Refining instance coreferencing results using belief propagation

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. The problem of coreference resolution (finding individuals, which describe the same entity but have different URIs) is crucial when dealing with semantic data coming from different sources. Specific features of Semantic Web data (ontological constraints, data sparseness, varying quality of sources) are all significant for coreference resolution and must be exploited. In this paper we present a framework, which uses Dempster-Shafer belief propagation to capture these features and refine coreference resolution results produced by simpler string similarity techniques.

1 Introduction

A major problem, which needs to be solved during information integration, is coreference resolution: finding data instances, which refer to the same real-world entity. This is a non-trivial problem due to many factors: different naming conventions used by the authors of different sources, usage of abbreviations, ambiguous names, data variations over time. This problem for a long time has been studied in the domains of database research and machine learning and multiple solutions have been developed. Although in the Semantic Web community information integration has always been considered as one of the most important research directions, so far the research has been primarily concentrated on resolving schema-level issues. However, semantic data represented in RDF and formatted according to OWL ontologies, has its specific features: instances often have only a few properties, relevant information is distributed between interlinked instances of different classes, an OWL ontology allows expressing a wider range of data restrictions than a standard database schema, different sources may significantly differ in quality. Some of these features make it hard to directly reuse the algorithms developed in the database domain, while others may provide valuable clues, which should be exploited.

The main motivation for our work comes from the enterprise-level knowledge management use case. In this scenario a shared corporate ontology is populated automatically with information extracted from multiple sources: text documents, images, database tables. Although there is no schema alignment required in this scenario, the data-level integration problems listed above are present. In addition to the usual issues related to heterogeneity, the data may also contain noise

caused by incorrect extraction results. Data sparseness often prevents the use of sophisticated machine-learning algorithms and requires simple techniques such as string similarity metrics applied to instance labels. The output of these techniques is not completely reliable. In order to improve coreferencing results we have to utilize the links between data instances, to take into account uncertainty of sources and coreferencing algorithms and to consider logical restrictions defined in the domain ontology. In this paper we describe an approach, which uses the Dempster-Shafer belief propagation in order to achieve this goal.

The rest of the paper is organized as follows: in the section 2 we briefly discuss the most relevant existing approaches. Section 3 provides a short description of the approach and its place in the overall integration architecture. Section 4 summarizes the theoretical background of our belief propagation algorithm. Section 5 describes in detail the usage of belief networks and provides examples. In the section 6 we present the results of our experiments performed with test datasets. Finally, section 7 summarizes our contribution and outlines directions for future work.

2 Related Work

The problem of coreference resolution during data integration has been studied for a long time [1]. In different communities it has been referred to as record linkage [1], object identification [2] and reference reconciliation [3]. A large number of approaches (see [4] for a survey) are based on a vector similarity model initially proposed in [1]: similarity scores are calculated for each pair of instances' attributes and their aggregation is used to make a decision about whether two instances are the same. This procedure is performed for instances of each single class in isolation. Different string similarity techniques have been proposed to measure the similarity between attribute values (e.g., edit distance, Jaro, Jaro-Winkler, Monge-Elkan [5]) and different machine learning algorithms to adjust the parameters of decision models have been developed (e.g., [6], [2]).

Such approaches assume that all attributes, which are relevant for determining the equivalence of two instances, are contained in the attribute vector. This assumption does not hold for scenarios where relevant data is distributed between different instances, which are related to each other. Thus, approaches, which analyze relations between data instances of different classes, have received significant attention in recent years (e.g., [3], [7], [8], [9]). One algorithm focusing on exploiting links between data objects for personal information management was proposed in [3], where the similarities between interlinked entities are propagated using dependency graphs. RelDC [7] proposes an approach based on analyzing entity-relationship graphs to choose the best pair of coreferent entities in case when several options are possible. The authors of these algorithms reported good performance on evaluation datasets and, in particular, significant increase in performance achieved by relation analysis. These algorithms, however, assume data representation similar to relational databases. The OWL language used for formatting Semantic Web data allows more advanced restrictions over data to be

defined (e.g., class disjointness, cardinality restrictions, etc.), which are relevant for the validation of coreference mappings. Given the variable quality of semantic annotations, information about provenance of the data is also valuable: if a mapping between two individuals violates an ontological restriction, it is possible that some piece of data is wrong, rather than a mapping. These factors require development of specific solutions adjusted to the needs of the Semantic Web domain.

