Computers in Human Behavior 30 (2014) 745–752

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Establishing agent trust for contradictory evidence by means of fuzzy voting model: An ontology mapping case study

Maria Vargas-Vera [a,*], Miklos Nagy [b]

[a] Facultad de Ingenieria y Ciencias, Centro de Investigaciones en Informatica y Telecomunicaciones, Universidad Adolfo Ibanez, Vinia del Mar, Chile
[b] The Open University (OU), Milton Keynes, UK

ARTICLE INFO

Article history:
Available online 5 November 2013

Keywords:
Ontology mapping
Semantic Web
Multi-agent systems
Uncertain reasoning

ABSTRACT

This paper introduces a novel trust assessment formalism for contradicting evidence in the context of multi-agent ontology mapping. Evidence combination using the Dempster rule tend to ignore contradictory evidence and the contemporary approaches for managing these conflicts introduce additional computation complexity i.e. increased response time of the system. On the Semantic Web, ontology mapping systems that need to interact with end users in real time cannot afford prolonged computation. In this work, we have made a step towards the formalisation of eliminating contradicting evidence, to utilise the original Dempster's combination rule without introducing additional complexity. Our proposed solution incorporates the fuzzy voting model to the Dempster–Shafer theory. Finally, we present a case study where we show how our approach improves the ontology mapping problem.

1. Introduction

High quality semantic meta data is a crucial part of the envisioned Semantic Web. However, the possible number of applications (Memon & Khoja, 2009) that can be developed on the Semantic Web heavily relies on how data can be integrated from distributed and heterogeneous data sources. There are several challenges, which have been identified in the context of ontology mapping by Shvaiko and Euzenat (2008) and Euzenat and Shvaiko (2007). These challenges are considered as roadblocks for developing real applications and they are explained in Section 3. Conflicting information that is inherent to interpreting Semantic Web data is one of these challenges. This conflict can be a result of insufficient or contradicting information of different terms that are similar or even the same. For example, consider two ontologies, which describe scientific publications. Both of these ontologies describe the concepts "paper". In ontology 1 the paper is represented as "Scientific Paper" in the context of "Conference participant" and in Ontology 2 as "Chapter" in the context of "Book". When mapping algorithms extend these contexts using any kind of background knowledge e.g. WordNet sister terms, one can derive that both describes a printed work of someone. The problem is though that this knowledge cannot directly be deduced from the ontologies, because "Scientific Paper" refers to participant, while "Chapter" refers to portion of the book, that has been published. Naturally human experts could easily resolve this contradiction through discussing their point of views and decide if the mapping can be made or not. However, this is not the case for ontology mapping applications that operate without human intervention. In the case of applications, sufficient conflict resolution processes need to be in place, to improve the quality of the mappings by eliminating the contradictions. In this paper, we propose a conflict elimination approach using a fuzzy voting model. Based on our initial approach for eliminating conflicts (Nagy, Vargas-Vera, & Motta, 2008), we propose different fuzzy variables, membership functions and a customised voting algorithm in order to provide more reliable results. The fuzzy voting model allows to detect and eliminate contradictory evidence, instead of discarding the whole scenario or combining them with contradictions. These contradictions can occur on any entities in the ontology e.g. classes, objects, data properties and instances. The main contribution of this paper is that it proposes a conflict elimination method, based on trust and fuzzy voting, before any conflicting beliefs are combined.

The paper is organised as follows. In Section 2 we present the related work. Section 3 describes why conflicts occur in the ontology mapping context and our proposed solution is explained in Section 4. In order to validate our approach we have carried out experiments, which is presented in Section 5. Finally in Section 6 we describe the future research directions and the conclusions of our work.

2. Related work

Managing contradictions in the context of ontology mapping in general has lead to different approaches. We review the most relevant approaches that were also identified as state-of-the-art (Shvaiko & Euzenat, 2013), before we introduce our approach.

* Corresponding author.
E-mail addresses: maria.vargas-vera@uai.cl (M. Vargas-Vera), m.nagy@open.ac.uk (M. Nagy).

0747-5632/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.chb.2013.10.032

Since our approach eliminates the contradictions before the judgement of the mapping is established, the relevant work on this area can be found in other ontology mapping approaches. Therefore, in our scenario the test bed that we have used is the OAEI-2008 datasets. The biggest challenge was how to compare our approach with other solutions. It is clear that the only qualitative comparison on the mapping system level can only be made through the Ontology Alignment Initiative, which is an international effort to compare ontology mappings systems. Different approaches to eliminate contradictions for ontology mapping have been proposed by the ontology mapping community. These approaches can be classified into two distinct categories.

First group include solutions that consider uncertainty and fuzziness as an inherent nature of the ontology mapping and tries to describe it accordingly. Ferrara et al. (2008) model the whole ontology mapping problem as fuzzy where conflicts can occur therefore, their approach models the whole mapping process as an uncertain reasoning task, where the mapping results need to be validated at the end of the reasoning process. The reasoning is supported by fuzzy Description Logic approaches. As a consequence, their mapping validation algorithm interprets the output mapping pairs as fuzzy and tries to eliminate the inconsistencies from them.

