A toward framework for generic uncertainty management

Ernesto Damiani   Paolo Ceravolo   Marcello Leida

Università degli Studi di Milano, Dipartimento di Tecnologie dell'Informazione
via Bramante 65, 26013 Crema (CR), Italy
Email: {ernesto.damiani, paolo.ceravolo, marcello.leida}@unimi.it

Abstract— *The need for an automatic inference process able to deal with information coming from unreliable sources is becoming a relevant issue both on corporate networks and on the open Web. Mathematical theories to reason with uncertain information have been successfully applied in several situations, but each one of these models is tailored to deal with a specific semantics of uncertainty. In this paper, we put forward the idea of using explicit representations of the different types of uncertainty for partitioning the inference process into parts. By coordinating multiple independent reasoning processes, we are sometimes able to apply a specific model to each type of uncertain information, and recombine the final results via a suitable reconciliation process. We validated our approach applying it to the classic* ***schema matching problem****, and using the* ***Ontology Alignment Evaluation Initiative****, (OAEI) tests to assess the results.*

Keywords— Uncertainty, Ontology Matching, Reasoning, Rules.

1 Introduction

The problem of automatic inference is one of the most challenging problem in computer science [1], becoming even harder when knowledge is uncertain, due to lack of reliability of the source of information, approximation, dependencies and other factors. While many mathematical models for reasoning on uncertain information have been proposed, the general problem of handling and interpretation of uncertain knowledge is still to be solved. In this paper, we put forward the idea of using explicit representations of the different uncertainties present in the knowledge base according to different uncertainty models, coordinating multiple independent reasoning processes. By splitting the inference process into parts, we are able to apply a specific model to each type of uncertain information, recombining the final results via a suitable reconciliation process. Although the interoperability among multiple inference models been studied [2, 3], in the literature we are not aware of any hybrid reasoning processes which can handle the flexible integration of different models. As a proof of concept of this approach we present a semantics-aware matching strategy, that we apply to the well-known problem of ontology alignment [4]. The paper is structured as follows: Section 2 introduces the problem of uncertain information in knowledge management, briefly presenting the different types of uncertainty and the mathematical models used for the inference process. The section also introduces the need for an explicit representation of the various types of uncertainty, referring to the Ontology of Uncertainty [5] proposed by W3C's UR3W-XG incubator group[1]. Section 3 presents a case study, applying our technique to the classic *schema matching problem*, testing it via the Ontology Alignment Evaluation Initiative (OAEI) tests and comparing the results to the participants to the OAEI 2007 contest [4]. Conclusions and future work on our framework are presented in Section 5.

[1] `http://www.w3.org/2005/Incubator/urw3/`

2 Uncertain Information Representation and Reasoning

Experience has shown that the open Web and other platforms for hosting user-generated content can provide little quality control at content production time. As a result, most publicly available information can be considered *uncertain* to some degree. In order to clarify the notion of uncertainty, it is important to distinguish between **degrees of truth** and **degrees of uncertainty** in the information [6]. A degree of truth can be defined as the degree of compatibility between a statement and a knowledge base, which is limited to what the system knows about reality: a statement S is true if this assumption agrees with the set of statements in the knowledge base. Instead, Uncertainty of a statement arises when the knowledge base does not provide sufficient information to decide if a statement is true or false. Therefore uncertainty falls at a meta-level with respect to truth [7]. In case of truth values, we briefly mention two major theories [7]: *Classical two-valued logic* and *Fuzzy Logic*. In the first case, a statement's truth value can only assume one of two values [8], namely `0` or `1`. In the second case [9], truth values belong to the entire interval $[0, 1]$. For uncertainty representation, we distinguish between *Probability* and *Possibility theory*. The degree of probability associated to a statement is a typical example of gradual uncertainty. In sentences like: *"The player tossing a coin* wins *with 50% probability"*, "The player *wins*" is a true statement: it cannot happen that a player "half wins". The "50%" at the end of the sentence is not the statement truth value, but its level of uncertainty. *Possibility theory* is an alternative to probability theory, which separates the uncertainty of statements in possibilities and necessities [7]. Uncertainty can be classified as **Epistemic**, if it comes from the limited knowledge of the agent that generates the assertion or **Aleatory** if it is intrinsic in the observed world. Depending on the features of the agent that generates uncertain statements, is possible to identify two different types of uncertainty: **Objective** if the uncertainty derives from a repeatable observation and **Subjective** if the uncertainty in the information is derived from an informal evaluation. Furthermore, uncertainty can depend on the type of statement it is associated to: **Ambiguous**, **Inconsistent**,**Vague**, **Incomplete** and **Empiric**.

For our purposes, uncertainty can be represented as an annotation about a statement, expressing the level of certainty about it. We shall call "uncertain information" the triple

