

Using the Dempster-Shafer theory of evidence to resolve ABox inconsistencies

Andriy Nikolov, Victoria Uren, Enrico Motta and Anne de Roeck

Knowledge Media Institute, The Open University, Milton Keynes, UK
{a.nikolov, v.s.uren, e.motta, a.deroeck}@open.ac.uk

Abstract. Automated ontology population using information extraction algorithms can produce inconsistent knowledge bases. Confidence values assigned by the extraction algorithms may serve as evidence helping to repair produced inconsistencies. The Dempster-Shafer theory of evidence is a formalism, which allows appropriate interpretation of extractors' confidence values. The paper presents an algorithm for translating the subontologies containing conflicts into belief propagation networks and repairing conflicts based on the Dempster-Shafer plausibility.

1 Introduction

One of the approaches for ontology population considers using automatic information extraction algorithms to annotate natural language data already available on the Web [1, 2]. Unsupervised information extraction algorithms do not produce 100% correct output, which may lead to inconsistency of the whole knowledge base produced in this way. Also information extracted from different sources can be genuinely contradictory. So when performing fusion of knowledge extracted from different sources it is important to resolve such inconsistencies automatically or provide the user with a ranking of conflicting options estimating how likely each statement is to be wrong. Extraction algorithms can often estimate the reliability of their output by attaching confidence values to produced statements [3]. Uncertain reasoning using these confidence values can help to evaluate the plausibility of statements and rank the conflicting options. Most of the ongoing research in the field of applying uncertain reasoning to the Semantic Web focuses on fuzzy logic and probabilistic approaches. Fuzzy logic was designed to deal with representation of vagueness and imprecision. This interpretation is not relevant for the problem occurring during population of crisp OWL knowledge bases, where we need to assess the likelihood for a statement to be true or false. The probabilistic approach is more appropriate for dealing with such problems. However, as stated in [4], axioms of probability theory are implied by seven properties of belief measures. One of them is *completeness*, which states that “a degree of belief can be assigned to any well-defined proposition”. However, this property cannot be ensured when dealing with confidence degrees assigned by extractors, because they do not always carry information about the probability of a statement being *false*. The Dempster-Shafer theory of

evidence [5] presents a formalism that helps to overcome this problem. It allows belief measurements to be assigned to sets of propositions, thus specifying explicitly degrees of ignorance. In this paper, we describe an algorithm for resolving conflicts using the Dempster-Shafer belief propagation approach.

2 Related Work

There are several studies dealing with inconsistency handling in OWL ontologies, among others [6] and [7]. The general algorithm for the task of repairing inconsistent ontologies consists of two steps:

- Ontology diagnosis: finding sets of axioms, which contribute to inconsistency;
- Repairing inconsistencies: changing/removing the axioms most likely to be erroneous.

Choosing the axioms for change and removal is a non-trivial task. Existing algorithms working with crisp ontologies (e.g., [7]) utilize such criteria as syntactic relevance (how often each entity is referenced in the ontology), impact (the influence of removal of the axiom on the ontology should be minimized) and provenance (reliability of the source of the axiom). The last criterion is especially interesting for the automatic ontology population scenario since extraction algorithms do not extract information with 100% accuracy. A study described in [8] specifies an algorithm which utilizes the confidence value assigned by the extraction algorithm. The strategy employed there was to order the axioms according to their confidence and add them incrementally, starting from the most certain one. If adding the axiom led to inconsistency of the ontology then a minimal subconsistent ontology was determined and the axiom with the lowest confidence was removed from it. A disadvantage of such a technique is that it does not take into account the impact of an axiom: e.g., when an axiom violates several restrictions, it does not increase its chances to be removed. Also it does not consider the influence of redundancy: if the same statement was extracted from several sources, this should increase its reliability. Using uncertain reasoning would provide a more sound approach to rank potentially erroneous statements and resolve inconsistencies.

In the Semantic Web domain the studies on uncertain reasoning are mostly focused on two formalisms: probability theory and fuzzy logic. Existing implementations of fuzzy description logic [9, 10] are based on the notion of fuzzy set representing a vague concept. The uncertainty value in this context denotes a membership function $\mu_F(x)$ which specifies the degree to which an object x belongs to a fuzzy class F . Probabilistic adaptations of OWL-DL include Bayes OWL [11] and PR-OWL [12]. However, both of these formalisms do not fully reflect the properties of the problems we are dealing with in the fusion scenario. In [4] a framework for choosing an appropriate uncertainty handling formalism was presented. The framework is based on the following seven properties of belief measurements:

1. *Clarity*: Propositions should be well-defined.
2. *Scalar continuity*: A single real number is both necessary and sufficient for representing a degree of belief.
3. *Completeness*: A degree of belief can be assigned to any well-defined proposition.
4. *Context dependency*: The belief assigned to a proposition can depend on the belief in other propositions.
5. *Hypothetical conditioning*: There exists some function that allows the belief in a conjunction of propositions to be calculated from the belief in one proposition and the belief in the other proposition given that the first proposition is true.
6. *Complementarity*: The belief in the negation of a proposition is a monotonically decreasing function of the belief in the proposition itself.
7. *Consistency*: There will be equal belief in propositions that have the same truth value.

