology and Schema Matchers

Paolo Besana

School of Informatics
University of Edinburgh

Abstract Ontologies, at least in the form of taxonomies, have proved rather successful, and are employed in many fields, as far apart as biology and finance. Reaching an agreement over a single ontology, or a set of shared ontologies, has proved difficult, and to obtain actual interoperability it is necessary to map the different ontologies. Mapping one entity between a source ontology and one in a target ontology means to compare the first entity with all the entities in second ontology: matchers analyse different aspects of the entities to identify the similarities. A single matcher can analyse only some aspects, and often has to rely on incomplete, imprecise or vague information. Therefore combining the outcomes of different matchers can yield better results.
In this paper I present a framework that uses Dempster-Shafer as a model for interpreting and combining results computed by the matchers.

1 Introduction

Ontologies have proved to be a powerful tool, and they have become rather common. For example, ontologies in the form of taxonomies are used by Google and Yahoo to categorise websites and by Amazon, eBay and other vendors to classify their products. A more complex ontology, such as GeneOntology, is used by biologists and bioinformaticians to uniformly name biological functions, processes and locations. Other ontologies are used in medicine, to classify illnesses and drugs, in finance, in software engineering and so on.

However, the development and the acceptance of a common ontology has failed to occur, and consequently a number of different ontologies are used. This heterogeneity can be both positive and negative. It is positive, because every ontology represents a different side of reality, and the overall representation of the world is richer. It is negative because it causes difficulties in mutual understanding. To exploit the richness provided by the ontologies it is necessary to build bridges between them. The various attempts to reconcile ontologies can be divided into [9]:

merging: the act of building a new ontology by unifying several ontologies into a single one
aligning or mapping: used when sources must be made coherent and consistent, but must be kept separated

integrating: entails building a new ontology composing parts of other ontologies.

Mapping ontologies lays at the basis of both merging and integration. This paper presents a framework for ontology mapping that uses Dempster-Shafer to interpret and combine the results computed by different matchers.

Section 2 introduces ontologies and the possible mismatches that can occur between them. Section 3 introduces the problem of ontology mapping and discusses a possible classification for the approaches discussed in literature. Section 4 discusses some of the issues that mapping algorithms encounter, and section 5 presents the framework, based on Dempster-Shafer, for tackling some of the issues. Finally, section 6 shows some partial results.

2 Ontology Formalisation

Ontologies specify the terminology used to describe a domain:

> An ontology is a specification of a conceptualisation.[8]

or more in detail:

> An ontology is [...] a description of the concepts that exist or can exist in a particular domain as well as of the relationships in which the various concepts may enter.[8]

According to [16], an ontology is composed by definitions of classes, relations or instances. The definitions of these entities are tuples:

$Def = \langle T, D, C \rangle$

where T is the term that identifies the entity to define (*definiendum* in [16]) and it is an atomic formula in a formal language; D is the formal definition (*definiens* in [16]) and it is a possibly compound formula in a formal language; C is the concept description, obtained in the conceptualisation step, and can be expressed in natural language. For example, using Description Logic as formal language:

T : *Human*
D : $Human \equiv Animal \sqcap Rational$
C : "A human being is a rational animal"

If the ontology is a simple taxonomy of classes, the definition is the hierarchy of the subsuming classes of the entity to define. The concept description C can either be explicitly written in the ontology (for example using the tag `rdfs:comment` in a rdf/owl ontology), or can be an implicit meaning conventionally associated to the term.

Database schemas, like ontologies, provide a vocabulary of terms that describe the domain of interest, and constraint the meaning of terms used in the vocabulary, but usually do not provide an explicit definitions for their data.

According to [16], ontologies can differ because of two main categories of mismatches: *conceptualisation* and *explication* mismatches. The first category of mismatches originates from the initial phase of conceptualisation of the domain. Conceptualisation mismatches include class and relation mismatches: for example, classes can be divided into different subclasses, or attributes can be assigned to different classes. Explication mismatches are caused by differences in the way the conceptualisation is specified in a formal language: for example, there might be ambiguities derived from using the same term to identify different entities, or from using different terms to identify the same entity.

3 Ontology mapping as decision making under uncertainty

An ontology mapping algorithm is a function that receives two ontologies and returns the relations between their concepts:

$$map : O \times O \rightarrow R \tag{1}$$

where R contains all the binary relations (*equivalence, generalisation, specialisation, similarity, disjointness*) between concepts in O_1 and O_2.

