Assessing Trust in Uncertain Information

Achille Fokoue[1], Mudhakar Srivatsa[1], and Rob Young[2]

[1] IBM Research, USA achille, msrivats@us.ibm.com
[2] Defense Science and Technology Lab, UK riyoung@dstl.gov.uk

Abstract. On the Semantic Web, decision makers (humans or software agents alike) are faced with the challenge of examining large volumes of information originating from heterogeneous sources with the goal of ascertaining trust in various pieces of information. While previous work has focused on simple models for review and rating systems, we introduce a new trust model for rich, complex and uncertain information.We present the challenges raised by the new model, and the results of an evaluation of the first prototype implementation under a variety of scenarios.

1 Introduction

Decision makers (humans or software agents alike) relying on information available on the web are increasingly faced with the challenge of examining large volumes of information originating from heterogeneous sources with the goal of ascertaining trust in various pieces of information. Several authors have explored various trust computation models (e.g., eBay recommendation system [14], NetFlix movie ratings [13], EigenTrust [10], PeerTrust [15], etc.) to assess trust in various entities. A common data model subsumed by several trust computation models (as succinctly captured in Kuter and Golbeck [11]) is the ability of an entity to assign a *numeric* trust score to another entity (e.g., eBay recommendation, Netflix movie ratings, etc.). Such pair-wise numeric ratings contribute to a (dis)similarity score (e.g., based on $\mathcal{L}_1$ norm, $\mathcal{L}_2$ norm, cosine distance, etc.) which is used to compute personalized trust scores (as in PeerTrust) or recursively propagated throughout the network to compute global trust scores (as in EigenTrust).

A pair-wise numeric score based data model may impose severe limitations in several real-world applications. For example, let us suppose that information sources $\{S_1, S_2, S_3\}$ assert axioms ϕ_1 = *all men are mortal*, ϕ_2 = *Socrates is a man* and ϕ_3 = *Socrates is not mortal* respectively. While there is an obvious conflict when all the three axioms are put together, we note that: (i) there is no pair-wise conflict, and (ii) there is no obvious numeric measure that captures (dis)similarity between two information sources.

This problem becomes even more challenging because of uncertainty associated with real-world data and applications. Uncertainty manifests itself in several diverse forms: from measurement errors (e.g., sensor readings) and stochasticity in physical processes (e.g., weather conditions) to reliability/trustworthiness of data sources; regardless of its nature, it is common to adopt a probabilistic measure for uncertainty. Reusing the *Socrates* example above, each information source S_i may assert the axiom ϕ_i with a

certain probability p_i = 0.6. Further, probabilities associated with various axioms need not be (statistically) independent. In such situations, the key challenge is develop trust computation models for rich (beyond pair-wise numeric ratings) and uncertain (probabilistic) information.

The contributions of this paper are three fold. First, our approach offers a rich data model for trust. We allow information items to be encoded in inconsistency-tolerant extension of Bayesian Description Logics [3] (BDL)[3] with axioms of the form $\phi : X$ [4] where ϕ is a classical axiom (in Description Logics (DL [1])) that is annotated with a boolean random variable from a Bayesian network [7]. Intuitively, $\phi : X$ can be read as follows: the axiom ϕ holds when the Boolean random variable X is true. Dependencies between axioms (e.g., $\phi_1 : X_1$ and $\phi_2 : X_2$) are captured using the Bayesian network that represents a set of random variables (corresponding to the annotations; e.g., X_1, X_2) and their conditional probability distribution functions (e.g., $Pr(X_2|X_1)$).

Second, our approach offers a trust computation model over uncertain information (encoded as BDL axioms). Intuitively, our approach allows us to compute a degree of inconsistency over a probabilistic knowledge base. We note that inconsistencies correspond to conflicts in information items reported by one or more information sources. Our approach assigns numeric weights to the degree of inconsistency using the *possible world* semantics (the formal semantics is given in section 3). Revisiting the *Socrates* example, three probabilistic axioms $\phi_i : p_i$[5] correspond to eight possible worlds (the power set of the set of axioms without annotations) corresponding to $\{\{\phi_1, \phi_2, \phi_3\}, \{\phi_1, \phi_2\}, \cdots, \emptyset\}$. Each possible world has probability measure that can be derived from p_i. For instance, the probability of a possible world $\{\phi_1, \phi_2\}$ is given by $p_1 * p_2 * (1 - p_3)$. The degree of inconsistency of a knowledge base is then computed as the sum of the probabilities associated with possible worlds that are inconsistent.

In the presence of inconsistencies, our approach extracts justifications $-$ minimal sets of axioms that together imply an inconsistency [9]. Our trust computation model essentially propagates the degree of inconsistency as blames (or penalties) to the axioms contributing to the inconsistency via justifications. This approach essentially allows us to compute trust in information at the granularity of an axiom. Indeed one may aggregate trust scores at different levels of granularity; e.g., axioms about a specific topic (e.g., birds), one information source (e.g., John), groups of information sources (e.g., all members affiliated with ACM), etc. Intuitively, our trust computation model works as follows. First, we compute a probability measure for each justification as the sum of the probabilities associated with possible worlds in which the justification holds (namely, all the axioms in the justification are present). Second, we partition the degree of inconsistency across all justifications; for instance, if a justification J_1 holds in 80% of the possible worlds then it is assigned four times the blame as a justification J_2 that holds in 20% of the possible worlds. Third, we partition the penalty associated with a justification across all axioms in the justification using a biased (on prior trust assessments)

[3] BDL is a simple probabilistic extension of Description Logics, the foundation of OWL DL.

[4] This is a very simplified version of the BDL formulation given here for ease of the presentation. The complete and formal definition of BDL is presented in section 2

[5] $\phi_i : p_i$ is a shorthand notation for $\phi_i : X_i$ and $Pr(X_i = true) = p_i$ for some independent random variable X_i

or an unbiased partitioning scheme. We note that there may be alternate approaches to derive trust scores from inconsistency measures and justifications; indeed, our approach is flexible and extensible to such trust computation models.

A naive implementation of our trust computation model requires *all* justifications. While computing a justification is an easy problem, exhaustively enumerating all possible justifications is known to be hard problem [9]. We formulate exhaustive enumeration of justifications as a tree traversal problem and develop an *importance sampling* approach to uniformly and randomly sample justifications without completely enumerating them. Unbiased sampling of justifications ensures that the malicious entities cannot game the trust computation model; say, selectively hide justifications that include axioms from malicious entities (and thus evade penalties) from the sampling process. For scalability reasons, our trust computation model operates on a random sample of justifications. A malicious entity may escape penalties due to incompleteness of justifications; however, across multiple inconsistency checks a malicious entity is likely to incur higher penalties (and thus lower trust score) than the honest entities.