It was proven that accepting all seven properties logically necessitates the axioms of probability theory. Alternative formalisms allow weakening of some properties. Fuzzy logic deals with the case when the *clarity* property does not hold, i.e., when concepts and relations are vague. Such an interpretation differs from the one we are dealing with in the fusion scenario, where the ontology TBox contains crisp concepts and properties. Confidence value attached to a type assertion $ClassA(Individual1)$ denotes a degree of belief that the statement is true in the real world rather than the degree of inclusion of the entity $Individual1$ into a fuzzy concept $ClassA$. This makes fuzzy interpretation inappropriate for our case.

Probabilistic interpretation of the extraction algorithm's confidence may lead to a potential problem. If we interpret the confidence value c attached to a statement returned by an extraction algorithm as a Bayesian probability value p , we, at the same time, introduce a belief that the statement is false with a probability $1 - p$. However, the confidence of an extraction algorithm reflects only the belief that the document supports the statement and does not itself reflect the probability of a statement being false in the real world. Also while statistical extraction algorithms ([13]) are able to assign a degree of probability to each extracted statement, rule-based algorithms ([14, 15]) can only assign the same confidence value to all statements extracted by the same rule based on the rule's performance on some evaluation set. Any extraction produced by a rule with a low confidence value in this case will serve as a negative evidence rather than simply lack of evidence. This issue becomes more important if the reliability of sources is included into analysis: it is hard to assign the conditional probability of a statement being false given that the document supports it. It means that the *completeness* property does not always hold.

The Dempster-Shafer theory of evidence [5] allows weakening of the completeness property. Belief can be assigned to sets of alternative options rather than only to atomic elements. In the case of binary logic, it means that the degree of ignorance can be explicitly represented by assigning a non-zero belief to the

set $\{\text{true};\text{false}\}$. On the other hand, it still allows the Bayesian interpretation of confidence to be used, when it is appropriate (in this case the belief assigned to the set $\{\text{true};\text{false}\}$ is set to 0). This paper presents an algorithm for resolving inconsistencies by translating the inconsistency-preserving subset of ontology into the Dempster-Shafer belief network and choosing the axioms to remove based on their plausibility. We are not aware of other studies adapting the Dempster-Shafer approach to the Semantic Web domain.

Alternative approaches to uncertainty representation, which were not applied so far to ontological modelling, include probability intervals [16] and higher-order probability [17]. However, the first of these approaches uses min and max operators for aggregation, which makes it hard to represent cumulative evidence, and the second focuses on resolving different kinds of problems (namely expressing probability estimations of other probability estimations). There are also other approaches to belief fusion in the Semantic Web (e.g., [18] and [19]). These studies deal with social issues of representing trust and provenance in a distributed knowledge base and focus on the problem of establishing the certainty of statements asserted by other people. These approaches, however, do not focus on resolving the inconsistencies and just deal with direct conflicts (i.e., statement A is true vs statement A is false). They do not take into account ontological inference and mutual influence of statements in the knowledge base. In this way, they can be considered complementary to ours.

3 The Dempster-Shafer Belief Theory

Dempster-Shafer theory of evidence differs from the Bayesian probability theory as it allows assigning beliefs not only to atomic elements but to sets of elements as well. The base of the Dempster's belief distribution is the frame of discernment (Ω) - an exhaustive set of mutually exclusive alternatives. A belief distribution function (also called mass function or belief potential) $m(A)$ represents the influence of a piece of evidence on subsets of Ω and has the following constraints:

- $m(\emptyset) = 0$ and
- $\sum_{A \subseteq \Omega} m(A) = 1$

$m(A)$ defines the amount of belief assigned to the subset A . When $m(A) > 0$, A is referred to as a focal element. If each focal element A contains only a single element, the mass function is reduced to be a probability distribution. Mass also can be assigned to the whole set of Ω . This represents the uncertainty of the piece of evidence about which of the elements in Ω is true. In our case each mass function is defined on a set of variables $D = \{x_1, \dots, x_n\}$ called the domain of m . Each variable is boolean and represents an assertion in the knowledge base. For a single variable we can get degree of support $Sup(x) = m(\{\text{true}\})$ and degree of plausibility $Pl(x) = m(\{\text{true}\}) + m(\{\text{true}; \text{false}\})$. Plausibility specifies how likely it is that the statement is false. Based on plausibility it is possible to select from a set of statements the one to be removed.