Stated otherwise, the function *map* finds the subsets $\Phi_1, \ldots, \Phi_n$ of the Cartesian product $O_1 \times O_2$ that contain the binary relations between the items in the two ontologies:

$\Phi_{equivalence} = \left\{ \left\langle t^a_{O_1}, t^b_{O_2} \right\rangle, \left\langle t^c_{O_1}, t^d_{O_2} \right\rangle \ldots \right\} \subseteq O_1 \times O_2$
$\Phi_{subsumedBy} = \left\{ \left\langle t^g_{O_1}, t^h_{O_2} \right\rangle, \left\langle t^n_{O_1}, t^m_{O_2} \right\rangle \ldots \right\} \subseteq O_1 \times O_2$
...

This is obtained calling a *matcher* function that verifies to what degree μ each pair $\left\langle t^i_{O_1}, t^j_{O_2} \right\rangle$ belongs to the subset Φ_{rel}:

$$matcher \; : \; t \times t \times \Phi_{rel} \rightarrow \mu \tag{2}$$

The problem is how to verify the existence of a particular relation between a pair of terms from two different ontologies. If the ontologies are inconsistent, as it is often the case, it may be impossible to prove the relations using logic reasoning from the definitions in the ontologies: $O_1, O_2 \vdash r\left(t^i_{O_1}, t^j_{O_2}\right)$ may not be derivable or, even worse, wrong relations may be derived. Therefore, mapping algorithms need to use other methods to identify relations between terms in the different ontologies. These methods usually assume that ontologies share some similarities that can be found. For example, the similarities can be in the label used to identify the entities T, in their formal definition D, or in the description (possibly implicit) of the concepts attached to the entities.

The membership of a pair $\left\langle t^i_{O_1}, t^j_{O_2} \right\rangle$ to a subset Φ_i often cannot be precisely stated. This may due to the vagueness or ambiguity of the terms (for instance, the terms may have many different senses, with only a few overlapping), to the lack or the imprecision of the information available in the decision process (for example a term or a sense may not be included in a thesaurus), or to actual differences in the meanings (if the system is looking only for equivalence, *book* and *booklet* are similar, but not completely equivalent).

Not all the relations are interesting. To obtain a working interoperability, some relations are more useful than others (for example, knowing that term t_i is generically related to a term t_j is less useful than knowing that t_i is equivalent to t_k). Moreover, the pair with the highest degree of membership is a more useful - and possibly more correct - mapping than the other pairs with lower degrees: the membership reflects how correct is the relation between the terms.

A general method for finding a mapping between a given entity $t \in O_{source}$ and an unknown entity $t_j \in O_{target}$ is found comparing the given t with all the entities in O_{target}, and keeping the pair that belongs to the most significant relation (for example *equivalence*), with the highest membership degree μ:

$$mapper\ :\ t \times O_{target} \rightarrow \langle t_j, rel, \mu \rangle \tag{3}$$

More sophisticated methods can verify the consistency of the choice, and keep the strongest mapping that does not conflict with other mappings.

Different approaches of ontology mapping in literature can be classified by:

- *the binary relations they search*: some look only for similarity [13], other look for more complex ontological relations [7].
- *the methods they use for taking the decision*: some use only string comparison between the terms, others use thesauruses and compare the semantic similarities of the conventional meaning attached to the terms, others analyse the similarities in the structure of the ontologies [2], others learn to classify the instances of the concepts [5], while most of the recent ones combine these techniques [4,6,7].
- *The type of membership degree they use*, often expressed as confidence level. Some use hard thresholds: the subsets $\Phi_1, \ldots, \Phi_n$ of $O_1 \times O_2$ are crisp sets, and a pair either belongs to the set or does not [7]. Others implicitly consider these subsets as fuzzy sets, and pair can belong to these sets with different degrees of membership [4].

A more detailed review of these approaches can be found in [14,11].

4 Mapping issues

4.1 Combining matchers

A matcher analyses only some aspects of the hypothetical relation between two terms, and may lack or omit important information. For example, comparing strings omits the fact that terms have a conventional meaning attached to them. Similarly, comparing senses of the terms using an external thesaurus omits both the fact that the formal definitions of the terms influence their meaning, and the uncertainty due to the incompleteness of the thesaurus (terms may not be listed, or senses may not be present. This is particularly true when the matcher uses a general-purpose thesaurus such as WordNet for technical expressions)

It therefore becomes important to combine the results from different matchers, in order to exploit all the information available.