Tang et al. (2006) formalise the ontology mapping problem as making decisions on mappings using Bayesian reasoning. Their system *RiMOM* Tang et al. (2006) has participated in the OAEI competition as well. Their solution do consider two kinds of conflicts in the ontologies, namely the structure and naming conflicts. However, they use thesaurus and statistical techniques to eliminate them before combining the results. *RiMOM* approach produces ontology mapping using well-defined processing steps like *ontology pre-processing*, *strategy selection*, *strategy execution* and *alignment combination*. *RiMOM* has been very successful during OAEI competitions; however, its strategies have to be defined in advance together with their rules, which are selected during execution time. As a result, it is questionable how the system can be adapted to the Semantic Web environment, where domains can change dynamically. Furthermore, the assumption that ontologies with *similar features* are similar in reality might not be valid in all cases. Another weak point is that large ontologies cannot easily be loaded into the internal model and the approach does not consider optimisation for the mapping process. Nevertheless, the main idea is remarkable since it builds up its own structure and, hence, tries to interpret the ontology before processing it.

The second group, however, differ conceptually because they mainly utilise data mining and logic reasoning techniques in pre and post processing stages of the mapping.

For example, Liu, Wang, and Wang (2006), split the ontology mapping process into four different phases. Their approach first exploits the available labels in the ontologies then it compares the instances. After it recalls mappings from the previous mapping tasks and compares it with the structure of the ontologies. Their approach also tries to eliminate contradictions, using the previous experience and data mining techniques on the relations that are defined on the ontologies.

Similar solution has been proposed by the Jean-Mary, Shironoshita, and Kabuka (2009) and Jean-Mary and Kabuka (2008). *Automated Semantic Mapping of Ontologies with Validation (ASMOV)* automates the ontology alignment process using a weighted average of measurements of similarity along four different features of ontologies, and performs semantic validation of resulting alignments. This system acknowledges that conflicting mappings are produced during the mapping process but they use an iterative post processing logic validation in order to filter out the conflicting mappings.

Anchor-Flood (Seddiqui & Aono, 2009), is an ontology mapping tool conceived in the context of the *International Patent Classification (IPC)*. The mapping approach itself was designed to work as part of a *patent mining* system that assigns patent abstracts to existing IPC ontologies and it also uses a multi-phase approach to create the mapping results. These phases are *pre-processing*, *anchoring*, *neighbouring block collection* and *similarity measures*. Anchor-Flood also uses an internal representation form to which the ontologies are transformed before processing. The system is also reliant on the availability of individuals, which might not be always present in real life scenarios. There are also a number of weaknesses that are related to the fact that the approach is highly dependent on the correctness of the initial anchoring. Inconsistencies might not be eliminated and missed links might not be discovered it they do not fall into the context of already linked entities.

TaxoMap (Hamdi, Niraula, & Reynaud, 2009) is an approach that is based on the assumption that large scale ontologies contain very extensive textual descriptions and well defined class structures but do not contain a large number of properties or individuals. The similarity assessment uses various *Natural Language Processing* techniques and frameworks like *TreeTagger* Schmid (1994) and *structural heuristic-based* similarity algorithms like *Semantic Cotopy* (Ehrig, Koschmider, & Oberweis, 2007). In order to filter out inconsistent mappings, it uses a so-called *refinement module*. End users have the possibility to define constraints and solutions using a logic-based language called *Mapping Refinement Pattern Language (MRPL)*. For example, this language allows the end users to express domain specific constraints that can remove a mapping pair on condition that the classes involved in the mapping do not have an equivalence relation in the source or target ontology. One weakness of the system is that it requires the fine-tuning of nine different threshold values, which is a challenge given the possible combinations and the possible impacts on the result set.

Lily (Wang & Xu, 2009) mapping approach carries out the mapping in different phases. These phases are *pre-processing*, *match computing* and *post processing*. In the last phase, the system extracts the final mapping set based on the similarity assessments, and then verifies that inconsistent mappings are indicated to the user, who can remove them manually. It is important to point out that the mapping approach recognises the fact that the interpretation of the ontologies involves dealing with uncertainty. However, the objective is only to reduce the amount of uncertainty instead of dealing or reasoning with it. As a result, the mapping process only reduces the negative effect of the matching uncertainty. Lily can also deal with large-scale ontology matching tasks thanks to its scalable ontology matching strategy.

3. Conflicting and trust related to Semantic Web data

3.1. Sources of conflicts

As we briefly mentioned earlier (in the paper), in the context of ontology mapping different challenges had been recognised by Shvaiko and Euzenat (2008). These challenges are viewed as roadblocks for implementing ontology mapping applications that can be applied with high confidence in different contexts i.e. real world domains. We have chosen two, which we believe are mostly related to problem of contradictions. In these two cases, the ontology mapping systems have to establish a certain degree of understanding of the meaning of the data that is present in the different ontologies.

Firstly, the uncertainty related to different representations stems from the fact that W3C has proposed different languages that can be used on the Semantic Web e.g. RDF(S),[1] OWL[2] and SKOS.[3] The problem is that ontology engineers can choose any

[1] http://www.w3.org/RDF/.
[2] http://www.w3.org/TR/owl-features/.
[3] http://www.w3.org/TR/skos-reference/.

language, depending on their application requirements. As a result any attempt to find matching between terms expressed in two different ontologies will end up in situations where uncertain and conflicting hypotheses need to be processed, to find correspondences between classes and properties of these ontologies.

Secondly, Semantic Web data quality issues need to be considered in any application context. In these cases, a manual input from the ontology designer is hard to foresee therefore, applications itself need to be able to process low quality information when establishing a certain degree of understanding, through defining different hypotheses related to the meaning of this data.

