10th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2014)

Proceedings

edited by

Fernando Bobillo
Rommel Carvalho
Davide Ceolin
Paulo C. G. da Costa
Claudia d'Amato
Nicola Fanizzi
Kathryn B. Laskey
Kenneth J. Laskey
Thomas Lukasiewicz
Trevor Martin
Matthias Nickles
Michael Pool
Tom De Nies
Olaf Hartig
Paul Groth
Stephen Marsh

Riva del Garda, Italy, October 19, 2014

collocated with
the 13th International Semantic Web Conference
(ISWC 2014)

Foreword

This volume contains the papers presented at the 10th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2014), held as a part of the 13th International Semantic Web Conference (ISWC 2014) at Riva del Garda, Italy, October 19, 2014. 4 technical papers and 1 position paper were accepted at URSW 2014. Furthermore, there was a special session on Methods for Establishing Trust of (Open) Data (METHOD 2014) including 1 technical paper and 2 position papers. All the papers were selected in a rigorous reviewing process, where each paper was reviewed by three program committee members.

The International Semantic Web Conference is a major international forum for presenting visionary research on all aspects of the Semantic Web. The International Workshop on Uncertainty Reasoning for the Semantic Web provides an opportunity for collaboration and cross-fertilization between the uncertainty reasoning community and the Semantic Web community.

We wish to thank all authors who submitted papers and all workshops participants for fruitful discussions. We would like to thank the program committee members for their timely expertise in carefully reviewing the submissions.

September 2014

Fernando Bobillo
Rommel Carvalho
Davide Ceolin
Paulo C. G. da Costa
Claudia d'Amato
Nicola Fanizzi
Kathryn B. Laskey
Kenneth J. Laskey
Thomas Lukasiewicz
Trevor Martin
Matthias Nickles
Michael Pool
Tom De Nies
Olaf Hartig
Paul Groth
Stephen Marsh

URSW 2014
Workshop Organization

Organizing Committee

Fernando Bobillo (University of Zaragoza, Spain)
Rommel Carvalho (Universidade de Brasilia, Brazil)
Paulo C. G. da Costa (George Mason University, USA)
Davide Ceolin (VU University Amsterdam, The Netherlands)
Claudia d'Amato (University of Bari, Italy)
Nicola Fanizzi (University of Bari, Italy)
Kathryn B. Laskey (George Mason University, USA)
Kenneth J. Laskey (MITRE Corporation, USA)
Thomas Lukasiewicz (University of Oxford, UK)
Trevor Martin (University of Bristol, UK)
Matthias Nickles (National University of Ireland, Ireland)
Michael Pool (Goldman Sachs, USA)

Program Committee

Fernando Bobillo (University of Zaragoza, Spain)
Rommel Carvalho (Universidade de Brasilia, Brazil)
Davide Ceolin (VU University Amsterdam, The Netherlands)
Paulo C. G. da Costa (George Mason University, USA)
Fabio Gagliardi Cozman (Universidade de São Paulo, Brazil)
Claudia d'Amato (University of Bari, Italy)
Nicola Fanizzi (University of Bari, Italy)
Marcelo Ladeira (Universidade de Brasília, Brazil)
Kathryn B. Laskey (George Mason University, USA)
Kenneth J. Laskey (MITRE Corporation, USA)
Thomas Lukasiewicz (University of Oxford, UK)
Trevor Martin (University of Bristol, UK)
Alessandra Mileo (DERI Galway, Ireland)
Matthias Nickles (National University of Ireland, Ireland)
Jeff Z. Pan (University of Aberdeen, UK)
Rafael Peñaloza (TU Dresden, Germany)
Michael Pool (Goldman Sachs, USA)
Livia Predoiu (University of Mannheim, Germany)
Guilin Qi (Southeast University, China)
David Robertson (University of Edinburgh, UK)
Thomas Scharrenbach (University of Zurich, Switzerland)

Giorgos Stoilos (National Technical University of Athens, Greece)
Umberto Straccia (ISTI-CNR, Italy)
Matthias Thimm (Universität Koblenz-Landau, Germany)
Peter Vojtáš (Charles University Prague, Czech Republic)

METHOD 2014

Organizing Committee

Davide Ceolin (VU University Amsterdam, The Netherlands)
Tom De Nies (Ghent University, Belgium)
Paul Groth (VU University Amsterdam, The Netherlands)
Olaf Hartig (University of Waterloo, Canada)
Stephen Marsh (University of Ontario Institute of Technology, Canada)

Program Committee

Edzard Hfig (Beuth University of Applied Sciences Berlin, Germany)
Rino Falcone (Institute of Cognitive Sciences and Technologies, Italy)
Jean-Francois Lalande (INSA Centre Val de Loire, France)
Zhendong Ma (Austrian Institute of Technology, Austria)
Uwe Nestmann (Technische Universität Berlin, Germany)
Florian Skopik (Austrian Institute of Technology, Austria)
Gabriele Lenzini (University of Luxembourg, Luxembourg)
Erik Mannens (Ghent University, Belgium)
Matthias Flgge (Fraunhofer FOKUS, Germany)
Trung Dong Huynh (University of Southampton, UK)
Paolo Missier (Newcastle University, UK)
Khalid Belhajjame (Paris-Dauphine University, France)
James Cheney (University of Edinburgh, UK)
Christian Bizer (Mannheim University Library, Germany)
Yolanda Gil (University of Southern California, USA)
Daniel Garijo (Universidad Politécnica de Madrid, Spain)
Tim Lebo (Rensselaer Polytechnic Institute, USA)
Simon Miles (Kings College of London, UK)
Andreas Schreiber (German Aerospace Center, Cologne, Germany)
Kieron O'Hara (University of Southampton, UK)
Tim Storer (University of Glasgow, Scotland)
Babak Esfandiari (Carleton University, Canada)
Christian Damsgaard Jensen (Technical University of Denmark, Denmark)
Khalil El-Khatib (University of Ontario Institute of Technology, Canada)
Kai Eckert (Mannheim University Library, Germany)

Table of Contents

URSW 2014 Technical Papers

URSW 2014 Position Papers

Special Session: Methods for Establishing Trust of (Open) Data

Overview

Research Papers

Short Papers

A Probabilistic OWL Reasoner for Intelligent Environments

David Ausín[1], Diego López-de-Ipiña[1], and Federico Castanedo[2]

[1] Deusto Institute of Technology, DeustoTech. University of Deusto, Avda. de las Universidades, 24, 48007 Bilbao, Spain. {david.ausin, dipina}@deusto.es
[2] Wise Athena. 71 Stevenson Street, San Francisco, USA.
fcastanedo@wiseathena.com *

Abstract. OWL ontologies have gained great popularity as a context modelling tool for intelligent environments due to their expressivity. However, they present some disadvantages when it is necessary to deal with uncertainty, which is common in our daily life and affects the decisions that we take. To overcome this drawback, we have developed a novel framework to compute fact probabilities from the axioms in an OWL ontology. This proposal comprises the definition and description of our probabilistic ontology. Our probabilistic ontology extends OWL 2 DL with a new layer to model uncertainty. With this work we aim to overcome OWL limitations to reason with uncertainty, developing a novel framework that can be used in intelligent environments.

Keywords: OWL, Bayesian networks, probability, probabilistic ontology

1 Introduction

In Ambient Intelligence applications, context can be defined as any data which can be employed to describe the state of an entity (a user, a relevant object, the location, etc.) [6]. How this information is modelled and reasoned over time is a key component of an intelligent environment in order to assist users in their daily activities or execute the corresponding actions. An intelligent environment is any space in which daily activities are enhanced by computation [4].

One of the most popular techniques for context modelling is OWL ontologies [18]. They have been employed in several Ambient Intelligence projects such as SOUPA[3], CONON[20] or CoDAMoS [15], to name a few.

OWL is the common way to encode description logics in real world. However, when the domain information contains uncertainty, the employment of OWL ontologies is less suitable [11]. The need to handle uncertainty has created a growing interest in the development of solutions to deal with it.

As in other domains, uncertainty is also present in Ambient Intelligence [16] and affects to the decision making process. This task requires context information

* This work is supported by the Spanish MICINN project FRASEWARE (TIN2013-47152-C3-3-R)

in order to respond to the users' needs. Data in Ambient Intelligence applications are provided by several sensors and services in real time. Unfortunately, these sensors can fail, run out of battery or be forgotten by the user, in the case of wearable devices. On the other hand, the services can also be inaccessible due to network connectivity problems or technical difficulties on the remote server. Nonetheless, that unavailable information may be essential to answer correctly user's requirements.

For this reason, we present a novel approach to deal with uncertainty in intelligent environments. This work proposes a method to model uncertainty, that combines OWL ontologies with Bayesian networks. The rest of this article is organized as follows. The next section describes the problem that we address. Section 3 explains the semantics and syntax of our proposal. Section 4 gives an exemplary use case where our proposal is applied and describes how to model it. Finally, section 5 summarizes this work and addresses the future work.

2 Description of the Problem

In intelligent environments, the lack of information causes incomplete context information and it may be produced by several causes:

- Sensors that have run out of batteries. Several sensors, such as wearable devices, depend on batteries to work.
- Network problems. Sensors, actuators and computers involved in the environment sensing and monitoring are connected to local networks that can suffer network failures. In these cases, the context information may be lost, although the sensors, actuators and computers are working properly.
- Remote services' failures. Some systems rely on remote services to provide a functionality or to gather context information.
- A system device stops working. Computer, sensors and actuators can suffer software and hardware failures that hamper their proper operation.

When one of these issues occurs, the OWL reasoner will infer conclusions that are insufficient to attend the user's needs. Besides, taking into account that factors can improve several tasks carried in intelligent environments, such as ontology-based activity recognition. For instance, *WatchingTVActivity* can be defined as an activity performed by a *Person* who is watching the television in a room:

$$\begin{aligned} WatchingTVActivity \equiv \exists isDoneBy.(Person \sqcap \exists isIn(Room \sqcap \\ \exists containsAppliance.(TV \sqcap \exists isSwitched.(_{true})))) \end{aligned} \quad (1)$$

If the user is watching the television and the system receives the values of all the sensors, then it is able to conclude that the user's current activity is of the type *WatchingTVActivity*. In contrast, if the value of the presence sensor is not available, then it is not possible to infer that the user is watching the television.

In addition, sometimes there is not a rule of thumb to classify an individual as a member of a class. For instance, we can classify the action that the system has to perform regarding the current activity of the user. Thus, we can define that the system should turn off the television, when the user is not watching it:

$$TurnOffTV \equiv \exists requiredBy.(Person \sqcap \forall isDoing.\neg WatchingTVActivity \sqcap \exists hasAppliance.(TV \sqcap \exists isSwitched.(_{true}))) (2)$$

However, this concept definition does not accurately model the reality. The action can fulfil every condition expressed in the *TurnOffTV* definition, but the television should not be turned off. This situation may occur when the user goes to the toilet or answers a call phone in another room, among others.

In these cases in which the information of the domain comes with quantitative uncertainty or vagueness, ontology languages are less suitable [11]. Uncertainty is usually considered as the different aspects of the imperfect knowledge, such as vagueness or incompleteness. In addition, the uncertainty reasoning is defined as the collection of methods to model and reason with knowledge in which boolean truth values are unknown, unknowable or inapplicable [19]. Other authors [1] [11] consider that there are enough differences to distinguish between uncertainty and vague knowledge. According to them, uncertainty knowledge is comprised by statements that are either true or false, but we are not certain about them due to our lack of knowledge. In contrast, vagueness knowledge is composed of statements that are true to certain degree due to vague notions.

In our work, we are more interested in the uncertainty caused by the lack of information rather than the vague knowledge. For this reason, probabilistic approaches are more suitable to solve our problem.

3 Turambar Solution

Our proposal, called Turambar, combines a Bayesian network model with an OWL 2 DL ontology in order to handle uncertainty. A Bayesian network [13] is a graphical model that is defined as a directed acyclic graph. The nodes in the model represent the random variables and the edges define the dependencies between the random variables. Each variable is conditionally independent of its non descendants given the value of its parents.

Turambar is able to calculate the probability associated to a class, object property or data property assertions. These probabilistic assertions have only two constraints:

- The class expression employed in the class assertion should be a class.
- For positive and negative object property assertions, the object property expression should be an object property.

However, these limitations can be solved declaring a class equivalent to a class expression or an object property as the equivalent of an inverse object property. Examples of probabilistic assertions that can be calculated with Turambar are:

$$isIn(John, Bedroom1)\ 0.7 \tag{3}$$
$$WatchingTVActivity(Action1)\ 0.8 \tag{4}$$
$$isSwitched(TV1, true)\ 1 \tag{5}$$

The probabilistic object property assertion expressed in (3) states that John is in Bedroom1 with a probability of 0.7. On the other hand the probabilistic class assertion (4) describes that the Action1 is member of the class *WatchingTVActivity* with a probability of 0.2. Finally, the probabilistic data property assertion (5) defines that the television, *TV1*, is on with a probability of 1.0. The probability associated to these assertions is calculated through Bayesian networks that describe how other property and class assertions influence each other. In Turambar, the probabilistic relationships should be defined by an expert. In other words, the Bayesian networks must be generated by hand, since learning a Bayesian network is out of the scope of this paper and it is not the goal of this work.

3.1 Turambar Functionality Definition

The classes, object properties and data properties of the OWL 2 DL ontology involved in the probabilistic knowledge are connected to the random variables defined in the Bayesian network model. For example, the OWL class *WatchingTVActivity* is connected to at least one random variable, in order to be able to calculate probabilistic assertions about that class. The set of data properties, object properties and classes that are linked to random variables is called $\mathcal{V}_{prob}$ and a member of $\mathcal{V}_{prob}$, $vprob_i$; such that $vprob_i \in \mathcal{V}_{prob}$.

In Turambar, every random variable (RV) is associated to a $\mathcal{V}_{prob}$ and every RV's domain is composed of a set of functions that determine the values that a random variable can take, such as $Val(RV) = \{f_1, f_2...f_n\}$ and $f_i \in Val(RV)$. These functions require a property or class and individual to calculate the probabilistic assertion, such as $f_i :\ a_1, ex\ \rightarrow result$ where a_1 is an OWL individual, ex, a class, data property or object property; result, a class assertion, object property assertion, data property assertion or void (no assertion). In the case, that every function in the domain of a random variable returns void, it means that the random variable is auxiliary. In contrast, if any f_i in the domain of a random variable returns a probability associated to an assertion, then the random variable is called final.

For instance, the data property *lieOnBedTime* is linked to a random variable named *SleepTime* whose domain is composed of two functions f_1 that check if the user has been sleeping for less than 8 hours and f_2 function that checks if the user has been sleeping for more than 8 hours. Both functions are not able to generate assertions, so the random variable *SleepTime* is auxiliary. By contrast, *WatchingTVActivity* class is linked to a random variable called *WatchingTV* whose domain comprises f_3 function that checks if an individual is member of the class *WatchingTVActivity* (e.g. $WatchingTVActivity(Activity1)$ 0.8) and

the f_4 function which checks if an individual is a member of the complement of *WatchingTVActivity*.

It is also important to remark that a $vprob_i$ can be referenced from several random variables. For example, the *TurnOffTV* depends on the user's impairments, so if the blind user is deaf, it is more likely that the television needs to be turned off. Additionally, having an impairment also affects to the probability of having another impairment: deaf people have a higher probability of also being mute. In this case, we can link *hasImpairment* object property with two random variables in order to model it.

Apart from the conditional probability distribution, nodes connected between them may have an associated node context. The context defines how different random variables are related between them and the condition that must fulfil. This context establishes an unequivocal relationship in which every individual involved in that relationship should be gatherer before calculating the probability of an assertion. If the relationship is not fulfilled then the probabilistic query cannot be answered. For example, to estimate the probability for the *TurnOffTV*, the reasoner needs to know who is the user and in which room he is. For this case the relationship may be the following one $isIn(?user, ?room) \wedge requiredBy(?action, ?user) \wedge hasAppliance(?user, ?tv)$, being $?user, ?action, ?tv$ and $?room$ variables. So, if we ask for the probability that *Action1* is member of *TurnOffTV*, such as Pr(TurnOffTV(Action1)), then the first step to calculate it is to check its context. If everything is right the evaluation of this relationship should return that the *Action1* is required only by one user who is only in one room and has only one television. Otherwise, the probability cannot be calculated.

Our proposal can be viewed as a $\mathcal{SROIQ}(\mathcal{D})$ extension that includes a probabilistic function Pr which maps role assertions and concept assertions to a value between 0 and 1. The sum of the probabilities obtained for a random variable is equal to 1. In contrast, the sum of probabilities for the set of assertions obtained for a $vprob_i$ may be different from 1. For instance, the object property *hasImpairment* is related to two random variables one to calculate the probability of being deaf and another one to calculate the probability of being mute. If both random variables have a domain with two functions, we can get four probabilistic assertions that sums 2 instead of 1, but the sum of probabilities obtained in one random variable is 1:

- Random variable deaf's assertions: $hasImpairment(John, Deaf)0.8$ and $\neg hasImpairment(John, Deaf)0.2$.
- Random variable mute's assertions: $hasImpairment(John, Mute)0.7$ and $\neg hasImpairment(John, Mute)0.3$.

The probability of an assertion that exists in the OWL 2 DL ontology is always 1 although the data property, object properties or class is not member of V_{prob}. For example, if an assertion states that John is a *Person* ($Person(John)$) and we ask for the probability of this assertion, then its probability is 1, such as $Person(John)1$. However, if the data property, object properties or class is not member of V_{prob} and there is not an assertion in the OWL 2 DL ontology

that states it, then the probability for that assertion is unknown. We consider that the probabilistic ontology is satisfied if the OWL 2 DL ontology is satisfied and the Bayesian network model is not in contradiction with the OWL ontology knowledge.

3.2 Turambar Ontology Specification

In Turambar, a probabilistic ontology comprises an ordinary OWL 2 DL ontology and the dependency description ontology that defines the Bayesian network model.

