Combining Uncertain Outputs from Multiple Ontology Matchers

Ying Wang, Weiru Liu, and David Bell

School of Electronics, Electrical Engineering and Computer Science,
Queen's University Belfast, Belfast, BT7 1NN, UK
{ywang14,w.liu,da.bell}@qub.ac.uk

Abstract. An ontology matching method (or a matcher) aims at matching every entity (or concept) in one ontology to the most suitable entity (or entities) in another ontology. Usually it is almost impossible to find a perfect match in the second ontology for every entity in the first ontology, so a matcher generally returns a set of possible matches with some weights (uncertainty) attached to each pair of match. In order to improve a matching result, several matchers can be used and the matched results from these matchers are combined with suitable approaches. In this paper, we first propose two new matchers among three matchers we use. We then address the need of dealing with uncertainties in mapping by investigating how some uncertainty reasoning frameworks can be used to combine matching results. We apply both the Dempster Shafer theory of evidence (DS theory) and Possibility Theory to merge the results computed by different matchers. Our experimental results and comparisons with related work indicate that integrating these theories to deal with uncertain ontology matching is a promising way to improve the overall matching results.

1 Introduction

Ontology mapping (or matching) is a very important task in the Semantic Web and it has attracted a large amount of effort (e.g., [1,2,3,4,5,6]). Good surveys on recent developments of ontology mapping can be found in [7,8]. Most of the earlier work in this area did not consider uncertainty or imprecision in a mapping, however, in most cases, the mappings produced are imprecise and uncertain. For instance, most automatic ontology mapping tools use heuristics or machine-learning techniques, which are imprecise by their very nature. Even experts are sometimes unsure about the exact matches between concepts and typically assign some certainty rating to a match [9], so a matching result is often associated with a weight which can express how close the two entities are as a match. The need to consider the uncertainty in a mapping began to emerge in a number of papers (e.g., [10,11,12,13,14]) in which Dempster Shafer theory, Bayesian Networks, and rough sets theory are used to deal with different aspects of mapping or ontology descriptions (e.g., concept subsumptions).

In this paper, we further investigate how to combine the weights associated with matchers. We first propose two new matchers, a *linguistic-based matcher*

H. Prade and V.S. Subrahmanian (Eds.): SUM 2007, LNAI 4772, pp. 201–214, 2007.

which extends Lin's approach [15] by considering the path length of two words in the WordNet as a punishment coefficient to adjust a similarity measure from Lin's approach, and a *structure-based matcher* which utilizes the similarity measures between two words (w_1 and w_2), a father node of w_1 with w_2 and all the child nodes of w_1 with w_2. This matcher takes both the semantics and the structure of an ontology into account. We then discuss how the mapping results from different matchers can be combined. We consider both the Dempster Shafer theory of evidence (DS theory) and Possibility Theory and apply them to combine the outcomes obtained by three different and independent matchers (the above two plus the standard *edit distance-based matcher*).

Each matcher returns a match with a weight. We interpret these weights in terms of both DS theory and Possibility Theory and then use their corresponding merging operators to merge the matched results. Our study shows that these two theories are suitable for different situations and using both theories significantly improves the matching results in terms of precision and recall, as illustrated in our experiments. Therefore, integrating uncertainty merging methods into ontology mapping is promising to improve the quality of mapping.

The rest of the paper is organized as follows. Section 2 introduces the basic concepts. Section 3 describes the main ideas in our approach and the mapping matchers used. Section 4 gives the background information about the experiments and the results. Section 5 discusses related work. Section 6 concludes the paper with discussions on future research.

2 Background

2.1 Ontologies and Ontology Mapping

There are many definitions about ontologies and a commonly used one is "An ontology is a formal, explicit specification of a shared conceptualization." [16]. We use the following notation to formally define an ontology. An ontology O is defined as a tuple: $O = (C, R, F, A, I)$ where C is a set of concepts, such as cars or persons; R is a set of relations, such as $mother-of(x, y)$ denotes that y is x's mother; F is a set of functions; A is a set of axioms and I is a set of instances, namely objects appearing in concepts in C, such as *Alan*. In this paper an entity of an ontology is defined as follows: e_{ij} are entities of O_i with $e_{ij} \in \{C_i, R_i, I_i\}$, and entity index $j \in N$ [1].