3.2. Trust in beliefs

Trust in general is in the focus of research from different fields of computer science. As a result, different research communities have come up with different definitions (Artz & Gil, 2007), depending on the use and representation of this trust. Considering multi-agent systems that operate in the Semantic Web environment, trust plays a very important role. During communicating results, agents need to decide if the information that is coming from an another agent can be trusted or not given a particular scenario. As a consequence, the interpretation of opinions like in our case beliefs in similarities between terms, can easily contradict and in these cases, agents need to establish, which other agents' belief can be accepted i.e. trusted. Such trust has been defined differently by various researchers (Grandison & Sloman, 2000; Mui, Mohtashemi, & Halberstadt, 2002; Olmedilla, Rana, Matthews, & Nejdl, 2005), however, we have adapted into our ontology mapping context the most relevant definition from Olmedilla et al. (2005). According to this adapted definition, in a multi-agent ontology mapping system, trust in the beliefs of mapping agents represent the amount of support that mapping agent 1 can assign to mapping agent 2 for each proposed mapping pair, given a concrete belief, similarity and belief difference.

In multi-agent environments that create ontology mappings, trust can have its own meanings. For example, we can define trust as a way to accept the reliability of information sources given the quality of information and background knowledge the different agents have used. As a result, this trust definition incorporates the concepts of both machine understanding and Semantic Web data quality, for mapping agents that need to propose a mapping between concepts collectively. Additionally, trust is important concept for mapping agent collaboration, because each proposed mapping is based on a subjective trust evaluation related to other agents'' belief.

Our approach involves a voting mechanism to establish these trusts, without carrying out complex investigations i.e. without creating additional computational burden.

4. Introducing fuzzy voting into ontology mapping

4.1. Fuzzy voting definition

Voting in order to eliminate conflicting opinions and decide on a commonly agreed solution, has been advocated in the context of political science Austen-Smith and Banks (1996) and Young (1988) long before it appeared in the context of computer science. The very same ideas can be transplanted to a multi-agent environments, where agents need to come up with a collectively accepted solution. For example, during the ontology mapping process, various similarity algorithms will produce different similarity measures. These differences will result in dissimilar degrees of belief that can be conceived as alternatives for the proposed mapping scenario. The difficulty coming up with an agreed solution lies in the fact, that these beliefs cannot easily be associated with past experiences or the reliability of the similarity algorithms. Therefore, we propose a voting mechanism that facilitates the process of reaching such agreement by evaluating trust in the established beliefs. Voting is not only a very effective way to reconcile differences, but also a mechanism that can be used to represent the collective preferences of individuals or software agents. Unfortunately, given different input variables, stating that a particular belief can or cannot be trusted is not always evident. The difficulty is associated with the fact that each voter's opinion is derived from subjective probabilities over similarities. In such situations, the trust will always involve a certain level of vagueness because the dividing line between trust and distrust cannot be fixed for all cases in the mapping process. In addition to this, there is no easy way of describing a belief in similarity as highly or less trustful. Therefore, the most appropriate representation of trust for each voter in this context is a fuzzy model as depicted in Fig. 1. Before evaluating trust in other agent's belief in the correctness of the mappings, each agent needs to calculate the difference between its own and others beliefs. Depending on the belief difference, the mapping agents can choose a trust level. For example, if the difference is 0.7 then the available trust level can be medium or low. In the proposed mapping system, trust is modelled as overlapping fuzzy membership output functions. Similarly to any fuzzy systems, each membership function (belief, belief difference or similarity) $\mu(x)$ is defined on the possible set of values U, where the membership function shows the degree of membership (between 0 and 1) of a particular input value. In order to carry out the fuzzy voting, we have defined the following fuzzy elements.

4.1.1. Input and output variables

For resolving conflict correctly during the ontology mapping, the proper selection of input and output variables is important. Given the fact that the best membership function can only be selected based on experiments with the given fuzzy system, a series of experiments have been executed with triangular, trapezoidal and gauss shaped membership functions. Fig. 1 shows inputs (belief difference, belief and similarity for mapping agents) with triangular and gauss output (trust) membership functions. The proposed fuzzy sets are described with their membership functions. They are overlapping in order to allow sufficient value selection for each input. Linguistic terms describing the fuzzy variables like "low similarity" ensure that complex rules can be applied if necessary. For example, when a mapping agent belief in similarities is 0.5 then, a voting agent whose membership functions are depicted in Fig. 1, can assign both "strong" and "weak" linguistic values. Analogically, when "similarity" between terms is 0.12, then a voting agent whose membership functions are depicted in Fig. 1 can only assign "low" linguistic value for this variable.

These input variables and their linguistic representations are defined as follows.

Definition 1. Similarity of entities in two different ontologies is defined as a numerical measure that is produced by a certain metrics. These metrics can vary from simple syntactic string distances to more complex semantic sub-graph comparisons. We propose three values for the similarity fuzzy membership value $\xi(x) = \{low, average, high\}$.

Definition 2. Belief in the correctness of the mapping is an input variable, which describes the amount of justified support to A that is the lower probability function of Dempster, which accounts for all evidence E_k that supports the given proposition A.