The ordinary OWL ontology imports the Turambar annotations ontology, which defines the following annotations:

- *turambarOntology* annotation defines the URI of the dependency description ontology.
- *turambarClass* annotation links OWL classes in the ordinary ontology to random variables in the dependency description ontology.
- *turambarProperty* annotation connects OWL data properties and object properties in the ordinary ontology to random variables in the dependency description ontology.

We choose to separate the Bayesian network definition from the ordinary ontology in order to isolate the probabilistic knowledge definition from the OWL knowledge. We define isolation as the ability of exposing an ontology with an unique URI that locates the traditional ontology and the probabilistic one. So, given the URI of a probabilistic ontology, a compatible reasoner loads the ordinary ontology and the dependency description ontology it. In contrast, a traditional reasoner only loads the ordinary ontology. So, if the Turambar probabilistic ontology is loaded by a traditional reasoner, the traditional reasoner does not have access to the knowledge encoded in the dependency description ontology. In this way, we also want to promote the re-utilization of probabilistic ontologies as simple OWL 2 DL ontologies by traditional systems and the interoperability between our proposal and them.

On the other hand, the dependency description ontology defines the probabilistic model employed to estimate the probabilistic assertions. To model that knowledge, it imports the Turambar ontology, which defines the vocabulary to describe the probabilistic model. As the figure 1 shows, the main classes and properties in the Turambar ontology are the following ones:

- *Node* class represents the nodes in Bayesian networks. *Node* instances are defined as auxiliary random variables through the property *isAuxiliar*. The *hasProbabilityDistibution* object property links *Node* instances with their corresponding probability distributions and *hasState* object property associates *Node* instances with their domains. Furthermore, *hasChildren* object property and its inverse *hasParent* set the dependencies between *Node* instances. Finally, *hasContext* object property defines the context for a node and *hasVariable* object property, the value of the variable that the node requires.

- *MetaNode* is a special type of *Node* that is employed with non functional object properties and data properties. Its main functionality is to group several nodes that share a context and are related to the same property. For instance, in the case of the *hasImpairment* object property we can model a *MetaNode* with two nodes: *Deaf* and *Mute*. Both nodes share the same context but have different parents and states. The object property *compriseNode* identifies the nodes that share a context.
- *State* class defines the values of random variables' domain. In other words, it describes the functions which generate probabilistic assertions. These functions are expressed as a string through the data property *stateCondition*.
- *ProbabilityDistribution* class represents a probability distribution. *Probability* distributions are given in form of conditional probability tables. Cells of the conditional probability table are associated to the instances of *ProbabilityDistribution* through *hasProbability* object property.
- *Probability* class represents a cell of a conditional probability table, such $P(x_1 \mid x_2, x_3) = value$, where x_1, x_2 and x_3 are *State* individuals and *value* is the probability value. x_1 *State* is assigned to an instance of *Probability* class through the *hasValue* object property and x_2 and x_3 conditions through *hasCondition* object property. Finally, the data property *hasProbabilityValue* sets the probability value for that cell.
- *Context* class establishes the relationships between the nodes of a Bayesian network. Relationships between nodes are expressed as a SPARQL-DL query through the data property *relationship*.
- *Variable* class represents the variables of the context. Their instances identify the SPARQL-DL variables defined in the context SPARQL-DL query. The *variableName* data property establishes the name of the variable. For example, if the context has been defined with the following SPARQL-DL expression: *select ?a ?b where* { *PropertyValue(p:livesIn, ?a, ?b)*} , then we should create two instances of *Variable* with the *variableName* property value of *a* and *b*, respectively.
- *Plugin* class defines a library that provides some functions that are referenced by *State* class instances and are not included as member of the Turambar core. The core functions are the following ones: (i) numbers and strings comparison, (ii) ranges of number and string comparison, (iii) individual instances comparison, (iv) boolean comparison, (v) class memberships checking and (vi) the void assertion to define the probability that no assertion involves an individual. Only i, iii and iv are able to generate probabilistic assertions. Every function, except the void function, has their inverse function to check if that value has been asserted as false.

4 Related Works

We can classify probabilistic approaches to deal with uncertainty in two groups: probabilistic description logics approaches and probabilistic web ontology languages [11].

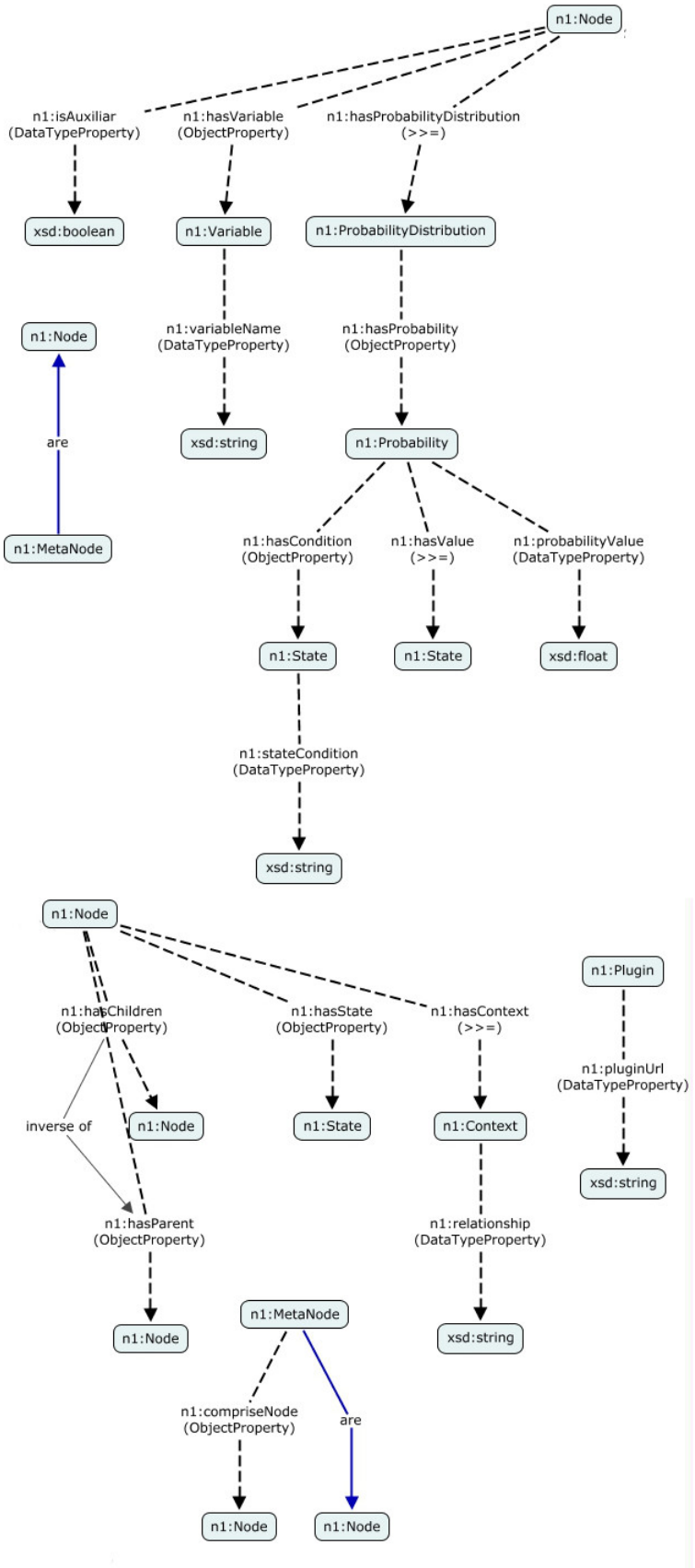


Fig. 1. Classes and properties defined by the Turambar ontology

In the first group, P-CLASSIC [9] extends description logic CLASSIC to add probability. In contrast, Pronto [8] is a probabilistic reasoner for P-$SROIQ$, a probabilistic extension of $SROIQ$. Pronto models probability intervals with its custom OWL annotation *pronto#certainty*. Apart from the previously described works, there are several other approaches that have been explained in different surveys such as [14].

In contrast, probabilistic web ontology languages combine OWL with probabilistic formalisms based on Bayesian networks. Since our proposal falls under this group, we will review in depth the most important works in this category: BayesOWL, OntoBayes and PR-OWL. The BayesOWL [7] framework extends OWL capacities for modelling and reasoning with uncertainty. It applies a set of rules to transform the class hierarchy defined in an OWL ontology into a Bayesian network. In the generated network there are two types of nodes: concept nodes and L-Nodes. The former one represents OWL classes and the latter one is a special kind of node that is employed to model the relationships defined by *owl:intersectionOf*, *owl:unionOf*, *owl:complementOf*, *owl:equivalentClass* and *owl:disjointWith* constructors. Concept nodes are connected between them by directed arcs that link superclasses with their classes. On the other hand, L-Nodes and concept nodes involved in a relationship are linked following the rules established for each constructor. The probabilities are defined with the classes *PriorProb*, for prior probabilities, and *CondProb*, for conditional probabilities. For instance, BayesOWL [22] recognizes some limitations: (i) variables should be binaries, (ii) probabilities should contain only one prior variable, (iii) probabilities should be complete and (iv) in case of inconsistency the result may not satisfy the constraints offered. BayesOWL approach is not valid for our purpose, because it only supports uncertainty to determine the class membership of an individual and this may not be enough for context modelling. For example, sensors' values may be represented as data and object properties values and knowing the probability that a sensor has certain value may be very useful for answering user's needs.

In contrast to BayesOWL, OntoBayes [21] focuses on properties. In OntoBayes, every random variable is a data or object property. Dependencies between them are described via the *rdfs:dependsOn* property. It supports to describe prior and conditional probabilities, besides it contains a property to specify the full disjoint probability distribution. Another improvement of OntoBayes over BayesOWL is that it supports multi-valued random variables. However, it is not possible to model relationships between classes in order to prevent errors when extracting Bayesian network structure from ontologies. OntoBayes offers us a solution for the limitation presented in BayesOWL regarding OWL properties, but its lack of OWL class support makes it unsuitable for our goal.

PR-OWL [5] is an OWL extension to describe complex bayesian models. It is based on the Multi-Entity Bayesian newtworks (MEBN) logic. MEBN [10] defines the probabilistic knowledge as a collection of MEBN fragments, named MFrags. A set of MFrags configures a MTheory and every PR-OWL ontology must contain at least one MTheory. To consider a MFrag set as a MTheory, it

must satisfy consistency constraints ensuring that it only exists a joint probability distribution over MFrags' random variables. In PR-OWL, probabilistic concepts can coexist with non probabilistic concepts, but these are only benefited by the advantages of the probabilistic ontology. Each MFrag is composed of a set of nodes which are classified in three groups: resident, input and context node. Resident nodes are random variables whose probability distribution is defined in the MFrag. Input nodes are random variables whose probability distribution is defined in a distinct MFrag than the one where is mentioned. In contrast, context nodes specify the constraints that must be satisfied by an entity to substitute an ordinary variable. Finally, node states are modelled with the object property named *hasPossibleValues*.

The last version of PR-OWL [2], PR-OWL 2, addresses the PR-OWL 1 limitations regarding to its compatibility with OWL: no mapping to properties of OWL and the lack of compatibility with existing types in OWL. Although, PR-OWL offers a good solution to deal with uncertainty, it does not provide some characteristics that we covet for our systems, such as isolation.

Our proposal is focused on computing the probability of data properties assertions, object properties assertions and class assertions. This issue is only covered by PR-OWL, because BayesOWL only takes into account class membership and OntoBayes, object and data properties.

In addition, we pretend to offer a way to keep the uncertainty information isolated as much as possible from the traditional ontology. With this policy, we want to ease the reutilization of our probabilistic ontologies by traditional systems that do not offer support for uncertainty and the interoperability between them. Furthermore, we aim to avoid that traditional reasoners load unnecessary information about the probabilistic knowledge that they do not need. Thus, if we load the Turambar probabilistic ontology located in http://www.example.org/ont.owl, traditional OWL reasoners load only the knowledge defined in the ordinary OWL ontology and do not have access to the probabilistic knowledge. In contrast, Turambar reasoner is able to load the ordinary OWL ontology and the dependency description ontology. The Turambar reasoner needs to access to the ordinary OWL ontology to answer traditional OWL queries and to find the evidences of the Bayesian networks defined in the dependency description ontology. It is also important to clarify that a class or property can have deterministic assertions and probabilistic assertions without duplicating them due to the links between Bayesian networks' nodes and OWL classes and properties through *turambarClass* and *turambarProperty*, respectively. Thanks to this feature, a Turambar ontology has a unique URI that allows it to be used as an ordinary OWL 2 DL ontology without loading the probabilistic knowledge. This characteristic is not offered by other approaches as far as we know.

Another difference with other approaches is that we have taken into account the extensibility of our approach through plug-ins to increase the basis functionalities. We believe that it is necessary to offer a straightforward, transparent and standard mechanism to extend reasoner functionality in order to cover heterogeneous domains' needs.

However, our approach has the shortcoming of assuming a simple attribute-value representation in comparison to PR-OWL. That means that each probabilistic query involves reasoning about the same fixed number of nodes, with only the evidence values changing from query to query. To solve this drawback, we can opt to employ situation specific Bayesian networks [12], as PR-OWL does. However, the development of custom plug-ins can overcome this limitation in some cases. Besides, thanks to this expressiveness restriction we are able to know the size of the Bayesian network and give a better estimation of the performance of the Turambar probabilistic ontology.

5 Conclusions and Future Work

In this work we have presented a proposal to deal with uncertainty in intelligent environments. Its main features are: a) it isolates the probabilistic information definition from traditional ontologies, b) it can be extended easily and c) it is oriented to intelligent environments.

As ongoing work, we are developing an extension to SPARQL-DL [17] in order to offer a simple mechanism to execute complex queries in a declarative way that abstracts developers from the reasoner implementation employed and its API. This extension proposes the addition of two new query atoms to query probabilistic knowledge: ProbType for probabilistic class assertions and ProbPropertyValue for probabilistic property assertions. We believe that this extension can ease the development of applications that employ Turambar.

As future work, we plan to create a graphical tool to ease the creation of probabilistic ontologies in order to promote its adoption. On the other hand, we plan to extend its expressivity and evaluate new and better ways to define the probabilistic description ontology in order to improve its maintainability. In addition, we are studying a formalism that allows us the definition of custom function for state evaluation that was independent of the programming language employed.

References

1. Baader, F., Küsters, R., Wolter, F.: The description logic handbook. chap. Extensions to description logics, pp. 219–261. Cambridge University Press, New York, NY, USA (2003), http://dl.acm.org/citation.cfm?id=885746.885753
2. Carvalho, R.N.: Probabilistic Ontology: Representation and Modeling Methodology. Ph.D. thesis, George Mason University (2011)
3. Chen, H., Perich, F., Finin, T., Joshi, A.: Soupa: standard ontology for ubiquitous and pervasive applications. In: Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on. pp. 258–267 (Aug 2004)
4. Coen, M.H.: Design principles for intelligent environments. In: Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence. pp. 547–554. AAAI '98/IAAI '98, American Association for Artificial Intelligence, Menlo Park, CA, USA (1998), http://dl.acm.org/citation.cfm?id=295240.295733

5. Costa, P.C.G.: Bayesian semantics for the Semantic Web. Ph.D. thesis, George Mason University (2005)
6. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. 5(1), 4–7 (Jan 2001), http://dx.doi.org/10.1007/s007790170019
7. Ding, Z.: Bayesowl: A Probabilistic Framework for Uncertainty in Semantic Web. Ph.D. thesis, Catonsville, MD, USA (2005)
8. Klinov, P.: Practical reasoning in probabilistic description logic. Ph.D. thesis, University of Manchester (2011)
9. Koller, D., Levy, A., Pfeffer, A.: P-CLASSIC: A tractable probabilistic description logic. In: In Proceedings of AAAI-97. pp. 390–397 (1997)
10. Laskey, K.: MEBN: A language for first-order bayesian knowledge bases. Artificial Intelligence 172(2-3), 140–178 (2008)
11. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web 6(4), 291–308 (2008)
12. Mahoney, S.M., Laskey, K.B.: Constructing situation specific belief networks. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. pp. 370–378. UAI'98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998), http://dl.acm.org/citation.cfm?id=2074094.2074138
13. Pearl, J.: Fusion, propagation, and structuring in belief networks. Artificial Intelligence 29(3), 241 – 288 (1986), http://www.sciencedirect.com/science/article/pii/000437028690072X
14. Predoiu, L., Stuckenschmidt, H.: Probabilistic models for the semantic web. The Semantic Web for Knowledge and Data Management: Technologies and Practices pp. 74–105 (2009)
15. Preuveneers, D., Van den Bergh, J., Wagelaar, D., Georges, A., Rigole, P., Clerckx, T., Berbers, Y., Coninx, K., Jonckers, V., De Bosschere, K.: Towards an extensible context ontology for ambient intelligence. In: Ambient Intelligence, Lecture Notes in Computer Science, vol. 3295, pp. 148–159. Springer Berlin Heidelberg (2004)
16. Ramos, C., Augusto, J.C., Shapiro, D.: Ambient intelligencethe next step for artificial intelligence. Intelligent Systems, IEEE 23(2), 15–18 (March 2008)
17. Sirin, E., Parsia, B.: SPARQL-DL: SPARQL query for OWL-DL. In: OWLED. vol. 258 (2007)
18. Strang, T., Linnhoff-Popien, C.: A context modeling survey. In: In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England (2004)
19. W3C: Uncertainty reasoning for the world wide web. Tech. rep., W3C (2008), http://www.w3.org/2005/Incubator/urw3/XGR-urw3/
20. Wang, X., Zhang, D., Gu, T., Pung, H.: Ontology based context modeling and reasoning using owl. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. pp. 18–22 (2004)
21. Yang, Y., Calmet, J.: Ontobayes: An ontology-driven uncertainty model. In: Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on. vol. 1, pp. 457–463. IEEE (2005)
22. Yun Peng, Z.D.: Bayesowl: Reference manual (Feb 2013), http://www.csee.umbc.edu/ ypeng/BayesOWL/manual/index.html

Learning to Propagate Knowledge in Web Ontologies

Pasquale Minervini[1], Claudia d'Amato[1], Nicola Fanizzi[1], Volker Tresp[2]

[1] Department of Computer Science - University of Bari, Italy
{pasquale.minervini|claudia.damato|nicola.fanizzi}@uniba.it
[2] Siemens AG, Corporate Technology, Munich, Germany
volker.tresp@siemens.com

Abstract. The increasing availability of structured machine-processable knowledge in the WEB OF DATA calls for machine learning methods to support standard pattern matching and reasoning based services (such as query-answering and inference). Statistical regularities can be efficiently exploited to overcome the limitations of the inherently incomplete knowledge bases distributed across the Web. This paper focuses on the problem of predicting missing class-memberships and property values of individual resources in Web ontologies. We propose a transductive inference method for inferring missing properties about individuals: given a class-membership/property value prediction problem, we address the task of identifying relations encoding similarities between individuals, and efficiently propagating knowledge across their relations.