The overall objective of ontology mapping can be described as in [6]: given two ontologyies O_1 and O_2, for each entity e (or element, concept) in ontology O_1 finding the corresponding element(s) in ontology O_2, which has/have the same or similar semantics with e, and vice versa. Ontology mapping functions and some relative functions that will be used are:

- $map\ O_{i_1} \rightarrow O_{i_2}$: representing the mapping function between the two ontologies
- $map(e_{i_1 j_1}) = e_{i_2 j_2}$: representing the mapping of two entities

- $sim(e_{i_1j_1}, e_{i_2j_2})$: representing the degree of similarity between two entities computed by a matcher
- $sim(e^f_{i_1j_1}, e_{i_2j_2})$: representing the degree of similarity between father node ($e^f_{i_1j_1}$) of $e_{i_1j_1}$ and $e_{i_2j_2}$ computed by a matcher
- $sim(e^c_{i_1j_1}, e_{i_2j_2})$: representing the degree of similarity between a child node ($e^c_{i_1j_1}$) of $e_{i_1j_1}$ and $e_{i_2j_2}$ computed by a matcher

2.2 Uncertainty Theories

Uncertainty is pervasive in information. Uncertain information is usually modeled numerically using Probability Theory, Possibility Theory, or DS theory.

The Dempster-Shafer theory of evidence: DS theory defines mass functions on frame of discernment denoted $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ which contains mutually exclusive and exhaustive possible answers to a question. A mass function assigns some positive values in $[0, 1]$ to some subsets of Θ. If a mass function gives a positive value to a subset A, then this value represents the probability mass of an agent's belief that the true value of the answer is exactly in A excluding any of its subsets. Since A can be a subset with more than one element, DS theory can be regarded as a generalization of probability theory in which a probability value has to be assigned to individual elements. When multiple mass functions are provided from independent sources on the same frame of discernment, the combined impact of these mass functions is obtained using a mathematical formula called *Dempster's combination rule*. DS theory provides a flexible way to model uncertain information and a convenient mechanism to combine two or more distinct pieces of evidence [17,18].

Possibility Theory: Possibility Theory was developed out of Zadeh's fuzzy set theory [19], it is a simple yet powerful theory for modeling and reasoning with uncertain and imprecise knowledge or information. At the semantic level, a basic function in Possibility Theory [20] is a possibility distribution denoted as π which assigns each possible word in the frame of discernment Ω - a value in $[0, 1]$ (or a set of graded values). From a possibility distribution, two measures are derived, a possibility measure (demoted as Π) and a necessity measure (denoted as N). The former estimates to what extent the true event is believed to be in the subset and the latter evaluates the degree of necessity that the subset is true. In terms of merging, there are two main families of merging operators for merging possibility distributions, namely, conjunctive and disjunctive. A typical conjunctive merging operation is the minimum (min) and a typical disjunctive one is the maximum (max).

3 Ontology Matching

Many mapping approaches make use of different aspects of information to discover mappings between ontologies. In this paper, we design our mapping method by utilizing three different matchers, two of which are name-based matchers and one is a structure-based matcher.

3.1 Name-Based Matchers

They are often used to match names and name descriptions of ontology entities. The names of ontology entities are composed of several words, so first we adopt two different matchers based on name-based method: *Edit distance-based matcher* and *Linguistic-based matcher* to compute similarity of two words, then we exploit a method to compute a similarity of the names of ontology entities based on this.

Edit distance-based matcher: *Edit distance* is a simply implemented method to compare the degree of similarity of two words. It takes two strings and computes the edit distance between these two strings. That is, the number of insertions, deletions, and substitutions of characters required to transform one string into another. For example, the edit distance between **test** and **tent** is 1. In this paper, we develop an edit distance-based matcher which uses edit distance method to compute the similarity between two words. The similarity measurement between words w_1 and w_2 is defined as:

$$sim_{ed}(w_1, w_2) = \frac{1}{1 + ed(w_1, w_2)} \tag{1}$$

where $ed(w_1, w_2)$ denotes the edit distance of two words. We choose the form stated above because it returns a similarity value in [0,1].

Linguistic-based matcher: *Linguistic-based matcher* uses common knowledge or domain specific thesauri to match words and this kind of matchers has been used in many papers [21,22]. In this paper, we use an electronic lexicon WordNet for calculating the similarity values between words. WordNet is a lexical database developed by Princeton University which is now commonly viewed as an ontology for natural language concepts. It is organized into taxonomic hierarchies. Nouns, verbs, adjectives and adverbs are grouped into synonym sets (synsets), and the synsets are organized into senses (i.e., corresponding to different meanings of the same concept). The synsets are related to other synsets at the higher or lower levels in the hierarchy by different types of relationships. The most common relationships are the Hyponym/Hypernym (i.e., Is-A relationships) and the Meronym/Holonym (i.e., Part-Of relationships) [23]. In this paper, we only use the Hyponym/Hypernym relationships from WordNet.