$$belief_i(A) = \sum_{E_k \subseteq A} m_i(E_k) \tag{1}$$

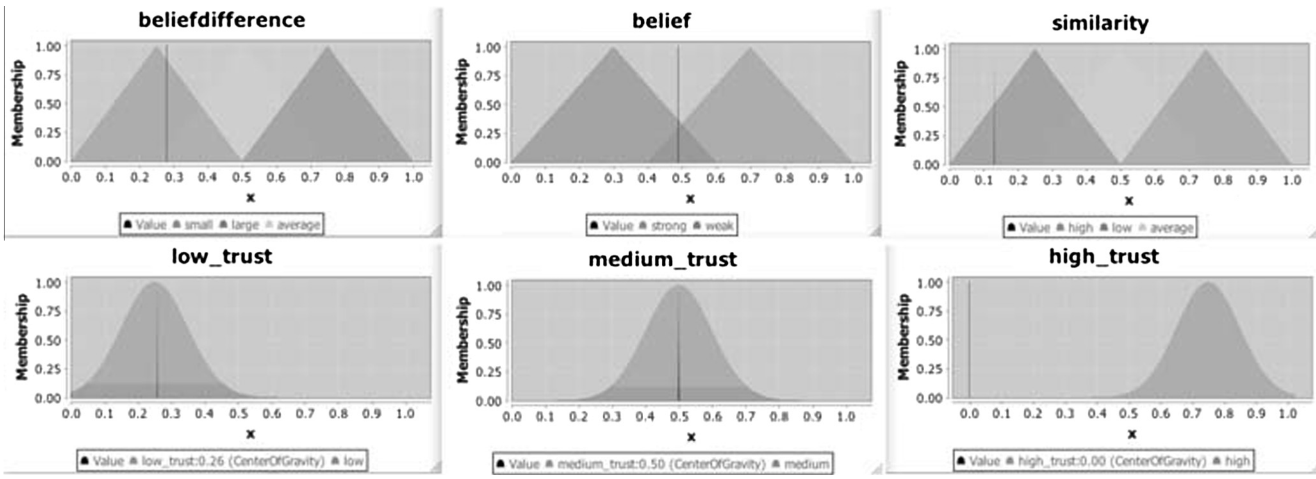

Fig. 1. Example fuzzy membership functions.

where *m* Demster belief mass function represents the strength of some evidence i.e. $m(A)$ is our exact belief in a proposition represented by *A*. The similarity algorithms itself produce these assignments, based on different similarity measures. We propose two values for the fuzzy membership value $v(x) = \{weak, strong\}$.

Definition 3. Belief difference is an input variable, which represents the agents' own belief over the correctness of a mapping, to establish mappings between concepts and properties in the ontology. During conflict resolution, we need to be able to determine the level of difference. We propose three values for the fuzzy membership value $\mu(x) = \{small, average, large\}$.

Definition 4. Low, medium and high trusts in other agents' belief are output variables and represent the level of trust we can assign to the combination of our input variables. We propose three values for the fuzzy membership value $\tau(x) = \{low, medium, high\}$.

4.1.2. Fuzzy rules

The rules of a fuzzy system define how the system should combine the available inputs to produce the necessary output. As such the fuzzy systems use *condition* → *action* rules i.e. rules that can be expressed in *If* ⟨*condition*⟩ *Then* form.

For our conflict resolution problem, we have defined four simple rules (Fig. 2) that ensure that each combination of the input variables produce output on more than one output i.e. there is always more than one initial trust level assigned to any input variables. For example, rule number three defines that the trust is high when the belief difference is either small or average, the belief is strong and the similarity is either high or average.

4.1.3. Defuzzification

Once the reasoning step has been finished, the fuzzy variables need to be converted back to quantifiable outputs (i.e. real numbers, not fuzzy sets). This step ensures that the system will come up with a meaningful number, like probability of the trust that best describes the output fuzzy values. In practice, defuzzification of the trust is the opposite of fuzzificating belief differences and the last step in the process. There are various methods for achieving this defuzzification. We have opted for the centre of area method, because it calculates an weighted average of the output membership functions. Therefore, in the proposed system the trust levels are proportional with the area of the membership functions, while other defuzzification methods like Center of Maximum or Mean of Maximum do not correspond well to our requirements.

Definition 5. For representing trust in beliefs over similarities we have defined three membership functions, $\tau(x) = \{low, average, high\}$.

The main objective of the proposed fuzzy voting mechanism, is to eliminate conflicts that can be encountered using the belief functions. This conflict can be detected when one belief supports a particular mapping hypothesis, but an another belief does not support it, or supports an another hypothesis. Consider for example a situation, where three agents have used WordNet as background knowledge to build their beliefs considering different concepts context. Each belief was derived from a background knowledge e.g. agent 1 used the direct hypernyms, agent 2 the sister terms and agent 3 the inherited hypernyms. Based on string similarity measures, a numerical belief value is calculated, which represents the strength of the confidence that the two terms are related to each other. The scenario is depicted in Table 1.

The values given in Table 1 are demonstrative numbers just for the purpose of providing an example. In our ontology mapping framework Dempster–Shafer Similarity (DSSim), the similarities are considered as subjective beliefs, which is represented by belief mass functions that can be combined using the Dempster's combination rule. This subjective belief is the outcome of a similarity algorithm, which is applied by a software agent for creating mapping between two concepts in different ontologies. In our ontology mapping framework, different agents assess similarities and their beliefs in the similarities need to be combined into a more coherent result. However, these individual beliefs in practice are often conflicting. In this scenario applying Dempster's combination rule to conflicting beliefs can lead to an almost impossible choice because the combination rule strongly emphasises the common support of beliefs that have been established for the various sources. Additionally, several criticisms have been formulated because due to the normalisation of the beliefs the combination disregards contradictory evidence. The counter-intuitive results that can occur with Dempster's rule of combination are well known and have generated a great deal of debate within the uncertainty reasoning community. Different variants of the combination rule (Sentz & Ferson, 2002) have been proposed, to achieve more realistic combined belief. Instead of proposing an additional combination rule, we turned our attention to the root cause of the conflict itself, namely how the uncertain information was produced in our model.