Third, we have developed a prototype of our trust assessment system by implementing a probabilistic extension, PSHER, to our publicly available highly scalable DL reasoner SHER [6]. To avoid the exponential blow up due to the fact that the number of possible worlds in the worst case is exponential in the number of axioms, we use an error-bounded approximation algorithm to compute the degree of inconsistency of a probabilistic knowledge base and the weight of its justifications. Finally, we empirically evaluate the efficacy of our scheme (on a publicly available UOBM dataset) when malicious sources use an oscillating behavior to milk the trust computation model and when honest sources are faced with measurement errors (high uncertainty) or commit honest mistakes.

The remainder of the paper is organized as follows. After a brief introduction of Bayesian Description Logics (BDL) in Section 2, Section 3 describes an inconsistency-tolerant extension of BDL and presents solutions to effectively compute justifications (a proxy for (dis)similarity scores in our trust computation model). Section 4 describes our trust computation model. Section 5 presents an experimental evaluation of our system. We finally conclude in Section 6.

2 Background

In this section, we briefly describe our data model for uncertain information.

2.1 Bayesian Network Notation

We briefly recall notations for a Bayesian Network, used in the remainder of the paper. V: set of all random variables in a Bayesian network (e.g., V = $\{V_1, V_2\}$). $D(V_i)$ (for some variable $V_i \in V$): finite set of values that V_i can take (e.g., $D(V_1) = \{0, 1\}$ and $D(V_2) = \{0, 1\}$). v: assignment of all random variables to a possible value (e.g., v = $\{V_1 = 0, V_2 = 1\}$). $v|X$ (for some $X \subseteq V$): projection of v that only includes the random variables in X (e.g., $v|\{V_2\}$ = $\{V_2 = 1\}$). $D(X)$ (for some $X \subseteq V$): Cartesian product of the domains $D(X_i)$ for all $X_i \in X$.

2.2 Bayesian Description Logics

Bayesian Description Logics [3] is a class of probabilistic description logic wherein each logical axiom is annotated with an event which is associated with a probability value via a Bayesian Network. In this section, we describe Bayesian DL at a syntactic level followed by a detailed example. A probabilistic axiom over a Bayesian Network BN over a set of variables V is of the form $\phi : e$, where ϕ is a classical DL axiom, and the probabilistic annotation e is an expression of one of the following forms: $X = x$ or $X \neq x$ where $X \subseteq V$ and $x \in D(X)$. Intuitively, every probabilistic annotation represents a scenario (or an event) which is associated with the set of all value assignments $V = v$ with $v \in D(V)$ that are compatible with $X = x$ (that is, $v|X = x$) and their probability value $Pr_{BN}(V = v)$ in the Bayesian network BN over V. Simply put, the semantics of a probabilistic axiom $\phi : X = x$ is as follows: when event $X = x$ occurs then ϕ holds. $\phi : p$, where $p \in [0, 1]$, is often used to directly assign a probability value to an classical axiom ϕ. This is an abbreviation for $\phi : X_0 = true$, where X_0 is a boolean random variable which is independent from all other variables and such that $Pr_{BN}(X_0 = true) = p$. We abbreviate the probabilistic axiom of the form $\top : e$ (resp. $\phi : \top$) as e (resp. ϕ).

A probabilistic knowledge base (KB) $K = (\mathcal{A}, \mathcal{T}, BN)$ consists of: 1) a Bayesian Network BN over a set of random variables V, 2) a set of probabilistic Abox axioms $\mathcal{A}$ of the form $\phi : e$, where ϕ is a classical Abox axiom, and 3) a set of probabilistic Tbox axioms $\mathcal{T}$ of the form $\phi : e$, where ϕ is a classical Tbox axiom. The following example illustrates how this formalism can be used to describe road conditions influenced by probabilistic events such as weather conditions:

$$\mathcal{T} = \{SlipperyRoad \sqcap OpenedRoad \sqsubseteq HazardousCondition,$$
$$Road \sqsubseteq SlipperyRoad : Rain = true\}$$
$$\mathcal{A} = \{Road(route9A), OpenedRoad(route9A) : TrustSource = true\}$$

In this example, the Bayesian network BN consists of three variables: $Rain$, a boolean variable which is true when it rains; $TrustSource$, a boolean variable which is true when the source of the axiom $OpenedRoad(route9A)$ can be trusted; and $Source$, a variable which indicates the provenance of the axiom $OpenedRoad(route9A)$. The probabilities specified by BN are as follows:

$$Pr_{BN}(TrustSource = true|Source = `Mary') = 0.8,\ Pr_{BN}(Rain = true) = 0.7$$

$$Pr_{BN}(TrustSource = true|Source = `John') = 0.5,\ Pr_{BN}(Source = `John') = 1$$

The first Tbox axiom asserts that a opened road that is slippery is a hazardous condition. The second Tbox axiom indicates that when it rains, roads are slippery. The Abox axioms assert that $route9A$ is a road and, assuming that the source of the statement $OpenedRoad(route9A)$ is trusted, $route9A$ is opened.

Informally, probability values computed through the Bayesian network 'propagate' to the 'DL side' as follows. Each assignment v of all random variables in BN (e.g.,v = {$Rain$ = true, $TrustSource$ = false, $Source$= 'John'}) corresponds to a primitive event ev (or a scenario). A primitive event ev is associated, through BN, to a probability

value p_{ev} and a classical DL KB K_{ev}[6] which consists of all classical axioms annotated with a compatible probabilistic annotation (e.g., $SlipperyRoad \sqcap OpenedRoad \sqsubseteq HazardousCondition, Road \sqsubseteq SlipperyRoad, Road(route9A)$). The probability value associated with the statement ϕ (e.g., $\phi = HazardousCondition(route9A)$) is obtained by summing p_{ev} for all ev such that the classical KB K_{ev} entails ϕ (e.g., $Pr(HazardousCondition(route9A)) = 0.35$).

3 Inconsistency and Justification

The ability to detect contradicting statements and measure the relative importance of the resulting conflict is a key prerequisite to estimate the (dis)similarity between information sources providing rich, complex and probabilistic assertions expressed as BDL axioms. Unfortunately, in the traditional BDL semantics [3], consistency is still categorically defined, i.e., a probabilistic KB is either completely satisfied or completely unsatisfied. In this section, we address this significant shortcoming by using a refined semantics which introduces the notion of degree of inconsistency. We start by presenting the traditional BDL semantics, which does not tolerate inconsistency.