The problem of data integration in the Semantic Web context also requires dealing with data sparseness and the distribution of data between several linked individuals. In the Semantic Web community so far the research effort has been primarily concentrated on the schema-level ontology matching problem [10]. Some of the schema-matching systems utilize links between concepts and properties to update initial mappings created using other techniques. One such technique is similarity flooding [11], which uses links in the graph to propagate similarity estimations. It is, however, more suitable to schema matching rather than data integration: it relies, for example, on the assumption that the graph is complete. Ontological restrictions and uncertainty of mappings between concepts are analyzed in [12]. Now, with a constantly increasing amount of RDF data being published and the emergence of the Linked Data initiative, the problem of instance-level integration is also gaining importance. The issue of recognizing coreferent individuals coming from different sources and having different URIs has been raised by different research groups and several architectural solutions were developed, such as OKKAM [13], Sindice [14], RKBExplorer [15]. Sindice [14] relies on inverse functional properties explicitly defined in corresponding ontologies. The authors of OKKAM entity name service [13] have employed Monge-Elkan string similarity for their prototype implementation. Data aggregation for RKBExplorer [15], to our knowledge, was performed using techniques specially developed for the scientific publication domain (e.g., analyzing co-authorship, etc.). The L2R/N2R algorithm recently proposed in [16] and [17] focuses on employing ontological restrictions (in particular, functionality and disjointness) in order to make coreferencing decisions. Their approach is probably the most similar to ours, but emerged as a purely logical inference-based algorithm and treats some aspects in a different way. In particular, data uncertainty is not considered (data statements are treated as correct) and similarity between individuals is aggregated using maximum function, which does not allow capturing cumulative evidence.

In our view, there is still a need for data integration methods adjusted to the needs of the Semantic Web domain. First, as was said, the algorithms developed in the database community do not take into account the specific properties of semantic data. Ontology matching techniques, on the other hand, focus primarily on the schema-matching issues. Our approach tries to analyze together relations between individuals of multiple classes, logical restrictions imposed by ontologies and data uncertainty in order to improve the quality of instance coreferencing.

3 Overview

The algorithm described in the paper represents a module of the knowledge fusion architecture KnoFuss initially developed to integrate semantic annotations produced from different sources using automatic information extraction algorithms. The architecture receives as its input a source knowledge base (KB) containing a set of RDF assertions extracted from a particular source. The system processes this source KB and integrates it into the target KB. KnoFuss aims to solve two main problems: find and merge coreferent individuals and ensure consistency of the integrated KB. The structure of the KnoFuss system and the initial stage of its workflow is described in [18]. This stage involves producing mappings between individuals (interpreted as *owl:sameAs* relations) using a library of coreferencing algorithms. In this paper we focus on the second stage of the fusion workflow where these initially produced mappings are refined using additional factors, which are not considered by attribute-based similarity algorithms but can serve as evidence for revising and refining the results of coreferencing stage. We consider three kinds of such factors:

- *Ontological schema restrictions.* Constraints and restrictions defined by the schema (e.g., functionality relations) may provide both positive and negative evidence. For instance, having two individuals as objects of a functional property with the same subject should reinforce a mapping between these individuals. The reverse also applies: the fact that two potentially identical individuals belong to two disjoint classes should be considered negative evidence.
- *Coreference mappings between other entities.* Even if there is no explicit functionality restriction defined for an ontological property, related individuals still may reduce the ambiguity: the fact that two similar individuals are both related to a third one may reinforce the confidence of the mapping.
- *Provenance data.* Knowledge about the quality of data may be used to assign the confidence to class and property assertions. This is important when we need to judge whether a mapping, which violates the domain ontology, is wrong or the conflict is caused by a wrong data statement. Knowledge about the "cleanness" of a source (e.g., whether duplicates occur in a given source) provides additional evidence about potential mappings.

Most information, which we have to deal with in the fusion scenario, is uncertain. Mappings are created by attribute-based matching algorithms, which do not provide 100% precision. Class and property assertions may come from unreliable sources or be extracted incorrectly. Various ontological relations provide different impact as evidence for mappings: if two similar *foaf:Person* individuals are both connected to a *sweto:Publication* individual via a *sweto:author* relation, it is a much stronger evidence for identity mapping than if they were related to a *tap:Country* individual *#USA* via a *#citizenOf* relation. In order to manage uncertainty adequately, the framework needs to have well-defined rules for reasoning about the confidence of both data statements and coreference mappings, combining multiple uncertain pieces of evidence and propagating beliefs. This

can be achieved by employing an uncertainty representation formalism. Our architecture utilizes the Dempster-Shafer theory of evidence [19], which generalizes the Bayesian probability theory. We proposed the initial version of the algorithm as a means to resolve ABox inconsistencies in knowledge bases [20]. The next section briefly summarizes our previous work.