RULE 1 : IF (beliefdifference IS large OR beliefdifference IS average) AND belief IS weak AND (similarity IS low OR similarity IS average) THEN low_trust IS low;
RULE 2 : IF (beliefdifference IS large OR beliefdifference IS average) AND belief IS weak AND (similarity IS low OR similarity IS average) THEN medium_trust IS medium;
RULE 3 : IF (beliefdifference IS small OR beliefdifference IS average) AND belief IS strong AND (similarity IS high OR similarity IS average) THEN high_trust IS high;
RULE 4 : IF (beliefdifference IS small OR beliefdifference IS average) AND belief IS strong AND (similarity IS high OR similarity IS average) THEN medium_trust IS medium;

Fig. 2. Fuzzy rules for trust assessment.

Table 1
Belief conflict detection.

Conflict detection	Belief 1	Belief 2	Belief 3
Obvious	0.85	0.80	0.1
Difficult	0.85	0.65	0.45

The fuzzy voting model was developed by Baldwin (1999) and has been used in Fuzzy logic applications. However, to our knowledge it has not been introduced in the context of trust management on the Semantic Web. In this section, we will briefly introduce the fuzzy voting model theory using a simple example of 10 voters voting against or in favour of the trustfulness of an another agents' belief over the correctness of mapping. In our ontology mapping framework, each mapping agent can request a number of voting agents to help assessing how trustful the other mapping agents' belief is.

According to Baldwin (1999) a fuzzy linguistic variable can be defined as a a list of elements (L, $T(L)$, U, G, μ), where L represents a variable name, $T(L)$ is the set of labels U is all possible values for the variable and G, μ a syntactic and semantic rule. For trust assessment in ontology mapping, we assume that G is equivalent to null and the variable labels are described by $T(L)$. A formalisation of the fuzzy voting model can be found in (Lawry, 1998).

To define trust, we define three labels for the linguistic variable {Low_trust (L_t), Medium_trust (M_t) and High_trust (H_t)}, where the possible values can be in $U = [0,1]$. Further assume that we have n number of voters where each voter need to pick different labels from the possible labels represented by $T(L)$ for a particular input value represented by u. In these cases, the membership value $\chi_{\mu(w)(u)}$ is equivalent to the proportion of voters who can choose u in the labels represented by w.

To eliminate conflict in the beliefs, a number of independent voters need to be considered and their opinions need to be consolidated. Therefore, in our case and for demonstrating the approach we define 10 voters. The argument is that 10 voters should be sufficient to represent the different opinions, and ensure the best possible outcome of the voting.

Formally, let us define

$$V = \{A1, A2, A3, A4, A5, A6, A7, A8, A9, A10\} \quad (2)$$
$$T(L) = \{L_t, M_t, H_t\}$$

The number of voters can differ, however, assuming 10 voters can ensure that

1. The overlap between the membership functions can proportionally be distributed on the possible scale of the belief difference $[0 \cdots 1]$.
2. The work load of the voters does not slow the mapping process down.

Let us start illustrating the previous ideas with a small example. By definition consider three linguistic output variables L representing trust levels and $T(L)$ the set of linguistic values as $T(L)$ = {*Low_trust*, *Medium_trust*, *High_trust*}. Then, we define the membership functions per output variable i.e. {(*Low_trust*), (*Medium_trust*), (*High_trust*)}. Further each voter has different overlapping gaussian membership functions as depicted in Fig. 1.

The overlap between the membership functions represented by the different vertices in Fig. 1, ensures that voters can introduce different opinions as they pick the possible trust levels for the same difference in belief.

The possible set of trust levels $L = TRUST$ is defined by Table 2. Note that in the table we use a short notation L_t means Low_trust, M_t means Medium_trust and H_t means High_trust. Once the fuzzy sets have been defined, the system is ready to assess the output trust memberships for the input values. Both input and output variables are real numbers on the range between $[0 \cdots 1]$. Based on the difference of beliefs, own belief and similarity of the different voters the system evaluates the scenario. The evaluation includes the fuzzification, which makes the conversion between the input and the fuzzy values, the reasoning process that is operates with the defined fuzzy rules to assign fuzzy output to the inputs. In the last step the defuzzification is carried out, which transforms the fuzzy outputs into values that can be used for further processing. Therefore, each input (belief difference, belief and similarity) produces a possible defuzzified output (low, medium or high trust) for the possible output variables. Each defuzzified value can be interpreted as a possible trust level, where the linguistic variable with the highest defuzzified value is retained in case more than one output variable is selected. As an example consider a case where the defuzzified output for belief difference between agent 1 (A1) and agent 2 (A2) with a value 0.67 has resulted in the situation described in Table 2. Note that each voter has its own membership function, where the level of overlap is different for each voter. Based on a concrete input, the first voting agent could map the defuzzified variables into high, medium and low trust whereas tenth voting agent to only low trust.

Note that behind each trust level there is a real number, which represents the defuzzified value. These values are used to reduce the number of possible linguistic variables in order to obtain the vote for each voting agent. Each agent retains the linguistic variable that represents the highest value and is depicted in Table 3.

Taken as a function of x these probabilities form probability functions. They should therefore satisfy:

$$\sum_{w \in T(L)} Pr(L = w|x) = 1 \quad (3)$$

which gives a probability distribution on words:

$$\sum Pr(L = Low\_trust|x) = 0.6 \quad (4)$$
$$\sum Pr(L = Medium\_trust|x) = 0.3 \quad (5)$$
$$\sum Pr(L = High\_trust|x) = 0.1 \quad (6)$$

Table 2
Possible values for the voting.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
L_t	L_t	L_t	L_t	L_t	L_t	L_t	L_t	L_t	L_t
M_t	M_t	M_t	M_t	M_t	M_t				
H_t	H_t	H_t							

Table 3
Voting.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
H_t	M_t	L_t	L_t	M_t	M_t	L_t	L_t	L_t	L_t

As a result of voting, we can conclude that given the particular difference in beliefs (represented by a real number 0.67 in this example) the combination should not consider this belief of agent 2 (A2). This is because based on its difference compared to belief of agent 1 (A1) it turns out to be a distrustful assessment. The before mentioned process of taking the "probability distributions on words" is then repeated as many times as needed. In fact, the process is repeated as many different beliefs we have for the similarity (i.e. as many as different similarity measures exist in the ontology mapping system).