For $v \in V$, we say that v is compatible with the probabilistic annotation $X = x$ (resp. $X \neq x$), denoted $v \models X = x$ (resp. $v \models X \neq x$), iff $v|X = x$ (resp. $v|X \neq x$).

Recall that BDL axioms ($\phi : e$) are extensions of classical axioms (ϕ) with a probabilistic annotation (e). BDL semantics defines an annotated interpretation as an extension of a first-order interpretation by assigning a value $v \in D(V)$ to V. An annotated interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ is defined in a similar way as a first-order interpretation except that the interpretation function $.^{\mathcal{I}}$ also maps the set of variables V in the Bayesian Network to a value $v \in D(V)$. An annotated interpretation $\mathcal{I}$ satisfies a probabilistic axiom $\phi : e$, denoted $\mathcal{I} \models \phi : e$, iff $V^{\mathcal{I}} \models e \Rightarrow \mathcal{I} \models \phi$ [7]. Now, a probabilistic interpretation is defined as a probabilistic distribution over annotated interpretations.

Definition 1 *(From [3]) A* probabilistic interpretation Pr *is a probability function over the set of all annotated interpretations that associates only a finite number of annotated interpretations with a positive probability. The probability of a probabilistic axiom $\phi : e$ in Pr, denoted $Pr(\phi : e)$, is the sum of all $Pr(\mathcal{I})$ such that $\mathcal{I}$ is an annotated interpretation that satisfies $\phi : e$. A probabilistic interpretation Pr satisfies (or is a model of) a probabilistic axiom $\phi : e$ iff $Pr(\phi : e) = 1$. We say Pr satisfies (or is a model of) a set of probabilistic axioms F iff Pr satisfies all $f \in F$.*

Finally, we define the notion of consistency of a probabilistic knowledge base.

Definition 2 *(From [3]) The probabilistic interpretation Pr satisfies (or is a model of) a probabilistic knowledge base $K = (\mathcal{T}, \mathcal{A}, BN)$ iff (i) Pr is a model of $\mathcal{T} \cup \mathcal{A}$ and (ii) $Pr_{BN}(V = v) = \sum_{\mathcal{I} \text{ s.t. } V^{\mathcal{I}} = v} Pr(\mathcal{I})$ for all $v \in D(V)$. We say KB is consistent iff it has a model Pr.*

[6] K_{ev} was informally referred to as a 'possible world' in the introduction

[7] This more expressive implication semantics differs from the equivalence semantics of [3]

We note that condition (ii) in the previous definition ensures that the sum of probability values for annotated interpretations mapping V to $v \in D(V)$ is the same probability value assigned to $V = v$ by the Bayesian Network.

3.1 Degree of Inconsistency

In the previously presented traditional BDL semantics, consistency is still categorically defined. We now address this significant shortcoming for our trust application using a refined semantics which introduces the notion of degree of inconsistency.

First, we illustrate using a simple example the intuition behind the notion of degree of inconsistency for a KB. Let K be the probabilistic KB defined as follows: $K = (\mathcal{T}, \mathcal{A} \cup \{\top \sqsubseteq \bot : X = true\}, BN)$ where $\mathcal{T}$ is a classical Tbox and $\mathcal{A}$ is a classical Abox such that the classical KB $cK = (\mathcal{T}, \mathcal{A})$ is consistent; BN is a Bayesian Network over a single boolean random variable X, and the probability $Pr_{BN}(X = true) = 10^{-6}$ that X is true is extremely low. Under past probabilistic extensions to DL, the K is completely inconsistent, and nothing meaningful can be inferred from it. This stems from the fact that when X is true, the set of classical axioms that must hold (i.e., $\mathcal{T} \cup \mathcal{A} \cup \{\top \sqsubseteq \bot\}$) is inconsistent. However, the event $X = true$ is extremely unlikely, and, therefore, it is unreasonable to consider the whole probabilistic KB inconsistent. Intuitively, the likelihood of events, whose set of associated classical axioms is inconsistent, represents the degree of inconsistency of a probabilistic KB.

We now formally define a degree of inconsistency and present an inconsistency-tolerant refinement of the semantics of a Bayesian DL.

Definition 3 *An annotated interpretation $\mathcal{I}$ is an annotated model of a probabilistic KB $K = (\mathcal{T}, \mathcal{A}, BN)$ where BN is a Bayesian Network over a set of variables V iff for each probabilistic axiom $\phi : e$, $\mathcal{I}$ satisfies $\phi : e$.*

In order, to measure the degree of inconsistency, we first need to find all primitive events v (i.e., elements of the domain $D(V)$ of the set of variables V) for which there are no annotated models $\mathcal{I}$ such that $V^{\mathcal{I}} = v$.

Definition 4 *For a probabilistic KB $K = (\mathcal{T}, \mathcal{A}, BN)$ where BN is a Bayesian Network over a set of variables V, the set of inconsistent primitive events, denoted $U(K)$, is the subset of $D(V)$, the domain of V, such that $v \in U(K)$ iff there is no annotated model $\mathcal{I}$ of K such that $V^{\mathcal{I}} = v$*

Finally, the degree of inconsistency of a probabilistic knowledge base is defined as the probability of occurrence of an inconsistent primitive event.

Definition 5 *Let $K = (\mathcal{T}, \mathcal{A}, BN)$ be a probabilistic KB such that BN is a Bayesian Network over a set of variables V. The degree of inconsistency of K, denoted $DU(K)$, is a real number between 0 and 1 defined as follows:*

$$DU(K) = \sum_{v \in U(K)} Pr_{BN}(V = v)$$

A probabilistic interpretation Pr (as per Definition 1) satisfies (or is a model of) a probabilistic KB $K = (\mathcal{T}, \mathcal{A}, BN)$ to a degree d, $0 < d \leq 1$ iff.:

– *(i)* Pr *is a model as* $\mathcal{T} \cup \mathcal{A}$ *(same as in Definition 2)*
– *(ii) for* $v \in V$,

$$\sum_{\mathcal{I} \text{ s.t. } V^{\mathcal{I}}=v} Pr(\mathcal{I}) = \begin{cases} 0 & \text{if } v \in U(K) \\ \frac{Pr_{BN}(V=v)}{d} & \text{if } v \notin U(K) \end{cases}$$

– *(iii)* $d = 1 - DU(K)$

A probabilistic knowledge base $K = (\mathcal{T}, \mathcal{A}, BN)$ *is consistent to the degree* d*, with* $0 < d \leq 1$*, iff there is a probabilistic interpretation that satisfies* K *to the degree* d*. It is completely inconsistent (or satisfiable to the degree* 0*), iff* $DU(K) = 1$.