4 Building Belief Networks

Our algorithm consists of four steps:

1. Inconsistency detection.
At this stage we select the subontology containing all axioms contributing to an inconsistency.
2. Constructing a belief network.
At this stage the subontology found at the previous step is translated into a belief network.
3. Assigning mass distributions.
At this stage we assign mass distribution functions to nodes.
4. Belief propagation.
At this stage we propagate uncertainties through the network and update the confidence degrees of ABox statements.

4.1 Illustrating Example

In order to illustrate our algorithm, we use an example from the banking domain. Supposedly, we have an ontology describing credit card applications, which defines two disjoint classes of applicants: reliable and risky. In order to be reliable, an applicant has to have UK citizenship and evidence that (s)he was never bankrupt in the past. For example, the TBox contains the following axioms:

T1: $RiskyApplicant \sqsubseteq CreditCardApplicant$

T2: $ReliableApplicant \sqsubseteq CreditCardApplicant$

T3: $RiskyApplicant \sqsubseteq \neg ReliableApplicant$

T4: $ReliableApplicant \equiv \exists wasBankrupt.False \sqcap \exists hasCitizenship.UK$

T5: $\top \sqsubseteq \leq 1 wasBankrupt (wasBankrupt \text{ is functional})$

The ABox contains the following axioms (with attached confidence values):

A1: $RiskyApplicant(Ind1) : 0.7$

A2: $wasBankrupt(Ind1, False) : 0.6$

A3: $hasCitizenship(Ind1, UK) : 0.4$

A4: $wasBankrupt(Ind1, True) : 0.5$

As given, the ontology is inconsistent: the individual *Ind1* is forced to belong to both classes *RiskyApplicant* and *ReliableApplicant*, which are disjoint, and the functional property *wasBankrupt* has two different values. If we choose to remove the axioms with the lowest confidence values, it will require removing A3 and A4. However, inconsistency can also be repaired by removing a single statement A2. The fact that A2 leads to the violation of two ontological constraints should increase the likelihood it is wrong.

4.2 Inconsistency Detection

The task of the inconsistency detection step is to retrieve all minimal inconsistent subontologies (MISO) of the ontology and combine them. As defined in [6], an ontology O' is a minimal inconsistent subontology of an ontology O , if $O' \sqsubseteq O$

and O' is inconsistent and for all O'' such that $O'' \subset O' \subseteq O$, O'' is consistent. OWL reasoner Pellet [7] is able to return the MISO for the first encountered inconsistency in the ontology. To calculate all MISO O'_1, \dots, O'_n in the ontology we employ Reiter’s hitting set tree algorithm [20]. After all conflict sets were identified, the next step involves constructing belief networks from each set. If for two subontologies $O'_i \cap O'_j \neq \emptyset$ then these two subontologies are replaced with $O' = O'_i \cup O'_j$.

For our illustrating example, the conflict detection algorithm is able to identify two conflict sets in this ontology: the first, consisting of $\{T3, T4, A1, A2, A3\}$ (individual *Ind1* belongs to two disjoint classes), and the second $\{T5, A2, A4\}$ (individual *Ind1* has two instantiations of a functional property). The statement A2 belongs to both sets and therefore the sets are merged.

4.3 Constructing Belief Networks

The networks for propagation of Dempster-Shafer belief functions (also called valuation networks) were described in [21]. By definition the valuation network is an undirected graph represented as a 5-tuple: $\{\Psi, \{\Omega_X\}_{X \in \Psi}, \{\tau_1, \dots, \tau_n\}, \downarrow, \otimes\}$, where Ψ is a set of variables, $\{\Omega_X\}_{X \in \Psi}$ is a collection of state spaces, $\{\tau_1, \dots, \tau_n\}$ is a collection of valuations (belief potentials of nodes), \downarrow is a marginalization operator and \otimes is a combination operator. In our case Ψ consists of ABox assertions, every $\{\Omega_X\}_{X \in \Psi} = \{0; 1\}$ and $\{\tau_1, \dots, \tau_n\}$ are created using rules described below. The operators are used for propagation of beliefs and are described in the following subsections. The network contains two kinds of nodes: variable nodes corresponding to explicit or inferred ABox assertions and valuation nodes representing TBox axioms. Variable nodes contain only one variable, while valuation nodes contain several variables.