4.2. Membership functions

The proper selection of a good membership function for a fuzzy system plays an important role in the overall applicability of the system for a particular problem. Therefore, for our conflict resolution problem we have carried out experiments, to select the best possible membership function combination that fit well to our problem.

We have experimented with the trapezoidal, triangular and gauss membership functions and their combinations, to represent our input and output variables. For each test, we have generated 300 scenarios, which contain random input variables (belief difference, belief and similarity) that maps to a single trust level i.e. output variable (high, medium or low trust). In addition, we have defined nine combinations of membership functions that describes our input and output variables. We repeated our experiment 1000 times regenerating the 300 scenarios for each iteration.

Based on out experiments, describing trust levels with trapezoidal triangular or gauss membership functions has proved to be successful (see Fig. 1), when it is combined with a triangular input values. The selection of the output membership functions does not influence the overall results. In this case, the average number of wrong answers are 121 out of 300 per scenarios therefore we can eliminate conflict in 2/3 of the cases. The second alternative with slightly worst precision is the trapezoidal input where 150 cases are wrongly resolved out of 300. The worst combination is the gauss input where 200 conflict resolution were wrong out of 300.

5. OAEI case study

Experimental comparison of the conflict resolution has been carried out with the Ontology Mapping Evaluation Initiative data sets from 2007 and 2008. In this section the results of these experiments are presented.

The OAEI evaluation uses widely accepted measures for search systems called recall and precision. Both precision and recall have a fixed range between 0 and 1 (i.e. 0% and 100%). Precision shows how many correct mappings have been returned by the search engine or a mapping system, while recall measures how many relevant mappings have been returned by the system.

The first experiments have been carried out with the benchmark ontologies of the Ontology Alignment Evaluation Initiative (OAEI),[4] which is an international initiative that has been set up for evaluating ontology matching algorithms. The experiments were carried out to assess how trust management influences results of our proposed mapping algorithm. The main objective was to evaluate the impact of establishing trust before combining beliefs in similarities between concepts and properties in the ontology.

The benchmark tests are easily processable for any system, because the original reference ontology contained 33 named classes and 64 properties. Additionally, each test contains the reference alignment, which can be used to validate the generated results.

The benchmark tests can be grouped into 3 different categories:

- Group 1xx: relatively simple tests such as comparing the reference with itself or with a totally different ontology.
- Group 2xx: these tests are the most interesting ones for the evaluation. Different tests have been generated using the reference ontology and change it using a specific rule. For example, replacing class names with random strings, introducing different naming conventions, remove class hierarchy or reducing the expressiveness of the ontology from OWL FULL to OWL Lite.
- Group 3xx: four real ontologies of bibliographic references that were defined by different research institutions.

Due to the fact that the benchmark is the only test set in the OAEI tracks where the results are also available, we have first run our experiments where DSSim applies the fuzzy voting model for evaluating trust and one without it. Therefore, as a basic comparison we have modified the mapping algorithm (without trust), which does not evaluate trust before conflicting belief combination, just combine them using Dempster's combination rule. The recall and precision graphs for the algorithm with trust and without trust over the whole benchmarks are depicted in Figs. 3 and 4. Experiments have proved that with establishing trust one can reach higher average precision and recall rate.

Fig. 3 depicts the recall values that were measures with and without applying trust assessment between mapping agents in the DSSim algorithm for the benchmark test categories.

Fig. 4 depicts the precision values that were measures with and without applying trust assessment between mapping agents in the DSSim algorithm for the benchmark test categories. Both recall and precision have been improved applying our trust model for combining contradictory evidence. From the precision point of view, the increased recall values have not impacted the results significantly, which is good because the objective is always the improvement of both recall and precision together. More specifically, we have measured the average improvement for the whole benchmark test set that contains 51 ontologies. Based on the experiments, the average recall has increased by 12% and the precision is by 16%. The relative high increase in precision compared to recall is attributed to the fact that in some cases the precision has been increased by 100% as a consequence of a small recall increase of 1%. This is perfectly normal because if the recall increases from 0% to 1% and the returned mappings are all correct (which is

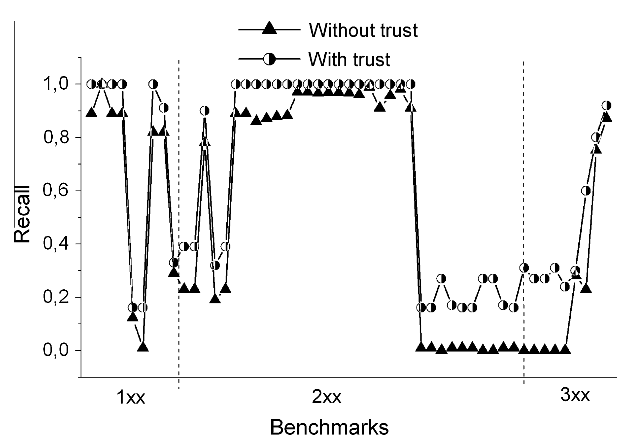

Fig. 3. Benchmarks-Recall graph with and without applying fuzzy voting.