Informally, by assigning a zero probability value to all annotated interpretations corresponding to inconsistent primitive events, (ii) in Definition 5 removes them from consideration, and it requires that the sum of the probability value assigned to interpretations mapping V to v for $v \notin U(K)$ is the same as the joint probability distribution Pr_{BN} defined by BN with a normalization factor d.

In practice, computing the degree of inconsistency of a Bayesian DL KB can be reduced to classical description logics consistency check as illustrated by Theorem 1. First we introduce an important notation used in the remainder of the paper:

Notation 1 *Let* $K = (\mathcal{T}, \mathcal{A}, BN)$ *be a probabilistic KB. For every* $v \in D(V)$*, let* $\mathcal{T}_v$ *(resp.,* $\mathcal{A}_v$*) be the set of all axioms* ϕ *for which there exists a probabilistic axiom* $\phi : e$ *in* $\mathcal{T}$ *(resp.,* $\mathcal{A}$*), such that* $v \models e$*.* K_v *denotes the classical KB* $(\mathcal{T}_v, \mathcal{A}_v)$*. Informally,* K_v *represents the classical KB that must hold when the primitive event* v *occurs.* $K_\top$ *denotes the classical KB obtained from* K *after removing all probabilistic annotations:* $K_\top = (\cup_{v \in D(V)} \mathcal{T}_v . \cup_{v \in D(V)} \mathcal{A}_v)$.

Theorem 1 *A probabilistic KB* $K = (\mathcal{T}, \mathcal{A}, BN)$ *is consistent to the degree* d *iff.*

$$d = 1 - \sum_{v \text{ s.t. } K_v \text{ inconsistent}} Pr_{BN}(V = v)$$

The proof of Theorem 1 is a consequence of Lemma 1.

Lemma 1 *Let* K *be a probabilistic KB.* $v \in U(K)$ *iff* K_v *is inconsistent.*

3.2 Inconsistency Justification

A conflict or contradiction is formally captured by the notion of an inconsistency justification − minimal inconsistency preserving subset of the KB.

Definition 6 *Let* $K = (\mathcal{T}, \mathcal{A}, BN)$ *be a probabilistic KB consistent to the degree* d *such that* BN *is a Bayesian Network over a set of variables* V*.* $\mathcal{J}$ *is an inconsistency justification iff. 1)* $\mathcal{J} \subseteq (\mathcal{T}, \mathcal{A})$*, 2)* $(\mathcal{J}, BN)$ *is probabilistic KB consistent to the degree* d' *such that* $d' < 1$*, and 3) for all* $\mathcal{J}' \subset \mathcal{J}$*,* $(\mathcal{J}', BN)$ *is probabilistic KB consistent to the degree* 1 *(i.e.* $(\mathcal{J}', BN)$ *is completely consistent). The degree* $DU(\mathcal{J})$ *of an inconsistency justification* $\mathcal{J}$ *is defined as the degree of inconsistency of the probabilistic KB made of its axioms:* $DU(\mathcal{J}) = DU((\mathcal{J}, BN))$

Justification computation in a probabilistic KB reduces to justification computation in classical KBs as shown by the following theorem, which is a direct consequence of Theorem 1 and Definition 6:

Theorem 2 *Let $K = (\mathcal{T}, \mathcal{A}, BN)$ be a probabilistic KB, where BN is a Bayesian network over a set V of random variables. $\mathcal{J}$ is an inconsistency justification of K iff. there exists $v \in D(V)$ such that $Pr_{BN}(V = v) > 0$ and $\mathcal{J}_\top$, the classical KB obtained from $\mathcal{J}$ by removing all probabilistic annotations, is an inconsistency justification of K_v. Furthermore, the degree, $DU(\mathcal{J})$, of an inconsistency justification $\mathcal{J}$ is as follows:*

$$DU(\mathcal{J}) = \sum_{v \text{ s.t. } \mathcal{J}_\top \subseteq K_v} Pr_{BN}(V = v)$$

Thus, once we have found a classical justification in a classical KB K_v for $v \in D(V)$ using, for example, the scalable approach described in our previous work [4], the degree of the corresponding probabilistic justification can be obtained through simple set inclusion tests.

Theorems 1 and 2 provide a concrete mechanism to compute degree of inconsistency of a probabilistic KB, and a degree of inconsistency of a justification. However, they are highly intractable since they require an exponential number, in the number of variables in BN, of corresponding classical tasks. We will address this issue in the next section.

3.3 Error-Bounded Approximate Reasoning

A Bayesian network based approach lends itself to fast Monte Carlo sampling algorithms for scalable partial consistency checks and query answering over a large probabilistic KB. In particular, we use a *forward sampling* approach described in [2] to estimate $pr = \sum_{v \in \Pi} Pr_{BN}(V = v)$ (recall theorem 1 and 2). The forward sampling approach generates a set of samples $v_1, \cdots, v_n$ from BN (each sample is generated in time that is linear in the size of BN) such that the probability pr can be estimated as $\widehat{pr_n} = \frac{1}{n} * \sum_{i=1}^{n} I(v_i \in \Pi)$, where $I(z) = 1$ if z is true; 0 otherwise. One can show that $\widehat{pr_n}$ is an unbiased estimator of pr such that $\lim_{n\to\infty} \sqrt{n} * (\widehat{pr_n} - pr) \rightarrow \mathcal{N}(0, \sigma_z^2)$, where $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2 and σ_z^2 denotes the variance of $I(z)$ for a boolean variable z. Hence, the sample size n which guarantees an absolute error of ϵ or less with a confidence level η is given by the following formula: $n = \frac{2*(erf^{-1}(\eta))^2 * \sigma^2_{z_{max}}}{\epsilon^2}$, where erf^{-1} denotes the inverse Gauss error function ($\sigma^2_{z_{max}} = 0.25$ for a boolean random variable). For example, to compute the degree of consistency of a probabilistic KB within $\pm 5\%$ error margin with a 95% confidence, the sample size $n = 396$ is necessary.