Lin in [15] proposed a probabilistic model which depends on corpus statistics to calculate the similarity values between words using the WordNet. This method is based on statistical analysis of corpora, so it considers the probability of $word_1$ ($sense_1$) and $word_2$ ($sense_2$) and their most specific common subsumer $lso(w_1, w_2)$ appearing in the general corpus. However, since the words in given ontologies are usually application specific, this general corpus statistics obtained using the WordNet can not reflect the real possibility of domain-specific words. To improve Lin's method, we propose to calculate a punishment coefficient according to the ideas in the path length method [24]. The path length method regards WordNet as a graph and measures the similarity between two concepts (words) by identifying the minimum number of edges linking the concepts. It provides a simple approach to calculating similarity values and does not suffer from the disadvantage that Lin's method does, so we integrate Lin's method and a punishment coefficient to calculate the similarity values between words. First, we outline Lin's approach. The main formulas in this method are as follows:

$$sim_{Lin}(s_1, s_2) = \frac{2 \cdot \log(p(s_1, s_2))}{\log(p(s_1)) + \log(p(s_2))} \tag{2}$$

$$p(s) = \frac{freq(s)}{N} \tag{3}$$

$$freq(s) = \sum_{n \in words(s)} count(n) \tag{4}$$

where: $p(s_1, s_2)$ is the probability that the same hypernym of sense s_1 and sense s_2 occurs, $freq(s)$ denotes the word counts in sense s, $p(s)$ expresses the probability that sense s occurs in some synset and N is the total number of words in WordNet.

The punishment coefficient which is based on the theory of path length of WordNet is denoted as: $\frac{1}{2}\alpha^l$. Its meaning is explained as follows: α is a constant between 0 and 1 and is used to adjust the decrease of the degree of similarity between two senses when the path length between them is deepened and l expresses the longest distance either sense s_1 or sense s_2 passes by in a hierarchical hypernym structure. Because sense s_1 and sense s_2 occupy one of the common branches, this value has to be halved.

Therefore in our method, the similarity value calculated by Lin's method is adjusted with this coefficient to reflect more accurate degree between two senses s_1 and s_2. The revised calculation is:

$$sim_{new}(s_1, s_2) = \frac{2 \cdot \log(p(s_1, s_2))}{\log(p(s_1)) + \log(p(s_2))} \bullet \frac{1}{2}\alpha^l \tag{5}$$

Word w_1 and word w_2 may have many senses, we use $s(w_1)$ and $s(w_2)$ to denote the sets of senses for word w_1 and word w_2 respectively as $s(w_1) = \{s_{1i} \mid i = 1, 2, ..., m\}, s(w_2) = \{s_{1j} \mid j = 1, 2, ..., n\}$. where the numbers of senses that word w_1 and word w_2 contain are m and n. We decide to choose the maximum similarity value between two words w_1 and w_2, so the similarity between words is:

$$sim(w_1, w_2) = \max(sim_{new}(s_{1i}, s_{2j})),\ 1 \leq i \leq m, 1 \leq j \leq n \tag{6}$$

Calculating similarities of names of ontology entities: We can compute similarities between pairs of words according to two matchers stated above, next we calculate similarities of names of ontology entities based on the results obtained from the two matchers separately. The names of ontology entities are composed of several words, for instance, **PersonList**, actually is **Person** and **List**. We preprocess these kinds of names before we start to calculate the similarities of these names. We split a phrase (name of entity) and put the individual words into a set like $set = \{Person, List\}$ and then we deal with these words as follows:

1. Calculate similarities of every pair of words within both sets by using one of the matchers (Edit distance-based matcher or Linguistic-based matcher).
2. For each word in one set, compute similarity values between this word and every word from the other set and then pick out the largest similarity value. Finally attach this value to the word. Repeat this step until all of the words in the two sets have their own values.
3. Compute the final degree of similarity of names using the sum of similarity values of all words from two sets divided by the total counts of all words.

For example, we calculate similarity of two phrases: **PersonName** and **Person-Sex**. First, we split these two phrases into two sets: $set_1 = \{Person, Name\}$, $set_2 = \{Person, Sex\}$. Second, we calculate similarity values of each pair from two sets, such as the similarity value between **Person** in set_1 and **Person** in set_2, the similarity value between **Person** in set_1 and **Sex** in set_2, then choose the largest value from these two values and attach this value to **Person** in set_1. Repeat this step until *Name*, *Person* (in set_2) and *Sex* have their own largest value. Finally, the sum of these four similarity values is divided by the total cardinality (i.e. four) of these words.

