An Argumentation Framework Based on *Strength* for Ontology Mapping

Cássia Trojahn[1], Paulo Quaresma[1], and Renata Vieira[2]

[1] Departamento de Informática, Universidade de Évora, Portugal
[2] Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, Brazil
cassia@di.uevora.pt, pq@di.uevora.pt, renata.vieira@pucrs.br

Abstract. In the field of ontology mapping, using argumentation to combine different mapping approaches is an innovative research area. We had extended the Value-based Argumentation Framework (VAF) in order to represent arguments with *confidence degrees*, according to the similarity degree between the terms being mapped. The mappings are computed by agents using different mapping approaches. Based on their preferences and confidences, the agents compute their preferred mapping sets. The arguments in such preferred sets are viewed as the set of globally acceptable arguments. In previous work we had used discrete classes to represent the *confidence degrees* (certainty and uncertainty). In this paper, we propose to use continuous values from the interval [0,1]. Here, *confidence* is treated as *strength*. Using a threshold for the *strength* we can reduce the set of mappings and adjust the values of precision. We evaluate the use of *strength* against the previous confidence as discrete classes. The results are promising, especially what concerns precision.

1 Introduction

Ontology mapping is the process of linking corresponding terms from different ontologies. The mapping result can be used for ontology merging, agent communication, query answering, or for navigation on the Semantic Web. [19], [20], and [7] present a broad overview of the various approaches on automated ontology matching. Basically, the ontology mapping problem involves to combine different approaches. Using argumentation to solve this problem is an innovative research.

We had extended an Argumentation Framework, namely Value-based Argumentation Framework (VAF)[3], in order to represent arguments with *confidence degrees*. The VAF allows to determine which arguments are acceptable, with respect to different *audiences* represented by different agents. We then associate to each argument a *confidence degree*, representing how confident an agent is in the similarity of two ontology terms.

Our agents apply different mapping approaches and cooperate in order to exchange their local results (arguments). Next, based on their preferences and confidence of the arguments, the agents compute their preferred mapping sets. The arguments in such preferred sets are viewed as the set of globally acceptable

I. Rahwan and P. Moraitis (Eds.): ArgMas 2008, LNAI 5384, pp. 57–71, 2009.

arguments. Our approach is able to give a formal motivation for the composite mapping approaches.

In previous work [24][25] we had used discrete classes to represent the *confidence degrees* (certainty and uncertainty). In this paper, we propose to use continuous values from the interval [0,1]. Here, *confidence* is treated as *strength*. Using a threshold for the *strength* we can reduce the set of mappings and adjust the values of precision. In a scenario where the mappings must be defined on the fly (i.e., web systems involving agent communication), precision is preferred than recall. On the other side, when the mapping system is used to help users in the mapping process, it is interesting to reduce the set of mappings. We evaluate the use of *strength* against the previous discrete classes. The results are promising, specially what concerns precision.

The paper is structured as follows. Firstly, in Section 2, we comment on ontologies and approaches for ontology mapping. Section 3 presents the Argumentation Framework, upon which our model rely. Section 4 presents our Strength based Argumentation Framework (S-VAF). Section 5 presents the evaluation. Section 6 comments on related work. Finally, section 7 presents the final remarks and future work.

2 Ontologies and Ontology Mapping Approaches

The standard definition of ontology is from [10]: "an explicit specification of the conceptualization of the domain". From this definition [8] point out that: (a) the ontology makes things explicit – without an ontology many design assumptions may be implicit in the executable representation; (b) the ontology is supposed to be formal: the notions it captures are thus precise and unambiguous; (c) the ontology concerns some specific domain; (d) the ontology represents a conceptualization – different people will conceptualize a domain differently according to experience, and their tasks in the domain – and there is no a single ontology applicable to a domain. Specifically, ontologies contain the types of *objects* in the domain; the *attributes* which these objects may have; the *relationships* which these objects may enter into; and the *values* that the attributes may have for particular types.

Ontology mapping is the process of finding correspondences between two ontologies, using as input their types of objects (classes), attributes, relationships or value of attributes. For instance, if two objects correspond, they mean the same thing, or closely related things. [19], [20], and [7] present a broad overview of the various approaches on automated ontology matching. In this paper, we focus in how to combine mapping approaches using argumentation. Three specific kinds of mapping approaches are considered: lexical ([22][18]), semantic and structural (see [11]). Lexical approaches apply metrics to compare string similarity. One well-known measure is the edit distance [14], which is given by the minimum number of operations (insertion, deletion, or substitution of a single character) needed to transform one string into another.