3.4 Sampling Justifications in a Classical KB

Ideally, it is desirable to find all classical justifications. Computing a single justification can be done fairly efficiently by 1) using tracing technique to obtain a significantly small set S of axioms that is responsible for an inconsistency discovered by a single

consistency test, and 2) performing additional $|S|$ consistency check on KBs of size at most $|S| - 1$ to remove extraneous elements from S. In our previous work [4], we presented a scalable approach to efficiently compute a large number of − but not all − justifications in large and expressive KBs through the technique of summarization and refinement [5]. The idea consists in looking for patterns of justifications in a dramatically reduced summary of the KB, and retrieve concrete instances of these patterns in the real KB.

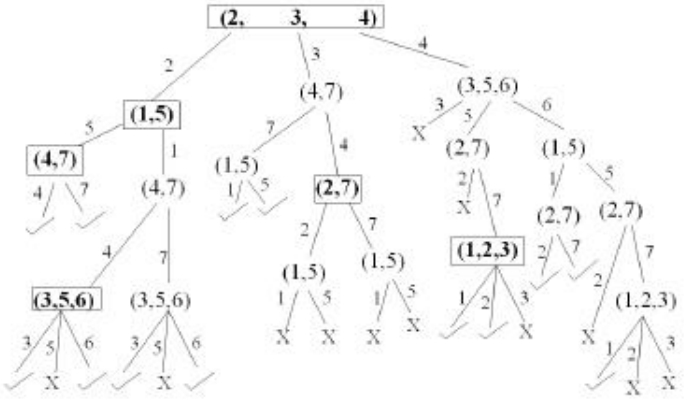

Fig. 1: Computing all justifications using Reiter's Hitting Set Tree Algorithm from [9]

Unfortunately, computing all justifications is well known to be intractable even for small and medium size expressive KBs [9]. [9] establishes a connection between the problem of finding all justifications and the hitting set problem (i.e., given n sets S_i, find sets that intersect each S_i). The intuition behind this result is the fact that in order to make an inconsistent KB consistent at least one axiom from each justification must be removed. Therefore, starting from a single justification a Reiter's Hitting Tree can be constructed in order to get all justifications as illustrated in Figure 1 from [9]: Starting from the first justification $J = \{2, 3, 4\}$ computed in the KB K (J is set to be the root v_0 of the tree), the algorithm arbitrary selects an axiom in J, say 2, and creates a new node w with an empty label in the tree and a new edge $< v_0, w >$ with axiom 2 in its label. The algorithm then tests the consistency of the $K - \{2\}$. If it is inconsistent, as in this case, a justification J' is obtained for $K - \{2\}$, say $\{1, 5\}$, and it is inserted in the label of the new node w. This process is repeated until the consistency test is positive in which case the new node is marked with a check mark. As an important optimization, we stop exploring super set of path discovered earlier and marked the node with 'X'.

In order to avoid the high cost associated with exploring the whole Hitting Set Tree to find all conflicts. One can find the first K conflicts by exploring the Reiter's Hitting Set Tree (HST) until K distinct justifications are found. The problem with this approach is that nodes in the HST are not equally likely to be selected with such a scheme: the probability $\pi(v_d)$ of a node v_d in a path $< v_0 v_1 ... v_d >$ to be selected is $\pi(v_d) = \prod_{0 \leq i < d}(1/|v_i|)$, where $|v_i|$ denotes the number of axioms in the justification v_i. As a result, a malicious source can use the bias in the sampling to 'hide' its conflicts.

However, since the bias can be precisely quantified, one can obtain an unbiased sample as follows. We select K nodes in the HST by exploring the HST in the normal way, but each time a node v_i is encountered, it is selected iff. a random number r generated uniformly from [0,1] is such that $r \leq \min(\beta/\pi(v_i), 1)$, where β is a strictly positive real number. The following Proposition shows that, in this approach, for a sample of K HST nodes, if β is properly chosen, then the expected number of time a node is selected is identical for all nodes.

Proposition 1 *Let N_v denotes the random variable representing the number of time the node v appears in a HST sample of size K. The expected value $E(N_v)$ of N_v is:*

$$E(N_v) = \begin{cases} K * \pi(v) & \text{if } \beta \geq \pi(v) \\ K * \beta & \text{if } 0 < \beta < \pi(v) \end{cases}$$

Thus, if β is chosen such that $0 < \beta < \min_{v \in HST}(\pi(v))$, then we obtain an unbiased sample from the HST. Unfortunately, the minimum value of $\pi(v)$ depends on the tree structure (branching factor and maximum depth), and cannot be computed precisely without exploring the whole HST. In practice, we use the following sampling approach to select K nodes (the trade-off between computation cost and bias in the sample is controlled by a parameter of the algorithm, α):

1. Let $visited$ denote the set of visited nodes. Set $visited$ to $\emptyset$,
2. Traverse the HST in any order, and add the first $\max(K - |visited|, 1)$ nodes visited to $visited$
3. Let π_{min} be the minimum value of $\pi(v)$ for $v \in visited$.
4. Set $\beta = \pi_{min}/\alpha$, where $\alpha > 1$ is a parameter of the sampling algorithm which controls the trade-off between computation cost and biased in the sampling. Higher values of α, while reducing the bias in our sampling, increase the computation cost by reducing the probability of a node selection − hence, increasing the length of tree traversal.
5. For each $v \in visited$, add it to the result set RS with a probability of $\beta/\pi(v)$
6. If $|RS| < K$ and the HST has not been completely explored, then set $RS = \emptyset$ and continue the exploration from step 2; otherwise return RS

4 Trust Computation Model

We now briefly formalize the problem of assessing trust in a set IS consisting of n information sources. The trust value assumed or known prior to any statement made by an information source i is specified by a probability distribution $PrTV(i)$ over the domain [0, 1]. For example, a uniform distribution is often assumed for new information source for which we have no prior knowledge. Statements made by each source i is specified in the form of a probabilistic KB $\mathcal{K}^i = (\mathcal{T}^i, \mathcal{A}^i, BN^i)$. The knowledge function C maps an information source i to the probabilistic KB $\mathcal{K}^i$ capturing all its statements. The trust update problem is a triple $(IS, PrTV, C)$ whose solution yields a posterior trust value function $PoTV$. $PoTV$ maps an information source i to a probability distribution over the domain [0, 1], which represents our new belief in the trustworthiness of i after processing statements in $\bigcup_{j \in IS} C(j)$.

In this paper, we only focus on trust computation based on direct observations, that is, on statements directly conveyed to us by the information sources. Inferring trust

from indirect observations (e.g., statements conveyed to us from IS_1 via IS_2) is an orthogonal problem; one could leverage solutions proposed in [10], [15], [11] to infer trust from indirect observations.