3.2 Structure-Based Matcher

We regard each ontology as a model of tree, and in terms of tree structure we propose a *Structure-based Matcher* which determines the similarity between two nodes (entities) based on the similarities of their father nodes and children nodes. Such similarity values are obtained using a path length method based on WordNet, so we first introduce the method. We take WordNet as a hierarchical structure and the idea of the path length method is to find the sum of the shortest path passing from two concepts (words) to their common hypernym. We measure the similarity between two words by using the inverse of the sum length of the shortest paths:

$$sim_{path}(w_1, w_2) = \frac{1}{llength + rlength} \tag{7}$$

where: *llength* is the shortest path from word node w_1 to its common hypernym with word node w_2 and *rlength* denotes the shortest path from w_2 to its common hypernym with w_1. After calculating similarities between words, we can obtain similarities between names of entities.

Given two names of entities which belong to different ontologies, we can calculate the values of $sim_{path}(e_{i_1j_1}, e_{i_2j_2})$, $sim_{path}(e^f_{i_1j_1}, e_{i_2j_2})$ and $sim_{path}(e^c_{i_1j_1}, e_{i_2j_2})$. Then our *Structure-based matcher* is defined to calculate similarities between two entities utilizing these values with suitable weights: α_1, α_2 and α_3

$$sim_{str}(e_{i_1j_1}, e_{i_2j_2}) = \begin{cases} \alpha_1 * sim_{path}(e_{i_1j_1}, e_{i_2j_2}) + \alpha_2 * sim_{path}(e^f_{i_1j_1}, e_{i_2j_2}) + \\ \alpha_3 * \sum sim_{path}(e^c_{i_1j_1}, e_{i_2j_2}) \\ \exists father\ node\ and\ children\ nodes; \\ \alpha_1 * sim_{path}(e_{i_1j_1}, e_{i_2j_2}) + \alpha_2 * sim_{path}(e^f_{i_1j_1}, e_{i_2j_2}) \\ \exists father\ node\ and\ \nexists children\ nodes; \\ \alpha_1 * sim_{path}(e_{i_1j_1}, e_{i_2j_2}) + \alpha_3 * \sum sim_{path}(e^c_{i_1j_1}, e_{i_2j_2}) \\ \exists children\ nodes\ and\ \nexists father\ nodes; \\ \alpha_1 * sim_{path}(e_{i_1j_1}, e_{i_2j_2}) \\ \nexists father\ node\ and\ children\ nodes. \end{cases} \tag{8}$$

where α_1,α_2,α_3 separately denotes different weights distributed to similarities between $e_{i_1j_1}$ and $e_{i_2j_2}$, the father node of $e_{i_1j_1}$ and $e_{i_2j_2}$, a child node of $e_{i_1j_1}$ and $e_{i_2j_2}$. In formula (8), $\sum \alpha_i = 1$. We assign different values to these three weights as follows

$$\begin{cases} \alpha_1 = 0.5, \alpha_2 = 0.3, \alpha_3 = 0.2 & \exists father\ node\ and\ children\ nodes; \\ \alpha_1 = 0.5, \alpha_2 = 0.5 & \exists father\ node\ and\ \nexists children\ nodes; \\ \alpha_1 = 0.5, \alpha_3 = 0.5 & \exists children\ nodes\ and\ \nexists father\ node; \\ \alpha_1 = 1, & \nexists father\ node\ and\ children\ nodes. \end{cases}$$

3.3 Combining Mapping Results from Three Matchers

Using Dempster Shafer Theory of Evidence to combine the three matchers: We deploy DS theory to model and combine the outputs from the three ontology matchers described above in Sections 3.1 and 3.2.

Definition 1 (Frame of Discernment). *A set is called a frame of discernment (or simply a frame) if it contains mutually exclusive and exhaustive possible answers to a question. The set is usually denoted as Θ.*

Definition 2 (Mass Function). *A function m: is called a mass function on frame Θ if it satisfies the following two conditions:*

1. $m(\emptyset) = 0$
2. $\sum_A m(A) = 1$

where $\emptyset$ is the empty set and A is a subset of Θ.

Definition 3 (Dempster's Combination Rule). *If m_1 and m_2 are two mass functions on frame Θ from distinct sources, then $m = m_1 \oplus m_2$ is the resulting mass function after combing m_1 and m_2.*

In terms of ontology mapping, let O_1 and O_2 be two ontologies. For an entity $e_{i_1j_1}$ in O_1, we get its mappings with all the names in O_2, and the frame of discernment is $\Theta = e_{i_1j_1} \times O_2$.