1 Introduction

Standard query answering and reasoning services for the Semantic Web [2] (SW) mainly rely on deductive inference. However, purely deductive inference suffers from several limitations [20]: reasoning tasks might be computationally complex, do not always address uncertainty and only rely on axiomatic prior knowledge. However, purely deductive reasoning with SW representations suffers from several limitations owing to its complexity and the inherent incompleteness and incoherence of distributed knowledge bases (KBs) in a Web-scale scenario modeled by formal ontologies. In such a context many complex tasks (e.g. query answering, clustering, ranking, etc.) are ultimately based on assessing the truth of ground facts. Deciding on the truth of specific facts (assertions) in SW knowledge bases requires to take into account the *open-world* form of reasoning adopted in this context: a failure on ascertaining the truth value of a given fact does not necessarily imply that such fact is false, but rather that its truth value cannot be deductively inferred from the KB (e.g. because of incomplete or uncertain knowledge). Other issues are related to the distributed nature of the data across the Web. Cases of contradictory answers or flawed inferences may be caused by distributed pieces of knowledge that may be mutually conflicting.

The prediction of the truth value of an assertion can be cast as a *classification* problem to be solved through *statistical learning*: individual resources in an ontology can be regarded as statistical units, and their properties can be statistically inferred even when they cannot be deduced from the KB. Several approaches have been proposed in the SW literature (see [20] for a recent survey). A major issue with the methods proposed so far

is that the induced statistical models (as those produced by kernel methods, tensor factorization, etc.) are either difficult to interpret by experts and to integrate in logic-based SW infrastructures, or computationally impractical.

1.1 Related Work

A variety of methods have been proposed for predicting the truth value of assertions in Web ontologies, including generative models [18,21], kernel methods [4,8,16], upgrading of propositional algorithms [15], matrix and tensor factorization methods [9,17,26]. An issue with existing methods is that they either rely on a possibly expensive search process, or induce statistical models that are often not easy to interpret by human experts. Kernel methods induce models (such as separating hyperplanes) in a high-dimensional feature space implicitly defined by a kernel function. The underlying kernel function itself usually relies on purely syntactic features of the neighborhood graphs of two individual resources (such as their common subtrees [16]). In both cases, there is not necessarily a direct translation in terms of domain knowledge. Latent variable and matrix or tensor factorization methods such as [9, 17, 21, 26] try to explain the observations in terms of latent classes or attributes, which also may be non-trivial to describe using the domain's vocabulary. The approaches in [15, 18] try to overcome this limitation by making use of more complex features and knowledge representation formalisms jointly with the ontology's terminology: these methods involve either a search process in a possibly very large feature space or complex probabilistic inference tasks, which might not be feasible in practice.

1.2 Contribution

We propose a transductive inference method for predicting the truth value of assertions, which is based on the following intuition: examples (each represented by a individual in the ontology) that are *similar* in some aspects tend to be linked by specific relations. Yet it may be not straightforward to find such relations for a given learning task. Our approach aims at identifying such relations, and permits the efficient propagation of information along chains of related entities. It turns out to be especially useful with real-world *shallow* ontologies [22] (i.e. those with a relatively simple fixed terminology and populated by very large amounts of instance data such as citation or social networks), in which the characteristics of related entities tend to be correlated. These features are particularly relevant in the context of the *Linked Open Data* [10] (LOD). Unlike other approaches, the proposed method can be used to elicit which relations link examples with similar characteristics. The proposed method is efficient, since the complexity of both inference and learning grows polynomially with the number of statistical units.

As in graph-based semi-supervised learning (SSL) methods [5], we rely on a similarity graph among examples for propagating knowledge. However, SSL methods are often designed for propositional representations; our method addresses the problem of learning from real ontologies, where examples can be interlinked by heterogeneous relations. In particular, this paper makes the following contributions:

- A method to efficiently *propagating* knowledge among similar examples: it leverages a similarity graph, which plays a critical role in the propagation process.

– An approach to efficiently *learning* the similarity matrix, by leveraging a set of semantically heterogeneous relations among examples in the ontology.

To the best of our knowledge, our approach is the first to explicitly identify relations that semantically encode similarities among examples w.r.t. a given learning task.

The remainder of the paper is organized as follows. In the next section, we review the basics of semantic knowledge representation and reasoning tasks, and we introduce the concept of *transductive learning* in the context of semantic KBs. In Sect. 3, we illustrate the proposed method based on an efficient propagation of information among related entities, and address the problem of identifying the relations relevant to the learning task. In Sect. 4, we provide empirical evidence for the effectiveness of the proposed method. Finally, in Sect. 5, we summarize the proposed approach, outline its limitations and discuss possible future research directions.

2 Transductive Learning with Web Ontologies

We consider ontological KBs using *Description Logics* (DLs) as a language to describe objects and their relations [1] Basics elements are *atomic concepts* $N_C = \{C, D, \ldots\}$ interpreted as subsets of a domain of objects (e.g. `Person` or `Article`), and *atomic roles* $N_R = \{R, S, \ldots\}$ interpreted as binary relations on such a domain (e.g. `friendOf` or `authorOf`). Domain objects are represented by *individuals* $N_I = \{a, b, \ldots\}$: each represents a domain entity (such as a person in a social network, or an article in a citation network).

In RDF(S)/OWL [1], concepts, roles and individuals are referred to as *classes*, *properties* and *resources*, respectively, and are identified by their URIs. Complex concept descriptions can be built using atomic concepts and roles by means of specific constructors offered by expressive DLs.

A *Knowledge Base* (KB) $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ contains a *TBox* $\mathcal{T}$, made by terminological axioms, and an *ABox* $\mathcal{A}$, made by ground axioms, named *assertions*, involving individuals. $\mathsf{Ind}(\mathcal{A})$ denotes the set of individuals occurring in $\mathcal{A}$. As inference procedure, *Instance Checking* consists in deciding whether $\mathcal{K} \models Q(a)$ (where Q is a query concept and a is an individual) holds. Because of the *Open-World Assumption* (OWA), instance checking may provide three possible outcomes, i.e. i) $\mathcal{K} \models Q(a)$, ii) $\mathcal{K} \models \neg Q(a)$ and iii) $\mathcal{K} \not\models Q(a) \wedge \mathcal{K} \not\models \neg Q(a)$. This means that failing to deductively infer the membership of an individual a to a concept Q does not imply that a is a member of its complement $\neg Q$.

Given the inherent incompleteness of deductive inference under open-world reasoning, in this work we focus on *transductive inference* [27]: this consists in reasoning from observed (training) cases to a specific set of test cases, without necessarily generalizing to unseen cases. This differs from *inductive inference*, where training cases are used to infer a general model, which is then applied to test cases.

The main motivation behind the choice of transductive learning is described by the *main principle* in [27]: "If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem

[1] OWL 2 W3C Recommendation: `http://www.w3.org/TR/owl-overview/`

as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem".

On the ground of the available information, the proposed approach aims at learning a *labeling function* for a given target class that can be used for predicting whether individuals belong to a class C (positive class) or to its complement $\neg C$ (negative class) when this cannot be inferred deductively. The problem can be defined as follows:

Definition 2.1 (Transductive Class-Membership Learning). Given:

- *a* target *class C in a KB $\mathcal{K}$;*
- *a set of examples $X \subseteq \mathsf{Ind}(\mathcal{A})$ partitioned into:*
  - *a set of* positive examples*: $X_+ \triangleq \{a \in X \mid \mathcal{K} \models C(a)\}$;*
  - *a set of* negative examples*: $X_- \triangleq \{a \in X \mid \mathcal{K} \models \neg C(a)\}$;*
  - *a set of* neutral *(unlabeled)* examples*: $X_0 \triangleq \{a \in X \mid a \notin X_+ \wedge a \notin X_-\}$;*
- *a space of* labeling functions *$\mathcal{F}$ with domain X and range $\{-1, +1\}$, i.e.*

$$\mathcal{F} \triangleq \{\mathbf{f} \mid \mathbf{f} : X \to \{+1, -1\}\};$$

- *a* cost function $\mathrm{cost}(\cdot) : \mathcal{F} \mapsto \mathbb{R}$*, specifying the cost associated to each labeling functions $\mathbf{f} \in \mathcal{F}$;*

Find $\mathbf{f}^* \in \mathcal{F}$ *minimizing* $\mathrm{cost}(\cdot)$ *w.r.t. X:*

$$\mathbf{f}^* \leftarrow \arg\min_{\mathbf{f} \in \mathcal{F}} \mathit{cost}(\mathbf{f}).$$

The transductive learning task is cast as the problem of finding a *labeling function* $\mathbf{f}^*$ for a target class C, defined over a finite set of labeled and unlabeled examples X (represented by a subset of the individuals in the KB), and minimizing some arbitrary cost criterion.

Example 2.1 (Transductive Class-Membership Learning). Consider an ontology modeling an academic domain. The problem of learning whether a set of persons is affiliated to a given research group or not, provided a set of positive and negative examples of affiliates, can be cast as a *transductive class-membership learning* problem: examples (a subset of the individuals in the ontology, each representing a person), represented by the set X, can be either *positive*, *negative* or *neutral* depending on their membership to a target class `ResearchGroupAffiliate`. Examples can be either *unlabeled* (if their membership to the target class cannot be determined) or *labeled* (if positive or negative). The transductive learning problem reduces to finding the best labeling function $\mathbf{f}$ (according to a given criterion, represented by the cost function) providing membership values for examples in X.

In this work, we exploit the relations holding among examples to *propagate* knowledge (in the form of label information) through chains of related examples. Inspired by graph-based semi-supervised transductive learning, the criterion on which the cost function will be defined follows the *semi-supervised smoothness assumption* [5]: if two points in a high-density region are close, then so should be the corresponding outputs. Transductive and semi-supervised learning are closely related: in both settings, test examples are available during the learning task (in the form of unlabeled examples). In the proposed method, the learning task is reduced to finding a labeling function $\mathbf{f}$ which *varies smoothly* across the similarity graph defined over examples.

3 Knowledge Propagation

In this section, we present our method for solving the learning problem in Def. 2.1 in the context of Web ontologies. In Sect. 3.1 we show that a similarity graph between examples can be used to efficiently propagate label information among similar examples. The effectiveness of this approach strongly depends on the choice of a similarity graph (represented by its adjacency matrix $\mathbf{W}$). In Sect. 3.2, we show how the matrix $\mathbf{W}$ can be learned from examples, by leveraging their relationship within the ontology.

3.1 Transductive Inference as an Optimization Problem

We now propose a solution to the transductive learning problem in Def. 2.1. As discussed in the end of Sect. 2, we need a labeling function $\mathbf{f}^*$ defined over examples X, which is both consistent with the training labels, and *varies smoothly* among similar examples (according to the semi-supervised smoothness assumption). In the following, we assume that a similarity graph over examples in X is already provided. Such a graph is represented by its adjacency matrix $\mathbf{W}$, such that $\mathbf{W}_{ij} = \mathbf{W}_{ji} \geq 0$ if $x_i, x_j \in X$ are *similar*, and 0 otherwise (for simplicity we assume that $\mathbf{W}_{ii} = 0$). In Sect. 3.2 we propose a solution to the problem of learning $\mathbf{W}$ from examples.

Formally, each labeling function $\mathbf{f}$ can be represented by a finite-size vector, where $\mathbf{f}_i \in \{-1, +1\}$ is the label for the i-th element in the set of examples X. According to [30], labels can be enforced to vary smoothly among similar examples by considering a cost function with the following form:

$$E(\mathbf{f}) \triangleq \frac{1}{2} \sum_{i=1}^{|X|} \sum_{j=1}^{|X|} \mathbf{W}_{ij} (\mathbf{f}_i - \mathbf{f}_j)^2 + \epsilon \sum_{i=1}^{|X|} \mathbf{f}_i^2, \tag{1}$$

where the first term enforces the labeling function to vary smoothly among similar examples (i.e. those connected by an edge in the similarity graph), and the second term is a L_2 regularizer (with weight $\epsilon > 0$) over $\mathbf{f}$. A labeling for unlabeled examples X_0 is obtained by minimizing the function $E(\cdot)$ in Eq. 1, constraining the value of $\mathbf{f}_i$ to 1 (resp. -1) if $x_i \in X_+$ (resp. $x_i \in X_-$).

Let $L \triangleq X_+ \cup X_-$ and $U \triangleq X_0$ represent labeled and unlabeled examples respectively. Constraining $\mathbf{f}$ to discrete values, i.e. $\mathbf{f}_i \in \{-1, +1\}, \forall x_i \in X_0$, has two main drawbacks:

- The function $\mathbf{f}$ only provides a *hard* classification (i.e. $\mathbf{f}_U \in \{-1, +1\}^{|U|}$), any measure of confidence;
- E defines the energy function of a discrete Markov Random Field, and calculating the marginal distribution over labels $\mathbf{f}_U$ is inherently difficult [13].

To overcome these problems, in [30] authors propose a continuous relaxation of $\mathbf{f}_U$, where labels for unlabeled examples are represented by real values, $\mathbf{f}_U \in \mathbb{R}^{|U|}$, which also express a measure of the classification confidence. This allows for a simple, closed-form solution to the problem of minimizing E for a fixed $\mathbf{f}_L$, where $\mathbf{f}_L$ represents the labels of labeled examples.

Application to Class-Membership Learning We can solve the learning problem in Def. 2.1 by minimizing the cost function $E(\cdot)$ in Eq. 1. It can be rewritten as [30]:

$$E(\mathbf{f}) = \mathbf{f}^T(\mathbf{D} - \mathbf{W})\mathbf{f} + \epsilon\mathbf{f}^T = \mathbf{f}^T(\mathbf{L} + \epsilon\mathbf{I})\mathbf{f}, \tag{2}$$

where $\mathbf{D}$ is a diagonal matrix such that $\mathbf{D}_{ii} = \sum_{j=1}^{|X|} \mathbf{W}_{ij}$ and $\mathbf{L} \triangleq \mathbf{D} - \mathbf{W}$ is the *graph Laplacian* of $\mathbf{W}$. Reordering the matrix $\mathbf{W}$, the graph Laplacian $\mathbf{L}$ and the vector $\mathbf{f}$ w.r.t. their membership to L and U, they can be rewritten as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{LL} & \mathbf{W}_{LU} \\ \mathbf{W}_{UL} & \mathbf{W}_{UU} \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \mathbf{L}_{LL} & \mathbf{L}_{LU} \\ \mathbf{L}_{UL} & \mathbf{L}_{UU} \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \mathbf{f}_L \\ \mathbf{f}_U \end{bmatrix}. \tag{3}$$

The problem of finding the $\mathbf{f}_U$ minimizing the cost function E for a fixed value for $\mathbf{f}_L$ has a closed form solution:

$$\mathbf{f}_U^* = (\mathbf{L}_{UU} + \epsilon\mathbf{I})^{-1}\mathbf{W}_{UL}\mathbf{f}_L. \tag{4}$$

Complexity A solution for Eq. 4 can be computed efficiently in nearly-linear time w.r.t. $|X|$. Indeed computing $\mathbf{f}_U^*$ can be reduced to solving a linear system in the form $\mathbf{A}\mathbf{x} = \mathbf{b}$, with $\mathbf{A} = (\mathbf{L}_{UU} + \epsilon\mathbf{I})$, $\mathbf{b} = \mathbf{W}_{UL}\mathbf{f}_L$ and $\mathbf{x} = \mathbf{f}_U^*$. A linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\mathbf{A} \in \mathbb{R}^{n \times n}$ can be solved in nearly linear time w.r.t. n if the coefficient matrix $\mathbf{A}$ is *symmetric diagonally dominant*[2] (SDD), e.g. using the algorithm in [6], whose time-complexity is $\approx \mathrm{O}\big(m \log^{1/2} n\big)$, where m is the number of non-zero entries in $\mathbf{A}$ and n is the number of variables in the system of linear equations. In Eq. 4, the matrix $(\mathbf{L}_{UU} + \epsilon\mathbf{I})$ is SDD (since $\mathbf{L}_{UU}$ is a principal submatrix of $\mathbf{L}$, which is SDD [25]). An efficient parallel solver for SDD linear systems is discussed in [19].

3.2 Learning to Propagate Knowledge in Web Ontologies

As discussed in Sect. 3.1, the proposed approach to knowledge propagation relies on a similarity graph, represented by its adjacency matrix $\mathbf{W}$.

The underlying assumption of this work is that some relations among examples in the KB might encode a similarity relation w.r.t. a specific target property or class. Identifying such relations can help propagate information through similar examples.

In the literature, this effect goes under the name of *Guilt-by-Association* [14]: related examples influence each other, and some relations (e.g. *friendship* in a social network) can encode some form of similarity w.r.t. a set of properties (such as political views, hobbies, religious beliefs). However, depending on the learning task at hand, not all relations are equally effective at encoding similarity relations. For example in a social network, friends may tend to share common interests, while quiet people may tend to prefer talkative friends and vice-versa [14].