Semantic approaches consider the semantic relations between concepts to measure the similarity between them, usually on the basis of semantic oriented linguistic resources. The well-known WordNet[1] database, a large repository of English semantically related items, has been used to provide these relations. This kind of mapping is complementary to the pure string similarity metrics. It is not uncommon the cases where string metrics fail to identify high similarity between strings that represent completely different concepts (i.e, the words "score" and "store"). Semantic-structural approaches have been explored [11]. In this case, the positions of the terms in the ontology hierarchy are considered, i.e, terms more generals and terms more specifics are also considered as input to the mapping process.

3 Argumentation Framework

Our argumentation framework for ontology mapping is based on the Value-based Argumentation Frameworks (VAF)[3], a development of the classical argument system of Dung [6]. First, we present the Dung's framework, upon which the VAF rely. Next, we present the VAF and our extended framework.

3.1 Classical Argumentation Framework

Dung [6] defines an argumentation framework as follows.

Definition 3.1.1. An Argumentation Framework is a pair $AF = (AR, \mathit{attacks})$, where AR is a set of arguments and *attacks* is a binary relation on AR, i.e., *attacks* $\subseteq$ AR $\times$ AR. An *attack*(A,B) means that the argument A attacks the argument B. A set of arguments S attacks an argument B if B is attacked by an argument in S.

The key question about the framework is whether a given argument A, $A \in AR$, should be accepted. One reasonable view is that an argument should be accepted only if every attack on it is rebutted by an accepted argument [6]. This notion produces the following definitions:

Definition 3.1.2. An argument $A \in AR$ is *acceptable* with respect to set arguments $S(acceptable(A,S))$, if $(\forall\ x)(x \in AR) \wedge (attacks(x,A)) \longrightarrow (\exists\ y)(y \in S) \wedge attacks(y,x)$

Definition 3.1.3. A set S of arguments is *conflict-free* if $\neg(\exists\ x)(\exists\ y)((x \in S) \wedge (y \in S) \wedge attacks(x,y))$

Definition 3.1.4. A conflict-free set of arguments S is *admissible* if $(\forall x)(x \in S) \longrightarrow acceptable(x,S)$

Definition 3.1.5. A set of arguments S is a *preferred extension* if it is a maximal (with respect to inclusion set) admissible set of AR.

[1] http://www.wordnet.princeton.edu

A *preferred extension* represent a consistent position within AF, which can defend itself against all attacks and which cannot be further extended without introducing a conflict. The purpose of [3] in extending the AF is to allow associate arguments with the social values they advance. Then, the attack of one argument on another is evaluated to say whether or not it succeeds by comparing the preferences of the values advanced by the arguments concerned.

3.2 Value-Based Argumentation Framework

In Dung's frameworks, attacks always succeed. However, in many domains, including the one under consideration, arguments lack this coercive force: they provide reasons which may be more or less persuasive [13]. Moreover, their persuasiveness may vary according to their audience. The VAF is able to distinguish attacks from successful attacks, those which defeat the attacked argument, with respect to an ordering on the preferences that are associated with the arguments. It allows accommodate different audiences with different interests and preferences.

Definition 3.2.1. A Value-based Argumentation Framework (VAF) is a 5-tuple $VAF = (AR, attacks, V, val, P)$ where $(AR, attacks)$ is an argumentation framework, V is a nonempty set of values, *val* is a function which maps from elements of AR to elements of V and P is a set of possible audiences. For each $A \in AR$, $val(A) \in V$.

Definition 3.2.2. An Audience-specific Value Based Argumentation Framework (AVAF) is a 5-tuple $VAF_a = (AR, attacks, V, val, valpref_a)$ where AR, emphattacks, V and val are as for a VAF, a is an audience and $valpref_a$ is a preference relation (transitive, irreflexive and asymmetric) $valpref_a \subseteq V \times V$, reflecting the value preferences of audience a. $valpref(v_1, v_2)$ means v_1 is preferred to v_2.

If V contains a single value, or no preference between the values has been defined, the AVAF becomes a standard AF. If each argument can map to a different value, a Preference Based Argumentation Framework is obtained [1].

Definition 3.2.3. An argument $A \in AR$ defeats$_a$ (or *successfully attacks*) an argument $B \in AR$ for audience a if and only if both $attacks(A,B)$ and not $valpref(val(B), val(A))$.

Definition 3.2.4. An argument $A \in AR$ is *acceptable* to audience a (*acceptable$_a$*) with respect to set of arguments S, $acceptable_a(A,S))$ if $(\forall\ x)\ ((x \in AR \wedge defeats_a\ (x,A)) \longrightarrow (\exists y)((y \in S) \wedge defeats_a(y,x)))$.