Based on this frame, we have three mass functions m_1, m_2 and m_3 representing the normalized similarity values which are in [0,1] of all the possible mappings between $e_{i_1j_1}$ and all the entities in O_2 form the three matchers respectively. In this situation, we interpret the similarity value between a pair of names as the mass value assigned to this pair, an element of the frame. After combining these three mass functions using Dempster's combination rule, a unified mapping result is obtained taking into account the result from each matcher.

Using Possibility Theory to combine the three matchers: Possibility theory and the body of aggregation operations from fuzzy set theory provide some tools to address the problem of merging information coming from several sources. In possibility theory, a possibility distribution $\pi_1(u) : \Theta \rightarrow [0, 1]$ assigns each element in Θ a value in $[0, 1]$ representing the possibility that this element is the true world, where Θ is a frame of discernment. There are two families of merging operators to combine two possibility distributions: the conjunctive operators (e.g., minimum operator) and the disjunctive operators (e.g., the maximum operator) [25]. We use the normalized minimum operator to combine two sets of matching data.

Definition 4. *Let π_1 and π_2 be two possibility distributions and π be the combined distribution with minimum operator, then*

$$\forall\omega, \pi(\omega) = min(\pi_1(\omega), \pi_2(\omega)) \tag{9}$$

Definition 5. *Let the degree of consistency of π_1 and π_2 be defined as*

$$h(\pi_1, \pi_2) = sup_{\omega\in\Omega}\pi_1(\omega) * \pi_2(\omega) = \max(\min(\pi_1(\omega), \pi_2(\omega))) \tag{10}$$

When using this theory, we interpret the similarity values as degrees of possibility of element in a frame - a frame of the form $e_{i_1j_1} \times O_2$, where $e_{i_1j_1}$ is an entity in O_1. From the three matchers, we get three possibility distributions π_1, π_2 and π_3 and we combine them using the minimum operator as showed above. An advantage of using this theory is that we do not have the restriction that the two pieces of information must come from distinct sources as required by Dempster's combination rule.

4 Experiments

4.1 Dataset

We have proposed two different ways to combine mapping results from three matchers. We now present the experimental results that demonstrate the performance of our matchers and combination methods on the OAEI 2006 Benchmark Tests. In our experiments, we only focus on classes and properties in ontologies.

Generally, almost all the benchmark tests in OAEI 2006 describe Bibliographic references except Test 102 which is about wine and they can be divided into five groups [26] in terms of their characteristics: Test 101-104, Test 201-210, Test 221-247, Test 248-266 and Test 301-304. A brief description is given below.

- **Test 101-104:** These tests contain classes and properties with either exactly the same or totally different names.
- **Test 201-210:** The tests in this group change some linguistic features compared to Test 101-104. For example, some of the ontologies in this group have no comments or names, names of some ontology have been replaced with synonyms.
- **Test 221-247:** The structures of the ontologies have been changed but the linguistic features have been maintained.
- **Test 248-266:** Both the structures and names of ontologies have been changed and the tests in this group are the most difficult cases in all the benchmark tests.
- **Test 301-304:** Four real-life ontologies about BibTeX.

In our evaluation, we choose **Test 101**, **Test 103**, **Test 104**, **Test 205**, **Test 223** and **Test 302** of OAEI 2006 Benchmark Tests and take **Test 101** as the reference ontology. All the other ontologies are compared with **Test 101**. The reason for selecting them as test cases are:

1. They are well known in the field of ontology mapping.
2. They have normal classes, object properties and datatype properties hierarchy, so we can obtain regular results by using these three matchers.
3. Test 101-104 have similar structures and names of entities, while the structures and names of Test 205, 223, 302 are different from the reference ontology, i.e. Test 101, so we can use these datasets to test performance of matchers and combination methods.

For Test 101, Test 103, Test 104 and Test 205 each test contains 33 classes and 64 properties; Test 223 has 66 classes and 65 properties; Test 302 has 13 classes and 30 properties.