In this work, we represent each relation by means of an *adjacency matrix* $\tilde{\mathbf{W}}$, such that $\tilde{\mathbf{W}}_{ij} = \tilde{\mathbf{W}}_{ji} = 1$ iff the relation $\texttt{rel}(x_i, x_j)$ between x_i and x_j holds in the ontology; $\texttt{wrel}$ might represent any generic relation between examples (e.g. friendship or co-authorship). For simplicity, we assume that $\tilde{\mathbf{W}}_{ii} = 0, \forall i$.

[2] A matrix $\mathbf{A}$ is SDD iff $\mathbf{A}$ is symmetric (i.e. $\mathbf{A} = \mathbf{A}^T$) and $\forall i : \mathbf{A}_{ii} \geq \sum_{i \neq j} |\mathbf{A}_{ij}|$.

Given a set of adjacency matrices $\mathcal{W} \triangleq \{\tilde{\mathbf{W}}_1, \ldots, \tilde{\mathbf{W}}_r\}$ (one for each relation type), we can define $\mathbf{W}$ as a linear combination of the matrices in $\mathcal{W}$:

$$\mathbf{W} \triangleq \sum_{i=1}^{r} \mu_i \tilde{\mathbf{W}}_i, \quad \text{with } \mu_i \geq 0, \forall i \tag{5}$$

where μ_i, is a parameter representing the contribution of $\tilde{\mathbf{W}}_i$ to the adjacency matrix of the similarity graph $\mathbf{W}$. Non-negativity in $\boldsymbol{\mu}$ ensures that $\mathbf{W}$ has non-negative weights, and therefore the corresponding graph Laplacian $\mathbf{L}$ is positive semidefinite [25] (PSD), leading to the unique, closed form solution in Eq. 4.

Probabilistic Interpretation as a Gaussian Random Field Let us consider the relaxation of the energy function in Eq. 2, such that labels $\mathbf{f}$ are allowed to range in $\mathbb{R}^{|X|}$ ($\mathbf{f} \in \mathbb{R}^{|X|}$). It corresponds to the following probability density function over $\mathbf{f}$:

$$p(\mathbf{f}) = (2\pi)^{-\frac{1}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}E(\mathbf{f})\right\} = \mathcal{N}\Big(\mathbf{0}, (\mathbf{L} + \epsilon\mathbf{I})^{-1}\Big). \tag{6}$$

The probability density function in Eq. 6 defines a Gaussian Markov Random Field [13] (GMRF) $\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = \boldsymbol{\Omega}^{-1}$ and $\boldsymbol{\Omega} = (\mathbf{L} + \epsilon\mathbf{I})$ are respectively its *covariance* and *inverse covariance* (or *precision*) matrix, and $|\boldsymbol{\Sigma}|$ indicates the determinant of the covariance matrix $\boldsymbol{\Sigma}$.

The covariance matrix and its inverse fully determine the independence relations among variables in a GMRF [13]: if $\boldsymbol{\Omega}_{ij} \neq 0$, then there is an edge between $\mathbf{f}_i$ and $\mathbf{f}_j$ in the minimal I-map GMRF of p. A zero element in the inverse covariance matrix implies that two variables are conditionally independent given all the other variables.

Parameters Learning The parametric form of $\mathbf{W}$ is fully specified by the parameters $\boldsymbol{\mu}$ in Eq. 5, which may be unknown. We will estimate the parameters by means of *Leave-One-Out* (LOO) *Error minimization*: given that propagation can be performed efficiently, we are able of directly minimizing the LOO error, consisting in the summation of reconstruction errors obtained by considering each labeled example, in turn, as unlabeled, and predicting its label (as in [29]). This leads to a computationally efficient procedure for evaluating the matrix $\mathbf{W}$, and yields more flexibility as arbitrary loss functions are allowed. Let $U_i \triangleq U \cup \{x_i\}$ and $L_i \triangleq L - \{x_i\}$: the labeling vector $\mathbf{f}$ and matrices $\mathbf{W}$ and $\mathbf{L}$, for any given $x_i \in L$, can be rewritten as follows:

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_{L_i} \\ \mathbf{f}_{U_i} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \mathbf{W}_{L_iL_i} & \mathbf{W}_{L_iU_i} \\ \mathbf{W}_{U_iL_i} & \mathbf{W}_{U_iU_i} \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \mathbf{L}_{L_iL_i} & \mathbf{L}_{L_iU_i} \\ \mathbf{L}_{U_iL_i} & \mathbf{L}_{U_iU_i} \end{bmatrix}, \tag{7}$$

where w.l.o.g. we assume that the left-out example $x_i \in L$ corresponds to the first element in U_i (in the enumeration used for the block representation in Eq. 7). Let $\ell(x, \hat{x})$ be a generic, differentiable loss function (e.g. $\ell(x, \hat{x}) = |x - \hat{x}|$ for the absolute loss, or $\ell(x, \hat{x}) = (x - \hat{x})^2/2$ for the quadratic loss). The LOO Error is defined as follows:

$$\mathcal{Q}(\boldsymbol{\Theta}) \triangleq \sum_{i=1}^{|L|} \ell(\mathbf{f}_i, \hat{\mathbf{f}}_i), \tag{8}$$

where $\mathbf{e}^T \triangleq (1, 0, \ldots, 0) \in \mathbb{R}^{u+1}$ and $\hat{\mathbf{f}}_i \triangleq \mathbf{e}^T(\mathbf{L}_{U_iU_i} + \epsilon\mathbf{I})^{-1}\mathbf{W}_{U_iL_i}\mathbf{f}_{L_i}$ represents the continuous label value assigned to x_i as if such a value was not known in advance. The vector $\mathbf{e}^T$ is needed to select the first value of $\mathbf{f}^*_{U_i}$ only, i.e. the inferred continuous label associated to the left-out example $x_i \in L$. This leads to the definition of the following criterion for learning the optimal set of parameters $\boldsymbol{\Theta} \triangleq \{\boldsymbol{\mu}, \epsilon\}$:

Definition 3.1 (Minimum LOO Error Parameters). *Given a set of labeled (resp. unlabeled) examples L (resp. U) and a similarity matrix $\mathbf{W}$ defined by parameters $\boldsymbol{\Theta}$ (according to the parametric form of $\mathbf{W}$ in Eq. 5), the* minimum LOO Error Parameters *$\boldsymbol{\Theta}^*_{LOO}$ are defined as follows:*

$$\boldsymbol{\Theta}^*_{LOO} \triangleq \arg\min_{\boldsymbol{\Theta}} \mathcal{Q}(\boldsymbol{\Theta}) + \lambda||\boldsymbol{\Theta}||^2, \tag{9}$$

where the function $\mathcal{Q}$ is defined as in Eq. 8 and $\lambda > 0$ is a small positive scalar that weights a regularization term over $\boldsymbol{\Theta}$ (useful for avoiding some parameters to diverge).

The objective function in Def. 3.1 is differentiable and can be efficiently minimized by using gradient-based function minimization approaches such as L-BFGS.

Let $\mathbf{Z}_i = (\mathbf{L}_{U_iU_i} + \epsilon\mathbf{I})$. The gradient of $\mathcal{Q}$ w.r.t. a parameter $\theta \in \boldsymbol{\Theta}$ is given by:

$$\frac{\partial \mathcal{Q}(\boldsymbol{\Theta})}{\partial \theta} = \sum_{i=1}^{|L|} \frac{\partial \ell(\mathbf{f}_i, \hat{\mathbf{f}}_i)}{\partial \hat{\mathbf{f}}_i} \left[\mathbf{e}^T \mathbf{Z}_i^{-1} \left(\frac{\partial \mathbf{W}_{U_iL_i}}{\partial \theta} \mathbf{f}_{L_i} - \frac{\partial \mathbf{Z}_i}{\partial \theta} \mathbf{f}^*_{U_i} \right) \right]. \tag{10}$$

Complexity of the Gradient Calculation Let $\mathbf{z}_i = \left(\frac{\partial \mathbf{W}_{U_iL_i}}{\partial \theta} \mathbf{f}_{L_i} - \frac{\partial \mathbf{Z}_i}{\partial \theta} \mathbf{f}^*_{U_i} \right)$. Calculating $\mathbf{Z}_i^{-1}\mathbf{z}_i$ can be reduced to solving a linear system in the form $\mathbf{A}\mathbf{x} = \mathbf{b}$, with $\mathbf{A} = \mathbf{Z}_i = (\mathbf{L}_{U_iU_i} + \epsilon\mathbf{I})$ and $\mathbf{b} = \mathbf{z}_i$. As discussed in Sect. 3.1, this calculation has a nearly-linear complexity in the number of non-zero elements in $\mathbf{A}$, since $\mathbf{Z}_i$ is SDD.

4 Empirical Evaluation

The transductive inference method discussed in Sect. 3, which we will refer to as *Adaptive Knowledge Propagation* (AKP), was experimentally evaluated [3] in comparison with other approaches proposed in the literature on a variety of assertion prediction problems. In the following, we describe the setup of experiments and their outcomes.

4.1 Setup

In empirical evaluations, we used an open source DL reasoner [4]. In experiments, we considered the DBPEDIA 3.9 Ontology [3]. DBPEDIA [3] makes available structured information extracted from Wikipedia the LOD cloud providing unique identifiers for the described entities that can be dereferenced over the Web. DBPEDIA 3.9, released in September 2013, describes 4.0 million entities.

[3] Sources and datasets are available at `http://lacam.di.uniba.it/phd/pmm.html`

[4] Pellet v2.3.1 – `http://clarkparsia.com/pellet/`

Experimental Setting As discussed in Sect. 3.2, parameters $\boldsymbol{\Theta} = \{\boldsymbol{\mu}, \epsilon\}$ in AKP are estimated by minimizing the Leave-One-Out error $\mathcal{Q}$, as described in Eq. 9. We solved the problem by using *Projected Gradient Descent*, according to the gradient formulation in Eq. 10 (enforcing $\boldsymbol{\mu} \geq \mathbf{0}$ and $\epsilon > 0$), together with an intermediate line search to assess the step size. The regularization parameter λ in Eq. 9 was fixed to $\lambda = 10^{-8}$. In this work, each of the adjacency matrices $\mathcal{W} = \{\tilde{\mathbf{W}}_1, \ldots, \tilde{\mathbf{W}}_r\}$ is associated to a distinct *atomic role* in the ontology, linking at least two examples.

Before each experiment, all knowledge inherent to the target class was removed from the ontology. Following the evaluation procedures in [16, 28], members of the target concepts were considered as *positive examples*, while an equal number of *negative examples* was randomly sampled from unlabeled examples. Remaining instances (i.e. neither positive nor negative) were considered as *neutral examples*.

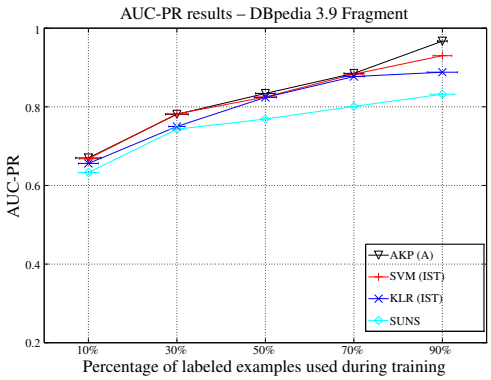
Results are reported in terms of *Area Under the Precision-Recall Curve* (AUC-PR), a measure to evaluate rankings also used in e.g. [17], and calculated using the procedure described in [7]. In each experiment, we considered the problem of predicting the membership to each of several classes; for each of such classes, we performed a 10-fold cross validation (CV), and report the average AUC-PR obtained using each of the considered methods. Since the folds used to evaluate each of the methods do not vary, we report statistical significance tests using a paired, non-parametric difference test (Wilcoxon T test). We also report diagrams showing how using a limited quantity of randomly sampled labeled training instances (i.e. $10\%, 30\%, 50\%, \ldots$, a plausible scenario for a number of real world settings with limited labeled training data), and using the remaining examples for testing, affects the results in terms of AUC-PR.

Setup of the Compared Methods We compared our method with state-of-the-art approaches proposed for learning from ontological KBs. Specifically, we selected two kernel methods: Soft-Margin SVM [23, pg. 223] (SM-SVM) and Kernel Logistic Regression (KLR), jointly with the *Intersection SubTree* [16] (IST) kernel for ontological KBs, and the SUNS [26] relational prediction model. The relational graph used by both the RDF kernel and SUNS was materialized as follows: all $\langle \mathtt{s}, \mathtt{p}, \mathtt{o} \rangle$ triples were retrieved by means of SPARQL-DL queries (where $\mathtt{p}$ was either an object or a data-type property) together with all *direct type* and *direct sub-class* relations.

As in [16], IST kernel parameters were ranging in $d \in \{1, 2, 3, 4\}$ and $\lambda_{ist} \in \{0.1, 0.3, \ldots, 0.9\}$). In order to obtain a ranking among instances (provided by soft-labels $\mathbf{f}$ in AKP), we applied the logistic function s to the decision boundary f instead of the standard sign function, commonly used in the classification context (thus obtaining $s(f(\cdot)) : \mathcal{X} \rightarrow [0, 1]$). In SM-SVM, $C \in \{0.0, 10^{-6}, 10^{-4}, \ldots, 10^{4}, 10^{6}\}$, while in KLR the weight λ_k associated to the L_2 regularizer was found considering $\lambda_k \in \{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$. In SUNS, parameters were selected by means of a 10-fold CV within the training set by grid optimization, with $t \in \{2, 4, 6, \ldots, 24\}$ and $\lambda_s \in \{0, 10^{-2}, 10^{-1}, \ldots, 10^{6}\}$.

4.2 Results

Similarly to [17], we evaluated the proposed approach on two prediction tasks, namely predicting party affiliations to either the Democratic and the Republican party for US



Method	AUC-PR (mean ± var.)	(S)
AKP (A)	0.967 ± 0.003	
SM-SVM (IST)	0.930 ± 0.011	$\triangledown$
KLR (IST)	0.888 ± 0.029	▼
SUNS	0.832 ± 0.019	▼

Fig. 1: DBPEDIA 3.9 Ontology – Left: AUC-PR results (mean, st.d.) estimated by 10-fold CV, obtained varying the percentage of examples used for training – Right: AUC-PR results estimated by 10-fold CV: ▼/$\triangledown$ (resp. ▲/$\triangle$) indicates that AKP's mean is significantly higher (resp. lower) in a paired Wilcoxon T test with $p < 0.05$ / $p < 0.10$

presidents and vice-presidents. The experiment illustrated in [17] uses a small RDF fragment containing the `president` and `vicePresident` predicates only. In this experiment, we used a real-life fragment of DBPEDIA 3.9 (obtained by means of a crawling process), containing a number of irrelevant and possibly noisy entities and relations. Following the procedure in [11], the DBPEDIA 3.9 RDF graph was traversed starting from resources representing US presidents and vice-presidents: all immediate neighbors, i.e. those with a recursion depth of 1, were retrieved, together with their related schema information (direct classes and their super-classes, together with their hierarchy). All extracted knowledge was used to create an $\mathcal{ALCH}$ ontology fragment, with 78795 axioms, 16606 individuals, 132 properties and 11 classes.

In this experiment, 82 individuals representing US presidents and vice-presidents were interlinked by 25 relations represented by atomic roles. The proposed method, denoted as AKP (A), makes use of such atomic roles to identify relations holding among the examples in the ontology.

Experimental results are summarized in Fig. 1. We observe that AUC-PR values obtained with AKP (A) are significantly higher than results obtained by other methods considered in comparison ($p < 0.05$, except for three cases in which $p < 0.10$). Results show how presidents and vice-presidents linked by simple relations such as `president` and `vicePresident` tend to be affiliated to the same political party.

AKP (A) is able to identify which atomic roles are likely to link same party affiliates. As expected, it recognizes that relations represented by the `president` and `vicePresident` atomic roles should be associated to higher weights, which means that presidents and their vice-presidents tend to have similar political party affiliations. AKP (A) also recognizes that presidents (or vice-presidents) linked by the `successor` atomic role are unlikely to have similar political party affiliations.

5 Conclusions and Future Works

In this work, we proposed a semi-supervised transductive inference method for statistical learning in the context of the WEB OF DATA. Starting from the assumption that some relations among entities in a Web ontology can encode similarity information w.r.t. a given prediction task (pertaining a particular property of examples, such as a class-membership relation), we proposed a method (named *Adaptive Knowledge Propagation*, or AKP) for efficiently learning the best way to propagate knowledge among related examples (each represented by an individual) in a Web ontology.

We empirically show that the proposed method is able to identify which relations encode similarity w.r.t. a given property, and that their identification can provide an effective method for predicting unknown characteristics of individuals. We also show that the proposed method can provide competitive results, in terms of AUC-PR, in comparison with other state-of-the-art methods in literature.

We only considered relations between statistical units (i.e. training examples) encoded by atomic roles. However, those do not always suffice: for example, in the research group affiliation prediction task discussed in [16], individuals representing researchers in the AIFB PORTAL ontology are not related by any atomic role. We are currently investigating other approaches to identifying meaningful relations among individuals, for example by means of Conjunctive Query Answering [12]. Other research directions involve the study of different objective functions and optimization methods.

Acknowledgments This work fulfills the objectives of the PON 02_00563_3489339 project "PUGLIA@SERVICE - Internet-based Service Engineering enabling Smart Territory structural development" funded by the Italian Ministry of University and Research (MIUR).