Definition 3.2.5. A set S of arguments is *conflict-free* for audience a if $(\forall\ x)(\forall\ y)((x \in S \wedge y \in S) \longrightarrow (\neg attacks(x,y) \vee valpref(val(y),val(x)) \in valpref_a))$.

Definition 3.2.6. A *conflict-free* set of argument S for audience a is *admissible* for an audience a if $(\forall x)(x \in S \longrightarrow acceptable_a(x,S))$.

Definition 3.2.7. A set of argument S in the VAF is a *preferred extension* for audience *a (preferred$_a$)* if it is a maximal (with respect to set inclusion) *admissible* for audience *a* of AR.

In order to determine the preferred extension with respect to a value ordering promoted by distinct audiences, [3] introduces the notion of *objective* and *subjective* acceptance.

Definition 3.2.8. An argument $x \in AR$ is *subjectively* acceptable if and only if x appears in the preferred extension for some specific audiences but not all. An argument $x \in AR$ is *objectively* acceptable if and only if, x appears in the preferred extension for every specific audience. An argument which is neither objectively nor subjectively acceptable is said to be *indefensible*.

3.3 *Strength* Based Argumentation Framework (S-VAF)

We extend the VAF in order to represent arguments with *strength*, which represents the confidence that an agent has in some argument. One element has been added to VAF: a function which maps from arguments to real values from the interval [0,1]. We assumed that the *strength* is a relevant criterion in the ontology mapping domain, representing the confidence measure by using the mapping approach.

Definition 3.3.1. A Strength based Argumentation Framework (S-VAF) is a 6-tuple *(AR, attacks, V, val, P, valS)* where *(AR, attacks, V, val, P)* is a value-based argumentation framework, and *valS* is a function which maps from elements of AR to real values from the interval [0,1] representing the *strength* of the argument.

Definition 3.3.2. In the S-VAF, an argument $x \in AR$ defeats$_a$ an argument $y \in AR$ for audience a if and only if $attacks(x,y) \wedge ((valS(x) > valS(y)) \vee (\neg\, valpref(val(y),val(x)) \wedge (\neg\, (valS(y) > valS(x)))$.

An attack succeeds if (a) the *strength* of the attacking argument is greater than the *strength* of the argument being attacked; or if (b) the argument being attacked does not have greater preference value than attacking argument (or if both arguments relate to the same preference values) and the *strength* of the argument being attacked is not greater than the attacking argument.

Definition 3.3.3. In the S-VAF, an argument $A \in AR$ is *acceptable* to audience a ($acceptable_a$) with respect to set of arguments S, $acceptable_a(A,S))$ if $(\forall x)\ ((x \in AR \wedge defeats_a\ (x,A)) \longrightarrow (\exists y)((y \in S) \wedge defeats_a(y,x)))$.

Definition 3.3.4. In the S-VAF, a set S of arguments is *conflict-free* for audience a if $(\forall x)(\forall y)\ ((x \in S \wedge y \in S) \longrightarrow (\neg attacks(x, y) \vee (\neg(valS(x) > valS(y)) \wedge (valpref(val(y), val(x))\ (\vee\ (valS(y) > valS(x))))))$.

Definition 3.3.5. A set of argument S in the S-VAF is a *preferred extension* for audience *a (preferred$_a$)* if it is a maximal (with respect to set inclusion) *admissible* for audience *a* of AR.

It is important to distinguish the difference between *values* and *strengths*. There are different types of agents representing different mapping approaches. Each approach represent a *value* and each agent represents an audience, with preferences between the *values*. The *values* are used to determine the preference between the different agents. Moreover, each agent generates arguments with a *strength*, based on the confidence returned by the mapping technique. So, we extended the VAF in order to define a new notion of argument acceptability which combines *values* (related with the agent's preference) and *strength* (confidence degree of an argument). If our criterion was based only on the *strength* of the arguments, a Preference Based Argumentation Framework could be used [1].

4 S-VAF for Ontology Mapping

In this paper we consider three *values*: lexical (L), semantic (S), and structural (E) (i.e. $V = \{L, S, E\}$, where $V \in$ S-VAF). These values represent the mapping approach used by the agent and are also used to represent the audiences. Each audience has an ordering preference between the *values*. For instance, the lexical agent represents an audience where the value L is preferred to the values S and E. Our idea is not to have an individual audience with preference between the agents (i.e., semantic agent is preferred to the other agents), but it is to try accommodate different audiences (agents) and their preferences.