4.1 Trust Computation

We model prior and posterior trust of a source i ($PrTV(i)$ and $PoTV(i)$) using a beta distribution $\mathcal{B}(\alpha, \beta)$ as proposed in several other trust computation models including [8]. Intuitively, the reward parameter α and the penalty parameter β correspond to good (non-conflicting) and bad (conflicting) axioms contributed to an information source respectively. The trust assessment problem now reduces to that of (periodically) updating the parameters α and β based on the axioms contributed by the information sources. One may bootstrap the model by setting $PrTV(i)$ to $\mathcal{B}(1, 1)$ – a uniform and random distribution over [0, 1], when we have no prior knowledge. In the rest of this section we focus on computing the reward (α) and penalty (β) parameters.

We use a simple reward structure wherein an information source receives unit reward for every axiom it contributes if the axiom is not in a justification for inconsistency [8]. We use a scaling parameter $\triangle$ to control the relative contribution of reward and penalty to the overall trust assessment; we typically set $\triangle > 1$, that is, penalty has higher impact on trust assessment than the reward. The rest of this section focuses on computing penalties from justifications for inconsistency.

Section 3.4 describes solutions to construct (a random sample of) justifications that explain inconsistencies in the KB; further, a justification J is associated with a weight $DU(J)$ that corresponds to the possible worlds in which the justification J holds (see section 3.2 for formal definition of $DU(J)$ and an algorithm to compute it). For each justification J_i we associate a penalty $\triangle(J_i) = \triangle * DU(J_i)$. The trust computation model traces a justification J_i, to conflicting information sources $\mathcal{S} = \{S_{i_1}, \cdots, S_{i_n}\}$ (for some $n \geq 2$) that contributed to the axioms in J_i. In this paper we examine three solutions to partition $\triangle(J_i)$ units of penalty amongst the contributing information sources as shown below. We use t_{i_j} to denote the expectation of $PrTV(i_j)$ for an information source i_j, that is, $t_{i_j} = \frac{\alpha_{i_j}}{\alpha_{i_j}+\beta_{i_j}}$.

$$\triangle(S_{i_j}) = \begin{cases} \frac{\triangle(J_i)}{n} & \textit{unbiased} \\ \frac{\triangle(J_i)}{n-1} * (1 - \frac{t_{i_j}}{\sum_{k=1}^{n} t_{i_k}}) & \textit{biased by trust in other sources} \\ \triangle(J_i) * \frac{\frac{1}{t_{i_j}}}{\sum_{k=1}^{n} \frac{1}{t_{i_k}}} & \textit{biased by inverse self trust} \end{cases}$$

The unbiased version distributes penalty for a justification equally across all conflicting information sources; the biased versions tend to penalize less trustworthy sources more. One possible approach is to weigh the penalty for a source S_{i_j} by the sum of the expected prior trust values for all the other conflicting sources, namely, $\mathcal{S} - \{S_{i_j}\}$. For instance, if we have three information sources S_{i_1}, S_{i_2} and S_{i_3} with expected prior trust t_{i_1} = 0.1 and $t_{i_2} = t_{i_3}$ = 0.9 then the penalty for source i_1 must be weighted by $\frac{1}{2} * \frac{0.9+0.9}{0.1+0.9+0.9}$ = 0.47, while that of sources i_2 and i_3 must be weighted by

[8] A preprocessing step weeds out trivial axioms (e.g., *sun rises in the east*)

0.265. Clearly, this approach penalizes the less trustworthy source more than the trusted sources; however, we note that even when the prior trust in i_1 is arbitrarily close to zero, the penalty for the honest source i_2 and i_2 is weighted by 0.25. A close observation reveals that a malicious source (with very low prior trust) may penalize honest nodes (with high prior trust) by simply injecting conflicts that involve the honest nodes; for instance, if sources i_2 and i_3 assert axioms ϕ_2 and ϕ_3 respectively, then the malicious source i_1 can assert an axiom $\phi_1 = \neg\phi_2 \vee \neg\phi_3$ and introduce an inconsistency whose justification spans all the three sources. To overcome this problem, this paper uses a third scheme that weights penalties for justifications by the inverse value of prior trust in the information source.

5 Experimental Evaluation

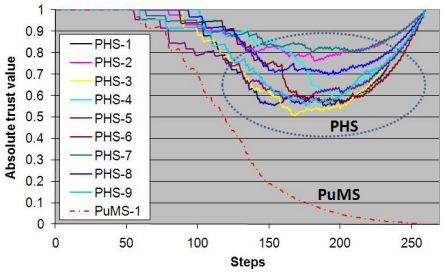

Fig. 2: Trust under single PuMS attack (No duplication)

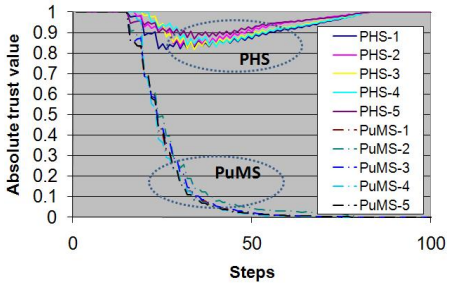

Fig. 3: Trust under 50% PuMS attack (No duplication)

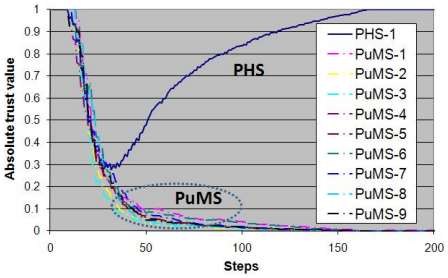

Fig. 4: Trust under 90% PuMS attack (No duplication)

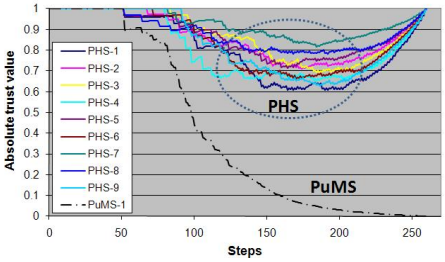

Fig. 5: Trust under single PuMS attack (25% duplication)

To evaluate our approach, we have developed a prototype implementation, PSHER, that extends SHER reasoner [6] to support Bayesian $\mathcal{SHIN}$ (the core of OWL 1.0 DL) reasoning. SHER was chosen for its unique ability to scale reasoning to very large and expressive KBs [5], and to efficiently detect large number of inconsistency justifications in a scalable way [4]. PSHER uses the results of sections 3.1, 3.2 and 3.3 to reduce the problem of computing justifications on a probabilistic KB to detecting those justifications on classical KBs using SHER.