4.2 Interpreting matcher's results

To combine the results it is necessary to interpret them in some semantically uniform way. Matchers return different types of results: some return natural numbers, other boolean values, other ratios. For example, EditDistance returns the number of changes necessary to transform one string into another (for example, the distance between *book* and *hook* is 1, between *course* and *curse* is still 1, between *thing* and *book* is 5), while a matcher that check if one string is an infix of another will return boolean values.

A possible interpretation, as described in [4], is to consider the result of a matcher a measure that gives the plausibility of the correspondence between the terms in a pair.

4.3 Indistinguishable results

We have seen in section 3 that to map a term t, a matcher is called to evaluate pairs of terms from $t \times O_{target}$. However, it may often be the case that a matcher cannot distinguish between pairs: for example, EditDistance will return the same result "1" for $\langle rate, race\rangle$,$\langle rate, rave\rangle$, $\langle rate, rage\rangle$, etc. According to the previous subsection, the interpretation for this outcome is that the the pairs must have the same plausibility.

Moreover, results that are close enough can be interpreted as sharing the same, or a very similar, plausibility level. For example, it is not meaningful to assign a different confidence to a pair with a distance of 5 and another one with distance 6: both are unlikely to be the mapping. Thus, it is possible to define intervals whose internal values correspond to the same confidence level.

4.4 Ignorance and Reliability

A matcher may also be unable to give evaluation for a pair, as it lacks information: in this case, all hypotheses are equally probable. For example, a matcher that uses a thesaurus will not be able to evaluate a pair if one of the two terms is not listed in its dictionary.

Matchers may also have different degrees of reliability: if EditDistance maintains that two terms are equivalent, the assertion can be false because the terms can be homonyms. The reliability measures how probable is that an assertion made by a matcher is correct [3].

5 A mathematical framework to combine the matchers

There are different mathematical theories that can be used as a framework for a system that must handle the uncertainty issues discussed in section 4, among which the Bayesian approach and Dempster-Shafer are the strongest candidates. However, Dempster-Shafer [17] is particularly adapt to tackle them: using this theory to model the mapping process it is possible to give a uniform interpretation, consistent with the uncertainty inherent in the problem, to the results of the matchers and to combine them in a mathematically sound way.

Dempster-Shafer has been considered for different uses, such as medical diagnosis [1], robot navigation through image processing [10], and it has been proposed for ontology mapping for query answering in [12].

While in the standard Bayesian approach, probabilities are assigned to single entities, in Dempster-Shafer the mass is distributed on *sets* of propositions. The mass distribution is a function $m(\cdot)$ that distributes a mass in the interval [0,1] to each element of the power set 2^{Θ} of the set of propositions $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ called *frame of discernment.* The total mass distributed is 1 and the *closed world assumption* is generally made: the frame Θ contains the true hypothesis. This is expressed assigning mass 0 to the empty set $\emptyset$, called contradiction. The mass $m(\Theta)$ assigned to the frame represents the mass that is not possible to assign to any particular subset of Θ. Different mass distributions can be combined using *Dempster's rule of combination* that computes the probability mass assigned to $C \subseteq \Theta$ given $A \subseteq \Theta$ and $B \subseteq \Theta$, where A is supported by m_1 and B is supported by m_2.

The model is applied to the function in expression 3 that searches an unknown entity t_j from an ontology O_{target} that best matches a given term t in an ontology O_{source}. The frame of discernment Θ of the problem becomes the Cartesian product $t \times O_{target}$, where each proposition is a pair $\langle t, t_i \rangle$.

5.1 Interpreting the results

In Dempster-Shafer, mass assigned to a proposition means support to the belief that the proposition is true. In this model, the matcher is considered an "expert" that uses a specific method to analyse a pair and gives an opinion about the similarity of the terms. The similarity measure must be converted into a measure of the belief in the correctness of the mapping.

As we have seen in section 4.3, the conversion is made considering that the matcher cannot distinguish pairs of terms that yield the same similarity results and it may be indifferent to pairs with similar results.