4.1 Argumentation Generation

First, the agents work in an independent manner, applying the mapping approaches and generating mapping sets. The mapping result will consist of a set of all possible correspondences between terms (type of objects) of two ontologies. A mapping m can be described as a 3-tuple $m = (t_1,t_2,h)$, where t_1 corresponds to a term in the ontology 1, t_2 corresponds to a term in the ontology 2, and h is one of {+,-} depending on whether the argument is that m does or does not hold. Now, we can define arguments as follows:

Definition 4.1. An *argument* $\in AR$ is a 3-tuple $x = (m,a,s)$, where m is a mapping; $a \in V$ is the value of the argument (lexical, semantic or structural); s is the *strength* of the argument.

Lexical Agent. This agent adopts the *lexical similarity* proposed by [18]. This metric is based on the Levenshtein distance [15] and considers the length of the compared terms to compute the final *lexical similarity*. A value from the interval [0,1] is returned, where 1 indicates high similarity between two terms.

Differently from the previous work [24][25], the agents are able to deal with compound terms. The first step in this process is the tokenization, where the terms are parsed into tokens by a tokenizer. The *strength* of an argument is computed according to the *lexical similarity* between each token of the two compared terms. Table 1 shows the possible values to s and h, where t_{Sn} correspond

Table 1. h and s to lexical audience

s	+ (h)
1	t_{S1} lexically similar to t_{T1}
calc-s	t_{S1} lexically similar to some t_{T1}, ..., t_{Tn} t_{S1}, ..., t_{Sn} some lexically similar to t_T t_{S1}, ..., t_{Sn} some lexically similar to some t_{T1}, ..., t_{Tn}
s	- (h)
0	otherwise

Table 2. h and s to semantic audience

s	+ (h)
1	t_{S1} semantic relation with t_{T1}
calc-s	t_{S1} some semantic relation with some t_{T1}, ..., t_{Tn} t_{S1}, ..., t_{Sn} some semantic relation with t_T t_{S1}, ..., t_{Sn} some semantic relation with some t_{T1}, ..., t_{Tn}
s	- (h)
0	otherwise

to some token of the source term (source ontology), and t_{Tn} correspond to some token of the target term (target ontology). Two tokens are *lexically similar* if the *lexical similarity* is greater than a *threshold r*.

When all tokens are *lexically similar* with each other, the terms match and the *strength* of the argument is 1. If *some* tokens of the terms are *lexically similar*, the *strength* is computed according to the number of tokens that matches, according to the *calc-s* formula, where T_S is the term from the source ontology, T_T is the term from the target ontology, and *nM* is the number of tokens that match between T_S and T_T:

$$calc\text{-}s = max\left(0, \frac{max(\mid T_S \mid, \mid T_T \mid) \ - \ nM)}{max(\mid T_S \mid, \mid T_S \mid)}\right)$$

If there are no lexically similar tokens between the terms, the agent is not sure that the terms map (i.e., *strength* equals to 0), because this agent knows that other agent can resolve this mapping. In the specific case, if there is no lexical similarity between the terms, the semantic agent can resolve that mapping.

Semantic Agent. This agent considers semantic relations (i.e., synonym, hyponym, and hypernym) between terms to measure the similarity between them, on the basis of WordNet[2] database. Table 2 shows the possible values to s and h according to the semantic similarity.

When all tokens have semantic relation with each other, the *strength* of the argument is 1. If *some* tokens have semantic relation, the *strength* is computed according to the number of semantically related tokens (formula presented above).

[2] http://www.wordnet.princeton.edu

Otherwise, if there are no semantic relation between the tokens, the agent is not sure that the terms map (i.e., *strength* equals to 0), because this agent knows that other agent can resolve the mapping. In the specific case, when the searched terms are not available in WordNet, the lexical agent can decide the mapping. It is common because there is no complete lexical database for every domain (i.e., WordNet is incomplete for some domains).

Structural Agent. The structural agent considers the positions of the terms in the ontology hierarchy to verify if the terms can be mapped. First, it is verified if the super-classes of the compared terms are lexically similar. If not, the semantic similarity is used. For instance, if the super-classes of the terms are not lexically similar, but they are synonymous, an argument $x = (\mathrm{m,E},s)$, where $m = (\mathrm{t_1,t_2,+})$, is generated, where s varies according to the rules from Tables 1 or 2.