Axioms asserted by various information sources in our experiments were taken from the UOBM benchmark [12] which was modified to $\mathcal{SHIN}$ expressivity, and its Abox

was modified by randomly annotating half of the axioms with probability values. Furthermore, we inserted additional Abox assertions in order to create inconsistencies involving axioms in the original UOBM KB. Note that not all axioms in the original UOBM KB end up being part of an inconsistency, which introduces an asymmetry in information source's knowledge (e.g., a malicious source is not assumed to have complete knowledge of all axioms asserted by other sources).

Due to space limitations, we only present an evaluation of our trust model under different scenarios. Scalability was already demonstrated in our previous work on SHER [4], where we presented a scalable approach to efficiently compute a large number of − but not all − justifications in large and expressive KBs through the technique of summarization and refinement [5]. Scalability of PSHER is achieved through parallelism since each probabilistic reasoning task performed by PSHER is reduced to n corresponding classical tasks evaluated using SHER, where n depends on the desired precision as explained in Section 3.3. In the rest of this section, we report experiments conducted on UOBM1 (one department $\sim$ 74,000 axioms, including added inconsistent axioms and excluding duplication across sources).

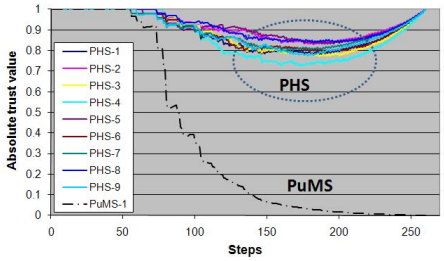

Fig. 6: Trust under single PuMS attack (50% duplication)

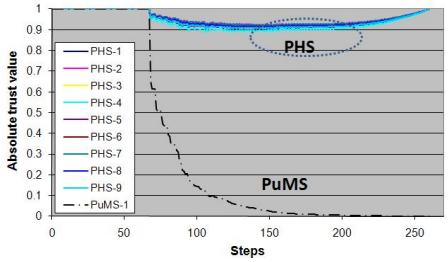

Fig. 7: Trust under single PuMS attack (100% duplication)

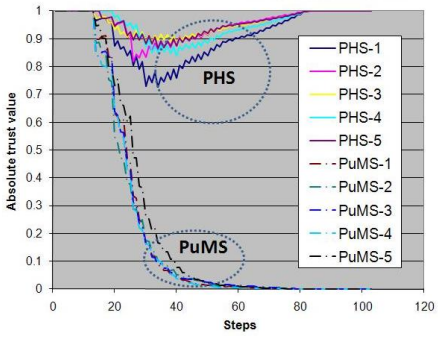

Fig. 8: Trust under 50% PuMS attack (25% duplication)

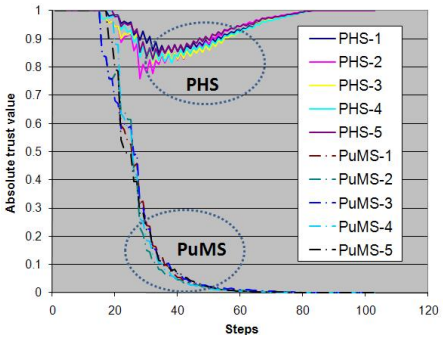

Fig. 9: Trust under 50% PuMS attack (50% duplication)

In our experiments, we considered 4 types of information sources:

– Perfect honest sources (PHS) whose axioms are taken from the UOBM KB before the introduction of inconsistencies.

- Purely malicious sources (PuMS) whose axioms are selected from the ones added to UOBM KB in order to create inconsistencies.
- Imperfect honest sources (IHS) have the majority of their axioms (more than 90%) from the UOBM KB before the introduction of inconsistencies. They allow us to simulate the behavior of our approach when honest sources are faced with measurement errors or commit honest mistakes.
- Partially malicious sources (PaMS) are such that between 10% to 90% of their axioms are selected from the axioms added to UOBM KB to create inconsistency. They are primarily used to simulate the behavior of our approach when malicious sources use an oscillating behavior to milk our trust computation scheme.

Axioms were randomly assigned to various sources without violating the proportion of conflicting vs. non-conflicting axioms for each type of source.

Our first experiment (Figure 2) measures the impact of a single purely malicious source (PuMS) on the trust values of 9 perfect honest sources. The PuMS asserts more and more incorrect axioms contradicting PHS's axioms (at each steps, each source asserts about 100 additional statements until all their axioms have been asserted) while the PHSs continue to assert more of what we consider as correct axioms. Axioms asserted by the PuMS do not necessarily yield an inconsistency in the same step in which they are asserted, but, by the end of the simulation, they are guaranteed to generate an inconsistency. For this experiment, there is no duplication of axioms across sources, and we do not assume any prior knowledge about the trustworthiness of the sources. Since each justification creates by the malicious source also involves at least one PuMS, initially, it manages to drop significantly the absolute trust value of some PHSs (up to 50% for PHS-3). However, a PuMS hurts its trust value significantly more than he hurts those of other sources. As a result of the fact that our scheme is such that less trustworthy sources get assigned a large portion of the penalty for a justification, the single PuMS eventually ends up receiving almost all the penalty for its inconsistencies, which allows the trust values of honest sources to recover. Due to information asymmetry (malicious sources do no have complete knowledge of informations in other sources and thus cannot contradict all the statements of an PHS), our scheme remains robust, in the sense that honest sources would recover, even when the proportion of PuMS increases (see Fig. 3 where 50% of the sources are PuMS and Fig. 4 where 90% of sources are PuMS).

In the previous experiments, although honest sources manage to recover from the attack, they can still be severely hurt before the credibility of the malicious sources decreased enough to enable a recovery for honest sources. This problem can be addressed in two ways: 1) by increasing the degree of redundancy between sources as illustrated in Figures 5, 6, 7, 8 and 9; and 2) by taking into account a priori knowledge of each source as illustrated in Figure 10.

In case of moderate to high redundancy between sources (Figures 5, 6, 7, 8 and 9), a justification generated by a malicious source to compromise a honest source is likely to hurt the malicious much more than the honest source because the axioms in the justification coming from the honest source are likely to be confirmed by (i.e. duplicated in) other honest sources. Therefore, the malicious source will be involved in as many justifications as there are corroborating honest sources, while each corroborating source will be involved in a single justification.