Therefore, the range of possible results of a matcher is split into intervals. An interval i_k corresponds to a mass m_k: pairs whose results fall into the interval are grouped in the same set s_k, and the belief in the fact that the correct mapping belongs to the set s_k is given by m_k. For example, for EditDistance the intervals, coupled with their masses, are:

$$I_{EditDist} = \{\langle [0,0], 0.48\rangle \, \langle [1,2], 0.3\rangle \, \langle [3,4], .14\rangle, \langle [4,5], 0.08\rangle, , \langle [5,..], 0.0\rangle\}$$

Figure 1 A shows how the pairs (marked by h_j) are divided into sets corresponding to the results of the matchers, and how sets generated by different matchers overlap.

Representing Belief in contradiction The closed world assumption in Classical Dempster-Shafer theory implies that the correct mapping must be in the frame of discernment $\Theta = t \times O_{target}$. However, it is possible that there is no proper mapping in O_{target} for $t \in O_{source}$. To make the theory consistent with the reality, we need to reject the closed world assumption and accept the open world one. This assumption will be required when combining the results, as will see in section 5.2.

Representing Ignorance and Reliability We have seen in section 4.4 that a matcher may lack the information needed to evaluate correctly a pair. For example, if the matcher has no information about the term $t \in O_{source}$ it is not able to evaluate any pair in the frame of discernment. In this case, none of the mass is allocated, and it should be transferred to the frame of discernment Θ. We have also seen, still in section 4.4, that the outcomes should be handled with care, as matchers can have different degrees of reliability. For example, EDITDISTANCE may wrongly support a particular set of pairs. The mass distributed by a matcher should be discounted by a reliability factor specific for the matcher. The discounted mass becomes unallocated mass, and should be interpreted as ignorance and transferred to the frame of discernment Θ.

Matcher interface The framework is independent of the the matchers used: they are considered as a function that compare a pair and they can be plugged in the framework. Their results are interpreted using an interface layer, that converts them into mass distributors. A matcher interface *MI* is the tuple:

$$MI_{matcher} = \langle I_{matcher}, \rho_{matcher} \rangle$$

where $I_{matcher}$ is the set of intervals $\{\langle r_o, m_1 \rangle, \ldots, \langle r_n, m_n \rangle\}$ of the results range with the corresponding mass, and $\rho_{matcher}$ is the reliability of the matcher, used to discount the distributed masses.

The intervals and their masses can be computed running the matchers over known mappings and counting how often the result for a correct mapping falls inside the intervals. The reliability of a matcher can be computed using a separate set of known mappings and counting the frequency of the errors made by the matcher in recognising the match.

5.2 Combining the results

The mass distributions generated by the matchers are combined using *Dempster's rule*:

$$m\left(C\right)=\frac{\sum\limits_{A\cap B=C}m_1\left(A\right)m_2\left(B\right)}{1-\sum\limits_{A\cap B\neq\emptyset}m_1\left(A\right)m_2\left(B\right)} \tag{4}$$

One of the main problems that arises using Dempster's rule is that normalisation can yield counterintuitive results when combining contradictory evidences [18]. The judgements made by the matchers can often be contradictory: for example, two entities can have very dissimilar strings, but can be easily recognised as synonyms by a thesaurus.

A possible solution is to avoid the normalisation. This means to drop the closed world assumption [15] by making $Bel\left(\emptyset\right)\neq 0$ possible. As we have seen in section 5.1, this is a useful feature for the process we need to model.

Figure 1 B shows how the sets obtained from different matchers are combined using Dempster's rule to generate a new mass distribution.

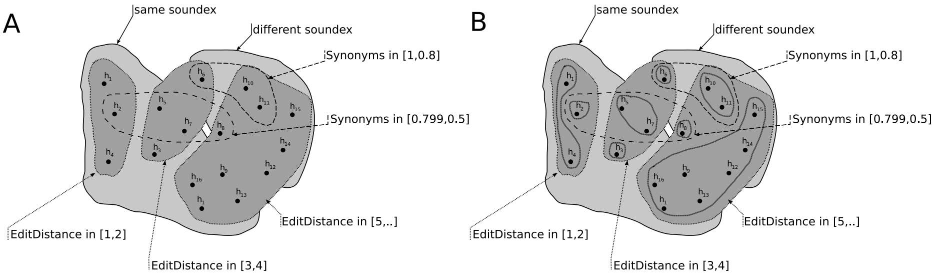

Figure1. Combining results

5.3 Choosing the mapping

Once the masses have been distributed and combined, it is necessary to extract the most likely entity from the mass distribution. Dempster-Shafer makes it possible to compute the *belief* about a set $A\subseteq\Theta$ of propositions, as the sum of all the basic masses that support its constituents:

$Bel(A)=\sum_{B\cap A}m(B)$

It also provides the formula for computing the *plausibility* of the set A, that is the measure of the extent to which A might be true:

$Pl(A)=1-Bel(\overline{A})=\sum_{B\cap A\neq\emptyset}m(B)$

The plausibility forms the upper bound for the belief in A. In some interpretation, the interval $[Bel(A),Pl(A)]$ is the ignorance about A, as in figure 2.