However, there are two main differences among the *strengths* returned by the lexical, semantic, and structural agents. As Table 1 and Table 2, when the agents can not resolve the mapping, the *strength* of the corresponding argument is 0. However, if the structural agent does not find similarity (lexical or semantic) between the super-classes of the compared terms, it is because the terms can not be mapped (i.e., the terms occurs in different contexts). Then, the *strength* for no mapping is 1. Otherwise, if the structural agent finds similarity between the super-classes of the compared terms, it is because they can be mapped, but it does not mean that the terms have lexical or semantic similarity, then the *strength* for the mapping is 0. For instance, for the terms "Publication/Topic" and "Publication/Proceedings", the structural agent indicates that the terms can be mapped because they have the same super-class, but not with *strength* 1 because it is not able to indicate that the terms are similar. Otherwise, for the terms "Digital-Camera/Accessories" and "Computer/ Accessories", the agent can indicate that the terms can not be mapped because they occur in different contexts (*no-mapping* with *strength* equal to 1).

4.2 Preferred Extension Generation

After generating their set of arguments, the agents exchange with each other their arguments and generate their *attacks* set. An *attack* (or counter-argument) will arise when we have arguments for the mapping between the same terms, but with conflicting values of h. For instance, an argument $x = (\mathrm{m_1,L,+})$ have as an *attack* an argument $y = (\mathrm{m_2,E,-})$, where $\mathrm{m_1}$ and $\mathrm{m_2}$ refer to the same terms in the ontologies. The argument y also represents an *attack* to the argument x.

As an example, consider the mapping between the terms "Subject" and "Topic" and the lexical and semantic agents. The lexical agent generates an argument $x = (\mathrm{m,L,0})$, where $m = (\mathrm{subject}_S,\mathrm{topic}_T,\text{-})$; and the semantic agent generates an argument $y = (\mathrm{m,S,1})$, where $m = (\mathrm{subject}_S,\mathrm{topic}_T,+)$. For both lexical and semantic audiences, the set of arguments is $AR = \{\mathrm{x,y}\}$ and the $attacks = \{(\mathrm{x,y}),(\mathrm{y,x})\}$.

When the set of arguments and attacks have been produced, the agents need to define which of them must be accepted. To do this, the agents compute their preferred extension, according to the agent's preferences and *strengths* of the arguments. A set of arguments is *globally subjectively acceptable* if each element appears in the preferred extension for some agent. A set of arguments is *globally objectively acceptable* if each element appears in the preferred extension for every agent. The arguments which are neither objectively nor subjectively acceptable are considered *indefensible.*

In the example above, considering the lexical(L) and semantic(S) audiences, where L $\succ$ S and S $\succ$ L, respectively, for the lexical audience, the argument y successfully attacks the argument x, while the argument x does not successfully attack the argument y for the semantic audience. Then, the preferred extension of both lexical and semantic agents is composed by the argument y.

5 Argumentation Model Evaluation

Let us consider that three agents need to obtain a consensus about mappings that link corresponding class names in two different ontologies. We have used three groups of ontologies: parts of Google and Yahoo web directories[3](Test 3), product schemas[4](Test 4), and company profiles[5](Test 8). In *Test 3*, the source ontology has 9 terms and the target ontology has 6 terms, resulting 54 possible mappings (comparisons term by term). The terms are formed from 1 to 2 tokens (for instance, "Art-History"). In *Test 4*, the source ontology has 5 terms and the target ontology has 6 terms, resulting 30 possible mappings. The terms are formed from 1 to 3 tokens. Finally, the source and target ontologies in *Test 8* have 10 and 16 classes, respectively, resulting 160 possible mappings. The terms are composed from 1 to 5 tokens (for instance "Oil-and-Gas-Exploration-and-Production" or "Petroleum-Product-Distribution").

As a mapping quality evaluation, the measures of precision, recall and f–measure were used. Precision is defined by the number of correct automated mappings divided by the number of mappings that the system returned. It measures the system's correctness or accuracy. Recall indicates the number of correct mappings returned by the system divided by the number of manual mappings. It measures how complete or comprehensive the system is in its extraction of relevant mappings. F–measure is a weighted harmonic mean of precision and recall.