ISBN: 978-989-95079-6-8

(S, t, l) formed by a statement S, its truth value t and its corresponding uncertainty level l. It is important to remark that in many practical scenarios, we may encounter statements whose uncertainty levels have diverse semantics, especially when uncertain information is generated by multiple unsupervised processes. For example, Web-based weather forecasting services provide uncertain information in different forms; the uncertainty can be possibilistic or probabilistic (*Cloudy:"50%"*, *Rain:"10%"*) while truth value ranges can be classical (*Rainy*, *Fair*) or fuzzy (*Partially Covered*, *Heavy Rain*). Traditional approaches to uncertain reasoning support extensions of logic models dealing with inference on statements and their truth values, including mathematical theories able to deal with the uncertainty levels. In many cases, however, handling uncertainty hs an impact on the complexity and even the decidability of the inference problem. Scientific areas very active in the integration of classical logic with mathematical theories dealing with uncertainty is Artificial Intelligence and Knowledge Representation. Recently the effort was concentrated on the languages for the Web, such as in particular Semantic Web standards. For example, several probabilistic extensions of *Description Logics* (DL) [10] and *First Order Logic* (FOL) are available. Here, we shall focus the discussion on Description Logics, which are the logical model underpinning the OWL-Lite and OWL-DL ontology languages [11] used on the Semantic Web [12]. OWL-Lite and OWL-DL correspond respectively to $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$ Description Logics respectively, which are known to be tractable. A sound and complete fragment of $\mathcal{SROIQ}(\mathbf{D})$ is $\mathcal{SHOQ}(\mathbf{D})$ [13]. This fragment can be extended to handle uncertainty; in [14] Lukasiewicz defines a Probabilistic Description Logic $\mathcal{P} - \mathcal{SHOQ}(\mathbf{D})$. Based on Lukasiewicz's Probabilistic Description Logics, Klinov in [15] has implemented the probabilistic reasoner Pronto, that can reason with ontologies where a probability interval is assigned to statements, specifying the probability that a certain statement is true. As far as truth-values are concerned, the work of Straccia and Bobillo [16] extends the classical two-valued Description Logics to fuzzy sets. Namely, the authors present an extension of $\mathcal{SHIF}(\mathbf{D})$ Description Logics to the fuzzy case, dealing with different definitions for the logic operators (Zadeh logic, Lukasiewicz logic and Classical Logic); this approach, moreover, provides support to backward reasoning in case of Classical Logic semantics.

At first sight, one might hope that uncertainty representations and truth value ranges can be freely mixed according to the characteristics of the problem at hand. On the Web, vague information is usually modelled using fuzzy truth-values, knowledge uncertainty due to incomplete or defective observations is represented by Probability theory, and uncertainty arising from common sense knowledge and guessing can be handled with Possibility theory [17].

Unfortunately, the problem of dealing simultaneously with probability-based uncertainty and fuzzy truth values has been widely treated in literature but, as stated in [7] probability and possibility theories are not fully compositional with respect to all the logical connectives, without a relevant loss of expressiveness. This consideration leads to the consequence that uncertain calculi and degrees of truth are not fully compositional either. Nevertheless, some work in this direction has been proposed, by imposing restrictions to the expressiveness of the logics. The most relevant studies are: [18, 19] where the authors define *Probabilistic Description Logics Programs* (PDLP) by combining stratified fuzzy Description Logics programs with respect to degrees of probabilities in a unified framework. In [20] a definition of possibilistic fuzzy Description Logics has been proposed by associating weights, representing degrees of uncertainty, to the fuzzy Description Logic formulas. An extension of the fuzzy Description Logics in the field of Possibility theory has been presented also in [21] by annotating logic axioms with possibilities and necessity measures; by extending the approach presented in [20].

It is also important to underline that different approaches can be used to tackle the same type of uncertainty (i.e. in case of incomplete information is possible to use Possibilistic or Probabilistic theory) the choice of the best theory depends on the context. For instance, in [22] the authors use Dempster-Shafer's beliefs theory to resolve inconsistencies.

2.1 *Ontology of Uncertainty*

As mentioned in Section 2 uncertainty is generated from different situations and represented under different semantics. it is possible to create a classification of assertions based on several criteria: *nature*, *derivation*, *temporal validity* and *type*. Nature of uncertainty can be divided in *epistemic* and *aleatory*; *objective* and *subjective*; based on the temporal validity of a statement that can be valid for a period of time or can be valid always. A statement is *contingent* if refers to a particular situation or instant; in the second case a statement refers to situations that summarize trends (e.g. laws of physics, common sense knowledge, statistical knowledge) and is classified as *generic* statement. The nature of uncertainty, moreover, also depends on the statement it is attached to. Statements can be *ambiguous* in case the statement can be represented in different worlds with more than one interpretation, *inconsistent* if there is no possible world where the statement can hold, *vague*, and *incomplete* in case the knowledge about the observed world do not provide enough information to take a decision. Finally, a statement is *empiric* when is satisfied at least in one world. A first effort toward capturing all aspects of uncertainty is the *Ontology of Uncertainty*, published by the UR3W-XG incubator group[5]. This ontology tries to capture the nature, type and source of uncertainty that are specific of an assertion and allows moreover to relate the assertion to the correct computational model of inference. Statements, in the Ontology of Uncertainty, are represented by the concept `Sentence`, that provides information about the source (`Agent`) of the assertion, the subject (`World`) of the assertion and the semantics information about the `Uncertainty` model related to the assertions. There is not much to say about the two concepts `Agent` and `World`, which respectively represent the producer and the subject of an assertion. More interesting is the case of concept `Uncertainty`, the central concept of the ontology. This concept is related to all the various elements used to classify a `Sentence` under precise semantics. Other concepts related to `Uncertainty` are then used to describe type, derivation, validity and nature of the statement[2]. The ontology provides a generic meta-model rep-

[2]Currently, the Ontology of Uncertainty defined by the URW3-XG does not include a concept of `Validity` related to the temporal

ISBN: 978-989-95079-6-8

resented in OWL-DL [11] for representing uncertainty associated to various assertions and provides some use case scenarios, where, according to the semantics of the uncertainty, the correct inference model is selected. Unfortunately, the document [5] produced by the URW3-XG incubator group does not specify how to deal with situations where more than one model is involved in the inference process.