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook. Cambridge University Press (2007)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284(5), 34–43 (May 2001)
3. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the Web of Data. J. Web Sem. 7(3), 154–165 (2009)
4. Bloehdorn, S., Sure, Y.: Kernel methods for mining instance data in ontologies. In: Aberer, K., et al. (eds.) Proceedings of ISWC'07. LNCS, vol. 4825, pp. 58–71. Springer (2007)
5. Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-Supervised Learning. MIT Press (2006)
6. Cohen, M.B., Kyng, R., Miller, G.L., Pachocki, J.W., Peng, R., Rao, A., Xu, S.C.: Solving SDD linear systems in nearly $m\log^{1/2}n$ time. In: Shmoys [24], pp. 343–352
7. Davis, J., Goadrich, M.: The relationship between Precision-Recall and ROC curves. In: Cohen, W., et al. (eds.) Proceedings of ICML'06. pp. 233–240. ACM (2006)
8. Fanizzi, N., d'Amato, C., Esposito, F.: Induction of robust classifiers for web ontologies through kernel machines. J. Web Sem. 11, 1–13 (2012)
9. Franz, T., Schultz, A., Sizov, S., Staab, S.: TripleRank: Ranking Semantic Web Data by Tensor Decomposition. In: Bernstein, A., et al. (eds.) International Semantic Web Conference. LNCS, vol. 5823, pp. 213–228. Springer (2009)

10. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web, Morgan & Claypool Publishers (2011)
11. Hellmann, S., Lehmann, J., Auer, S.: Learning of OWL Class Descriptions on Very Large Knowledge Bases. Int. J. Semantic Web Inf. Syst. 5(2), 25–48 (2009)
12. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
13. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
14. Koutra, D., Ke, T.Y., Kang, U., Chau, D.H., Pao, H.K.K., Faloutsos, C.: Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms. In: Gunopulos, D., et al. (eds.) Proceedings of ECML/PKDD'11. LNCS, vol. 6912, pp. 245–260. Springer (2011)
15. Lin, H.T., Koul, N., Honavar, V.: Learning Relational Bayesian Classifiers from RDF Data. In: Aroyo, L., et al. (eds.) International Semantic Web Conference (1). LNCS, vol. 7031, pp. 389–404. Springer (2011)
16. Lösch, U., Bloehdorn, S., Rettinger, A.: Graph kernels for RDF data. In: Simperl, E., et al. (eds.) Proceedings of ESWC'12. LNCS, vol. 7295, pp. 134–148. Springer (2012)
17. Nickel, M., Tresp, V., Kriegel, H.P.: A Three-Way Model for Collective Learning on Multi-Relational Data. In: Getoor, L., et al. (eds.) Proceedings of ICML'11. pp. 809–816. Omnipress (2011)
18. Ochoa-Luna, J.E., Cozman, F.G.: An Algorithm for Learning with Probabilistic Description Logics. In: Bobillo, F., et al. (eds.) Proceedings of the 5th International Workshop on Uncertainty Reasoning for the Semantic Web, URSW09. CEUR Workshop Proceedings, vol. 654, pp. 63–74. CEUR-WS.org (2009)
19. Peng, R., Spielman, D.A.: An efficient parallel solver for SDD linear systems. In: Shmoys [24], pp. 333–342
20. Rettinger, A., Lösch, U., Tresp, V., d'Amato, C., Fanizzi, N.: Mining the Semantic Web: Statistical learning for next generation knowledge bases. Data Min. Knowl. Discov. 24(3), 613–662 (2012)
21. Rettinger, A., Nickles, M., Tresp, V.: Statistical Relational Learning with Formal Ontologies. In: Buntine, W.L., et al. (eds.) Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, ECML/PKDD'09. LNCS, vol. 5782, pp. 286–301. Springer (2009)
22. Shadbolt, N., Berners-Lee, T., Hall, W.: The Semantic Web Revisited. IEEE Intelligent Systems 21(3), 96–101 (2006)
23. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
24. Shmoys, D.B. (ed.): Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014. ACM (2014)
25. Spielman, D.A.: Algorithms, Graph Theory, and Linear Equations in Laplacian Matrices. In: Proceedings of ICM'10. pp. 2698–2722 (2010)
26. Tresp, V., Huang, Y., Bundschus, M., Rettinger, A.: Materializing and querying learned knowledge. In: Proceedings of IRMLeS'09 (2009)
27. Vapnik, V.N.: Statistical learning theory. Wiley, 1 edn. (Sep 1998)
28. de Vries, G.K.D.: A Fast Approximation of the Weisfeiler-Lehman Graph Kernel for RDF Data. In: Blockeel, H., et al. (eds.) ECML/PKDD (1). LNCS, vol. 8188, pp. 606–621. Springer (2013)
29. Zhang, X., et al.: Hyperparameter Learning for Graph Based Semi-supervised Learning Algorithms. In: Schölkopf, B., et al. (eds.) NIPS. pp. 1585–1592. MIT Press (2006)
30. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In: Fawcett, T., et al. (eds.) Proceedings of ICML'03. pp. 912–919. AAAI Press (2003)

Automated Evaluation of Crowdsourced Annotations in the Cultural Heritage Domain

Archana Nottamkandath[1], Jasper Oosterman[2], Davide Ceolin[1], and Wan Fokkink[1]

[1] VU University Amsterdam, Amsterdam, The Netherlands
{a.nottamkandath,d.ceolin,w.j.fokkink}@vu.nl
[2] Delft University of Technology, Delft, The Netherlands
j.e.g.oosterman@tudelft.nl

Abstract. Cultural heritage institutions are employing crowdsourcing techniques to enrich their collection. However, assessing the quality of crowdsourced annotations is a challenge for these institutions and manually evaluating all annotations is not feasible. We employ Support Vector Machines and feature set selectors to understand which annotator and annotation properties are relevant to the annotation quality. In addition we propose a trust model to build an annotator reputation using subjective logic and assess the relevance of both annotator and annotation properties on the reputation. We applied our models to the Steve.museum dataset and found that a subset of annotation properties can identify useful annotations with a precision of 90%. However, our studied annotator properties were less predictive.

1 Introduction

Cultural heritage institutions have large collections which can be viewed in exhibitions and often are digitised and visible online. For these institutions the metadata of these artefacts (paintings, prints, sculptures etc.) are of the utmost importance. They notably cover the physical properties of the artefact (e.g. dimensions, material), provenance properties (e.g. creator, previous owners) and the subject matter (what is depicted on the artefact). Typically, cultural heritage institutions employ professionals, mostly art historians, who mostly provide high-quality annotations about art-historical properties of artefacts, but tend to lack domain expertise for other aspects such as names of depicted items (of e.g. flowers and birds). With regard to the large scale of collections, their annotation capacity is also limited to describe the subject matter in detail.

Due to these limitations institutions are looking into the knowledge and capacity of crowds. Projects such as Steve.museum [18], Your Paintings [4] and Waisda? [8], are all examples of cultural heritage or media institutions opening up their collection to the crowd for annotation. In these projects institutions engage people from the web in different tasks with the purpose of integrating the

obtained data within their collections. However, employed professional annotators are trained and follow strict guidelines on how to correctly and qualitatively annotate artefacts, to maintain the high quality standards these institutions have. Crowdsourced annotators are not trained in such a way and their quality cannot be guaranteed in a straightforward manner.

Crowdsourced annotations thus need to be assessed, to evaluate whether they meet the institution's quality criteria. However, manually evaluating such a large amount of annotations is likely as expensive as entering the information manually. Thus there is a need to develop algorithms which can automatically or semi-automatically predict the trustworthiness of crowd annotations. The goal of this study is to understand which kinds of properties are important in deciding this trustworthiness, so that in the future suitable annotators can be recruited, or annotation tasks can be tuned in such a way to more likely obtain desired information. The results from this study will thus have implications in the fields of expert finding and task formulation in the domain of crowdsourcing cultural heritage data. In this paper we answer the following research questions:

RQ1: Which annotation properties affect the trustworthiness of crowd-provided annotations?

RQ2: Can an annotator's profile information help in the estimation of annotation and annotator trustworthiness?

In this paper we make use of the Steve.museum dataset [18] containing reviewed annotations on museum objects and information about the annotators such as *age*, *museum and annotation familiarity* and *income*. We propose a trust model for annotator reputation and make prediction models for both annotation usefulness and annotator reputation. The contributions of this paper are: 1) A trust model for reputation based on subjective logic, and 2) insights into the relevance of annotation and annotator properties on the trustworthiness of cultural heritage annotations.

The remainder of the paper is structured as follows. Section 2 compares our work to existing methods. Section 3 describes our methodology and presents the trust model and semantic model. The Steve.museum case study and semantic representation of the data are described in Section 4. Experiments and evaluations are reported in Section 5 and Section 6 provides conclusions of the paper.

2 Related Work

The problem of assessing the trustworthiness of annotations and annotators is not new. There exist several ontologies for representing trust (e.g., those of Golbeck et al. [6] and of Alnemr et al. [1]). While these put emphasis on the social aspects of trust, we are more interested in the trustworthiness of annotations and annotators. Ceolin et al. [2] employed semantic similarity measures, clustering algorithms and subjective logic for the semi-automatic evaluation of annotations in the cultural heritage domain. A probabilistic model, based on a combination of an annotators reputation and the semantic similarity with already labelled

annotations, is used to assess the usefulness of new annotations, achieving 80% correctness. In this paper we take a different approach and employ machine learning algorithms to determine the usefulness of an annotation by using features of both annotator and annotations.

Majority voting [9] is a commonly used method to assess the quality of annotations. However, for domains with a broad vocabulary, such as the cultural heritage domain, this is not optimal. Adapted annotator agreement or disagreement measures have also been studied [11,5], by considering, for example, annotator history and agreement with aggregated label. In contrast, we employ subjective logic to build a user reputation based on his/her positive and negative contributions, and focus more on identifying features about the information and the user that may help to predict his/her trustworthiness.

Task design is also important to achieve qualitative annotations. Test questions or other specialised constructions should be employed to filter out low-quality and spam workers [14] and are necessary to approximate results from experts [15].

Annotation properties have also been studies in the context of Wikipedia [19] and Twitter [17]. Annotation quality has been shown to be related to properties of the annotator. The impact of user information such as *age*, *gender*, *education* and demographics in crowdsourcing tasks have been explored in [13]. They explored the relationship between worker characteristics and their work quality and showed a strong link between them. In this paper we continue in this direction and investigate the relationship between annotation quality and a more extensive set of user properties including *income* and *internet connection speed*.

3 Methodology

In this section we describe the methodology employed in this paper. Our methodology focusses around methods to understand the importance of annotator and annotation properties and is outlined in Algorithm 1. Firstly we identify the features which are relevant for predicting the *value*, in our case the evaluation of the annotation and the reputation of the annotator. Feature identification is done through three different methods: *process analysis*, *extended analysis* and using *feature selection algorithms*. Having identified the sets of features, we perform an independent correlation analysis of each of the identified features with the *value*. We split the dataset into a test and a training set and use the feature sets to predict the *value*. The result of the feature selection methods are then compared.

In Section 3.1 we describe the trust modelling of annotator reputation and in 3.2 we describe the semantic representation of our data model.

3.1 Trust Modelling

The annotation process involves an annotator who is either a user from the crowd or an employee of a cultural heritage institution who provides information about

Algorithm 1: Algorithm to perform predictions based on relevant features

```
Input: A finite set of features F and values used for training
    Input_set ={⟨F,value⟩}
Output: A finite set of relevant features and predicted values
    Output_set ={⟨F_relevant, predicted_value⟩}
F_relevant ← Identify_relevant_features(Input_set)
for F_relevant ← F_relevant_1 to F_relevant_n do
    Compute_correlation(F_relevant, value )
Train_set ← Build_train_set(F_relevant, value)
Test_set ← Build_test_set(F_relevant)
Output_set ← Employ_machine_learning(Train_set, Test_set)
return Output_set
```

digital artefacts. A digital artefact is an image of the actual physical artefact which is published online by the cultural heritage institution. An annotation is information describing some properties of the digital artefact such as what is depicted, who is the artist, etc. A reviewer is a trusted entity, usually an employee of a cultural heritage institution who evaluates the annotation and decides if it is to be accepted or not, based on review policy of the institution.

Aggregating the annotations and their evaluations per annotator helps us understand the reputation of the annotator in the system based on the total number of *useful* and *not useful* annotations. We define reputation of an annotator as a value representing the trustworthiness of a given annotator, based on the evaluation that a cultural heritage institution made of the tags that he or she contributed.

In order to properly model and represent the user expertise and reputation based on the evidence at our disposal, we use a probabilistic logic named subjective logic [12]. It models the truth of propositions as Beta probability distributions that represent both the probability of the proposition to be true (i.e., for instance, the probability of a user to be trustworthy) and the uncertainty about this probability. In subjective logic such a probability distribution is represented by means of the "opinion" (ω) construct. An opinion that a certain *institution* holds with respect to a given *annotator* is represented as follows:

$$\omega_{annotator}^{institution}(belief, disbelief, uncertainty, apriori)$$

where

$$belief + disbelief + uncertainty \;=\; 1, \qquad apriori \in [0...1]$$

and

$$belief = \frac{p}{p+n+2} \qquad disbelief = \frac{n}{p+n+2} \qquad uncertainty = \frac{2}{p+n+2}$$

Here p is the amount of positive evidence (e.g., annotations evaluated as *useful*), n the amount of negative evidence (e.g., annotations evaluated as *not useful*), and *apriori* is the prior knowledge about the reputation, which is set to $\frac{1}{2}$ by

default. The actual value that we use to represent an annotator's reputation is the expected value of the corresponding Beta distribution, that is computed as:

$$E = belief + apriori \cdot uncertainty$$

Subjective logic offers a wide range of operators that allow one to reason upon the evidence at our disposal and infer the reputation based on the different features considered. But we use it merely for a representation purpose. In fact, to apply such operators we would need to know a priori the kind of relations that occur between the features that we identify and the reputation. These relations will instead be discovered by means of a machine learning approach.

We use subjective logic to model both annotator and annotation reputations by means of the expected value E. In the case of the annotators, we collect evidence about them (i.e. reviews of the tags they contributed) and we estimate their reputations by means of the subjective opinions described above. In the case of annotation reputations, we use the expected value E to model them, but their prediction is made by means of the machine learning methods.

3.2 Semantic Modelling

We adopt semantic web technologies for representing the annotations and the related metadata. This is done for two reasons. First, they provide a uniform layer that allow us interoperability and prevents us from relying on the specific structure such as relational databases. Second, they provide a means to possibly share metadata and computation results in such a manner that other institutions could benefit from them, thus promoting the sharing of possibly precious information (precious both because of their specificity and of their quality).

A (crowd) annotator performs an annotation task. The annotator's features (e.g., age, country, education) are as much as possible represented by means of the standard FOAF ontology [3], while the annotation is represented by means of the Open Annotation Model [16].

The annotation entered by the user is reviewed by an employee of the cultural heritage institution. The annotation evaluation is yet again represented by means of the Open Annotation Model, as an annotation of the first annotation. All the features we adopt in our computation that are not representable by means of standard vocabularies are represented by means of an ad-hoc construct ("ex:" prefix). An illustration of the annotation (and related metadata) representation is provided in Figure 1, where it is also indicated that we use annotator and annotation features as a basis for estimating the value of an annotation evaluation.

4 Cultural Heritage Annotations: Steve.museum

The Steve.museum [18] dataset was created by a group of art museums with the aim to explore the role that user-contributed descriptions can play in improving

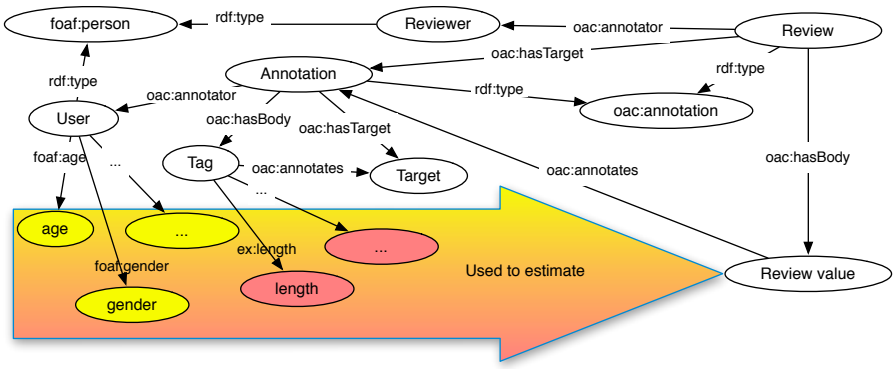


Fig. 1: Representation of an annotation and of the related metadata.

on-line access to works of art. Annotations were gathered for 1,784 artworks and the usefulness, either *useful* or *not useful*, of each annotation was evaluated by professional museum staff. The annotations including their evaluations and annotator information were published as a dataset to study[3].

We performed two pre-processing steps on the data. First, for the correct calculation of the annotator reputation we need at least five annotations per annotator and as such removed data from annotator with fewer annotations. It also occurred that multiple reviewers evaluated the same annotation. For those annotations we took the majority vote of the evaluations. In case of a tie we always chose *useful*, giving more weight to a potentially useful annotation.

The dataset contains both anonymous (730) and registered (488) annotators. Table 1 lists the annotator properties and the percentage of registered annotators who filled in each property. The distribution of the number of annotations per annotator follows a power law. The majority of the annotations (87%) were evaluated as useful. Considering other crowdsourcing initiatives this was a remarkably good crowd. Table 2 provides a summary of the complete dataset.

Table 1: Annotator properties and the percentage of registered annotators who filled in the property.

Community	**Experience**	**Education**	**Age**	**Gender**	**Household income**
431 (88%)	483 (99%)	483 (99%)	480 (98%)	447 (92%)	344 (70%)
Works in a museum	**Involvement level**	**Tagging experience**	**Internet connection**	**Internet usage**	
428 (88%)	411 (84%)	425 (87%)	406 (83%)	432 (89%)	

[3] `http://verne.steve.museum/steve-data-release.zip`

Table 2: Summary of the Steve.museum dataset.