4 Dempster-Shafer belief propagation

Our algorithm uses the Dempster-Shafer theory of evidence as a theoretical basis for uncertainty representation. The reason for this choice (in comparison with the more commonly used Bayesian probability) is its ability to represent a degree of ignorance in addition to the positive and negative belief [20]. This feature is valuable when we deal with the output of coreferencing algorithms. By default, these algorithms can only produce positive evidence: a positive result produced by a low-quality algorithm (e.g., with a precision 0.2) can only be considered as insufficient evidence rather than negative evidence. The uncertainty of a statement is described by belief masses, which can be assigned to sets of possible values. In our case each statement is described by three mass assignments: (i) belief that the statement is true $m(1)$, (ii) belief that the statement is false $m(0)$ and (iii) unassigned belief $m(\{0;1\})$, specifying the degree of our ignorance about the truth of the statement. Given that $\sum_i m_i = 1$, these assignments are usually represented using two values: *belief* (or *support*) ($m(1)$) and *plausibility* ($m(1) + m(\{0;1\})$). Bayesian probability is a special case, which postulates that no ignorance is allowed and $m(1) + m(0) = 1$. Our workflow for processing a conflict involves three steps:

- *Constructing a belief propagation network.* At this stage an OWL subontology is translated into a belief network.
- *Assigning mass distributions.* At this stage the belief mass distribution functions are assigned to nodes.
- *Belief propagation.* At this stage the uncertainties are propagated through the network and the confidence degrees of statements are updated.

As the theoretical base for belief propagation we used valuation networks as described in [21]. Valuation networks contain two kinds of nodes: *variable nodes*, which represent the uncertain assertions, and *valuation nodes*, which represent the belief propagation rules (converted from TBox axioms). We use a set of rules to convert an OWL subontology into a corresponding valuation network (this procedure is described in more detail in [20]). Then, initial beliefs are propagated through the network and updated values are produced according to the standard axioms for valuation networks formulated in [21]. 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 $C \subseteq D$. It extracts the belief distribution for a single variable or subset of variables from a complete distribution over a larger set.

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

For instance, if we have the function m defined on the 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 the domain $\{y\}$, we will get $m(0) = 0.2 + 0.3 = 0.5$ and $m(1) = 0.35 + 0.15 = 0.5$. Combination calculates an aggregated belief distribution based on several pieces of evidence. The combination operator is represented by Dempster's rule of combination [19]:

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

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

1. Each node sends a message to its inward neighbour (towards the arbitrary selected root of the tree). If $\mu^{A \to B}$ is a message from a node A to a node B, $N(A)$ is a set of neigbours 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: $\mu^{A \to B} = (\otimes\{\mu^{X \to A} | X \in (N(A) - \{B\}) \otimes m_A\})^{\downarrow A \cap B}$
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.

Loops must be eliminated by replacing all nodes in a loop with a single node combining their belief functions. The initial version of the algorithm deals with inconsistency resolution and does not consider coreference mappings and identity uncertainty. In the following section we describe how we further develop the same theoretical approach in order to reason about coreference mappings.

5 Refining coreference mappings

The algorithm receives as its input a set of candidate mappings between individuals of source and target KBs. In order to perform belief propagation, these mappings along with relevant parts from both knowledge bases must be translated into valuation networks. Building a large network from complete knowledge bases is both computationally expensive and unnecessary, as not all triples are valuable for analysis. We select only relevant triples, which include (i) values of object properties, which can be used to propagate belief between two *owl:sameAs* mappings (functional, inverse functional and "influential" as described in 5.2) and (ii) class and property assertions, which produce conflicts. Conflicts are detected by selecting all statements in the neighborhood of potentially mapped individuals and checking their consistency with respect to the domain ontology (we use the Pellet OWL reasoner with the explanation service). If the reasoner found an inconsistency, all statements which contribute to it are considered relevant. Then, belief networks are constructed by applying the rules defined in ([20] and the extended set described in subsection 5.1) and initial beliefs are assigned to variable nodes. For each *owl:sameAs* variable node the belief is determined according to the precision of the corresponding coreferencing algorithm, which

produced it. Each algorithm could produce two kinds of mappings: "probably correct" exceeding the optimal similarity threshold for the algorithm (the one, which maximized the algorithm's F-measure performance), and "possibly correct" with similarities below the optimal threshold, but achieving at least 0.1 precision. Each variable node representing a class or property assertion receives its initial belief based on its attached provenance data: the reliability of its source and/or its extraction algorithm. After that the beliefs are updated using belief propagation and for each mapping the decision about its acceptance is taken.

The most significant part of the algorithm is network construction. At this stage we exploit the factors listed in the section 3. In the following subsections we describe how it is done in more detail.