2.2 Using the Ontology of Uncertainty to support Reasoning

The Ontology of Uncertainty provides information on which mathematical model of uncertainty can be employed for managing a specific set of statements. When more than one model is involved in the reasoning process, the problem of integrating the results of multiple inference processes arises : if the subsets of statements handled by each model are disjoint, i.e. inferences are independent form each other, however, there are no particular problems in re-conciliating the results of the various reasoning processes;s. Some work in this direction has been carried out by the *Rule Interchange Format* (RIF[3]). In [2] the authors propose a framework for sharing information between three different models of uncertainty, where the fuzzy linguistic truth values are propagated through the three models in a non-monotonic way, by exploiting the extension principle [23] and aggregation of linguistic values. This approach is promising but is grounded to fixed fuzzy values (linguistic truth) that are used by all the different models and then aggregated according to non-monotonic rules.

In our approach, instead, we make use of the Ontology of Uncertainty as a way to model different types of uncertainty in a unified framework. The inference process involves three different steps: the first step is to partition the knowledge base in subsets according to the specific model, the second step is to carry out independent inferences, and the third one aggregates the results of the independent inference processes, following the *First Inference Then Aggregation* (FITA) approach [24], which also supports parallel reasoning. In our strategy the various reasoning processes are independent; we use the Ontology of Uncertainty classification to divide the various matching relations according to the uncertainty model to be used for the reasoning process. This way, the reasoners can be modelled as parallel processes. Directives on how to divide the heterogeneous knowledge base and how to recombine the results of the different reasoning processes are explicitly specified as DL-Safe Horn rules [25].

The knowledge base containing the information for our matching strategy is composed by a set of statements, generated independently by different sources, seen as instances of the concept `Sentence` in the Ontology of Uncertainty. To each statement S, information about uncertainty is associated by instantiating the ontology concept `Uncertainty` that defines the correct semantics. DL-Safe rules and SPARQL queries are largely involved in this process. A first set of DL-Safe rules is used to associate the statements to the correct type of uncertainty, and a set of SPARQL queries is used to divide the knowledge base according to the reasoning model associated to the statement. Finally a third set of rules is used to aggregate the results of the various reasoners. Currently, our approach requires the manual definition of rule sets. However, while the first set of rules is largely application dependent, the set of SPARQL queries and the third set of rules can be reused in many applications[4]. A detailed application of our approach is presented in Section 3.

3 A Semantics-aware Matching Strategy

Schema Matching is the time-honored problem of identifying the relations between the entities of two data source schemata. In case these schemata are represented as ontologies, this problem is also known as Ontology Alignment. In the literature different matching operators for a wide range of situations are available: the most exhaustive survey is [26].

Recent proposals tackle the schema matching problem by considering more than one matching operator at once and combining the final results through a matching strategy [27]. A **Matching Strategy** can be defined as *the process of transformation from a set of Matching Relations Mr to a new set Mr'*, while a **Matching Operator** can be defined as *a function that takes as input two schema elements and creates as output a Matching Relation between the two elements*.

So far, even if some logic-based approaches are available [28, 29], most strategies proposed in literature neither consider explicitly the semantics of the various matching operators nor the different meanings of the relations that they generate. Instead, our matching strategy explicitly models the semantics of different matching operators as a Description Logic. In our approach, matching relations are stored as instances of a domain ontology, describing our application scenario. This domain ontology is extended with the *Ontology of Uncertainty* [5], which is used to associate the respective uncertainty to the different types of assertions. The *classification* of the various uncertainty types is performed by applying SWRL rules [30, 25] to the knowledge base. Once the classification is performed, the knowledge base is divided (by a *splitting* process) according to the specific uncertainty model. Each one of this inference models is used to perform a classification of the various relations in order to discover the most reliable ones. Finally a *reconciliation* process aggregates the best results[5].

3.1 The Matching Ontology

A matching relation mr_i is a $1:1$ relation associating two elements (concepts or attributes) of the two ontologies to align by a relation r from a collection of set theory operators ($\equiv$,$\subset$,$\supset$,$\cap$,$\neq$) and a degree of matching δ represented as:

$$mr_i = \langle e_h, e_k, r, \delta \rangle \quad (1)$$

The relation r between the elements depends on the particular feature that is analysed by the matching operator: as an example, in case of *JaroWinkler* matching operator, the features are composed by the label of the elements e_h and e_k to

validity of an assertion, but its addition is straightforward.

[3]The mission of the Rule Interchange Format (RIF) Working Group is to produce W3C Recommendations for rules interchange. `http://www.w3.org/2005/rules/wiki/RIF_Working_Group`

[4]Although they may require some fine tuning. This especially true of the third set, which is closely related to the aggregation procedure.

[5]The *classification*, *splitting* and *reconciliation* process in our system are rule driven, but this is not a strict requirement: this phases of the strategy can also be inferred from a Description Logics reasoning process used to classify various instances.

ISBN: 978-989-95079-6-8

match, while in case of an *Instance-based* matching operator the features are composed by the instances related to the elements to match.