In Figure 10, we assume that we have a high a priori confidence in the trustworthiness of the honest sources: the prior distribution of the trust value of PHS in that experiment is a beta distribution with parameter $\alpha = 2000$ and $\beta = 1$. As expected, in Figure 10, the damage inflicted by the malicious source is significantly reduced compared to Figure 2 where no prior knowledge about the source trustworthiness was taken into account.

The next experiment evaluates the behavior of our scheme when partially malicious sources use an oscillating behavior. They alternate periods where they assert incorrect axioms, contradicting axioms asserted in the same period by other sources, with periods in which they assert only correct axioms. As opposed to previous experiments where malicious axioms asserted in a step were not guaranteed to yield an inconsistency in the same step, in the oscillation experiments, the inconsistency is observed at the same step. As shown in Figure 11 and 12, in absence of prior knowledge, the trust values of partially malicious sources (PaMS) and honest sources drop significantly at the first period in which incorrect axioms are stated. However, malicious sources, which due to information asymmetry, can only contradict limited set of statements from honest sources, never recover significantly, while honest sources quickly improve their trust values by asserting more axioms not involved in conflicts. As in the previous non-oscillating experiments, the negative impact on honest sources can be reduced considerably through axiom duplication and prior strong confidence in their trustworthiness.

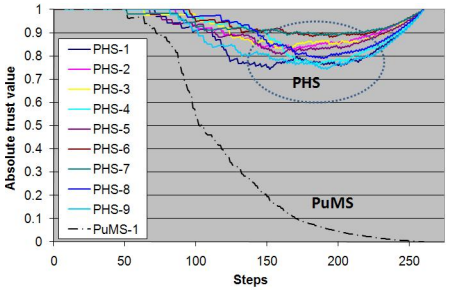

Fig. 10: Trust under single PuMS attack: No duplication - Prior = B(2000,1)

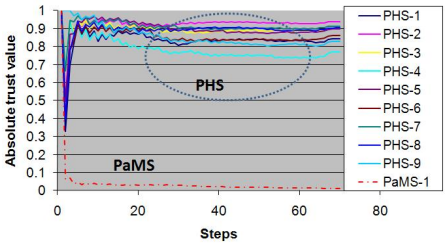

Fig. 11: Oscillating experiment - 90% PHS & 10% PaMS (No duplication)

The last experiment simulates an oscillating scenario where all four types of sources are present: 30% PHS, 20% PuMS, 30% IHS and 20%PaMS. Figure 13 shows how our scheme correctly separates the 4 types of sources as expected.

6 Conclusion

In this paper, we have introduced a new trust framework for rich, complex and uncertain information by leveraging the expressiveness of Bayesian Description Logics. We have demonstrated the robustness of the proposed framework under a variety of scenarios, and shown how duplication of assertions across different sources as well as prior knowledge of the trustworthiness of sources can further enhance it.

Acknowledgements Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this

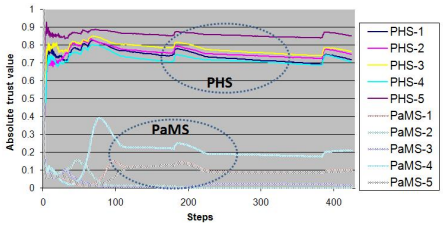

Fig. 12: Oscillating experiment - 50% PHS & 50% PaMS (No duplication)

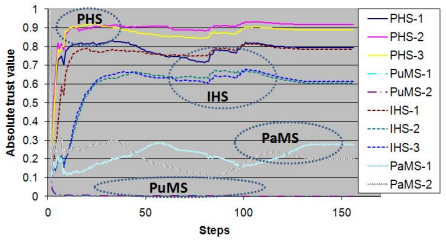

Fig. 13: Oscillating experiment - 30% PHS, 20% PuMS, 30% IHS & 20% PaMS

document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, 2003.

[2] J. Cheng and M. J. Druzdzel. AIS-BN: An Adaptive Importance Sampling Algorithm for Evidential Reasoning in Large Bayesian Networks. In *Journal of AI Research*, 2000.

[3] C. D'Amato, N. Fanizzi, and T. Lukasiewicz. Tractable reasoning with bayesian description logics. In *Scalable Uncertainty Management (SUM08)*, pages 146–159, 2008.

[4] J. Dolby, J. Fan, A. Fokoue, A. Kalyanpur, A. Kershenbaum, L. Ma, J. W. Murdock, K. Srinivas, and C. A. Welty. Scalable cleanup of information extraction data using ontologies. In *ISWC/ASWC*, pages 100–113, 2007.

[5] J. Dolby, A. Fokoue, A. Kalyanpur, A. Kershenbaum, E. Schonberg, K. Srinivas, and L. Ma. Scalable semantic retrieval through summarization and refinement. In *AAAI*, pages 299–304, 2007.

[6] J. Dolby, A. Fokoue, A. Kalyanpur, E. Schonberg, and K. Srinivas. Scalable highly expressive reasoner (sher). *J. Web Sem.*, 7(4):357–361, 2009.

[7] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference and Prediction. In *Springer Series in Statistics*, 2009.

[8] A. Josang and R. Ismail. The beta reputation system. In *15th Conference on Electronic Commerce*, 2002.

[9] A. Kalyanpur. *Debugging and Repair of OWL-DL Ontologies*. PhD thesis, University of Maryland, https://drum.umd.edu/dspace/bitstream/1903/3820/1/umi-umd-3665.pdf, 2006.

[10] S. Kamvar, M. Schlosser, and H. Garcia-Molina. EigenTrust: Reputation management in P2P networks. In *WWW Conference*, 2003.

[11] U. Kuter and J. Golbeck. SUNNY: A New Algorithm for Trust Inference in Social Networks, using Probabilistic Confidence Models. In *AAAI-07*, 2007.

[12] L. Ma, Y. Yang, Z. Qiu, G. Xie, and Y. Pan. Towards a complete owl ontology benchmark. In *ESWC, 2006*, pages 124–139, 2006.

[13] Netflix. Netflix Prize. http://www.netflixprize.com/.

[14] J. B. Schafer, J. Konstan, and J. Riedl. Recommender Systems in E-Commerce. In *ACM Conference on Electronic Commerce*, 1999.

[15] L. Xiong and L. Liu. Supporting reputation based trust in peer-to-peer communities. In *IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 71, 16(7)*, July 2004.