5.1 Exploiting ontological schema

Logical axioms defined by the schema may have both positive and negative influence on mappings. First, some OWL axioms impose restrictions on the data. If creating an *owl:sameAs* relation between two individuals violates a restriction, the confidence of the mapping should be reduced. Second, object properties defined as *owl:FunctionalProperty* and *owl:InverseFunctionalProperty* allow us to infer equivalence between individuals. The initial set of rules and possible network nodes we proposed in [20] does not capture instance equivalence and thus is insufficient for reasoning about coreference relations. Therefore, in this section we present a novel set of additional rules (Table 1), which allow us to reason about coreference mappings. Table 2 lists the additional belief assignment functions for corresponding valuation nodes.

Table 1. Belief network construction rules

N	Axiom	Pre-conditions	Nodes to create	Links to create
1	*sameAs*	$I_1 = I_2$	$N_1 : I_1 = I_2$ (variable)	
2	*differentFrom*	$I_1 \neq I_2$	$N_1 : I_1 \neq I_2$ (variable)	
3	*sameAs*	$N_1 : I_1 = I_2$ (variable), $N_2 : R(I_2, I_3)$	$N_3 : I_1 = I_2$ (valuation), $N_4 : R(I_1, I_3)$	(N_1,N_3),(N_2,N_3), (N_3,N_4)
4	*differentFrom*	$N_1 : I_1 = I_2$ (variable), $N_2 : I_1 \neq I_2$ (variable)	$N_3 : I_1 \neq I_2$ (valuation)	(N_1,N_3),(N_2,N_3)
5	*Functional Property*	$\top \sqsubseteq \leq 1R$, $N_1 : R(I_3, I_1)$, $N_2 : R(I_3, I_2)$, $N_3 : I_1 = I_2$	$N_4 : \top \sqsubseteq \leq 1R$	(N_1,N_4),(N_2,N_4), (N_3,N_4)
6	*InverseFunctional Property*	$\top \sqsubseteq \leq 1R^-$, $N_1 : R(I_1, I_3)$, $N_2 : R(I_2, I_3)$, $N_3 : I_1 = I_2$	$N_4 : \top \sqsubseteq \leq 1R^-$	(N_1,N_4),(N_2,N_4), (N_3,N_4)

The axioms *owl:sameAs* and *owl:differentFrom* (Table 1, rows 1-4) lead to the creation of both variable and valuation nodes. This is because each one represents both a schema-level rule, which allows new statements to be inferred, and a

data-level assertion, which has its own confidence (e.g., produced by a matching algorithm). *owl:FunctionalProperty* and *owl:InverseFunctionalProperty* (rows 5-6) can only be linked to already existing *owl:sameAs* nodes, so that they can only increase similarity between individuals, which were already considered potentially equal. Otherwise the functionality node is treated as in [20]: as a strict constraint violated by two property assertion statements. This is done to prevent the propagation of incorrect mappings.

Table 2. Belief distribution functions for valuation nodes

N	Axiom	Node type	Variables	Mass distribution
1	*sameAs*	$I_1 = I_2$	$I_1 = I_2$, $R(I_1, I_3)$, $R(I_2, I_3)$	m({0;0;0}, {0;0;1}, {0;1;0}, {0;1;1}, {1;0;0}, {1;1;1})=1
2	*differentFrom*	$I_1 \neq I_2$	$I_1 = I_2$, $I_1 \neq I_2$	m({0;1},{1;0})=1
3	*Functional Property*	$\top \sqsubseteq \leq 1R$	$R(I_3, I_1)$, $R(I_3, I_2)$, $I_1 = I_2$	m({0;0;0}, {0;0;1}, {0;1;0}, {1;0;0}, {1;1;1})=1
4	*Inverse Functional Property*	$\top \sqsubseteq \leq 1R^-$	$R(I_1, I_3)$, $R(I_2, I_3)$, $I_1 = I_2$	m({0;0;0}, {0;0;1}, {0;1;0}, {1;0;0}, {1;1;1})=1

To illustrate the work of the algorithm we will use an example from our experiments with datasets from the citations domain (see Section 6). One such dataset (DBLP) contains an individual *Ind1* describing the following paper:

> D. Corsar, D. H. Sleeman. Reusing JessTab Rules in Protege. Knowledge-Based Systems 19(5). (2006) 291-297.

Another one (EPrints) also contained a paper *Ind2* with the same title:

> Corsar, Mr. David and Sleeman, Prof. Derek. Reusing JessTab Rules in Protege. In Proceedings The Twenty-fifth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (2005), pages pp. 7-20, Cambridge, UK.