We generally consider a matching operator as a process that generates a matching relation mr_i represented by the relation r between the elements e_h, e_k in input, associated with the strength δ of the relation. A matching operator is defined by a function f used to extract a particular feature that is analysed by the operator, a function θ which generates the relation r and a function ϕ that is in charge of the creation of the value δ, which represents the *strength* of the relation:

$$\begin{aligned} mo &= \langle f, \theta, \phi \rangle \\ mo(e_h, e_k) &\rightarrow \langle e_h, e_k, r, \delta \rangle \end{aligned} \quad (2)$$

Obviously, different matching operators carry different semantics that needs to be considered in the matching generation process. Also, the same relation can be generated by different matching operators; but in this case, the syntax of the relations is identical while the semantics is different. Consequently, different theories can be adopted, in order to infer the most suitable matching relations: the relations generated by the data type based matching operator can be considered as *possibilities*, while the relations generated by the instance based and string based matching operator can be modelled as *necessities* [6]. As we have seen, the Matching Ontology models the information created by the various matching operators, as well as information describing each specific matching operator. The attributes, relations and concepts from the ontologies to align are stored in the Matching Ontology as instances of the concept `Element`, represented by a unique identifier (e.g. URI) which is used by the system to retrieve the element in the original ontologies. The concept `MatchingOperator`, in the Matching Ontology, represents a generic matching operator; the specific matching operators (e.g. JaroWinkler, WordNet, Instances) are modelled as subclasses of this concept. The various matching relations are defined as instances of the concept `MatchingRelation`, which is composed by the concept `RelatedElements` that has a `subject` and an `object` that represent the two elements that have been related by one or more matching operators; and by a `confidence` value representing the strength of the matching relation. Subclasses of the concept `Matching Relation` are created in order to specialize the matching relation according to the specific relation (e.g. `Equivalent`, `Subset`, `Superset`, `Disjoint`, `Intersection`).

Instances of the Matching Ontology are generated during the matching process. The various matching operators generate a set of matching relations in the form $\langle e_h, e_k, r, \delta \rangle$; these relations are stored in the Matching Ontology with a reference to the instance of the matching operator that has generated the relation. When the Matching Ontology is generated in this way, some information is certain (e.g. the information provided by the original ontologies), and some other information in uncertain under different semantics (e.g. the matching relations that can disagree or that may have a probability degree to consider). This consideration motivates the use of additional annotations able to model the semantics of uncertainty of the various matching relations.

[6] Note that this does not have a direct implication on the mappings to be created by the systems; for instance, it is clear that a necessary matching of strings does not imply a necessary mapping.

3.2 *Managing uncertainty in the matching strategy*

In a scenario like the one described in Section 3.1, we use the Ontology of Uncertainty to identify situations where is important to explicitly describe the type of uncertainty related to an assertion. First of all, the two ontologies (Matching Ontology and Ontology of Uncertainty) have to be linked somehow. Assertions in the Ontology of Uncertainty are represented by the concept `Sentence`. A sentence is `saidBy` an `Agent`, which we identify with the matching operator; moreover a sentence have also an object of the assertion (`saidAbout`) which is represented by the concept `World`. The two concepts `Agent` and `World` are then the linking point between the two ontologies; the link is defined by declaring `MatchingRelation` and `MatchingOperator` as sub-concepts of `World` and `Agent` respectively. Here, the Ontology of Uncertainty is used basically to drive the reasoning process: each type of uncertainty is processed by its specific reasoner and a final process, based on SWRL rules, integrates the results of the various reasoners. The application flow of the Matching Strategy operates as follows: a process takes as input the uncertain knowledge base generated by the matching operators, afterwards it divides the assertions according to their uncertainty and each sub part of the ontology is processed by its specific reasoner: in the system we consider a Probabilistic Description Logic reasoner, such as *Pronto* [15], a fuzzy Description Logic reasoner such as *FuzzyDL* [16] and a Defeasible Logic Reasoner such as *DR-Prolog* [3], but other models can be easily added. In our scenario the sources of information to be analysed, according to different uncertainty models are independent and no intersections among them has to be managed. This particular case allows a straightforward use of the Ontology of Uncertainty to drive the reasoning process, although in general the assumption of independence among the source of information is a lucky case.

The first part of the matching strategy is to assign to the various assertions (`Sentence`), the correct information about their uncertainty semantics. This information is classified according to a set of pre-defined SWRL rules that assigns the correct semantics in relation to several factors. The assignment is based on: (i) the `Agent` that has generated the relation; as an example: some agents can generate objective or subjective assertions: we can identify objective statements as necessities and subjective as possibilities; (ii) the presence of a degree of probability: as an example, sentences with a degree of probability can be handled with probabilistic theory models or with possibility theories. In this case is important to identify which statements need to be modelled with probability theory and which ones need to be modelled with possibility theory. (iii) the level of inconsistency among matching relations: as an example, if a sentence asserts that two elements are equivalent and another matching relation asserts that they are disjoint; (iv) the trustiness level of the matching operator: some operators are more reliable than others; (v) the level of detail of the assertion: the assertion created by a Data Type matching operator is more vague then an assertion created by a Regular Expression matching operator; (vi) the Data Type

ISBN: 978-989-95079-6-8

of the elements to match can be used to establish priorities between operators.

$$\begin{aligned} uncertainty &: Sentence(?sentence) \wedge \\ matching &: MatchingRelation(?matrel) \wedge \\ uncertainty &: saidBy(?sentence, ?operator) \wedge \\ matching &: MatchingOperator(?operator) \wedge \\ uncertainty &: saidAbout(?sentence, ?matrel) \wedge \\ uncertainty &: Objective(?derivation) \wedge \\ uncertainty &: Uncertainty(?uncer) \wedge \\ uncertainty &: hasUncertainty(?sentence, ?uncer) \rightarrow \\ uncertainty &: derivationType(?uncer, ?derivation) \end{aligned} \tag{3}$$

There are one or more rules for each specific uncertainty type, nature, model, derivation and temporal validity. An example of a rule that we use in this step of the matching strategy is reported in (3). In our strategy the rules are applied to the matching ontology with the use of a rule engine such as Jess[7]; the rules we use have to be restricted to the DL-Safe [25] subset to ensure tractable complexity of the reasoning processes. By applying the rules, all the matching relations are associated to their respective uncertainty.