First, we compared the results from using *confidence* as discrete classes (*certainty* and uncertainty), based on E-VAF, as proposed in [24][25], against the results from using *strength* as continuous values. When considering only the mappings (h equals $+$) with *certainty* (Figure 1 (a)) and the mappings with *strength* equals to 1 (Figure 2 (a)), the values for f-measure (and corresponding precision and recall) were the same, for the three tests. However, when considering both mappings with *certainty* and *uncertainty* (Figure 1 (b)) against the use of a *threshold* (0.70) (Figure 2 (b)), better values of precision were obtained using *strength.*

[3] http://dit.unitn.it/~accord/Experimentaldesign.html (Test 3)
[4] http://dit.unitn.it/~accord/Experimentaldesign.html (Test 4)
[5] http://dit.unitn.it/~accord/Experimentaldesign.html (Test 8)

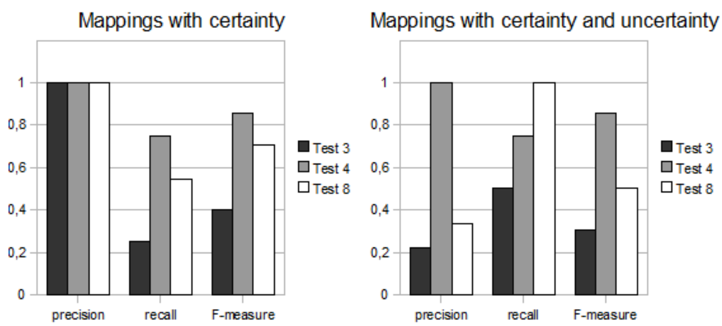

Fig. 1. Mappings with *confidence*: (a) *certainty*; (b) *certainty* + *uncertainty*

Next, we analyzed more specifically the use of different values of *threshold* (Figure 3). When using a low *threshold*, the recall is 1 and the precision is lower. When using a high *threshold* (0.70), the precision is 1 and the recall is lower. In a scenario where the mappings must be defined on the fly, the precision of the mappings is more valuable than the recall (i.e., web systems involving agent communication).

On the other side, when the mapping system is used to help users in the mapping process, it is interesting reduce the set of mappings. When using the *confidence uncertainty* it is not possible. However, we can do that using *thresholds* for *strength*. Specifically for *Test 8* (larger ontology), Figure 4 shows the number of mappings using different values for *threshold* (40 mappings returned when considering the mapping with *certainty* and *uncertainty*). In the scenario under consideration, there are two advantages to use *strength* and *thresholds*. First, the user can adjust the *threshold*. Second, when reducing the set of mapping, it is easier for the user to analyze the resulting mappings. As shown in Figure 4, using the *threshold* the set can be reduced to 26, 12 and 6 mappings. In this sense, our system can help the users to reduce the set of possible mappings, using different *thresholds* for *strength*.