Number of annotators / Registered	1,218 / 488 (40%)
Provided tags	45,733
Unique tags	13,949
Tags evaluated as *useful*	39,931 (87%)
Tags evaluated as *not useful*	5,802 (13%)

5 Evaluation

Annotations in the Steve.museum dataset have been assessed as either *useful* or *not useful*. Each annotator has a *reputation* score using the model described in Section 3.1. Using machine learning techniques, we aim to automatically predict the evaluation of the annotations based on features of annotators and annotations. Next to that we aim to predict the reputation of the annotator based on the annotator features. The first subsection describes the setup and tooling of our experiments. Section 5.2 contains the results of analysing the relation between annotation properties and *usefulness* of annotations and Section 5.3 between annotation properties and both annotation evaluation and user reputation.

5.1 Experimental Setup

In order to perform fair training, we randomly selected 1000 *useful* and 1000 *not useful* annotations as training set. The remainder of the dataset was used as test set. We used a Support Vector Machine (Sequential Minimal Optimisation[4], default PolyKernel[5]) on selected features to predict annotation usefulness, since that algorithm works for dichotomous variables, and is commonly used, fast and resistant against over-fitting. For prediction of the *reputation* of a user (an interval variable) we used a similar algorithm but adapted for regression. For automated selection of relevant features we used correlation-based feature subset selection [7]. This algorithm selects subsets of features that are highly correlated with the prediction class but have a low inter-correlation.

To calculate an independent correlation between different types of variables we used appropriate statistical tests; *Biserial* for interval, ordinal and nominal against dichotomous variables followed by *Wilcoxon rank sum* for ordinal and *Chi squared* for nominal; *Fisher's exact test* for two dichotomous variables; *Kendall* τ for ordinal against interval variables; and *Pearson* for both two interval variables and nominal against interval variables. Fisher's exact test signals a strong correlation above a score of 1.0.

[4] We used the implementation inside the tool WEKA `http://cs.waikato.ac.nz/ml/weka/`.

[5] There are specific kernels targeting RDF data, but these were, for simplicity reasons, not used.

5.2 Predicting Annotation Evaluation Using Annotation Features

Features Selection. We manually analysed the annotations in different evaluation categories of the Steve.museum so as to understand the evaluation policies depicted as ***F_ man***. From our observations, we found out that some of the evaluations were strongly influenced by certain features of the annotation. Annotations that did not describe something actually depicted, for example sentimental annotations such as "happy", were evaluated as *not useful*. Adjectives in general were not deemed useful. Also annotations in non-English languages or misspelled words were evaluated as *not useful*. To detect these problems we created the features *is_ adjective*, *is_ english* and *in_ wordnet*, where the latter signals a correctly spelled word. For detecting the language of a tag we used the n-gram based language detection from [10]. For detecting the adjective and spelling errors we used Wordnet,[6] where words not in Wordnet are treated as incorrectly spelled. For multi-words annotations we assessed whether either of the words matched the criteria. We explored the possibilities to extract more features which might be indicative of the evaluation of the annotation represented as ***F_ all*** . We regarded the creation time (both day and hour) of the annotation, how specific the annotation was (based on the depth a word occurs at in the Wordnet tree), the length and number of words of the annotation, and the frequency with which the annotation was created for the same object.
We applied the feature selection algorithm to the features from ***F_ all*** on the annotation data resulting in the feature set ***F_ ml***.
F_ man = *[is_ adjective, is_ english, in_ wordnet]*
F_ all = ***F_ man*** + *[created_ day, created_ hour, Length, Specificity, nrWords, Frequency]*
F_ ml = *[created_ day, in_ wordnet, Frequency]*

Independent correlation of annotation features. We performed an independent correlation analysis of the mentioned features with regard to the *evaluation* of the annotation. We observed a strong correlation (3.34, using Fisher's exact test) for *in_ wordnet*, significant at <0.01. We observed a weak correlation for *Specificity* (-0.11), *Frequency* (0.14), *is_ adjective*(0.67, Fisher) and *is_ English* (0.94, Fisher, not statistically significant).

Predicting annotation evaluation. Table 3 lists the precision, recall and F-measure of the three feature sets. We observe that the precision is high, ranging from 0.90 to 0.978 in all the cases of classifying *useful* annotations. All three methods for creating the feature sets result in a model that can predict *useful* annotations very well. However, the recall is high only for the feature set ***F_ man***, while the predictions using feature sets ***F_ all*** and ***F_ ml*** had a high number of false positives.

None of the classifiers performed well in predicting the annotations which were classified as *not useful*. There was a large number of false positives and the

[6] We used the NLTK library (`http://nltk.org/`) to query the Wordnet tree.

precision was very low in all cases, ranging from 0.13 to 0.21. Thus from our analysis we can observe that although the machine learning classifier using the three different features were comparably successful in identifying *useful* annotations, neither of them succeeded in identifying the *not useful* annotations.

Table 3: Comparison of results from SVM predictions using annotation features.

Feature set	**Class**	**Precision**	**Recall**	**F-measure**
F_man	useful	0.90	0.90	0.90
	not useful	0.21	0.20	0.20
F_all	useful	0.91	0.75	0.83
	not useful	0.18	0.42	0.25
F_ml	useful	0.98	0.20	0.34
	not useful	0.13	0.96	0.23

5.3 Predicting Annotation Evaluation And User Reputation Using Annotator Features

Feature Selection. The set ***F_man*** is based on the annotator properties listed in Table 1. Apart from the provided features for an annotator, we also compute certain features related to the annotations they provided, which may be useful for predicting the *evaluation* of an annotation. The computed features are the total number of annotations entered by the user (*#Annotations*), the vocabulary size and diversity of the annotator, and the number of matched annotations in Wordnet (*#matched_in_wordnet*). The vocabulary size of an annotator is the number of distinct annotations after stemming has been applied. The vocabulary diversity is computed as the vocabulary size divided by the total number of annotations provided by that annotator. The definition of vocabulary diversity is reasonable in view of the fact that the number and length of annotations is relatively small in Steve.museum dataset.
Two sets are obtained when the feature selection algorithm is applied in two instances, one to identify relevant features for the annotation evaluation, represented as ***F_ml_a***, and in the second case to identify relevant features for annotator reputation, represented as ***F_ml_u***. For the prediction of the annotation evaluation, we merged the annotation data with the corresponding annotator properties and performed a prediction of annotation evaluation. We applied the feature selection algorithm to the features from ***F_all*** on the annotation data (***F_ml_a***) and on the user data (***F_ml_u***) resulting in the following features.
F_man = [Features in Table 1]
F_all = [**F_man**, *vocabulary_size*, *vocabulary_diversity*, *is_anonymous*, *#Annotations_in_wordnet*]
F_ml_a = [*vocabulary_size, vocabulary_diversity*]
F_ml_u = [*Language*, *Education*, *Community*, *#tags_wordnet*, *Tagging_experience*]

Independent correlation analysis of annotator features. A statistical correlation analysis was performed to determine the relationship between the annotator features with the annotation reputation and annotation evaluation as shown in Table 4. For the annotation evaluation, Experience, Education, Tagging Experience, Internet connection and Internet usage had a weak correlation that was statistically significant. However, Community had a higher correlation compared to the other features. For the annotator reputation, the computed features such as *# Annotations, vocabulary size* and *#Annotations in Wordnet* were considered significant.

Table 4: Correlation of features with annotation evaluation and annotation reputation. In brackets the statistical test (See Section 5.1). * indicates significance at $p < 0.01$. Note: Fisher signals a high correlation for values > 1.

Annotator feature	**Correlation score Annotation evaluation**	**Correlation score Annotator reputation**
Community	0.22* (C+B)	0.22 (P)
Experience	0.02* (W+B)	0.02 (K)
Education	0.02* (W+B)	0.01 (K)
Age	0.01 (B)	-0.16 (P)
Gender	1.11 (F)	-0.004 (B)
Household income	-0.14 (W+B)	-0.14 (K)
Works in a museum	0.99 (F)	-0.34 (B)
Involvement level	0.04* (W+B)	-0.10 (K)
Tagging experience	1.22* (F)	-0.08 (B)
Internet connection	0.02* (W)	0.06 (K)
Internet usage	0.02* (W)	-0.16 (K)
# Annotations	-0.06 (B)	0.27* (P)
Vocabulary size	-0.06 (B)	0.27* (P)
Vocabulary diversity	0.05 (B)	-0.03 (P)
# Annotations in Wordnet	-0.08 (B)	0.31* (P)

Predicting annotation evaluation and annotator reputation. From Table 5 we can see that the features identified from the annotator profile and those identified by the feature selection algorithm are useful in classifying *useful* annotations and have a high precision of 0.91. However, these methods also have lower values of recall, indicating a high number of false negatives. Both methods have a low precision and recall in classifying *not useful* annotations, and thus are not successful in predicting *not useful* annotations.

We used a SVM for regression to estimate the reputation of the annotator since it was hard to perform a classification for reputation. This is because the reputation is highly right skewed with 90% of the annotators having a reputation > 0.7. This makes it hard to classify data and distinguish the classes when the distribution is highly skewed. Another point is that classification of reputation is highly use case dependent. Upon performing regression on the reputation, as shown in Table 6, we can observe that all the predictions have a very high

relative absolute error and low coefficients. Another observation is that relative weights assigned to the *#Annotations in Wordnet* feature are relatively high, showing consistency with our earlier analysis.

Table 5: Comparison of results from SVM predictions using annotator features.

Feature set	**Class**	**Precision**	**Recall**	**F-measure**
F_man	useful	0.90	0.29	0.44
	not useful	0.11	0.73	0.20
F_all	useful	0.91	0.69	0.78
	not useful	0.15	0.43	0.22
F_ml_a	useful	0.91	0.55	0.68
	not useful	0.13	0.53	0.21

Table 6: Comparison of results from predicting annotator reputation using SVM regression and 10-fold cross validation.

Feature set	**corr**	**Mean abs error**	**Root mean sq error**	**Rel abs error**
F_man	−0.02	0.10	0.15	97.8%
F_all	0.22	0.09	0.13	95.1%
F_ml_u	0.29	0.09	0.13	90.4%

6 Conclusion and Future Work

In this paper we described methods which can automatically evaluate annotations. The experiment was performed on the Steve.museum dataset and investigated the effect of annotation and annotator properties in predicting trustworthiness of annotations and reputation of annotator. We also devised a model using Support Vector Machines for predicting annotation evaluation and annotator reputation. Presence of an annotation in Wordnet is shown to be indicative for the perceived usefulness of that annotation. With a small set of features we were able to predict 98% of the *useful* and 13% of the *not useful* annotations correctly. The annotator reputation was computed using a model in subjective logic. Since the reputation of annotators is highly skewed in this dataset(with more than 90% having a reputation > 0.7), we could not make successful estimations of reputation from annotator profiles.

As part of future work, we would like to repeat the experiment on other cultural heritage datasets. We would also like to build a reputation for an annotator based on topics of expertise, to obtain more accurate correlations between the semantics of the annotation and the topical reputation of the annotator. Our analysis also indicated that there is relevance in aspects related to creation time of an annotation. A more sophisticated model, such as whether an annotation was created during work or during free-time might increase the predictive power.

Acknowledgements This publication was supported by Data2Semantics and SEALINCMedia projects from the Dutch National program COMMIT.

References

1. Alnemr, R., Paschke, A., Meinel, C.: Enabling reputation interoperability through semantic technologies. In: I-SEMANTICS. pp. 1–9. ACM (2010)
2. Ceolin, D., Nottamkandath, A., Fokkink, W.: Efficient semi-automated assessment of annotation trustworthiness. Journal of Trust Management 1, 1–31 (2014)
3. Dan Brickley, L.M.: FOAF. `http://xmlns.com/foaf/spec/` (Jan 2014)
4. Ellis, A., Gluckman, D., Cooper, A., Greg, A.: Your paintings: A nation's oil paintings go online, tagged by the public. In: Museums and the Web 2012. Online (2012)
5. Georgescu, M., Zhu, X.: Aggregation of crowdsourced labels based on worker history. In: Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14). pp. 37:1–37:11. WIMS '14, ACM (2014)
6. Golbeck, J., Parsia, B., Hendler, J.A.: Trust networks on the semantic web. In: CIA. pp. 238–249. Springer (2003)
7. Hall, M.A.: Correlation-based Feature Subset Selection for Machine Learning. Ph.D. thesis, University of Waikato, Hamilton, New Zealand (1998)
8. Hildebrand, M., Brinkerink, M., Gligorov, R., van Steenbergen, M., Huijkman, J., Oomen, J.: Waisda?: Video labeling game. In: Proceedings of the 21st ACM International Conference on Multimedia. pp. 823–826. MM '13, ACM (2013)
9. Hirth, M., Hossfeld, T., Tran-Gia, P.: Cost-optimal validation mechanisms and cheat-detection for crowdsourcing platforms. In: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on. pp. 316–321 (June 2011)
10. Hornik, K., Mair, P., Rauch, J., Geiger, W., Buchta, C., Feinerer, I.: The textcat package for n-gram based text categorization in R. Journal of Statistical Software 52(6), 1–17 (2013)
11. Inel, O., Aroyo, L., Welty, C., Sips, R.J.: Domain-independent quality measures for crowd truth disagreement. Journal of Detection, Representation, and Exploitation of Events in the Semantic Web pp. 2–13 (2013)
12. Jøsang, A.: A logic for uncertain probabilities. Intl. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(3), 279–212 (2001)
13. Kazai, G., Kamps, J., Milic-Frayling, N.: The face of quality in crowdsourcing relevance labels: Demographics, personality and labeling accuracy. pp. 2583–2586. CIKM '12, ACM (2012)
14. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with mechanical turk. pp. 453–456. CHI '08, ACM (2008)
15. Nowak, S., Rüger, S.: How reliable are annotations via crowdsourcing: A study about inter-annotator agreement for multi-label image annotation. pp. 557–566. MIR '10, ACM (2010)
16. Sanderson, R., Ciccarese, P., de Sompel, H.V., Clark, T., Cole, T., Hunter, J., Fraistat, N.: Open annotation core data model. Tech. rep., W3C Community (May 9 2012)
17. Suh, B., Hong, L., Pirolli, P., Chi, E.H.: Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In: Social Computing (SocialCom), 2010 IEEE Second International Conference on. pp. 177–184. IEEE (Aug 2010)
18. Trant, J.: Tagging, folksonomy and art museums: Early experiments and ongoing research. J. Digit. Inf. 10(1) (2009)
19. Warncke-Wang, M., Cosley, D., Riedl, J.: Tell me more: An actionable quality model for wikipedia. In: Proceedings of the 9th International Symposium on Open Collaboration. pp. 8:1–8:10. WikiSym '13, ACM (2013)

Probabilistic Relational Reasoning in Semantic Robot Navigation

Walter Mayor Toro[1], Fabio G. Cozman[1], Kate Revoredo[2], and Anna Helena Reali Costa[1]

[1] Escola Politécnica, Univ. de São Paulo
Av. Prof. Luciano Gualberto, trav.3, 158, 05508-970 São Paulo, SP, Brazil
{walter.mayortoro,fgcozman,anna.reali}@usp.br
[2] Depto. de Informática Aplicada, Univ. Federal do Estado do Rio de Janeiro
Rio de Janeiro, RJ, Brazil
katerevoredo@uniriotec.br

Abstract. We examine the use of semantic web resources in robot navigation; more specifically, in qualitative navigation where uncertain reasoning plays a significant role. We propose a framework for robot navigation that connects existing semantic web resources based on probabilistic description logics, with probabilistic relational learning and planning. We show the benefits of this framework in a real robot, presenting a case study on how semantic web resources can be used to face sensor and mapping uncertainty in a practical problem.

Keywords: Semantic robotics, KnowRob system, probabilistic description logics, Bayesian networks.

1 Introduction

Recent experience has shown that applications in robotics can benefit from semantic information carrying commonsense facts [1, 3, 12, 13] One particular example of semantic knowledge system for robotics is the KnowRob package (Knowledge Processing for Autonomous Personal Robots) [15, 16]. KnowRob operates on ontology databases such as OMICS (indoor common-sense knowledge database) [4], mixing description logics [2] and Bayesian networks [11].

However, it is not always easy to effectively bring these semantic web resources into practical use, as it is necessary to combine semantic information and low-level data, and to handle uncertain sensors and incomplete maps. In this paper we propose a framework for qualitative robot navigation that uses the probabilistic description logic knowlege base in KnowRob to learn and reason at a relational level. We explore a scheme where higher level descriptions are used to reason at an abstract level. This has important advantages. First, it saves computation as it handles sparser representations. Second, it is a perfect match to the level at which knowledge is stored (that is, relations are used throughout). Third, the use of information in a higher level of abstraction allows

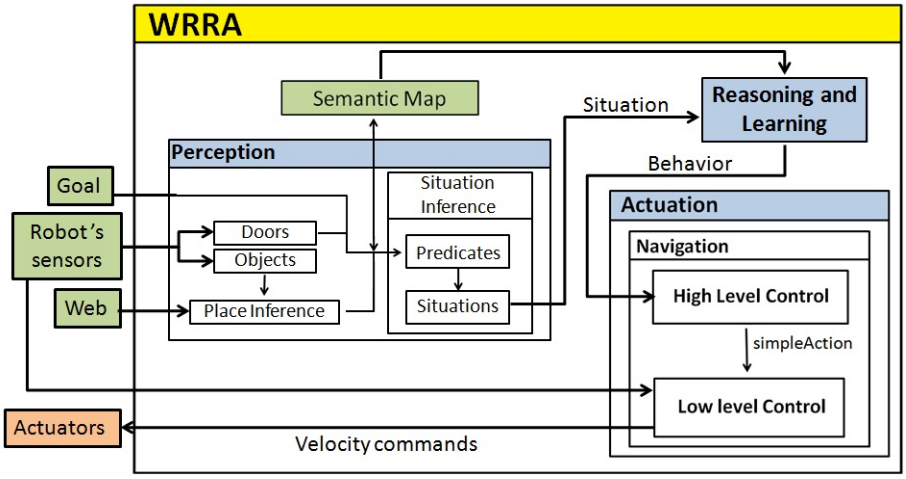


Fig. 1. Overview of the Web-based Relational Robotic Architecture.

knowledge to be generalized and transferred to other agents or used in similar tasks.