Translation of an inconsistent subontology into a belief propagation network is performed using a set of rules (Table 1). Each rule translates a specific OWL-DL construct into a set of network nodes and links between them. Rules 1 and 2 directly translate each ABox statement into a variable node. Other rules process TBox axioms and create two kinds of nodes: one valuation node to represent the TBox axiom and one or more variable nodes to represent inferred statements. Such rules only fire if the network already contains variable nodes for ABox axioms, which are necessary to make the inference. For example, a rule processing the class equivalence axiom (Rule 4) is interpreted as the following: “If there is a node N_1 representing the type assertion $I \in X$ and an *owl : equivalentClass* axiom $X \equiv Y$, then:

- Create a node N_2 representing the assertion $I \in Y$;
- Create a node N_3 representing the axiom $X \sqsubseteq Y$;
- Create links between N_1 and N_3 and between N_3 and N_2 .”

If a rule requires creating a node, which already exists in the network, then the existing node is used.

Applying the rules described above to our illustrating example (rules 1, 2, 4, 5, 6, 9, 20) will result in the following network (Fig. 1).

Table 1. Belief network construction rules

| N | Pre-conditions | Nodes to create | Links to create |
|----|--|--|---|
| 1 | $I \in X$ | $N_1 : I \in X$ | |
| 2 | $R(I_1, I_2)$ | $N_2 : R(I_1, I_2)$ | |
| 3 | $N_1 : I \in X, X \sqsubseteq Y$ | $N_2 : I \in Y, N_3 : X \sqsubseteq Y$ | $(N_1, N_3), (N_3, N_2)$ |
| 4 | $N_1 : I \in X, X \equiv Y$ | $N_2 : I \in Y, N_3 : X \sqsubseteq Y$ | $(N_1, N_3), (N_3, N_2)$ |
| 5 | $N_1 : I \in X, X \sqsubseteq \neg Y$ | $N_2 : I \in Y, N_3 : X \sqsubseteq \neg Y$ | $(N_1, N_3), (N_3, N_2)$ |
| 6 | $N_1 : I \in X, X \sqcap Y$ | $N_2 : I \in X \sqcap Y, N_3 : X \sqcap Y,$ $N_4 : I \in Y$ | $(N_1, N_3), (N_4, N_3),$ (N_3, N_2) |
| 7 | $N_1 : I \in X, X \sqcup Y$ | $N_2 : I \in X \sqcup Y, N_3 : X \sqcup Y,$ $N_4 : I \in Y$ | $(N_1, N_3), (N_4, N_3),$ (N_3, N_2) |
| 8 | $N_1 : I \in X, \neg X$ | $N_2 : I \in \neg X, N_3 : \neg X$ | $(N_1, N_3), (N_3, N_2)$ |
| 9 | $\top \sqsubseteq \leq 1R, N_1 : R(I, o_1),$ $N_2 : R(I, o_2)$ | $N_3 : \top \sqsubseteq \leq 1R$ | $(N_1, N_3), (N_2, N_3)$ |
| 10 | $\top \sqsubseteq \leq 1R^-, N_1 : R(I_2, I_1),$ $N_2 : R(I_3, I_1)$ | $N_3 : \top \sqsubseteq \leq 1R^-$ | $(N_1, N_3), (N_2, N_3)$ |
| 11 | $R \equiv R^-, N_1 : R(I_1, I_2)$ | $N_2 : R \equiv R^-, N_3 : R(I_2, I_1)$ | $(N_1, N_2), (N_2, N_3)$ |
| 12 | $R \equiv Q, N_1 : R(I_1, I_2)$ | $N_2 : R \equiv Q, N_3 : Q(I_1, I_2)$ | $(N_1, N_2), (N_2, N_3)$ |
| 13 | $R \sqsubseteq Q, N_1 : R(I_1, I_2)$ | $N_2 : R \sqsubseteq Q, N_3 : Q(I_1, I_2)$ | $(N_1, N_2), (N_2, N_3)$ |
| 14 | $R \equiv Q^-, N_1 : R(I_1, I_2)$ | $N_2 : R \equiv Q^-, N_3 : Q(I_2, I_1)$ | $(N_1, N_2), (N_2, N_3)$ |
| 15 | $Trans(R), N_1 : R(I_1, I_2),$ $N_2 : R(I_2, I_3)$ | $N_3 : Trans(R), N_4 : R(I_1, I_3)$ | $(N_1, N_3), (N_2, N_3),$ (N_3, N_4) |
| 16 | $\leq 1.R, N_1 : R(I_1, o_1),$ $N_2 : R(I_1, o_2)$ | $N_3 : \leq 1.R, N_4 : I \in \leq 1.R$ | $(N_1, N_3), (N_2, N_3),$ (N_3, N_4) |
| 17 | $\geq 1.R, N_1 : R(I_1, o_1),$ $N_2 : R(I_1, o_2)$ | $N_3 : \geq 1.R, N_4 : I \in \geq 1.R$ | $(N_1, N_3), (N_2, N_3)$ |
| 18 | $= 1.R, N_1 : R(I_1, o_1),$ $N_2 : R(I_1, o_2)$ | $N_3 : I \in = 1.R$ | $(N_1, N_3), (N_2, N_3)$ |
| 19 | $\forall R.X, N_1 : R(I_1, I_2),$ $N_2 : I_2 \in X$ | $N_3 : \forall R.X, N_4 : I_1 \in \forall R.X$ | $(N_1, N_3), (N_2, N_3),$ (N_3, N_4) |
| 20 | $\exists R.X, N_1 : R(I_1, I_2),$ $N_2 : I_2 \in X$ | $N_3 : \exists R.X, N_4 : I_1 \in \exists R.X$ | $(N_1, N_3), (N_2, N_3),$ (N_3, N_4) |
| 21 | $\exists R. \top \sqsubseteq X, N_1 : R(I_1, I_2),$ $N_2 : I_1 \in X$ | $N_3 : \exists R. \top \sqsubseteq X$ | $(N_1, N_3), (N_2, N_3)$ |
| 22 | $\top \sqsubseteq \forall R.X, N_1 : R(I_1, I_2),$ $N_2 : I_2 \in X$ | $N_3 : \top \sqsubseteq \forall R.X$ | $(N_1, N_3), (N_2, N_3)$ |