At this point, a set of sub-knowledge bases is created by dividing the various instances of the concept `sentence` according to their uncertainty model. A SPARQL [32] query is used in this case to select the instances of the concept `sentence` that respect the desired restrictions. As an example the following SPARQL query returns the instances of `Sentence` that are associated to a probability theory model.

```
SELECT ?sentence ?type ?derivation ?nature ?model
WHERE {?sentence uncertainty:hasUncertainty ?uncertainty.
?uncertainty uncertainty:nature ?nature.
?uncertainty uncertainty:derivationType ?derivation.
?uncertainty uncertainty:uncertaintyType ?type.
?uncertainty uncertainty:uncertaintyModel ?model.
?probabilistic rdf:type uncertainty:Probability.
FILTER (?model = ?probabilistic)}
```

Once the set of assertions has been partitioned, the parallel reasoning processes can be launched. The reasoning processes are performed locally, exploiting the information provided by each assertion and the information provided by the ontologies to align. According to the information that has been provided to each reasoner, the process has to return back to the matching strategy the set of assertions that they believe to be the most reliable ones. Each reasoning process returns the results as instances of its representative concept (sub-concepts of the concept `Sentence`).

When the parallel reasoning processes come to an end, results are propagated back to the matching ontology by a reconciliation process. This process can be another reasoning process; in the case of our matching strategy we make use of SWRL rules to aggregate the results. Basically the reconciliation process follows the priority between the various reasoners that need to be made explicit. Some models are more reliable than others and this preference is defined in our system by SWRL rules. In our matching strategy we have to deal with inconsistencies because different relations on the same pair of elements can be classified as reliable relations by different reasoning processes. In case the preferences between inference models can not solve this situation we handle inconsistencies by assigning preferences between relations and operators, according to contextual factors: (e.g., analysing the data types of the elements to match. *Equivalence* relations have the highest priority in case of Strings, because Instance-based equivalence between integers is less reliable than the one between Strings). Rules are also used to propagate the best matching relations to other elements of the ontologies exploiting structural information from the original ontologies to match. This case can again be managed using a defeasible rules system such as DR-Prolog [3], which provides different precedences between rules, to help the decision process in inconsistent situations.

[7] in the Java prototype we developed, we have used Jess Rule engine (`http://herzberg.ca.sandia.gov/`) with the support of JessTab [31] and Protégé (`http://protege.stanford.edu/`) to translate SWRL rules and assertions from OWL to Jess and backwards.

4 Experimental Evaluation

Let us now describe the use of OAEI as a validating test for our data integration system. The comparison has been carried out with the results of the 2007 contest [4]. The majority of the participants of the context are based on a linear combination of several matching operators. In some cases this aggregating function is adaptive (Asmov [33], Prior+ [34]), while in some other cases it is fixed and defined by several experimentation (RiMOM [35], Sambo [36], Soda [37], Ola [38], TaxoMap [39], X-Som [40]). Some approaches are based on Possibility theory such as DSSim [28] and OWL-CM [41]. The test is performed using a palette of five semantically different matching operators (*JaroWrinkler Matching Operator, WordNet Label Matching Operator, Description Matching Operator, Type-based Matching Operator* and *Instance-based Matching Operator*). The results are stored in a matching ontology and the most reliable relations are extracted by our matching strategy. Each relation is associated to the respective matching operator. The reasoners that we used are a Fuzzy Description Logic reasoner [16] for the matching relations classified as probabilities (e.g. JaroWinkler matching operator) and a classic Description Logic reasoner [42] to process certain matching relations that the rules classify as necessities (e.g. String matching operator). The matching strategy has been developed as a Java prototype that we used to run the tests.

4.1 Results

The testbed can be divided in five subcategories:

101-104 This test set is the easiest set of tests. The first task is to match the reference ontology 101 with itself, the second test requires to generate a matching from the reference ontology to an ontology totally irrelevant (102 is a wine ontology). The ontology 103 represents a language generalization (unavailable constraints are replaced by their generalization): this ontology is an OWL-Lite generalization of the ontology 101. Finally the ontology 104 represents a language restriction with respect to the reference ontology 101. The constraints that are not available in OWL-Lite are simply removed. In the case of this first set of tests, our algorithm does not generate perfect alignments. In the case of 103 the propagation mechanism of the strategy does not perform well because of the generalization of the ontology. The various matching operators generate several reliable relations that the strategy can not recombine correctly in relation to their priority.

201-210 In this set of test cases, the structure of ontology

ISBN: 978-989-95079-6-8

Test #	Name	Prec.	Rec.	fMeas.
101	Reference Alignment	1	1	1
102	Irrelevant Ontology	NaN	NaN	NaN
103	Language Generalization	0,81	0,56	0,66
104	Language Restriction	0,94	0,94	0,94

is preserved. Syntactical changes are introduced: labels and identifiers are replaced by random names, misspellings, synonyms, language translation and moreover the comments in some cases have been suppressed. Our matching strategy obtains good results in this set of tests, because of the variety of the palette of matching operators that are considered in the strategy. In some cases (202, 209, 210) the Recall value is low, this because in case a matching operator does not create a possible relation this can not be automatically generated by the strategy. SWRL rules are used to propagate the best results to related elements, but if the matching relation is not created by any matching operator the propagation mechanism is not effective.