4.2 Experimental Evaluation Metrics

To evaluate the performance of mapping, like many other papers that use retrieval metrics, *Precision*, *Recall* and *f-measure* to measure a mapping method, we use these measures to evaluate our methods as well. *Precision* describes the number of correctly identified mappings versus the number of all mappings discovered by the three approaches. *Recall* measures the number of correctly identified mappings versus the number of possible existing mappings discovered by hand. *f-measure* is defined as a combination of the *Precision* and *Recall*. Its score is in the range $[0, 1]$.

$$precision = \frac{|m_m \cap m_a|}{|m_a|} \tag{11}$$

$$recall = \frac{|m_m \cap m_a|}{|m_m|} \tag{12}$$

$$f - measure = \frac{2 * precision * recall}{precision + recall} \tag{13}$$

where m_m and m_a represent the mappings discovered by hand and by a method proposed in our paper respectively.

4.3 Single Matchers vs. Combination of Matchers

Figure 1 shows the *f-measure* of the three single matchers and combination methods on the five datasets, which includes Test 101 vs Test 103, Test 101 vs Test 104, Test 101 vs Test 205, Test 101 vs Test 223, Test 101 vs Test 302. Each single matcher is marked as follows: *Ed* for Edit distance-based matcher; *L* for Linguistic-base matcher; *S* for *Structure-based matcher*; *DS* for Dempster's combination rule; *PT* for the minimum merging operator in Possibility Theory.

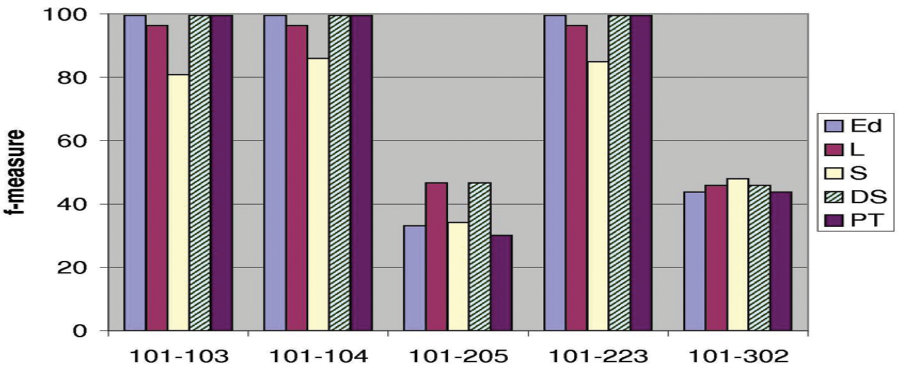

Fig. 1. Single matchers vs. combination methods

From Figure 1, we can see that almost for every group of tests, the f-measures of results using Dempster's combination rule is better than or equivalent to that of a single matcher, the minimum operator of Possibility Theory performs well except for Test101 vs Test 205, which has the results lower than other single matchers results. For Test 101 vs Test 103, Test 101 vs Test 104 and Test 101 vs Test 223, *Edit distance-based matcher* obtains better results than the other two single matchers because these three groups of tests have almost the same names of entities. For Test 101 vs Test 205 and Test 101 vs Test 302, *Linguistic-based matcher* gets better results than the other two single matchers because *Linguistic-based matcher* can obtain good results for those different names which have the same meaning.

4.4 Comparison of Systems Utilizing Different Matchers

We use the combination mechanisms in both DS theory and Possibility Theory to combine the matching results from our three matchers. We now compare the outputs from the two combination rules to the results obtained from *falcon, ola* and *ctxMatch2-1* algorithms which were used in the EON 2005 Ontology

Table 1. Comparison of Experiment Results

Datasets	DS			PT			falcon			ola			ctxMatch2-1		
	p	r	f	p	r	f	p	r	f	p	r	f	p	r	f
101-103	100	98.97	99.48	100	98.97	99.48	100	100	100	100	100	100	87	34	48
101-104	100	98.97	99.48	100	98.97	99.48	100	100	100	100	100	100	87	34	48.89
101-205	46.88	46.39	46.63	30.29	29.90	30.09	88	87	87.5	43	42	42.5	36	4	7.2
101-223	100	98.97	99.48	100	98.97	99.48	100	100	100	100	100	100	83	31	45.14
101-302	45.83	45.83	45.83	43.75	43.75	43.75	97	67	79.26	37	33	34.89	0	0	0

Alignment Contest [1], and the details are given in Table 1. In Table 1, p for precision, r for recall, f for f-measure, DS for Dempster's combination rule, and PT for the minimum merging operator in Possibility Theory. Overall, we believe that the two combination rules we use are very satisfactory, with Dempster's combination rule outperforming the minimum rule in Possibility Theory slightly for pair 101 vs 205. Although on every pair of ontologies, our results of two combination rules are less ideal than the *falcon* system, however, our results are better than *ola* system on two out of five pairs of matching, and the results are much better than the *ctxMatch2-1* system. The performances of these five different approaches are all very good for Test 101 vs 103 and vs 104 and Test 101 vs Test 223, but none of the systems performed exceptionally well for Test 205 and Test 302. Below we analyze the reasons for this.