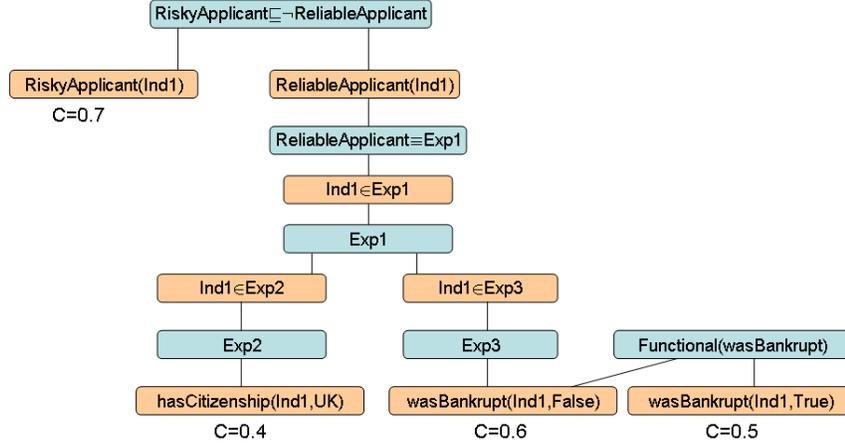


Fig. 1. Belief network example ($Exp1 = \exists wasBankrupt.False \sqcap \exists hasCitizenship.UK$, $Exp2 = \exists hasCitizenship.UK$, $Exp3 = \exists wasBankrupt.False$)

4.4 Assigning Mass Distributions

After the nodes were combined into the network, the next step is to assign the mass distribution functions to the nodes. There are two kinds of variable nodes: (i) nodes representing statements supported by the evidence and (ii) nodes representing inferred statements. Initial mass distribution for the nodes of the first type is assigned based on their extracted confidence. If a statement was extracted with a confidence degree c , it is assigned the following mass distribution: $m(True) = c, m(True; False) = 1 - c$. It is possible that the same statement is extracted from several sources. In this case, multiple pieces of evidence have to be combined using Dempster’s rule of combination.

Nodes created artificially during network construction are only used for propagation of beliefs from their neighbours and do not contain their own mass assignment. Valuation nodes specify the TBox axioms and are used to propagate beliefs through the network. For the crisp OWL ontologies only mass assignments of 0 and 1 are possible. The principle for assigning masses is to assign the mass of 1 to the set of all combinations of variable sets allowed by the corresponding axiom. Table 2 shows the mass assignment functions for OWL-DL T-Box axioms ¹.