Test #	Name	Prec.	Rec.	fMeas.
201	No Names	0,92	0,90	0,91
202	No Names, No Comments	0,88	0,15	0,26
203	No Comments	0,97	0,97	0,97
204	Naming Conventions	0,92	0,92	0,92
205	Synonyms	0.91	0,90	0,90
206	Translation	0,96	0,70	0,95
207		0,98	0,70	0,97
208		0,96	0,72	0,82
209		0,87	0,34	0,49
210		0,92	0,12	0,22

221-247 In this case the set of tests can be divided into two subgroups: **221-231** and **232-247**. The first subgroup contains several structural modifications, such as the hierarchy that is flattened or expanded, and individuals, restrictions and data types that are suppressed. Each one of the documents in this subgroup has been modified by a structural change. Because of the fact that the labels and comments are preserved, the modifications have little influence on our system. The use of *Description Matching Operator* and *Name Matching Operator* allows the strategy to find most of the correct alignments using just the labels and comments information. In the second subgroup (232-247), the modifications made by combinations of the single modifications used in 221-231. Our system obtains good results for 232-247 as well.

Test #	Name	Prec.	Rec.	fMeas.
221	No Specialization	0,92	0,92	0,92
222	Flattened Hierarchy	0,92	0,92	0,92
223	Expanded Hierarchy	0,93	0,93	0,93
224	No Instance	0,93	0,93	0,93
225	No Restrictions	0,91	0,91	0,91
228	No Properties	0,97	0,97	0,97
230	Flattened Classes	0,95	0,96	0,95
231	Expanded Classes	0,93	0,93	0,93
232		0,95	0,95	0,95
233		1	1	1
236		1	1	1
237		0,92	0,92	0,92
238		0,91	0,91	0,91
239		0,93	0,97	0,95
240		0,85	0,88	0,87
241		1	1	1
246		0,97	1	0,98
247		0,91	0,94	0,93

Note that the observations about the Recall value we made about the previous tests, here are not valid since the ontologies provide sufficient information to the matching operators that can create reliable relations. Also in case of 222 where there is no hierarchy the results are satisfactory because of the information provided by the ontology.

248-266 This set of documents represents the most challenging case. This set combines structural and syntactical suppressions. The single challenges represented by the two previous set of documents are mixed in this set. All labels and identifiers are replaced by random names, and the comments are also suppressed. In this case our system does not perform well because no hierarchical representation is preserved so the strategy can not propagate the few correct matching found. This results in a low level of Recall. However, not enough information is provided in the ontologies, and the matching strategy can only find few alignments. The tests from 254 to 262, are the most difficult since almost all literal (labels and comments) and structural information are removed. In this case the propagation of the results can not take place because the relations that the strategy discovers can not be associated to other elements because of the structural information that is missing. When some structural information is preserved, the strategy can exploit this information in order to create possible matches, starting from the relations discovered by the single matching operators.

Test #	Name	Prec.	Rec.	fMeas.
248		0,88	0,14	0,25
249		0,74	0,14	0,24
250		0,92	0,33	0,49
251		0,86	0,13	0,22
252		0,65	0,11	0,19
253		0,88	0,14	0,25
254		0,90	0,27	0,42
257		0,92	0,33	0,49
258		0,86	0,13	0,22
259		0,65	0,11	0,19
260		0,82	0,31	0,45
261		0,69	0,27	0,39
262		0,90	0,27	0,42
265		0,82	0,31	0,45
266		0,69	0,27	0,39

301-304 This test set is composed by real ontologies contextually related to the reference ontology 101. The ontologies in this test set represent bibliographical information and they have been defined independently each other. This test represents a real world case of ontology alignment. Our strategy performs in the average with respect to the other systems evaluated. In the case of ontologies 301 our approach finds most of the correct alignments, but it also returns some wrong results. The alignment results for 302 and 303 are far from satisfactory. The reason is that these ontologies do not provide individuals and with shallow class hierarchy, where classes and properties are not related. In this case as well, the recall value is low: the matching operators are very sensible to the noise in data; moreover without a useful hierarchy the few matching relations identified by the matching operators can not be propagated or enforced with the support of hierarchical information. The ontology 304 has similar structure and vocabularies to the reference ontology 101 and in this case, the results are slightly better than the previous alignments.

Test #	Name	Prec.	Rec.	fMeas.
301	Real: BibTeX/MIT	0,9	0,6	0,71
302	Real: BibTeX/UMBC	0,79	0,46	0,58
303	Real: Karlsruhe	0,83	0,49	0,62
304	Real: INRIA	0,81	0,62	0,70

ISBN: 978-989-95079-6-8

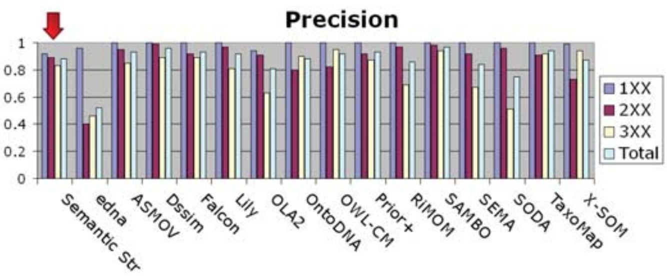

Figure 1: Graph of the Precision values of the different matching algorithms.