This illustrates a common case when the same group of researchers first publishes their research results at a conference and then submits the extended and revised paper to a journal. An attribute-based coreferencing algorithm (Jaro-Winkler similarity applied to the title), which had a good overall performance (precision about 0.92 and F-measure about 0.94), incorrectly considered these two papers identical. However, a mapping between these individuals violated two restrictions: the individual belonged to two disjoint classes simultaneously and had two different values for the functional property *year*. The inconsistencies were detected by the algorithm, which produced two sets of relevant statements: {*owl:sameAs(Ind1, Ind2)*; *Article(Ind1)*; *Article_in_Proceedings(Ind2)*; *owl:disjointWith(Article, Article_in_Proceedings)*} and {*owl:sameAs(Ind1, Ind2)*;

year(Ind1, 2006); *year(Ind2, 2005)*; *owl: Functional Property(year)*}. Since these sets share a common statement (*sameAs* link), they are translated into a single valuation network (Fig. 1). Although in our example the initial support of the

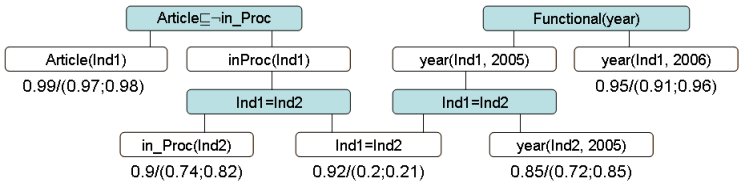

Fig. 1. Example of a belief network constructed during the experimental testing. The numbers show the support before propagation and support and plausibility after propagation for variable nodes (white). Leaf variable nodes are given in the KB while non-leaf ones are inferred using axioms corresponding to valuation nodes (blue).

mapping was higher than the support of both statements related to Ind2 (*Article_in_Proceedings(Ind2)* and *year(Ind2, 2005)*), after belief propagation the incorrect *owl:sameAs* mapping was properly recognized and received the lowest plausibility (0.21 - obtained as $m(1) + m(0;1) = 0.20 + 0.01$).

5.2 Influence of context mappings

Belief propagation for properties explicitly defined as functional is a trivial case. However, properties which allow multiple values are also valuable as a means to narrow the context of matched individuals and increase similarity between them. We have to estimate the impact of the relation and model this in the network. As shown in Table 2 (row 1), by default the valuation node for the *owl:sameAs* relation is defined in such a way that the belief in $I_1 = I_2$ is completely independent from a strong belief for both $R(I_3, I_1)$ and $R(I_3, I_2)$. The functionality axiom represents an opposite scenario: having a belief 1.0 for both $R(I_3, I_1)$ and $R(I_3, I_2)$ implies the belief 1.0 for $I_1 = I_2$. The actual strength of influence for a property may lay between these extreme cases. In order to utilize such links the network construction algorithm receives for each relevant property a vector $< n_1, n_2 >$, where n_1, n_2 determine the impact of the link in direct (subject to object) and reverse (object to subject) directions. The impact in two directions may be different: having two people as first authors of the same paper strongly implies people's equivalence, while having the same person as the first author of two papers with the similar title does not increase the probability of two papers being the same. The $owl : sameAs$ valuation node, combining variables $I_1 = I_2$, $R(I_2, I_3)$, $R(I_1, I_3)$ will receive two belief assignments instead of one: m({0;0;0}, {0;0;1}, {0;1;0}, {1;0;0}, {1;1;1})=n_1 and m({0;0;0}, {0;0;1}, {0;1;0}, {0;1;1}, {1;0;0}, {1;1;1})=$1 - n_1$. One possible way to determine coefficients $< n_1, n_2 >$ is to learn them from training data, as we did in our experiments, or to assign

them based on expert estimations or the number of statements per individual as in [11].

Also some relevant relations may be implicit and not defined in the ontology. For instance, the same group of people may be involved in different projects. If the link between a project and a person is specified using a property *akt:has-project-member*, when two knowledge bases describing two non-overlapping sets of projects are combined, the relations between people cannot be utilized. In order to capture these implicit relations we can add artificial properties, which connect individuals belonging to the same sets, into the ontology. Co-authorship analysis, commonly used in the citation matching domain, is a special case of this scenario (Fig. 2a).

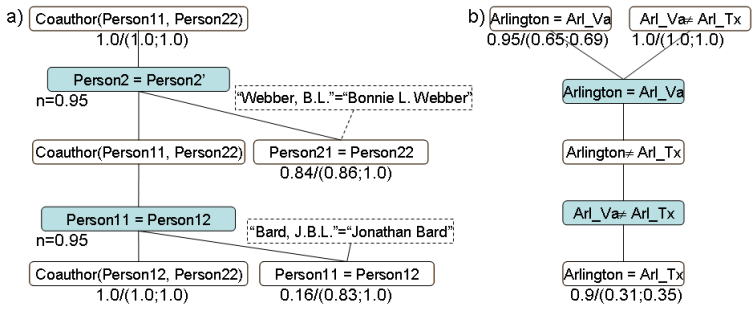

Fig. 2. Examples of belief networks illustrating (a) the usage of artificial set membership relations and (b) processing competing mappings knowing that a source does not contain duplicates. The numbers show the belief before propagation and belief and plausibility after propagation.