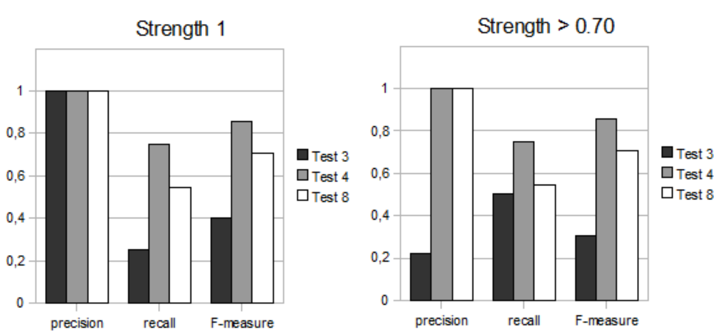

Fig. 2. Mappings with *strength*

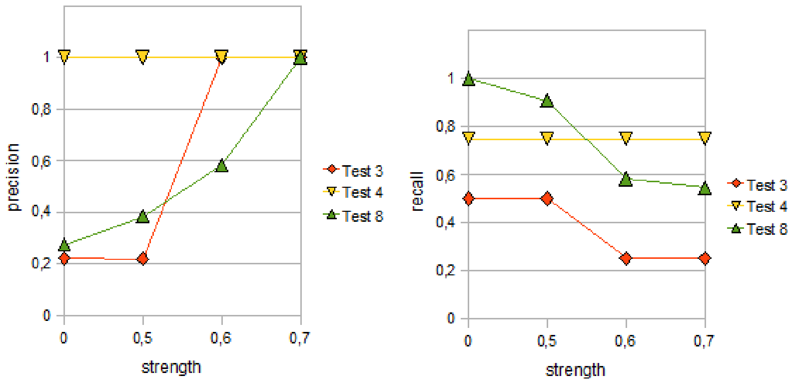

Fig. 3. Precision and recall for the three tests using different thresholds

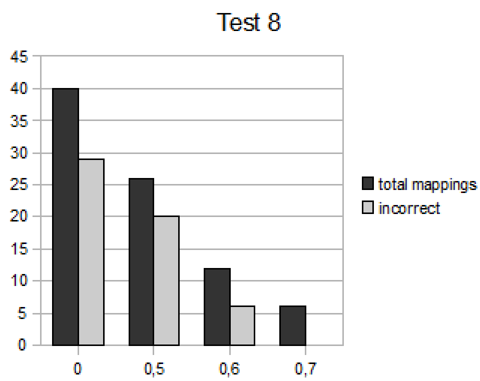

Fig. 4. Comparative results

Second, we compared our proposal with three mapping systems: Cupid[16], COMA[5], and S-Match[9]. The comparative results among these three systems are available in [9]. We utilized these results as criteria to evaluate our argumentation model, but the details of these tests (implementations, time of run, processor, etc) are not available. The evaluation of ontology mapping systems still lacks well established benchmarks, therefore our choices on evaluation were based on the availability of reported results of previous systems. Figure 5 shows the comparative results. We used a threshold r equals to 0.8 for the lexical agent classifies the mappings (terms with lexical similarity greater than 0.8 are considered similar) and a threshold to eliminate the mappings that have *strength* below 0.75. Our model returned better precision than Cupid and COMA, and equal precision when compared to S-Match (precision equal to 1). When comparing the f–measure values, our model had better result than Cupid.

Differently from these works, our model uses argumentation to combine mapping approaches. Cupid uses a weighted similarity which is a mean of linguistic and structural similarities. COMA represents a generic system to combine

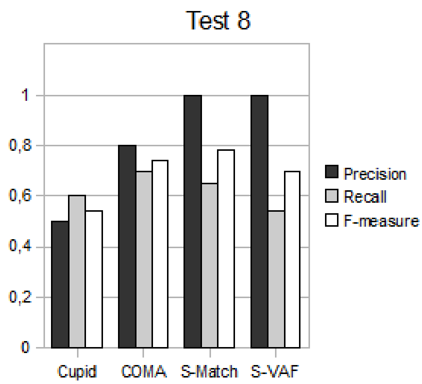

Fig. 5. Comparative results

matching results, which is a set of mapping elements specifying the matching schema elements together with a similarity 2 [0,1] indicating the plausibility of their correspondence. S-Match algorithm is based on the semantic and structural similarities, where the semantic matcher provides the input to the structural matcher.

Although our implementation does not provide the best solution for the ontology mapping problem for these experimental tests as yet, we claim that our main contribution is to propose a model that can be used to combine different approaches.

Using argumentation has the following advantages: the agents are independent to each other; many other agents can be easily added to our model, without having to modify the implementation; there are several techniques for ontology mappings, which can be adapted according to domain, kind of ontologies, and available resources (for instance, in the context of some languages, there is no lexical databases such WordNet).

6 Related Work

In the field of ontology argumentation few approaches are being proposed. Basically, the closer proposal is from [13][12], where an argument framework is used to deal with arguments that support or oppose candidate correspondences between ontologies. The candidate mappings are obtained from an Ontology Mapping Repository (OMR) – the focus is not how the mappings are computed – and argumentation is used to accommodate different agent's preferences. In our approach mappings are computed by the specialized agents described in this paper, and argumentation is used to solve conflicts between the individual results.

We find similar proposals in the field of ontology negotiation. [23] presents an ontology to serve as the basis for agent negotiation, the ontology itself is not the object being negotiated. A similar approach is proposed by [4], where agents agree on a common ontology in a decentralized way. Rather than being the goal of each agent, the ontology mapping is a common goal for every agent in the system. [2] presents an ontology negotiation model which aims to arrive at a common ontology which the agents can use in their particular interaction. We, on the other hand, are concerned with delivering mapping pairs found by a group of agents using argumentation. [21] describes an approach for ontology mapping negotiation, where the mapping is composed by a set of semantic bridges and their inter-relations, as proposed in [17]. The agents are able to achieve a consensus about the mapping through the evaluation of a confidence value that is obtained by utility functions. According to the confidence value the mapping rule is accepted, rejected or negotiated. Differently from [21], we do not use utility functions. Our model is based on cooperation and argumentation, where the agents change their arguments and by argumentation they select the preferred mapping.

7 Final Remarks and Future Work

In this paper we proposed to use continuous values to represent the *strength* of arguments, which represents the *confidence degree* that an agent has in the mapping, according to the similarity degree between the ontology terms. We had previously extended an Argumentation Framework, namely Value-based Argumentation Framework (VAF)[3], in order to represent arguments with *confidence* with discrete values.

Using a *threshold* for the *strength* we can reduce the set of mappings and adjust the values of *precision*. In a scenario where the mappings must be defined on the fly (i.e., web systems involving agent communication), the precision is preferred than the recall. On the other, when the mapping system is used to help users in the mapping process, it is interesting reduce the set of mappings, what cannot be done when using the discrete classes. Moreover, the *strength* as a continuous value is more expressive than the discrete classes, especially when dealing with compound terms.