For Test 101 vs 103 and vs 104, the two ontologies to be matched contain classes and properties with exactly the same names and structures, so every system that deploys the computation of similarities of names of entities can get good results. Test 223 has more classes than Test 101 to 104 and the structure of its ontology is changed although the linguistic features remains the same and its class names are generally the same as the reference ontology. These similarities in the linguistic features and class names enable these matching systems to perform well.

Test 205 describes the same kind of information as other ontologies, i.e. publications, however, the class names in it are very different from those in the reference ontology Test 101. Even though we employed three matchers to calculate similarities between names, the results are still not very satisfactory. Test 302 is a real-life BibTeX ontology which also includes different words compared to Test 101 describing publications so the results are similar to Test 205, so we do not get good results from these two datasets.

Our linguistic-based matcher does not consider the structures between words and assumes that all the words are equally important. However, different words in a name have different degrees of importance, therefore, this is one aspect that we will need to improve further. In our structure-based matcher, we adopt the idea of assigning different weights to different aspects when matching two words. The weights are predefined but we think these could be learned in our next step of research.

[1] http://oaei.ontologymatching.org/2005/results/

5 Related Work

In a mapping process, if only syntactic or element-level matching is performed, as in the case for name matching without the use of a thesaurus, inaccuracies can occur [27]. This affects the results of mapping, but so far only a few ontology mapping methods have considered dealing with the uncertainty issue.

Nagy et al [10] and Besana [11] both recognized the importance of uncertainty in ontology mapping, and both of them used DS theory to assist mapping. They believed that different matchers have uncertainties associated with them, so they combine the results obtained from different matchers using DS theory and it is possible to give a uniform interpretation, consistent with the uncertainty inherited in the problem. Although Nagy et al utilized Dempster's combination rule into ontology mapping, it is not clear how they applied the theory. For example, they did not explicitly define a *Frame of Discernment*. Besana [11] exploited DS theory into a more complicated process. He considers not only combining ontology matching results using DS theory, but also uncertain mappings using DS theory.

In [12] a Bayesian Networks based approach was designed and a system called BayesOWL was proposed. In this approach, the source and target ontologies are first translated into Bayesian networks (BN); the concept mapping between the two ontologies are treated as evidential reasoning between the two translated BNs. Probabilities, which are required for constructing conditional probability tables (CPT) during translation and for measuring semantic similarity during mapping, are learned using text classification techniques, where each concept in an ontology is associated with a set of semantically relevant text documents, which are obtained by ontology guided web mining. This approach used Bayesian Networks, but the networks are sophisticated and it is not easy to construct them from an ontology expressed by OWL.

Holi and Hyvönen [13] observed that in the real world, concepts are not always subsumed by each other, and cannot always be organized in a crisp subsumption hierarchies. Many concepts only partly overlap each other, so they present a new probabilistic method to model conceptual overlap in taxonomies, and an algorithm to compute the overlap between a selected concept and other concepts of a taxonomy by using Bayesian networks. This method focused on the uncertainty of description languages of ontologies. Although it is not related to the mapping, it can be used as a measure of semantic distance between concepts.

Zhao et al [14] proposed a novel similarity measure method based on rough set theory and formal concept analysis (RFCA) to realize ontology mapping tasks. The authors combined rough set theory into the similarity computation formula of formal concept analysis (FCA). Although the authors did not consider uncertainty in the process of mapping explicitly, they applied the rough set theory to measure the similarities of concepts of ontologies. So, in some case, they did consider the uncertainty problem.

6 Conclusion

In this paper, we utilize three independent matchers to deal with ontology mapping and they are: *Edit distance-based matcher*, *Linguistic-based matcher* and *Structure-based matcher*. In the *Linguistic-based matcher*, we improved Lin's method which computes similarity value between words. In the *Structure-based matcher*, we adopt the structure of ontology to calculate similarity values between two entities and it considers the impact of the direct relative nodes (father and/or children) to one entity.

Following this, we investigated how the problem of uncertainty in ontology mapping can be dealt with. We considered both the Dempster-Shafer theory and Possibility Theory to combine the uncertain mapping results from different matchers stated above. We applied our ontology mapping systems (two combination rules with three matchers) to a set of ontologies used for ontology mapping competitions. The experimental results show that it is efficient and feasible to exploit these uncertainty theories to deal with uncertainty factors in the process of ontology mapping.