5.3 Provenance data

The estimated reliability of a source is directly used at the starting stage when initial beliefs are assigned to variable nodes representing class and property assertions. Thus, if a violation of a functional restriction is caused by a property assertion with a low belief, its impact will be insufficient to break the *owl:sameAs* link. Another important factor is the knowledge about duplicate individuals in a knowledge base. For instance, one knowledge base (AGROVOC) contains an individual "fao:arlington". If we match this against the UTexas geographical ontology, which contains two individuals "arlingtonVa" and "arlingtonTx", then although the similarity of one pair is slightly greater than another one, both values are above the threshold and both these individuals can potentially be matched to the first individual. However, knowing that this particular knowledge base does not contain duplicates, allows us to add a corresponding *owl:differentFrom* variable node into the network (Fig.2b). Updating beliefs allows us to reject one of the two competing options.

6 Evaluation

In order to test the system we used the following datasets from the domain of scientific publications:

1. AKT EPrints archive[1]. This dataset contains information about papers produced within the AKT research project.
2. Rexa dataset[2]. The dataset extracted from the Rexa search server, which was constructed in the University of Massachusetts using automatic IE algorithms.
3. SWETO DBLP dataset[3]. This is a publicly available dataset listing publications from the computer science domain.
4. Cora(I) dataset[4]. A citation dataset used for machine learning tests.
5. Cora(II) dataset. Another version of the Cora dataset used in [3].

AKT, Rexa and SWETO-DBLP datasets were previously used by the authors in [18]. The SWETO-DBLP dataset was originally represented in RDF. AKT and Rexa datasets were extracted from the HTML sources using specially constructed wrappers and structured according to the SWETO-DBLP ontology (Fig. 3). The Cora(I) dataset was created in the University of Massachusetts for the purpose of testing machine-learning clustering algorithms. It contains 1295 references and is intentionally made noisy: e.g., the gold standard contains some obviously wrong mappings[5]. We translated this dataset into RDF using the SWETO-DBLP ontology. The authors of Cora(II)[3] translated the data from Cora(I) into RDF according to their own ontology and cleaned the gold standard by removing some spurious mappings, so the results achieved on Cora(I) and Cora(II) are not comparable. Data and gold standards mappings in Cora(II) are significantly cleaner than in Cora(I). Also in Cora(II) all *Person* individuals were initially considered different while in Cora(I) individuals with exactly the same name were assigned the same URI, which led to a significant difference in the number of individuals (305 vs 3521) and, consequently, in performance measurements. In our tests we tried to merge each pair of datasets 1-3 and to find duplicates in the Cora datasets. To the SWETO ontology we added the restrictions specifying that (i) classes *Article* and *Article_in_Proceedings* are disjoint, (ii) datatype property *year* describing the publication year is functional and (iii) object property *author* connecting a publication with a set of authors is functional. Given that both Cora datasets did not distinguish between journal and conference articles, instead we used venues as individuals and added functionality relations for them. Also

[1] http://eprints.aktors.org/
[2] http://www.rexa.info/
[3] http://lsdis.cs.uga.edu/projects/semdis/swetodblp/august2007/opus_august2007.rdf
[4] http://www.cs.utexas.edu/users/ml/riddle/data/cora.tar.gz
[5] For instance, two papers by N. Cesa-Bianchi et al. "How to use expert advice. 25th ACM Symposium on the theory of computing (1993) 382-391" and "On-line prediction and conversion strategies. Eurocolt'93 (1993) 205-216" were considered the same in Cora(I).

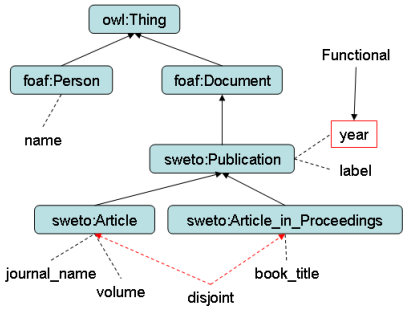

Fig. 3. Class hierarchy in the SWETO-DBLP ontology

the Cora(II) ontology described pages as two integer properties *pageFrom* and *pageTo*, which allowed us to add a functionality restriction on them as well.