In our example, we assign distributions based on the extractor’s confidence values to the variable nodes representing extracted statements: A1: ($m(1)=0.7, m(\{0;1\})=0.3$), A2: ($m(1)=0.6, m(\{0;1\})=0.4$), A3: ($m(1)=0.4, m(\{0;1\})=0.6$),

¹ For nodes allowing multiple operands (e.g., intersection or cardinality) only the case of two operands is given. If the node allows more than two children, then number of variables and the distribution function is adjusted to represent the restriction correctly

Table 2. Belief distribution functions for valuation nodes

| N | Node type | Variables | Mass distribution |
|----|--------------------------------|---|--|
| 1 | $X \sqsubseteq Y$ | $I \in X, I \in Y$ | $m(\{0;0\}, \{0;1\}, \{1;1\})=1$ |
| 2 | $X \equiv Y$ | $I \in X, I \in Y$ | $m(\{0;0\}, \{1;1\})=1$ |
| 3 | $X \sqsubseteq \neg Y$ | $I \in X, I \in Y$ | $m(\{0;0\}, \{0;1\}, \{1;0\})=1$ |
| 4 | $X \sqcap Y$ | $I \in X, I \in Y, I \in X \sqcap Y$ | $m(\{0;0;0\}, \{0;1;0\}, \{1;0;0\}, \{1;1;1\})=1$ |
| 5 | $X \sqcup Y$ | $I \in X, I \in Y, I \in X \sqcup Y$ | $m(\{0;0;0\}, \{0;1;1\}, \{1;0;1\}, \{1;1;1\})=1$ |
| 6 | $\neg X$ | $I \in X, I \in \neg X$ | $m(\{0;1\}, \{1;0\})=1$ |
| 7 | $\top \sqsubseteq \leq 1R$ | $R(I, o_1), R(I, o_2)$ | $m(\{0;0\}, \{0;1\}, \{1;0\})=1$ |
| 8 | $\top \sqsubseteq \leq 1R^-$ | $R(I_2, I_1), R(I_3, I_1)$ | $m(\{0;0\}, \{0;1\}, \{1;0\})=1$ |
| 9 | $R \equiv R^-$ | $R(I_1, I_2), R(I_2, I_1)$ | $m(\{0;0\}, \{1;1\})=1$ |
| 10 | $R \equiv Q$ | $R(I_1, I_2), Q(I_1, I_2)$ | $m(\{0;0\}, \{1;1\})=1$ |
| 11 | $R \sqsubseteq Q$ | $R(I_1, I_2), Q(I_1, I_2)$ | $m(\{0;0\}, \{0;1\}, \{1;1\})=1$ |
| 12 | $R \equiv Q^-$ | $R(I_1, I_2), Q(I_2, I_1)$ | $m(\{0;0\}, \{1;1\})=1$ |
| 13 | $Trans(R)$ | $R(I_1, I_2), R(I_2, I_3), R(I_1, I_3)$ | $m(\{0;0;0\}, \{0;0;1\}, \{0;1;0\}, \{0;1;1\}, \{1;0;0\}, \{1;0;1\}, \{1;1;1\})=1$ |
| 14 | $\leq 1.R$ | $R(I_1, o_1), R(I_1, o_2), I_1 \in \leq 1.R$ | $m(\{0;0;1\}, \{0;1;1\}, \{1;0;1\}, \{1;1;0\})=1$ |
| 15 | $\geq 1.R$ | $R(I_1, o_1), R(I_1, o_2), I_1 \in \geq 1.R$ | $m(\{0;0;0\}, \{0;1;1\}, \{1;0;1\}, \{1;1;1\})=1$ |
| 16 | $= 1.R$ | $R(I_1, o_1), R(I_1, o_2), I_1 \in = 1.R$ | $m(\{0;0;0\}, \{0;1;1\}, \{1;0;1\}, \{1;1;0\})=1$ |
| 17 | $\forall R.X$ | $R(I_1, I_2), I_2 \in X, I_1 \in \forall R.X$ | $m(\{0;0;1\}, \{0;1;1\}, \{1;0;0\}, \{1;1;1\})=1$ |
| 18 | $\exists R.X$ | $R(I_1, I_2), I_2 \in X, I_1 \in \exists R.X$ | $m(\{0;0;1\}, \{0;1;1\}, \{1;0;0\}, \{1;1;1\})=1$ |
| 19 | $\exists R.\top \sqsubseteq X$ | $R(I_1, I_2), I_1 \in X$ | $m(\{0;0\}, \{0;1\}, \{1;1\})=1$ |
| 20 | $\top \sqsubseteq \forall R.X$ | $R(I_1, I_2), I_2 \in X$ | $m(\{0;0\}, \{0;1\}, \{1;1\})=1$ |

A4: ($m(1)=0.5, m(\{0;1\})=0.5$). The valuation nodes obtain their distributions according to the rules specified in the Table 2: T3 (rule 3), T4 (rules 2, 4, 18) and T5 (rule 7).