[4] http://oaei.ontologymatching.org/.

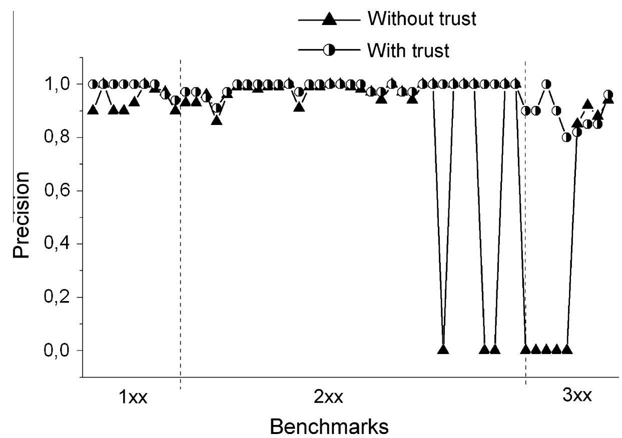

Fig. 4. Benchmarks-Precision graph with and without applying fuzzy voting.

possible since the number of mappings are small) then the precision increases from 0% to 100%. Furthermore the increase in recall and precision greatly varies from test to test. Surprisingly, the precision has decreased in some cases (5 out of 51). The maximum decrease in precision was 7% and maximum increase was 100%. The recall has never decreased in any of the tests and the minimum increase was 0.02% whereas the maximum increase was 37%.

As mentioned above, in our ontology mapping algorithm there are number of mapping agents that carry out similarity assessments, hence create belief mass assignments for the evidence. Before the belief mass function is combined, each mapping agent need to calculate dynamically a trust value, which describes how confident the particular mapping agent is about the other mapping agents' assessment. This dynamic trust assessment is based on the fuzzy voting model, and depends on its own and other agents' belief mass function. In our ontology mapping framework, we assess trust between the mapping agents' beliefs and determine, which agents' belief cannot be trusted, rejecting the one, which is as the result of trust assessment become distrustful.

The weakness of our system is to provide good mappings, when only semantic similarity can be exploited is the direct consequence of our mapping architecture. At the moment, we are using 4 mapping agents where 3 carries our syntactic similarity comparisons and only 1 is specialised in semantics. However, it is worth to note that our approach seems to be stable compared to our performance in 2006 and 2007. The precision and recall values were similar, in spite of the fact that more and more difficult tests have been introduced in 2008. As the DSSim architecture is easily expandable with adding more mapping agents, it is possible to enhance our semantic mapping performance in the future.

5.1. Directory

The directory track is a large and challenging track, because the tests were generated from existing Web directories i.e. real world ontologies. The size of the ontologies are relatively small, however, the number of tests are large. Further the generated ontologies do not have a deep class hierarchy and they do not contain any properties.

The specific characteristics of the dataset are as follows:

- The root of the web directories has been included with a small number of classes for more than 4500 tests. Expert mappings for all the matching tasks.
- Each test contains only simple ontology relationships i.e. subclass
- The generated tests contain mistakes concerning the terminology in order to mimic the real world modelling scenario.

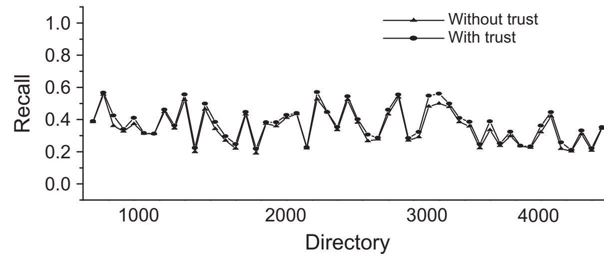

Fig. 5. Directory-Recall graph with and without applying fuzzy voting.

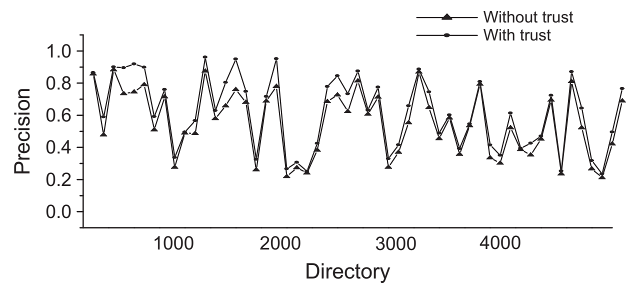

Fig. 6. Directory-Precision graph with and without applying fuzzy voting.

Figs. 5 and 6 display the result of the mapping with and without applying trust into the belief combination for the directory track for each test case. In case of the directories, the measures without applying trust have been calculated based on the original results submitted to the OAEI organisers. During the OAEI evaluation DSSim has produced only one mapping file that included the trust assessment algorithm. Based on the results communicated by the organisers we have run the our mapping algorithm and compared the mapping file with the one that was submitted to OAEI. The library track shows large differences in some mappings (e.g. 50% better with applying trust), however, it is important to note that these large differences can be attributed to the fact that the ontologies contain only a couple of classes. In these cases, even improving the mapping with two new mapping pairs can result in a 50% increase in precision or recall. Therefore, the results should be interpreted considering this bias. Thanks to the large number of tests i.e. more than 4500 mapping tasks, it is possible to deduce an average improvement for both precision and recall. Considering recall, the average improvement was 8% and the precision increase was 11%.