We also describe an implementation with a real robot that demonstrates how semantic web resources can work within applications that demand substantial uncertain reasoning. We present our knowledge representation strategy, our sensor gathering blocks, and our ground and abstract reasoning modules. Overall, our goal is to contribute with a case study on how semantic web resources can be used in practice. The paper is organized as follows. In Section 2 we give an overview of the different subsystems that our proposal combines. Section 3 presents the implementation of the system and our experiments. Section 4 concludes the paper.

2 A Framework for Robot Navigation

We consider a robot with three important sources of information. First, sensors that detect objects in the environment. Second, a map with semantic information, that is updated during navigation. Third, and most important, a web database with commonsense knowledge (for example, likely location of objects in rooms). In our framework, information processing flows through three major modules: Perception, Reasoning and Learning, and Actuation. We call the whole architecture by *Web-based Relational Robotic Architecture* (WRRA), as depicted in Figure 1. From the perspective of uncertain reasoning with semantic web resources, the Perception module (Section 2.1) is the most significant contribution of this paper. The other two modules are only briefly described as relevant information can be found in previous publications.

2.1 Perception: semantic information and probabilistic reasoning

This module receives a description of the *goal*. In the case study of interest here, the goal is to find a *target room* inside a house. The Perception module receives sensory information (detected objects), and must abstract the data into compact symbolic representations. Upon receiving data, the robot accesses its Semantic Web resources to determine the most likely room that generated the data, as described in this section.

Unlike most existing robotic systems, we pursue reasoning at a high level of abstraction, employing concepts, roles and relations between then as expressed within KnowRob. It is due to this decision that we can effectively employ semantic web resources. The *situation* of the robot is described in terms of a relational representation that not only allows for abstraction of metric and sensory details, but also enables knowledge to be generalized and reused in new tasks. During navigation, the robot uses sensor data to build a relational representation of the environment (the semantic map).

The output of the Perception module is a description of the robot's situation, which is specified by a conjunction of active predicates (with truth value TRUE) such as: `seeDoor`() that indicates that the robot sees one or more doors in the room; `seeNonVisitedDoor`($\mathtt{d}_1$), meaning that the robot sees door $\mathtt{d}_1$ that has not yet been visited; `inTargetRoom`(), which indicates that the target room is where the robot is; `nonTargetRoom`($\mathtt{p}_1$), meaning that the robot is in $\mathtt{p}_1$ and it is not the target room; `inRoom`($\mathtt{p}_1$) that indicates that $\mathtt{p}$ is the most likely room where the robot is; and others. The truth value of `inRoom`($\mathtt{p}$) is computed by *Place Inference* block, as we explain now.

The Perception module is heavily based on reasoning facilities available in the KnowRob package. The knowledge base in KnowRob uses rdf triples to represent a large ontology, with relationships between objects such as Drawer, a subclass of StorageConstruct, or Refrigerator – Freezer, a subclass of FurniturePiece [16]. Additionally, sentences in OWL indicate relationships between objects. Sentences are attached to probabilities, and for inference they are grounded into Bayesian networks using facilities in the ProbCog system [5].

Just as an example of rdf triple in the knowledge base, consider the fact, contained in the OMICS database, that a kitchen contains a refrigerator (`XXX` denotes the string `http://ias.cs.tum.edu/kb/knowrob.owl`):

```
<rdf:Description rdf:about="XXX#OmicsLocations-1">
    <ns1:object rdf:resource="XXX#Kitchen"></ns1:object>
    <ns1:subject rdf:resource="XXX#Refrigerator"></ns1:subject>
    <rdf:type rdf:resource="XXX#OmicsLocations"></rdf:type>
</rdf:Description>
```

The Perception module queries KnowRob, which returns, for each observed object, the probability that the location is each possible room, given the observed object. Queries are sent to KnowRob through Prolog sentences via function calls in the Python language; as an example, consider (a complete query is given in Section 3):

```
for obj in DetectedObjects:
    q="bayes_probability_given(knowrob:'OmicsLocations',
          Room,knowrob:'"+obj+"',Pr)"
    query = prolog.query(q)
```

Such a query returns probabilities such as

```
Room = 'knowrob.owl#Kitchen'
Pr = 0.1031101853182014 ;
```

That is, given a perceived object o_i, KnowRob uses inference with its probabilistic description logic [8, 7] to return $P(r_j|o_i)$ for each room r_j. The problem now is to combine these pieces of information into a probability that the robot is in room r_j, given *all* detected objects $o_1, \ldots, o_n$. We have:

$$\begin{aligned} P(r_j|o_1,\ldots,o_n) &= \frac{P(o_1,\ldots,o_n|r_j)P(r_j)}{P(o_1,\ldots,o_n)} \\ &= \frac{P(o_1|r_j,o_2,\ldots,o_n)P(o_2|r_j,o_3,\ldots,o_n)\ldots P(o_1|r_j)P(r_j)}{P(o_1,\ldots,o_n)}. \end{aligned}$$

We now assume that, given r_j, an observation (of an object) is independent of other observations (of other objects in the same room). Hence:

$$\begin{aligned} P(r_j|o_1,\ldots,o_n) &= \frac{P(o_1|r_j)P(o_2|r_j)\ldots P(o_1|r_j)P(r_j)}{P(o_1,\ldots,o_n)} \\ &= \frac{(P(r_j|o_1)P(o_1)/P(r_j))\ldots(P(r_j|o_n)P(o_n)/P(r_j))P(r_j)}{P(o_1,\ldots,o_n)} \\ &= \left(\prod_{i=1}^{n} P(r_j|o_i)\right)\frac{\prod_{i=1}^{n} P(o_i)}{P(o_1,\ldots,o_n)(P(r_j))^{n-1}}. \end{aligned}$$

We now introduce a substantive assumption, namely, that every room has identical a priori probability $P(r_j)$. So, $P(r_j|o_1,\ldots,o_n)$ is proportional to $\prod_{i=1}^{n} P(r_j|o_i)$. Once the Perception module gets, for each room, each term of this product from KnowRob, it compares each room with respect to this product, setting the truth value of `inRoom(p)` as TRUE for: $\mathtt{p} = \arg\max_{r_j} \prod_{i=1}^{n} P(r_j|o_i)$, and FALSE otherwise.

During navigation, a semantic map of the environment is created. Each visited room and each observed object are represented as vertices of a graph that describes the topological map (left side of Figure 2). Connectivity between rooms is represented by graph edges, which are defined through doors conecting the rooms. While this topological map is built, edges are created by connecting vertices of the topological map to vertices of the conceptual map (right side of Figure 2). Unlike other approaches [1, 3], our map does not involve metric representation of the environment. Still, our semantic map inserts probabilistic information in the representation. Every inference and reasoning in WRRA occurs at the level of objects, rooms and relationships and properties thereof.

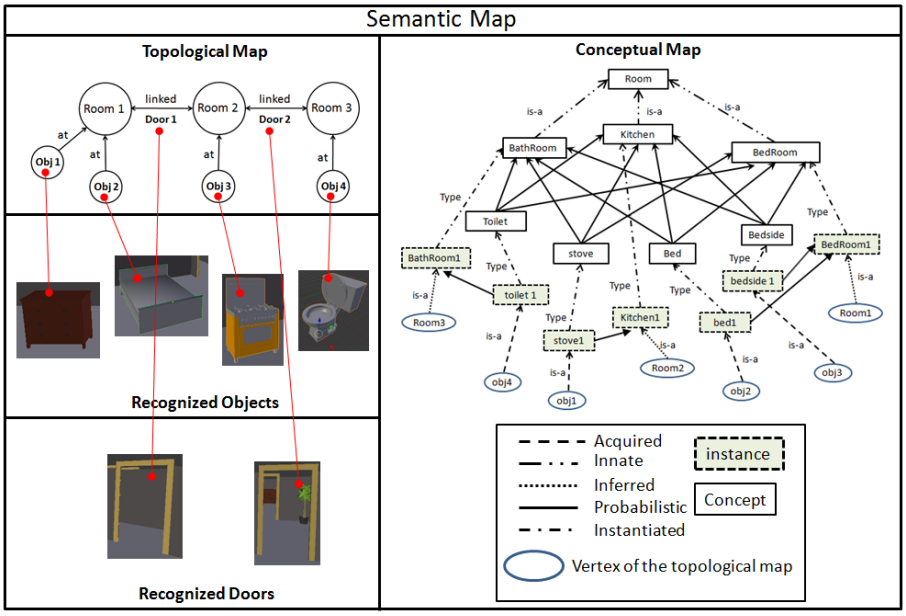


Fig. 2. The semantic map built by WRRA.

2.2 Reasoning and Learning, and Actuation

The WRRA uses reinforcement learning (RL) to refine behavior through interactions with the environment [14]. Typical RL solutions learn from scratch; we instead employ two levels of RL [6], where an abstract and a ground policy are learned simultaneously. The stochastic abstract policy learned in a source task is then used in new similar tasks. Our robot navigation problem is modeled as a Relational Markov Decision Process (RMDP) [10], in which situations $s \in \mathcal{S}$ are represented as a conjunction of predicates describing properties of and relations among objects, such as: $s_1 = \texttt{inRoom}(\texttt{livingroom}) \wedge \texttt{nonTargetRoom}(\texttt{livingroom}) \wedge \texttt{seeNoDoors}() \wedge \texttt{notAllDoorsVisited}()$. Other formalisms are possible to represent decisions and transitions [9].

A conjunction is a *ground conjunction* if it contains only ground atoms (such as s_1 given in the example). In our discussion each variable in a conjunction is implicitly assumed to be existentially quantified. An *abstract situation* σ (and *abstract behavior* α) is a conjunction with no ground atom. A relational representation enables us to aggregate situations and behaviors by using variables instead of constants in the predicate terms. For example, ground situation s_1 is covered by abstract situation σ by replacing *livingroom* with variable X; in this case, other situation could also be covered by σ, e.g., $s_1 = \texttt{inRoom}(\texttt{kitchen}) \wedge \texttt{nonTargetRoom}(\texttt{kitchen}) \wedge \texttt{seeNoDoors}() \wedge \texttt{notAllDoorsVisited}()$.

Denote by $\mathcal{S}_\sigma$ the set of ground situations $s \in \mathcal{S}$ covered by abstract situation σ. We assume that each ground situation s is abstracted to only one abstract situation σ. Similarly, we define $\mathcal{A}_\alpha(s)$ as the set of all ground behaviors $a \in \mathcal{A}$ covered by an abstract behavior α in ground situation s. We also define $\mathcal{S}_{ab}$ and $\mathcal{A}_{ab}$ as the set of all abstract situations and the set of all abstract behaviors in an RMDP, respectively. To simplify notation, here we use the assumption that if an atom does not appear in a ground sentence, the negated atom is assumed.

To solve an RMDP is to find an *optimal policy* π^* that maximizes a function R_t of future rewards. In RL tasks the agent does not know the dynamics of the process and a series of RL algorithms can be used to find a policy [14]. To translate from ground to abstract level, we define two operations: *abstraction* and *grounding*. Abstraction is the translation from the ground level (perceived by the robot's sensors) to the abstract level by replacing constants with variables, $\phi_s : \mathcal{S} \to \mathcal{S}_{ab}$. For a ground situation s, the corresponding abstract situation σ is given by $\phi_s(s) = \sigma$ so that $s \in \mathcal{S}_\sigma$. Grounding is the translation from the abstract level to the ground level, $a = \mathit{grounding}(\alpha, s)$. Clearly only ground states are sensed and visited by the robot, and only ground actions can be actually applied. Laerning and reasoning must proceed by processing, at time t, the (ground) experience $\langle s_t, a_t, r_t, s_{t+1}, a_{t+1}\rangle$, which is related to the tuple $\langle \sigma_t, \alpha_t, r_t, \sigma_{t+1}, \alpha_{t+1}\rangle$.

We propose the following scheme to apply an abstract policy in a ground problem. Consider a stochastic abstract policy defined as $\pi_{ab} : \mathcal{S}_{ab} \times \mathcal{A}_{ab} \to [0, 1]$. After the abstract situation $\sigma = \phi_s(s)$ is derived from the observed ground situation s, a transferred abstract policy (learned from source tasks) yields probabilities $\pi_{ab}(\sigma, \alpha_k) = P(\alpha_k|\sigma)$ for all $\alpha_k \in \mathcal{A}_{ab}$. We select an abstract behavior $\alpha_k \in \mathcal{A}_{ab}$ according to these probabilities. Then the process remains the same, with $a = \mathit{grounding}(\alpha_k, s)$.

Thus, in our system, the robot initially receives an abstract policy and applies it. As its knowledge about the new environment increases, due to its perception and action in the environment, the robot creates and improves a semantic map, which places restrictions on the actions defined by the policy initially received, adapting it to the new environment and to the new task. For example, consider the robot identifies it is in the living room, which is not the target room, and the living room has two doors, d_1 and d_2. The abstract policy indicates that it can randomly choose any one of the two doors and go through it, hoping to reach the target room. Assume the robot circulates in other rooms, after going through the chosen door, say d_1, and represents what is discovered about the environment in a semantic map. Upon returning to the living room without having reached the target room, the reasoning process now indicates that it should choose another door (d_2).

Finally, the Actuation module is divided into a High Level Control (HLC) and Low Level Control (LLC). HLC receives a behavior selected by the Reasoning and Learning module. The behavior is divided into simple actions that can be executed by specific hardware modules. Each simple action is sent to LLC, to be actually executed. Low-level commands are issued by the Actuation module.

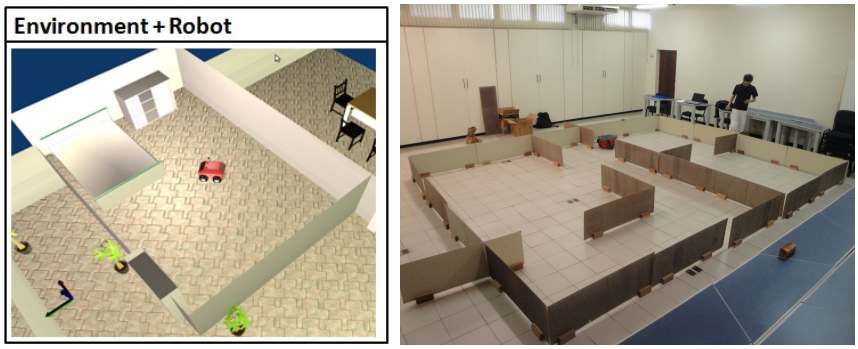


Fig. 3. *Left*: Simulated house. *Right*: Experimental setup with real robot.

3 Implementation and Discussion

We now describe our implementation and experiments. The robot we consider is a wheeled base equipped with 5 sensors: 3 semantic cameras, 1 odometer and 1 laser range scanner. We run tests both in a simulated environment and with the real robot. The simulated scenario (see Figure 3-*Left*) was designed with the open source tool for 3D creation, BLENDER[3], with which we have created a 3D CAD representation of the house and the objects it contains, including the robot (an ATRV 4-wheeled base); the representation was integrated into the MORSE simulator[4] and the Robot Operating System (ROS)[5]. The environment is a house that has eight types of rooms: 1 hallway (with some potted plants), 1 kitchen (with 1 stove, 1 fridge, and a dishwasher), 1 living room (with 1 sofa and 2 armchairs), 1 bathroom (with 1 toilet and 1 sink), 3 bedrooms (with 1 bed and 1 bedside), and 1 dining room (with 1 table and 6 chairs). The real robot is a Pioneer 2DX, and with the real robot we used QR codes to identify doors and objects, so as to obtain functionality similar to a semantic camera.

The semantic camera is, in essence, a sensor that allows to recognize objects that the robot sees and the relative position between the robot and objects viewed. The robot was equipped with two semantic cameras that recognize general objects and one semantic camera that recognizes only doors. The odometer and the laser scanner are used in the Navigation module.

For a better understanding of how the architecture WRRA works, how is its integration with the information of the semantic web and the technologies employed, we describe a simple case study executed in the simulated scenario, where we have great flexibility in defining tasks and measuring behavior. WRRA was implemented using the Python programming language and was integrated with the ROS framework. Initially, the robot knows nothing about the house

[3] http://www.blender.org/
[4] http://www.openrobots.org/wiki/morse/
[5] http://wiki.ros.org/

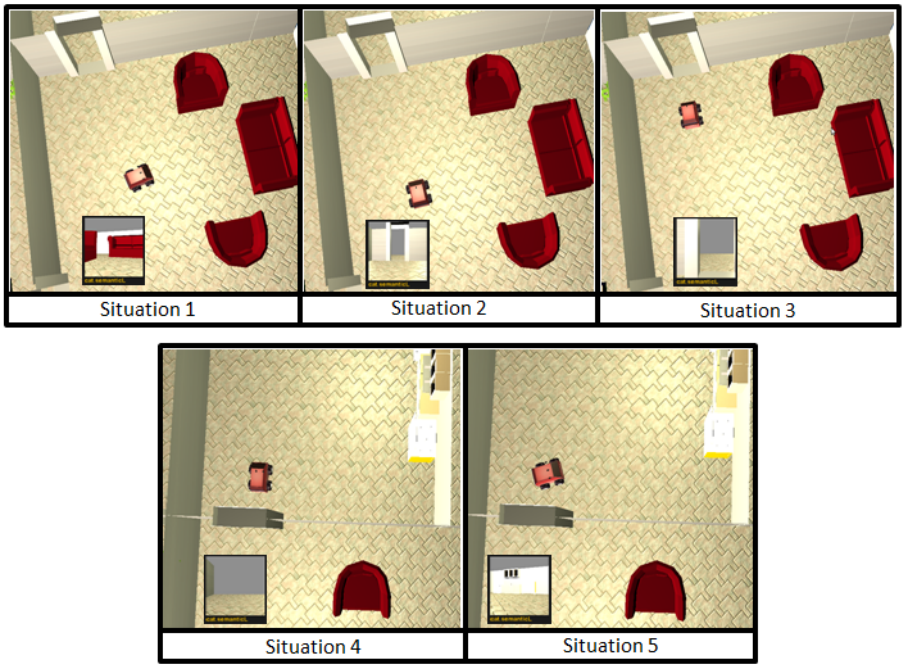


Fig. 4. Sequence of situations faced by the mobile robot in case study.

and has only a generic abstract policy that defines mappings from abstract situations into abstract behaviors, such as $\pi_{abs}(\texttt{inRoom(X)} \wedge \texttt{nonTargetRoom(X)} \wedge \texttt{seeNoDoors}() \wedge \texttt{notAllDoorsVisited}()) = \texttt{findDoor}()$, Indicating that if the robot is in a certain room that is not the target room and it did not detect any door in this room and the list of doors visited by it is empty, then the robot must find a door. The robot is given the goal of reaching the home kitchen. Then, the robot perceives situations, and selects the appropriate behavior for each situation and performs it, until the target room is reached. Figure 4 describes a sequence of five situations faced by the robot.