We evaluated the use of *strength* against the previous discrete classes using three groups of ontologies. The results are promising, especially what concerns *precision*.

In the future, we intend to develop further tests considering a benchmark of ontologies[6]; verify the impact of using only *strengths* in our model; and use the mapping as input to an ontology merge process in the question answering domain.

References

1. Amgoud, L., Cayrol, C.: On the acceptability of arguments in preference-based argumentation. In: 14th Conference on Uncertainty in Artificial Intelligence (UAI 1998), San Francisco, California, juillet 1998, pp. 1–7. Morgan Kaufmann, San Francisco (1998)
2. Bailin, S., Truszkowski, W.: Ontology negotiation between intelligent information agents. The Knowledge Engineering Review 17(1), 7–19 (2002)

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

3. Bench-Capon, T.: Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation 13, 429–448 (2003)
4. van Diggelen, J., Beun, R., Dignum, F., van Eijk, R., Meyer, J.C.: Anemone: An effective minimal ontology negotiation environment. In: Proceedings of the Fiftheen International Conference on Autonomous Agents and Multi-Agent Systems, pp. 899–906 (2006)
5. Do, H.H., Rahm, E.: Coma - a system for flexible combination of schema matching approaches. In: Proceedings of the 28th Conference on Very Large Databases, pp. 610–621 (2002)
6. Dung, P.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n–person games. Artificial Intelligence 77, 321–358 (1995)
7. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, Heidelberg (2007)
8. Gangemi, A., Pisanelli, D.M., Steve, G.: A formal ontology framework to represent norm dynamics. In: Congreso Internacional de Culturas y Sistemas Jurídicos Comparados (2005)
9. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-match: an algorithm and an implementation of semantic matching. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 61–75. Springer, Heidelberg (2004)
10. Gruber, T.R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In: Guarino, N., Poli, R. (eds.) Formal Ontology in Conceptual Analysis and Knowledge Representation, Deventer, The Netherlands. Kluwer Academic Publishers, Dordrecht (1993)
11. Hakimpour, F., Geppert, A.: Resolving semantic heterogeneity in schema integration: an ontology approach. In: Proceedings of the International Conference on Formal Ontology in Informational Systems, pp. 297–308 (2001)
12. Laera, L., Blacoe, I., Tamma, V., Payne, T., Euzenat, J., Bench-Capon, T.: Argumentation over ontology correspondences in mas. In: Durfee, M., Yokoo, E.H. (eds.) Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (2007)
13. Laera, L., Tamma, V., Euzenat, J., Bench-Capon, T., Payne, T.R.: Reaching agreement over ontology alignments. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 371–384. Springer, Heidelberg (2006)
14. Levenshtein, I.: Binary codes capable of correcting deletions, insertions an reversals. In: Cybernetics and Control Theory (1966)
15. Levenshtein, V.: Binary Codes Capable of Correcting Deletions and Insertions and Reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
16. Madhavan, J., Bernstein, P., Rahm, E.: Generic schema matching with cupid. In: Proceedings of the Very Large Data Bases Conference, pp. 49–58 (2001)
17. Maedche, A., Motik, B., Silva, N., Volz, R.: Mafra - a mapping framework for distributed ontologies. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS, vol. 2473, pp. 235–250. Springer, Heidelberg (2002)
18. Maedche, A., Staab, S.: Measuring similarity between ontologies. In: Proceedings of the European Conference on Knowledge Acquisition and Management, pp. 251–263 (2002)
19. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB 10, 334–350 (2001)
20. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Technical report, Informatica e Telecomunicazioni, University of Trento (2004)

21. Silva, N., Maio, P., Rocha, J.: An approach to ontology mapping negotiation. In: Proceedings of the K-CAP Workshop on Integrating Ontologies (2005)
22. Stoilos, G., Stamou, G., Kollias, S.: A string metric for ontology alignment. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 624–637. Springer, Heidelberg (2005)
23. Tamma, V., Wooldridge, M., Blacoe, I., Dickinson, I.: An ontology based approach to automated negotiation. In: Proceedings of the IV Workshop on Agent Mediated Electronic Commerce, pp. 219–237 (2002)
24. Trojahn, C., Quaresma, P., Vieira, R.: A cooperative approach for composite ontology mapping. LNCS Journal of Data Semantic (to appear, 2007)
25. Trojahn, C., Quaresma, P., Vieira, R.: An extended value-based argumentation framework for ontology mapping with confidence degrees. In: Fourth International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2007). Workshop at International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2007)