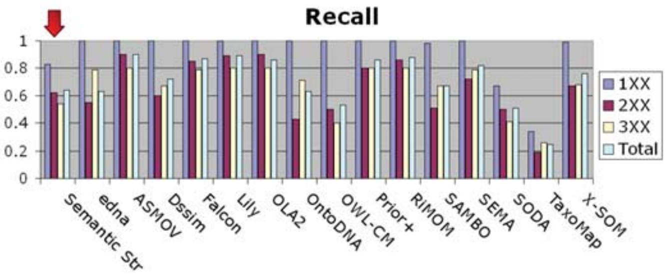

Figure 2: Graph of the Recall values of the different matching algorithms.

As one can see from the graphs in Figure 1 and 2 , our algorithm has acceptable results but under the best scores. These results are anyway reasonably encouraging, because our strategy is neither entailed to a particular reasoning model nor to a set of specific matching operator. More important our strategy does not imply a training phase that is typically required by the other systems. Our semantics-aware strategy is a new approach to ontology matching problem that provide a approach generally valid and not dependent on the domain. At this stage the results are very sensible to factors independent to the strategy; such as the quality of the matching operators and the available reasoners.

For this test we used a Probabilistic Description Logics reasoner and a traditional Description Logic reasoner in case of matching relations without a confidence degree. Moreover the inconsistencies, in the knowledge base, are not treated by a reasoning process but with the use of SWRL rules. As soon as new reasoners or new models appears we can exploit their use with our strategy. Moreover or strategy is highly customizable: if a new matching operator is plugged in the matching strategy is just necessary to create the related concept in the matching ontology and the SWRL rules reacting to the semantics of the new matching operator.

In case of the first series of test the strategy do not obtain the best results but, instead provides results under the average. Even the most simple matching operators, such as Edna, performs better than our strategy. The reason of this behaviour has been identified in the confusion of the strategy that considers the relations generated by different operator all reliable at the same level. We had run the same test only with JaroWinker matching operator and we obtained an average value of Precision 1 and Recall 1 for all the first series; this result confirm our previous assumption. In the case of the second and the third series the results are in the average of the approaches presented to the contest. The graph in Figure 1 shows the comparison of the Precision of our strategy with respect to the other algorithms. As is possible to see the values are very closed to 1, except in some isolated cases (Edna). These results show that the various strategies generate correct relations. The graph in Figure 2 shows the comparison of the Recall of our strategy with respect to the other algorithms. As is possible to see the values this time instead are far from the limit. Only few strategies (Asmonv, Falcon, Lily, OLA2) obtain results around 0.8, while the others still remain over the $0,5$. This generally low value of Recall means that the algorithm does not propagate well the good matching relations by exploiting the structure of the ontologies.

5 Conclusions

In this paper we presented preliminary work on a framework for managing different types of uncertainty and a possible application to the Schema Matching problem. The Ontology of Uncertainty, proposed by the W3C's UR3W-XG incubator group, provides a vocabulary to annotate different sources of information with different types of uncertainty. We argue that such annotations should be clearly mapped to corresponding reasoning and representation strategies. This mapping allows the system to analyse the information on the basis of its uncertainty model, running the inference process according to the respective uncertainty. In this way we can also deal with complex situations that do not tailor to the traditional strategies: as an example, matching relations provided by a user with her specific level of expertise. In our scenario the sources of information analysed, according to different uncertainty models are independent and no intersection among them has to be managed. This particular case allows a straight-forward use of the Ontology of Uncertainty to drive the reasoning process, although in general the assumption of independence among the source of information is a lucky case. The need of additional work on the Ontology of Uncertainty is necessary in order to support reasoning processes when combinations of uncertainty models are applied to a single source of information. This outlook is promising in order to provide more expressive frameworks for reasoning under different types of uncertainties, but definitely the need of more research in this direction is evident.

Acknowledgements

This work was partly funded by the Italian Ministry of Research under `FIRB` contract n. `RBNE05FKZ2_004`, `TEKNE`.

References

[1] Maurizio Lenzerini. Data integration is harder than you thought. In *CooplS '01: Proceedings of the 9th International Conference on Cooperative Information Systems*, pages 22–26, London, UK, 2001. Springer-Verlag.

[2] X. Luo, C. Zhang, and N. R. Jennings. A hybrid model for sharing information between fuzzy, uncertain and default reasoning models in multi-agent systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(4):401–450, 2002.

[3] Grigoris Antoniou and Antonis Bikakis. Dr-prolog: A system for defeasible reasoning with rules and ontologies on the semantic web. *IEEE Trans. Knowl. Data Eng.*, 19(2):233–245, 2007.

[4] Jérôme Euzenat, Antoine Isaac, Christian Meilicke, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2007. In *Proc. 2nd ISWC 2007 international workshop on ontology matching (OM), Busan (KR)*, pages 96–132, 2007.

[5] I Kenneth J. Laskey, Kathryn B. Laskey, Paulo C. G. Costa, Mieczyslaw M. Kokar, Trevor Martin, and Thomas Lukasiewicz. Uncertainty reasoning for the world wide web. *W3C Incubator Group Report*, 2008.

[6] B. de Finetti. La logique de la probabilité. In *Actes du Congres International de Philosophie Scientifique a Paris 1935, Tome IV*, pages 1–9. 1936.

[7] Didier Dubois and Henri Prade. Can we enforce full compositionality in uncertainty calculi. In *In Proc. of the 11th Nat. Conf. on Artificial Intelligence (AAAI-94*, pages 149–154. AAAI Press / MIT Press, 1994.

[8] George Boole. The calculus of logic, 1848.

[9] L.A. Zadeh, K.S Fu, K. Tanaka, and M. Shimura. *Fuzzy sets and their applications to Cognitive and Decision Processes*. 1975.