For attribute-based coreferencing we used string similarity metrics applied to a paper title or person's name. In particular, we used Jaro-Winkler and Monge-Elkan metrics applied to the whole strings or tokenized strings (L2 Jaro-Winkler). L2 Jaro-Winkler is a mixture of string similarity and set similarity measures: it tokenizes both compared values, then each pair of tokens is compared using the standard Jaro-Winkler algorithm and the maximal total score is selected. Initial belief mass distribution for each *owl:sameAs* relation was assigned according to the precision of the algorithm, which produced it. Initial belief assignments for the class and property assertions are shown in the Table 3. We assigned the values based on our knowledge about how each dataset

Table 3. Initial belief mass assignment

Dataset	Class assertions	Datatype assertions
DBLP	0.99	0.95
Rexa	0.95	0.81 (<2 citations) 0.855 (>2 citations)
EPrints	0.9	0.85
Cora(I & II)	N/A	0.6

was produced and manual reviewing of the datasets. We did not further classify publications in Cora datasets into journal and conference articles, so class assertions were not relevant. Knowing that the data in Cora datasets was noisy, we assigned beliefs in such a way that disagreement on a single property value was not sufficient to break the mapping. We measured the quality of coreference before and after belief propagation. The results of the tests are shown in the Table 4. As expected, in almost all cases the refinement procedure led to an improvement in overall performance expressed by the F1-measure. For *sweto:Publication* instances (rows 1, 2, 4, 6, 7, 8) the recall has decreased: the

Table 4. Test results

Dataset	No	Matching algorithm	*sweto:Publication*					
			Before			After		
			Precision	Recall	F1	Precision	Recall	F1
EPrints/Rexa	1	Jaro-Winkler	0.950	0.833	0.887	0.969	0.832	0.895
	2	L2 Jaro-Winkler	0.879	0.956	0.916	0.923	0.956	0.939
EPrints/DBLP	3	Jaro-Winkler	0.922	0.952	0.937	0.992	0.952	0.971
	4	L2 Jaro-Winkler	0.389	0.984	0.558	0.838	0.983	0.905
Rexa/DBLP	5	Jaro-Winkler	0.899	0.933	0.916	0.944	0.932	0.938
	6	L2 Jaro-Winkler	0.546	0.982	0.702	0.823	0.981	0.895
Cora(I)	7	Monge-Elkan	0.735	0.931	0.821	0.939	0.836	0.884
Cora(II)	8	Monge-Elkan	0.698	0.986	0.817	0.958	0.956	0.957
			foaf:Person					
EPrints/Rexa	9	L2 Jaro-Winkler	0.738	0.888	0.806	0.788	0.935	0.855
EPrints/DBLP	10	L2 Jaro-Winkler	0.532	0.746	0.621	0.583	0.921	0.714
Rexa/DBLP	11	Jaro-Winkler	0.965	0.755	0.846	0.968	0.876	0.920
Cora(I)	12	L2 Jaro-Winkler	0.983	0.879	0.928	0.981	0.895	0.936
Cora(II)	13	L2 Jaro-Winkler	0.999	0.994	0.997	0.999	0.994	0.997

algorithm incorrectly resolved some inconsistencies, which in fact occurred due to wrong data statements. The decrease was slight for AKT/Rexa/DBLP datasets and more significant for Cora where the degree of noise was higher. However, in all cases this decrease was more than compensated by the increase in precision. For *foaf:Person* individuals the effect of belief propagation primarily influenced recall: links between instances reinforced the potential mappings, which would otherwise be rejected. Because Cora(II) was better formatted than Cora(I) there were very few "dubious" mappings produced during initial coreferencing and belief propagation was not able to catch them.

Considering the F1 measure obtained for Cora(I) publication (row 7) in comparison with the state-of-the art algorithms from the database and machine learning communities, we found that it is higher than those reported in [22] (0.867), [9] (0.87), but lower than in [2] (0.93)[6]. As was said before, in order to minimize the number of attributes processed by basic coreferencing methods, in our tests we only used the title comparison for determining candidate individuals. This was the main factor, which reduced the performance: e.g., the algorithm used in [2] achieved similar F-measure (0.88) on the test set when trained only on the *title*, *year* and *venue* attributes. For Cora(II) the F-measure was similar to that reported for [3]: slightly higher for publications (0.957 vs 0.954) while slightly lower for people (0.997 vs 0.999). The difference is due to the fact that the authors of [3] used better similarity measures (reported F-measure for publi-

[6] Note that the authors of [8] and [7], and [16] used different versions of the Cora dataset where, in particular, more mappings were removed from the gold standard so that the dataset contained 132 clusters [8] rather than 125 in Cora(II), and papers with the same title and year were considered identical [16]. This does not allow direct comparison of reported performance with our algorithms.

cations 0.948 without exploiting links) while exploiting data uncertainty by our approach increased recall (e.g., having different years for papers was not enough to break the mapping if there was an agreement for the venue name and pages).

7 Conclusion and future work

In the paper we have presented an approach which uses Dempster-Shafer belief propagation in order to improve the quality of data integration, in particular coreferencing of individuals. We consider this extension and application of the Dempster-Shafer belief propagation mechanism as the main contribution of this paper. Our initial experiments performed with test datasets have shown an improvement in the output quality of basic string similarity algorithms. However, there are still issues which have to be resolved in the future work.