In the current version of the framework, belief and plausibility are computed for singletons, as it is exponentially complex to compute them for different combinations of sets.

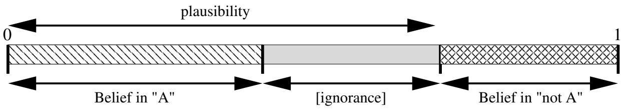

Figure2. Interpretation of Dempster-Shafer

Currently, the best mapping is chosen ordering the pairs by plausibility, and discarding all the pairs with plausibility and belief below an arbitrary threshold, and with ignorance higher than another arbitrary threshold. This thresholding guarantees that pairs with very high plausibility, but very low belief are discarded.

6 Testing

The framework described in this paper is independent of the matchers used: other ontology matching systems could be used as specific and expensive matchers. However, to test the general concept, the algorithm has been implemented and it is available at the address:

`http://pyontomap.sourceforge.net`

In the current version the matchers in table 1 have been implemented. The tests were executed comparing two pairs of ontologies, after manually creating the mappings between their entities for comparison.

The first pair of ontologies belongs to the benchmark in the Ontology Alignment Evaluation Initiative (`http://oaei.ontologymatching.org/2006/`) : the first ontology is the number 101, while the second is number 205 and replaces the terms in 101 with synonyms. Both ontologies have about 100 entities.

The second pair of ontologies were created for experiments of interaction between agents. The taxonomy and the properties are taken manually from Amazon and eBay. Both ontologies consist of about 150-200 terms, and are available at the url `http://pyontomap.sourceforge.net/testing.`

The tests were run using different sets of matchers. The first battery of tests, shown in figure 3, was run disabling the mass redistribution for less reliable matchers (see section 5.1), while the second battery, shown in figure 4, was run with the mass distribution. Although the outcomes are provisional and the matchers need calibration, it is possible to notice that combining the matcher improve both precision and recall, and that taking into account the different reliability of different matchers improve the results.

7 Conclusion

In this paper I have discussed the issues that ontology mapping systems must address, and I have proposed a generic framework that allows to combine different matching algorithms. The framework is independent of the actual matchers

String based matchers: verify the similarity of the labels used for the entities.

- **EditDistanceMatch:** counts the number of changes required to transform a string into another one
- **InfixMatch:** checks if a string is inside another one ("GPS" matches "Eletronics, GPS and Cameras")
- **PrefixMatch:** checks if one string starts with the other one ("Mac" matches "MacOS")
- **PostfixMatch:** checks if one string ends with the other one ("OS" matches "MacOS")
- **InitialsMatch:** checks if the initials in one string match the other string ("Operating_System" matches "OS")

Conventional Meaning based matchers: use an external thesaurus to compare the meaning conventionally attached to the label

- **WordNetMatch:** counts the number of intersecting senses between the terms, its synonyms, their direct hypernyms and hyponyms.

Ontology Structure based matchers: verify how similar are the entities surrounding the two entities in their ontologies.

- **AncestorMatch:** checks how many ancestors are similar. Similarity is computed using string based matchers
- **ChildrenMatch:** checks how many children are similar.
- **SiblingMatch:** checks how many siblings are similar
- **RoleMatch:** checks if the entities have the same role (class, property or instance) in the two ontologies

PastMappingMatch: uses mapping approved by an external reviewer.

Table1. Implemented matchers

used. The main result of the framework is to give a consistent interpretation to results returned by different matchers and to provide a mechanism for combining them. The framework's implementation is under development, and uses an *ad hoc* set of matchers, and while the results are still provisional and need improvement, the framework behaviour is consistent with the goals.

References

1. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, chapter 13, pages 272–292. Addison-Wesley, 1984.
2. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *ICDE '02: Proceedings of the 18th International Conference on Data Engineering (ICDE'02)*, page 117, Washington, DC, USA, 2002. IEEE Computer Society.
3. Paolo Besana, Dave Robertson, and Michael Rovatsos. Exploiting interaction contexts in p2p ontology mapping. In *P2PKM*, 2005.
4. Hong Hai Do and Erhard Rahm. Coma - a system for flexible combination of schema matching approaches. In *VLDB*, pages 610–621, 2002.