[10] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*, 2003.

[11] Deborah L. McGuinness and Frank van Harmelen. Owl web ontology language overview. *W3C Recommendation*, 2004.

[12] Shiyong Lu, Ming Dong, and Farshad Fotouhi. The semantic web: Opportunities and challenges for next-generation web applications. *International Journal of Information Research*, 7:2002, 2002.

[13] I. Horrocks and U. Sattler. Ontology reasoning in the SHOQ(D) description logic. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

[14] Thomas Lukasiewicz. Expressive probabilistic description logics. *Artif. Intell.*, 172(6-7):852–883, 2008.

[15] Pavel Klinov. Pronto: Probabilistic dl reasoning with pellet, 2008.

[16] Fernando Bobillo and Umberto Straccia. fuzzydl: An expressive fuzzy description logic reasoner. In *2008 International Conference on Fuzzy Systems (FUZZ-08)*, pages –. IEEE Computer Society, 2008.

[17] Janusz Kacprzyk and Slawomir Zadrozny. Computing with words in intelligent database querying: standalone and internet-based applications. *Inf. Sci.*, 134(1-4):71–109, 2001.

[18] Thomas Lukasiewicz and Umberto Straccia. Description logic programs under probabilistic uncertainty and fuzzy vagueness. In *ECSQARU*, pages 187–198, 2007.

[19] Thomas Lukasiewicz and Umberto Straccia. Uncertainty and vagueness in description logic programs for the semantic web, 2007.

[20] Didier Dubois, Jérôme Mengin, and Henri Prade. Possibilistic uncertainty and fuzzy features in description logic. A preliminary discussion. In E. Sanchez, editor, *Fuzzy logic and the semantic web*, pages 101–113. http://www.elsevier.com/, 2006. DMenP001.

[21] Fernando Bobillo, Miguel Delgado, and Juan Gómez-Romero. Extending fuzzy description logics with a possibilistic layer. In *URSW*, volume 327 of *CEUR Workshop Proceedings*, 2007.

[22] Andriy Nikolov, Victoria S. Uren, Enrico Motta, and Anne N. De Roeck. Using the dempster-shafer theory of evidence to resolve abox inconsistencies. In *URSW*, volume 327, 2007.

[23] L.A. Zadeh. Fuzzy sets. *Information Control*, 8:338–353, 1965.

[24] Hiroshi Maeda and Yuji Nobusada. A study on the parallel computability of multi-fold approximate reasoning. *Fuzzy Sets Syst.*, 97(2):129–144, 1998.

[25] Boris Motik, Ulrike Sattler, and Rudi Studer. Query answering for owl-dl with rules. In *Journal of Web Semantics*, pages 549–563, 2004.

[26] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal: Very Large Data Bases*, 10(4):334–350, 2001.

[27] Pavel Shvaiko and Jérôme Euzenat. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

[28] Miklos Nagy, Maria Vargas-Vera, and Enrico Motta. Dssim-ontology mapping with uncertainty. In *Ontology Matching*, 2006.

[29] Andrea Calì, Thomas Lukasiewicz, Livia Predoiu, and Heiner Stuckenschmidt. A framework for representing ontology mappings under probabilities and inconsistency. In *URSW*, 2007.

[30] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member Submission*, 2004.

[31] Henrik Eriksson. Using jesstab to integrate protégé and jess. *IEEE Intelligent Systems*, 18(2):43–50, 2003.

[32] Eric Prud'hommeaux and Andy Seaborne. Sparql query language for rdf. *W3C Member Submission*, 2008.

[33] Yves Jean-Mary and Mansur Kabuka. Asmov results for oaei 2007. In *Proc. 2nd Ontology matching workshop, Busan (KR)*, pages 141–150, 2007.

[34] Ming Mao and Yefei Peng. The prior+: Results for oaei campaign 2007. In *2nd International Workshop on Ontology Matching (OM) at 6th International Semantic Web Conference (ISWC), Busan, Korea, 2007*, 2007.

[35] Yi Li, Qian Zhong, Juanzi Li, and Jie Tang. Result of ontology alignment with rimom at oaei'07. In *The Second International Workshop on Ontology Matching*, 2007.

[36] Patrick Lambrix and He Tan. Sambo-a system for aligning and merging biomedical ontologies. *Web Semant.*, 4(3):196–206, 2006.

[37] Sami ZGHAL, S. Ben Yahia, Engelbert MEPHU NGUIFO, and Y. Slimani. Soda: an owl-dl based ontology matching system. In *OAEI'07 - OM'07 - ISWC'07*, Busan (Korea), nov 2007.

[38] Jérôme Euzenat, David Loup, Mohamed Touzani, and Petko Valtchev. Ontology alignment with ola. pages 333–337, 2004.

[39] Haifa Zargayouna, Brigitte Safar, and Chantal Reynaud. TaxoMap in the OAEI 2007 alignment contest . In *Ontology Alignment Evaluation Inititiative (OAEI) 2007 Campaign - Workshop on Ontology Matching*, 2007.

[40] Carlo Curino, Giorgio Orsi, and Letizia Tanca. X-som: A flexible ontology mapper. In *DEXA '07: Proceedings of the 18th International Conference on Database and Expert Systems Applications*, pages 424–428, Washington, DC, USA, 2007. IEEE Computer Society.

[41] Boutheina Ben Yaghlane and Najoua Laamari. Owl-cm: Owl combining matcher based on belief functions theory.

[42] Evren Sirin, Bijan Parsia, Bernardo C. Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53, June 2007.

ISBN: 978-989-95079-6-8