First, the Dempster-Shafer belief propagation mechanism is sensitive to the initial belief distribution, which may be an issue if initial belief does not adequately reflect the actual data, e.g., if the estimated precision of a coreferencing algorithm was measured using a test set with a different distribution of data. Second, at the moment the algorithm assumes that the data to be merged is formatted according to the same ontology. In order to be employed on a Web scale, the ability to work in a multi-ontology environment is necessary. In particular, the output of ontology matching algorithms must be considered. Another important feature would be automatic discovery of ontological restrictions by retrieving other ontologies covering the same domain (e.g., using Watson[7] or Swoogle[8] engines) and analyzing them.

8 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. The authors would like to thank Steffen Rendle and Karen Tso for providing their object identification tool [2], Luna Dong for providing the Cora(II) dataset and Fatiha Saïs for providing materials about L2R/N2R algorithm [17].

References

1. Fellegi, I.P., Sunter, A.B.: A theory for record linkage. Journal of American Statistical Association **64**(328) (1969) 1183–1210
2. Rendle, S., Schmidt-Thieme, L.: Object identification with constraints. In: 6th IEEE International Conference on Data Mining (ICDM). (2006)
3. Dong, X., Halevy, A., Madhavan, J.: Reference reconciliation in complex information spaces. In: SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM (2005) 85–96

[7] http://watson.kmi.open.ac.uk/WatsonWUI/
[8] http://swoogle.umbc.edu/

4. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. IEEE Transactions on Knowledge and Data Engineering **19**(1) (2007) 1–16
5. Winkler, W.E.: Overview of record linkage and current research directions. Technical Report RRS2006/02, US Bureau of the Census, Washington, DC 20233 (2006)
6. Sarawagi, S., Bhamidipaty, A.: Interactive deduplication using active learning. In: 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002), Edmonton, Alberta, Canada, ACM (2002)
7. Chen, Z., Kalashnikov, D.V., Mehrotra, S.: Adaptive graphical approach to entity resolution. In: ACM IEEE Joint Conference on Digital Libraries 2007 (ACM IEEE JCDL 2007), Vancouver, British Columbia, Canada (2007) 204–213
8. Singla, P., Domingos, P.: Object identification with attribute-mediated dependences. In: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PAKDD-2005), Porto, Portugal (2005) 297–308
9. Parag, Domingos, P.: Multi-relational record linkage. In: KDD Workshop on Multi-Relational Data Mining, Seattle, CA, USA, ACM Press (2004) 31–48
10. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (2007)
11. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: A versatile graph matching algorithm. In: 18th International Conference on Data Engineering (ICDE), San Jose (CA US) (2002) 117–128
12. Castano, S., Ferrara, A., Lorusso, D., Näth, T.H., Möller, R.: Mapping validation by probabilistic reasoning. In: 5th Annual European Semantic Web Conference (ESWC 2008), Tenerife, Spain (2008) 170–184
13. Bouquet, P., Stoermer, H., Bazzanella, B.: An Entity Name System (ENS) for the Semantic Web. In: 5th Annual European Semantic Web Conference (ESWC 2008). (2008) 258–272
14. Tummarello, G., Delbru, R., Oren, E.: Sindice.com: Weaving the open linked data. In: 6th International Semantic Web Conference (ISWC/ASWC 2007). (2007) 552–565
15. Glaser, H., Millard, I., Jaffri, A.: RKBExplorer.com: A knowledge driven infrastructure for linked data providers. In: 5th Annual European Semantic Web Conference (ESWC 2008). (2008)
16. Saïs, F., Pernelle, N., Rousset, M.C.: L2R: a logical method for reference reconciliation. In: 22nd AAAI Conference on Artificial Intelligence (AAAI-07), Vancouver, BC, Canada, AAAI Press (2007) 329–334
17. Saïs, F., Pernelle, N., Rousset, M.C.: Combining a logical and a numerical method for data reconciliation. Journal of Data Semantics **12** (2008)
18. Nikolov, A., Uren, V., Motta, E., de Roeck, A.: Handling instance coreferencing in the KnoFuss architecture. In: Workshop on Identity and Reference on the Semantic Web, ESWC 2008, Tenerife, Spain (2008)
19. Shafer, G.: A mathematical theory of evidence. Princeton University Press (1976)
20. Nikolov, A., Uren, V., Motta, E., de Roeck, A.: Using the Dempster-Shafer theory of evidence to resolve ABox inconsistencies. In: Workshop on Uncertainty Reasoning for the Semantic Web, ISWC 2007, Busan, Korea (2007)
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. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003), Washington DC (2003) 39–48