4.5 Belief Propagation

The axioms for belief propagation were formulated in [22]. The basic operators for belief potentials are marginalization \downarrow and combination \otimes . Marginalization takes a mass distribution function m on domain D and produces a new mass distribution on domain $C \subseteq D$.

$$m^{\downarrow C}(X) = \sum_{Y \uparrow C = X} m(Y)$$

For instance, if we have the function m defined on domain $\{x, y\}$ as $m(\{0;0\}) = 0.2, m(\{0;1\}) = 0.35, m(\{1;0\}) = 0.3, m(\{1;1\}) = 0.15$ and we want to find a marginalization on domain $\{y\}$, we will get $m(0) = 0.2 + 0.3 = 0.5$ and $m(1) = 0.35 + 0.15 = 0.5$. The combination operator is represented by the Dempster's rule of combination:

$$m_1 \otimes m_2(X) = \frac{\sum_{X_1 \cap X_2 = X} m_1(X_1)m_2(X_2)}{1 - \sum_{X_1 \cap X_2 = \emptyset} m_1(X_1)m_2(X_2)}$$

Belief propagation is performed by passing messages between nodes according to the following rules:

1. Each node sends a message to its inward neighbour (towards the root of the tree). If $\mu^{A \rightarrow B}$ is a message from A to B , $N(A)$ is a set of neighbours of A and the potential of A is m_A , then the message is specified as a combination of messages from all neighbours except B and the potential of A :

$$\mu^{A \rightarrow B} = (\otimes \{ \mu^{X \rightarrow A} | X \in (N(A) - \{B\}) \} \otimes m_A) \downarrow^{A \cap B}$$

2. After a node A has received a message from all its neighbors, it combines all messages with its own potential and reports the result as its marginal.

As the message-passing algorithm assumes that the graph is a tree, it is necessary to eliminate loops. All valuation nodes constituting the loop are replaced by a single node with the mass distribution equal to the combination of mass distributions of its constituents. The marginals obtained after propagation for the nodes corresponding to initial ABox assertions will reflect updated mass distributions. After the propagation we can remove the statement with the lowest plausibility from each of the MISO found at the diagnosis stage.

Calculating the beliefs for our example gives the following Dempster-Shafer plausibility values for ABox statements: $Pl(A1)=0.94$, $Pl(A2)=0.58$, $Pl(A3)=0.8$, $Pl(A4)=0.65$. In order to make the ontology consistent it is sufficient to remove from both conflict sets an axiom with the lowest plausibility value (A2). In this example, we can see how the results using Dempster-Shafer belief propagation differ from the Bayesian interpretation. Bayesian probabilities, in this case, are calculated in the same way as Dempster-Shafer support values. If we use confidence values as probabilities and propagate them using the same valuation network we will obtain the results: $P(A1)=0.66$, $P(A2)=0.35$, $P(A3)=0.32$ and $P(A4)=0.33$. In this scenario, we would remove A3 and A4 because of the negative belief bias. Also we can see that all three statements A2, A3 and A4 will be considered wrong in such a scenario (resulting probability is less than 0.5). The Dempster-Shafer approach provides more flexibility by making it possible to reason about both support (“harsh” queries) and plausibility (“lenient” queries).

5 Conclusion and Future Work

In this paper, we described how the Dempster-Shafer theory of evidence can be used for dealing with ABox-level inconsistencies produced by inaccurate information extraction. It would be interesting to investigate if the capabilities of the Dempster-Shafer uncertainty representation (e.g., explicit representation of ignorance) can be utilized for knowledge modelling at the TBox level. In [23] it was shown that the Dempster-Shafer approach may lead to problems when it is used to represent uncertainty of inferencing rules (i.e., TBox-level) and not only of pieces of evidence (ABox assertions). These problems occur if the ontology contains contradictory pieces of knowledge, and are caused by the fact that

the Dempster-Shafer approach does not distinguish pieces of evidence regarding specific individuals from generic rules applicable to all individuals. It will be interesting to investigate if these problems can be avoided when modelling description logic axioms.