6. Conclusions and future work

In this paper, we have proposed a fuzzy voting model to eliminate conflicts in beliefs in the context of ontology mapping. We have defined what fuzzy trust means in this context, and have defined the input and output variables of a fuzzy system that can be used to manage the contradictory beliefs. Through fuzzy trust, we have defined different trust levels, represented by linguistic variables. Recognising the importance of different trust levels is relevant, not just for ontology mapping but for the Semantic Web as a whole. Based on the experiments with the different OAEI tests, we can conclude that by the addition to the combination rule, we had improved average recall up to 12% and the precision between 3% and 16% as depicted in Figs. 3 and 4. In practice these improvements are significant in terms of recall and precision, because in the ontology mapping community researchers are trying to push the limits of the existing matching algorithms to achieve around 10% and 30% improvement. In addition, our proposed approach also turns out to be flexible concerning the improvements in precision and recall. The reason is that the membership functions for the voters can be changed dynamically in order to influence the outputs, according to the different similarity measures that can be used in the mapping system. We have described initial experimental results with the benchmarks of the Ontology Alignment Initiative, which demonstrates the effectiveness of our approach through the improved recall and precision rates. There are many areas of ongoing work, with our primary focus considering the effect of the changing number of voters and the impact on precision and recall or applying our algorithm in different application areas. In the future, we also try to investigate how multilingual background knowledge can impact the mapping process for ontologies that are not in English e.g. library. We also aim to measure the proportion of the obvious and difficult conflicts (see Table 1) that can occur during the mapping process and how these affect the overall performance of our solution.

References

Artz, D., & Gil, Y. (2007). A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web, 5*, 58–71.

Austen-Smith, D., & Banks, J. S. (1996). Information aggregation, rationality, and the Condorcet jury theorem. *The American Political Science Review, 90*, 34–45.

Baldwin, J. F. (1999). Mass assignment fundamentals for computing with words. In *Selected and invited papers from the workshop on fuzzy logic in artificial intelligence*. *Lecture notes in computer science* (Vol. 1566, pp. 22–44). Springer-Verlag.

Ehrig, M., Koschmider, A., & Oberweis, A. (2007). Measuring similarity between semantic business process models. In *Proceedings of the fourth Asia–Pacific conference on Conceptual modelling* (pp. 71–80). Australian Computer Society, Inc..

Euzenat, J., & Shvaiko, P. (2007). *Ontology matching*. Heidelberg (DE): Springer-Verlag.

Ferrara, A., Lorusso, D., Stamou, G., Stoilos, G., Tzouvaras, V., & Venetis, T. (2008). Resolution of conflicts among ontology mappings: A fuzzy approach. In *Proceedings of the 3rd international workshop on ontology matching*.

Grandison, T., & Sloman, M. (2000). A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials, 3*.

Hamdi, F., Niraula, B. S. N. B., & Reynaud, C. (2009). TaxoMap in the OAEI 2009 alignment contest. In Proceedings of the 4th international workshop on ontology matching (OM-2009). CEUR workshop proceedings (Vol. 551).

Jean-Mary, Y. R., & Kabuka, M. R. (2008). Asmov: Results for oaei 2008. In *Proceedings of the 3rd international workshop on ontology matching*.

Jean-Mary, Y. R., Shironoshita, E. P., & Kabuka, M. R. (2009). Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web, 7*, 235–251.

Lawry, J. (1998). A voting mechanism for fuzzy logic. *International Journal of Approximate Reasoning, 19*, 315–333.

Liu, X. -J., Wang, Y. -L., & Wang, J. (2006). Towards a semi-automatic ontology mapping – An approach using instance based learning and logic relation mining. In *Fifth Mexican international conference (MICAI 2006) on artificial intelligence*.

Memon, Q. A., & Khoja, S. A. (2009). Semantic web approach to academic program assessment. *International Journal of Engineering Education, 25*, 1020–1028.

Mui, L., Mohtashemi, M., & Halberstadt, A. (2002). A computational model of trust and reputation. In *Proceedings of the 35th annual Hawaii international conference on system sciences* (pp. 2431–2439).

Nagy, M., Vargas-Vera, M., & Motta, E. (2008). Managing conflicting beliefs with fuzzy trust on the semantic web. In *MICAI 2008: Advances in artificial intelligence* (Vol. 5317/2008, pp. 827–837).

Olmedilla, D., Rana, O. F., Matthews, B., & Nejdl, W. (2005). Security and trust issues in semantic grids. In *Semantic grid*.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In Proceedings of international conference on new methods in language processing (Vol. 12, pp. 44–49).

Seddiqui, M. H., & Aono, M. (2009). An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Web Semantics: Science, Services and Agents on the World Wide Web, 7*, 344–356.

Sentz, K., & Ferson, S. (2002). *Combination of evidence in Dempster–Shafer theory*. Technical report, Systems Science and Industrial Engineering Department, Binghamton University.

Shvaiko, P., & Euzenat, J. (2008). *Ten challenges for ontology matching*. Technical report DISI-08-042, University of Trento.

Shvaiko, P., & Euzenat, J. (2013). Ontology matching: State of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering, 25*, 158–176.

Tang, J., Li, J., Liang, B., Huang, X., Li, Y., & Wang, K. (2006). Using bayesian decision for ontology mapping. *Web Semantics: Science, Services and Agents on the World Wide Web, 4*.

Wang, P., & Xu, B. (2009). Lily: Ontology alignment results for OAEI 2009. In Proceedings of the 4th international workshop on ontology matching (OM-2009). CEUR workshop proceedings (Vol. 551).

Young, H. P. (1988). Condorcet's theory of voting. *The American Political Science Review, 82*, 1231–1244.