As future work, on the one hand, we will design new matchers to handle some situations that are not considered here, for example, how to get accurate *n:1*, *1:n* or *n:n* mapping results. On the other hand, we will continue investigating the uncertainty issues in ontology mapping and consider how to use different uncertainty theories to deal with different situations in ontology mapping.

References

1. Ehrig, M., Sure, Y.: Ontology mapping - an integrated approach. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 76–91. Springer, Heidelberg (2004)
2. Ehrig, M., Staab, S.: Qom - quick ontology mapping. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 683–697. Springer, Heidelberg (2004)
3. Noy, N.F., Musen, M.A.: Prompt: Algorithm and tool for automated ontology merging and alignment. In: Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI 2000), pp. 450–455 (2000)
4. Noy, N.F., Musen, M.A.: Anchor-prompt: Using non- local context for semantic matching. In: Workshop on Ontologies and Information Sharing at the 17th International Joint Conference on Articial Intelligence (IJCAI 2001) (2001)
5. Kalfoglou, Y., Schorlemmer, W.M.: Information-flow-based ontology mapping. In: Proceedings of the International Federated Conferences (CoopIS/DOA/ODBASE 2002), pp. 1132–1151 (2002)
6. Su, X., Gulla, J.A.: Semantic enrichment for ontology mapping. In: Meziane, F., Métais, E. (eds.) NLDB 2004. LNCS, vol. 3136, pp. 217–228. Springer, Heidelberg (2004)
7. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. Journal of VLDB 10(4), 334–350 (2001)
8. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal of Data Semantics 4, 146–171 (2005)

9. Choi, N., Song, I.Y., Han, H.: A survey on ontology mapping. SIGMOD Record. 35(3), 34–41 (2006)
10. Nagy, M., Maria Vargas-Vera, E.M.: Dssim-ontology mapping with uncertainty. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L. (eds.) ISWC 2006. LNCS, vol. 4273, Springer, Heidelberg (2006)
11. Besana, P.: A framework for combining ontology and schema matchers with dempster shafer. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 196–200. Springer, Heidelberg (2006)
12. Pan, R., Ding, Z., Yu, Y., Peng, Y.: A bayesian network approach to ontology mapping. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 563–577. Springer, Heidelberg (2005)
13. Markus Holi, E.H.: Modeling degrees of conceptual overlap in semantic web ontologies. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 98–99. Springer, Heidelberg (2005)
14. Zhao, Y., Wang, X., Halang, W.A.: Ontology mapping based on rough formal concept analysis. In: Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW'06), p. 180 (2006)
15. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning (ICML'98), pp. 296–304 (1998)
16. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition Journal 5(2), 199–220 (1993)
17. Liu, W.: Propositional, Probabilistic and Evidential Reasoning: Integrating Numerical and Symbolic Approaches (2001)
18. Liu, W.: Analyzing the degree of conflict among belief functions. Journal of Artificial Intelligence. 170(11), 909–924 (2006)
19. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. Journal of Fuzzy Sets and Systems 100, 9–34 (1999)
20. Dubois, D., Prade, H.: Possibility Theory (1988)
21. Tang, J., 0003, Y.L., Li, Z.: Multiple strategies detection in ontology mapping. In: Proceedings of the 14th international conference on World Wide Web (WWW'05) (Special interest tracks and posters) pp. 1040–1041 (2005)
22. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with cupid. In: Proceedings of 27th International Conference on Very Large Data Bases (VLDB'01), pp. 49–58 (2001)
23. Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E.G.M., Milios, E.E.: Semantic similarity methods in wordnet and their application to information retrieval on the web. In: Proceedings of 7th ACM International Workshop on Web Information and Data Management (WIDM 2005), pp. 10–16 (2005)
24. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of 14th International Joint Conference for Artificial Intelligence (IJCAI'95), pp. 448–453 (1995)
25. Dubois, D., Prade, H.: Possibility theory in information fusion. In: the 11th International Workshop on Non-Monotonic Reasoning (NMR'02), pp. 103–116 (2002)
26. Qu, Y., Hu, W., Cheng, G.: Constructing virtual documents for ontology matching. In: Proceedings of 15th World Wide Web Conference (WWW'06), pp. 23–31 (2006)
27. Cross, V.: Uncertainty in the automation of ontology matching. In: Proceedings of the 4th International Symposium on Uncertainty Modeling and Analysis (ISUMA'03), pp. 135–140 (2003)