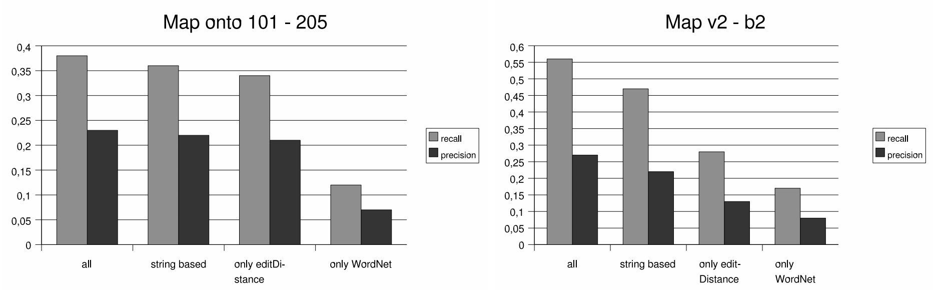

Figure3. Mapping statistics

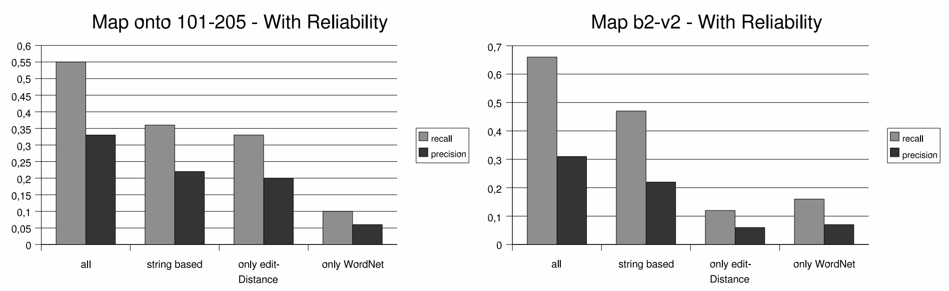

Figure4. Mapping statistics with reliability

5. AnHai Doan, Jayant Madhavan, Robin Dhamankarse, Pedro Domingos, and Alon Halevy. Learning to match ontologies on the semantic web. *The VLDB Journal*, 12(4):303–319, 2003.
6. Marc Ehrig and Steffen Staab. Qom - quick ontology mapping. In *International Semantic Web Conference*, pages 683–697, 2004.
7. Fausto Giunchiglia, Mikalai Yatskevich, and Enrico Giunchiglia. Efficient semantic matching. In *ESWC*, pages 272–289, 2005.
8. Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
9. Adil Hameed, Alun Preece, and Derek Sleeman. *Ontology Reconciliation*, pages 231– 250. Springer Verlag, Germany, 02 2003.
10. Andress K.M.-Lopez-Abadia C Caroll M.S. Kak, A.C and J.R Lewis. Hierarchical ecidence accumulation in the pseiki system. In *Uncertainty in Artificial Intelligence 5*. Elsevier Science Publishers (North-Holland, 1990.
11. Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. *Knowledge Engineering Review*, 2003.
12. Enrico Motta Miklos Nagy, Maria Vargas-Vera. Ontology mapping with domain specific agents in aqua. In *1st Workshop on End User Aspects of the Semantic Web, Heraklion, Crete*, pages 69–83, 2005.
13. Silva Nuno and Joao Rocha. Mafra - an ontology mapping framework for the semantic web. In *Proc. of the 13th European Conf. on Knowledge*, 1999.
14. Pavel Shvaiko and Jérôme Euzenat. A survey of schema-based matching approaches. pages 146–171, 2005.

15. P. Smets. The combination of evidence in the transferable belief model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(5):447–458, 1990.
16. Pepjijn R. S. Visser, Dean M. Jones, T. J. M. Bench-Capon, and M. J. R. Shave. An analysis of ontological mismatches: Heterogeneity versus interoperability. In *AAAI 1997 Spring Symposium on Ontological Engineering*, Stanford, USA, 1997.
17. Ronald Yager. *Advances in the Dempster-Shafer Theory of Evidence.* John Wiley, New York, 1994.
18. L.A. Zadeh. Review of shafer's a mathematical theory of evidence. *AI-Magazine*, (5):81–83, 1984.