The algorithm described in the paper focuses on only one aspect of provenance information: confidence values assigned by extraction algorithms. However, such an approach has its limitations: for instance, we know that rule-based extractors tend to repeat their errors when applied to several documents. Such reoccurring errors lead to erroneous inconsistency resolution if interpreted as independent pieces of evidence. In order to improve the quality of the fusion procedure, it would be useful to take into account other kinds of provenance information, in particular, the reliability of the extraction algorithm itself, the reliability of sources from which statements were extracted, and the timestamp reflecting when each statement was produced. This we consider our primary direction for the future work.

6 Acknowledgements

This work was funded by the X-Media project (www.x-media-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978.

References

1. Welty, C., Murdock, W.: Towards knowledge acquisition from information extraction. In: 5th International Semantic Web Conference, Athens, GA, USA, November 5-9, 2006. (2006) 709–722
2. Popov, B.: KIM - a semantic platform for information extraction and retrieval. *Natural language engineering* **10** (2004) 375–392
3. Iria, J.: Relation extraction for mining the Semantic Web. In: Dagstuhl Seminar on Machine Learning for the Semantic Web, Dagstuhl, Germany (2005)
4. Horvitz, E.J., Heckerman, D.E., Langlotz, C.P.: A framework for comparing alternative formalisms for plausible reasoning. In: AAAI-86. (1986) 210–214
5. Shafer, G.: A mathematical theory of evidence. Princeton University Press (1976)
6. Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A framework for handling inconsistency in changing ontologies. In: Proceedings of the International Semantic Web Conference (ISWC2005). (2005) 353–367
7. Kalyanpur, A., Parsia, B., Sirin, E., Grau, B.C.: Repairing unsatisfiable concepts in OWL ontologies. In: ESWC2006. (2006) 170–184
8. Haase, P., Volker, J.: Ontology learning and reasoning - dealing with uncertainty and inconsistency. In: International Workshop on Uncertainty Reasoning for the Semantic Web (URSW). (2005) 45–55
9. Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J.Z., Horrocks, I.: Fuzzy OWL: Uncertainty and the semantic web. In: International Workshop of OWL: Experiences and Directions, ISWC 2005. (2005)

10. Straccia, U.: Towards a fuzzy description logic for the Semantic Web (preliminary report). In: 2nd European Semantic Web Conference (ESWC-05). Number 3532 in Lecture Notes in Computer Science, Crete, Springer Verlag (2005) 167–181
11. Ding, Z., Peng, Y.: A probabilistic extension to ontology language OWL. In: 37th Hawaii International Conference On System Sciences (HICSS-37). (2004)
12. da Costa, P.C.G., Laskey, K.B., Laskey, K.J.: PR-OWL: A Bayesian ontology language for the semantic web. In: Workshop on Uncertainty Reasoning for the Semantic Web, ISWC 2005. (2005)
13. Li, Y., Bontcheva, K., Cunningham, H.: SVM based learning system for information extraction. In: Deterministic and Statistical Methods in Machine Learning. (2005) 319–339
14. Ciravegna, F., Wilks, Y.: Designing Adaptive Information Extraction for the Semantic Web in Amilcare. In: Annotation for the Semantic Web. (2003)
15. Zhu, J., Uren, V., Motta, E.: ESpotter: Adaptive named entity recognition for web browsing. In: Professional Knowledge Management Conference (WM2005). (2005)
16. de Campos, L.M., Huete, J.F., Moral, S.: Uncertainty management using probability intervals. In: Advances in Intelligent Computing IPMU '94. (1994) 190–199
17. Gaifman, H.: A theory of higher order probabilities. In: TARK '86: Proceedings of the 1986 conference on Theoretical aspects of reasoning about knowledge, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1986) 275–292
18. Richardson, M., Agrawal, R., Domingos, P.: Trust management for the Semantic Web. In: 2nd International Semantic Web Conference, ISWC 2003, Sanibel Island, Florida (2003) 351–368
19. Nickles, M.: Social acquisition of ontologies from communication processes. *Journal of Applied Ontology* (2007)
20. Reiter, R.: A theory of diagnosis from first principles. *Artificial Intelligence* **32**(1) (1987) 57–95
21. Shenoy, P.P.: Valuation-based systems: a framework for managing uncertainty in expert systems. In: *Fuzzy logic for the management of uncertainty*. John Wiley & Sons, Inc., New York, NY, USA (1992) 83–104
22. Shenoy, P.P., Shafer, G.: Axioms for probability and belief-function propagation. In: *Readings in uncertain reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1990) 575–610
23. Pearl, J.: Bayesian and belief-function formalisms for evidential reasoning: a conceptual analysis. In: *Readings in uncertain reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1990) 575–610