Situation 1: The Perception module collects information from the environment using the robot semantic cameras. From the position where the robot is, two objects are detected: $\texttt{obj}_1 = \texttt{sofa}$ and $\texttt{obj}_2 = \texttt{armchair}$. Then the Place Inference submodule performs a query to the integrated ROS library KnowRobOMICS, which estimates the most likely room where the robot is taking into account the objects detected by the robot:

```
prolog = json_prolog.Prolog()
for obj in objets:
    q="bayes_probability_given(knowrob:'OmicsLocations',
    Room,knowrob:'"+obj+"',Pr)"
```

```
    query = prolog.query(q)
    for solution in query.solutions():
        room=str(solution['Room'])[37::]
        places.place[room].append(solution['Pr'])
query.finish()
```

As the queries are implemented using Python and the KnowRob-OMICS is implemented using the PROLOG logic programming language, the WRRA uses the ROS library JSON-PROLOG [6] to send the queries from Python code to PROLOG. When the KnowRob-OMICS is queried, it returns the following information for each recognized object:

```
Room = 'knowrob.owl#Kitchen'
Pr = 0.1031101853182014 ;
Room = 'knowrob.owl#DiningRoom'
Pr = 0.12185749173969258 ;
Room = 'knowrob.owl#BedRoom'
Pr = 0.12185749173969258 ;
Room = 'knowrob.owl#LivingRoom'
Pr = 0.3655724752190777 ;
Room = 'knowrob.owl#Hallway'
Pr = 0.14893693434851316 ;
Room = 'knowrob.owl#BathRoom'
Pr = 0.1386654216348226 ;
```

This information gives the probability that each room is the current location of the robot, given only one recognized object. Figure 5 shows some probabilities that a room type has certain object type. These probabilities come from the ROS library KnowRob-OMICS that uses the Lidstone's law, which redistributes the probability mass assigned to the seen tuples to the unseen tuples like (bed,bathroom). The Lidstone's law uses a parameter $\lambda < 1$ and when the parameter $\lambda \to 1$ much probability is distributed to unseen tuples [4]. In our experiments, we set $\lambda = 0.5$.

Then the Place Inference submodule uses the probabilistic reasoning process explained in Section 2.1 to infer the most likely place where the robot is by taking into account all objects recognized by the robot. In $situation_1$ of the case study reported here, the place inferred as the most likely place is $\mathtt{p}_1 = \mathtt{livingroom}$.

When objects and doors are detected and the place is inferred, the semantic map is updated. For this instance the map is updated with the objects $\mathtt{obj}_1$ and $\mathtt{obj}_2$ and the place $\mathtt{p}_1$ by building the relations $\mathtt{obj}_1$ at $\mathtt{p}_1$ and $\mathtt{obj}_2$ at $\mathtt{p}_1$. Next, the Inference Situation submodule receives information about detected doors, the inferred place and the updated semantic map. With this information, the truth values of the predicates are calculated and the conjunction of predicates with truth value TRUE forms a situation description using the SMACH library [7],

[6] http://wiki.ros.org/json_prolog
[7] http://wiki.ros.org/smach

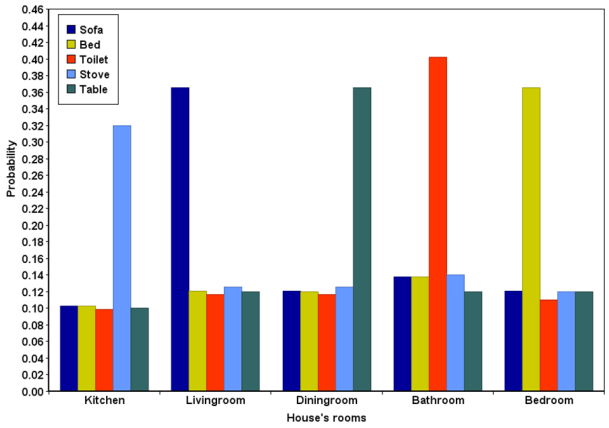


Fig. 5. Probabilities of room given observed object, by KnowRob-OMICS.

which is a ROS-independent Python library to build hierarchical state machines. Finally that inferred situation is the output of the Perception module that for this case is the $situation_1 = \texttt{inRoom}(\texttt{livingroom}) \wedge \texttt{nonTargetRoom}(\texttt{livingroom}) \wedge \texttt{seeNoDoors}() \wedge \texttt{notAllDoorsVisited}()$.

Then $situation_1$ is sent as input to the Reasoning and Learning module, and it is transformed in its corresponding abstract situation. The abstract policy is then used to define the output to the Actuation module: $\pi_{abs}(\texttt{inRoom}(\texttt{X}) \wedge \texttt{nonTargetRoom}(\texttt{X}) \wedge \texttt{seeNoDoors}() \wedge \texttt{notAllDoorsVisited}()) = \texttt{findDoor}()$ (see subsection 2.2). In the current version the output of this module is one of four behaviors: `goToDoor(d)` meaning that the robot should go to `d`; `goToNextRoom(p)` meaning that the robot should go into the next place `p`; `findDoor()` meaning that the robot should search for doors in the room; `exploration()` meaning that the robot should search for objects in the room.

Finally, in the Actuation module, the HLC submodule uses the ActionLIB ROS library to decompose each behavior into a sequence of simple actions, which are in turn translated by the LLC submodule to respective velocities of translation and rotation by using the Move Base ROS library [8], which allows the robot to reach a certain target pose. The Move Base library in turn uses other ROS libraries to avoid obstacles during navigation and to build a local path for the execution of each simple action sent by the HLC. Usually the Move Base library works with a static metric map of the environment, but in our case the Move Base library was set to work without it, since WRRA only reasons in relational level. The robot, controlled by the actuation module, search for a door in the environment. At the moment its semantics camera detects a door, a new situation is defined.

Situation 2: When the robot perceives a door (in this case, it sees door $\texttt{d}_1$), the Perception module updates the semantic map with the door $\texttt{d}_1$ and the

[8] http://wiki.ros.org/move_base

relation $\mathtt{d}_1$ at $\mathtt{p}_1$. So a new ground situation is determined by the Perception module: $situation_2$ = $\mathtt{inRoom(livingroom)} \wedge \mathtt{nonTargetRoom(livingroom)} \wedge \mathtt{seeDoors}() \wedge \mathtt{notAllDoorsVisited}() \wedge \mathtt{seeNonVisitedDoor}(\mathtt{d}_1)$. This situation is turned into its corresponding abstract situation in the Reasoning and Learning module, and the abstract policy gives: $\pi_{abs}(\mathtt{inRoom(X)} \wedge \mathtt{nonTargetRoom(X)} \wedge \mathtt{seeDoors}() \wedge \mathtt{notAllDoorsVisited}() \wedge \mathtt{seeNonVisitedDoor(Y)}) = \mathtt{goToDoor(Y)}$. In this case, the grounding of behavior $\mathtt{goToDoor(Y)}$ gives $\mathtt{goToDoor}(\mathtt{d}_1)$. Then, the Navigation module operates properly, using sensory information that gives the relative position of the robot to $\mathtt{d}_1$ and to obstacles, in order to drive the robot to the front door.

Situation 3: In this situation the robot knows it still is in the living room, it still sees the door $\mathtt{d}_1$ that has not been visited, and now it can see the adjacent room $\mathtt{p}_2$ through door $\mathtt{d}_1$. The map is updated by building the relation $\mathtt{p}_2$ connected to $\mathtt{p}_1$ through $\mathtt{d}_1$. This situation is:
$situation_3$ = $\mathtt{inRoom(livingroom)} \wedge \mathtt{nonTargetRoom(livingroom)} \wedge \mathtt{seeDoors}() \wedge \mathtt{notAllDoorsVisited}() \wedge \mathtt{seeNonVisitedDoor}(\mathtt{d}_1) \wedge \mathtt{seeNextPlace}(\mathtt{p}_2)$.
Given the abstraction of $situation_3$, the behavior indicated by the abstract policy is $goToNextPlace(Z)$ and the grounding of it results in $\mathtt{goToNextPlace}(\mathtt{p}_2)$. In this case, the Navigation module drives the robot through the door and the robot reaches the adjacent room $\mathtt{p}_2$.

Situation 4: As the robot has not observed any object in this new room, then $\mathtt{p}_2 = \mathtt{unknown}$. In this case, the map does not need to be updated and the only predicate with truth-value TRUE is $\mathtt{inUnknownRoom}()$. Then the Reasoning and Learning module outputs the behavior $\mathtt{exploration}()$, meaning that the robot must explore the room, looking for objects.

Situation 5: Finally, the robot observes objects $\mathtt{o}obj_3 = \mathtt{oven}$ and $\mathtt{o}obj_4 = \mathtt{freezer}$ that allow inferring that it is in the $\mathtt{p}_2 = \mathtt{kitchen}$. Then the map is updated by bulding the relations $\mathtt{o}obj_3$ at $\mathtt{p}_2$, $\mathtt{o}obj_4$ at $\mathtt{p}_2$. As the kitchen is the target room, the episode ends and the task is fulfilled.

4 Conclusion

In this paper we have explored the use of semantic web resources to conduct probabilistic relational learning and reasoning in robot navigation. We have presented an architecture (WRRA) for robot navigation that employs the KnowRob system and its knowledge base of probabilistic description logic sentences, together with relational reinforcement learning. The resulting framework shows how to use, in practice, semantic web resources that can deal with uncertainty.

We have implemented the WRRA, first in a simulated environment, then in a real robot. The fact that the WRRA operates with abstract semantic information, both in its knowledge base, and in its inputs and outputs, simplifies the whole process and leads to effective qualitative navigation. Moreover, the acquired abstract knowledge base can be transferred to other scenarios. Our experiments indicate that indoor navigation can actually benefit from such a framework.

Several avenues are open to future work. Additional predicates can be tested to infer the robot location, and web resources can be mixed with human intervention. Furthermore, we intend to include measures of uncertainty about the situation of the robot, by associating probabilities with predicates. We also plan to conduct more extensive tests with the real robot.

Acknowledgments. The authors are partially supported by the National Council for Scientific and Technological Development (CNPq), Brazil.

References

1. Aydemir, A., Pronobis, A., Gobelbecker, M., Jensfelt, P.: Active visual object search in unknown environments using uncertain semantics. Robotics, IEEE Transactions on, 29(4), 986–1002 (2013)
2. Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., Patel-Schneider, P. F.: Description Logic Handbook, Cambridge University Press (2002)
3. Galindo, C., Saffiotti, A.: Inferring robot goals from violations of semantic knowledge. Robotics and Autonomous Systems, 61(10), 1131–1143 (2013)
4. Gupta, Rakesh, Kochenderfer, Mykel J. : Commonsense data acquisition for indoor mobile robots. In: 19th National Conference on Artificial Intelligence (AAAI-04), pp. 605–610. AAAI Press (2004)
5. Jain, D., Waldherr, S., and Beetz, M.: Bayesian Logic Networks. Technical report, IAS Group, Fakultat fur Informatik, Technische Universitat Munchen (2009)
6. Koga, M.L., Silva, V.F.d., Cozman, F.G., Costa, A.H.R.: Speeding-up reinforcement learning through abstraction and transfer learning. In: Conf. Autonomous Agents and Multiagent Systems, pp.119–126 (2013)
7. Lukasiewicz, T.: Expressive probabilistic description logics. Artificial Intelligence, 172(6-7), 852–883 (2008)
8. Lukasiewicz, T., Straccia, U.: Managing Uncertainty and Vagueness in Description Logics for the Semantic Web. Journal of Web Semantics, 6, 291–308 (2008)
9. Mateus, P., Pacheco, A., Pinto, J., Sernadas, A.:Probabilistic Situation Calculus. Annals of Mathematics and Artificial Intelligence, 32, 393–431 (2001)
10. Otterlo,M. V.: The Logic of Adaptative Behaviour. IOS Press, Amsterdam (2009)
11. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann (1988)
12. Riazuelo, L., Tenorth, M., Di Marco, D., Salas, M., Msenlechner, L., Kunze, L., Montiel, J. M. M.: RoboEarth Web-Enabled and Knowledge-Based Active Perception. In: IROS Workshop on AI-based Robotics (2013)
13. Saito, M., Chen, H., Okada, K., Inaba, M., Kunze, L.,Beetz, M.: Semantic object search in large-scale indoor environments. In: IROS Workshop on active Semantic Perception and Object Search in the Real World (2011)
14. Sutton, Richard S., Barto, Andrew G.: Introduction to Reinforcement Learning, MIT Press, Cambridge, MA (1998)
15. Tenorth, M., Klank, U., Pangercic, D., Beetz, M.: Web-enabled robots. Robotics & Automation Magazine, IEEE, 18(2), 58–68 (2011)
16. Tenorth, M., Beetz, M.: KnowRob – A Knowledge Processing Infrastructure for Cognition-enabled Robots. Part 1: The KnowRob System. International Journal of Robotics Research (IJRR) 32(5), 566–590 (2013)

Towards a Distributional Semantic Web Stack

André Freitas[1], Edward Curry[1], Siegfried Handschuh[1,2]

[1]Insight Centre for Data Analytics, National University of Ireland, Galway
[2]School of Computer Science and Mathematics, University of Passau

Abstract. The capacity of distributional semantic models (DSMs) to discover similarities over large scale heterogeneous and poorly structured data brings them as a promising universal and low-effort framework to support semantic approximation and knowledge discovery. This position paper explores the role of distributional semantics in the Semantic Web vision, based on state-of-the-art distributional-relational models, categorizing and generalizing existing approaches into a Distributional Semantic Web stack.

1 Introduction

Distributional semantics is based on the idea that semantic information can be extracted from lexical co-occurrence from large-scale data corpora. The simplicity of its vector space representation, its ability to automatically derive meaning from large-scale unstructured and heterogeneous data and its built-in semantic approximation capabilities are bringing distributional semantic models as a promising approach to bring additional flexibility into existing knowledge representation frameworks.

Distributional semantic approaches are being used to complement the semantics of structured knowledge bases, generating hybrid *distributional-relational models*. These hybrid models are built to support *semantic approximation*, and can be applied to selective reasoning mechanisms, reasoning over incomplete KBs, semantic search, schema-agnostic queries over structured knowledge bases and knowledge discovery.

2 Distributional Semantic Models

Distributional semantic models (DSMs) are semantic models which are based on the statistical analysis of co-occurrences of words in large corpora. Distributional semantics allows the construction of a *quantitative model of meaning*, where the degree of the semantic association between different words can be quantified in relation to a *reference corpus*. With the availability of large Web corpora, comprehensive distributional models can effectively be built.

DSMs are represented as a *vector space model*, where each dimension represents a *context* $\mathcal{C}$ for the linguistic or data context in which the *target term* $\mathcal{T}$ occurs. A *context* can be defined using documents, co-occurrence window sizes

(number of neighboring words or data elements) or syntactic features. The *distributional interpretation* of a target term is defined by a weighted vector of the contexts in which the term occurs, defining a geometric interpretation under a distributional vector space. The weights associated with the vectors are defined using an *associated weighting scheme* $\mathcal{W}$, which can re-calibrates the relevance of more generic or discriminative contexts. A *semantic relatedness measure* $\mathcal{S}$ between two words in the dataset can be calculated by using different *similarity/distance* measures such as the *cosine similarity* or *Euclidean distance*. As the dimensionality of the distributional space can grow large, dimensionality reduction approaches d can be applied.

Different DSMs are built by varying the parameters of the tuple $(\mathcal{T}, \mathcal{C}, \mathcal{W}, d, \mathcal{S})$. Examples of distributional models are *Latent Semantic Analysis*, *Random Indexing*, *Dependency Vectors*, *Explicit Semantic Analysis*, among others. Distributional semantic models can be specialized to different application areas using different corpora.

3 Distributional-Relational Models (DRMs)

Distributional-Relational Models (DRMs) are models in which the semantics of a *structured knowledge base* (KB) is complemented by a *distributional semantic model*.

A *Distributional-Relational Model (DRM)* is a tuple $(\mathcal{DSM}, \mathcal{KB}, \mathcal{RC}, \mathcal{F}, \mathcal{H}, \mathcal{OP})$, where: $\mathcal{DSM}$ is the *associated distributional semantic model*; $\mathcal{KB}$ is the *structured dataset*, with elements E and tuples Ω; $\mathcal{RC}$ is the *reference corpora* which can be unstructured, structured or both. The reference corpora can be internal (based on the co-occurrence of elements within the $\mathcal{KB}$) or external (a separate reference corpora); $\mathcal{F}$ is a *map* which translates the elements $e_i \in E$ into vectors $\overrightarrow{\mathbf{e_i}}$ in the the distributional vector space $VS^{\mathcal{DSM}}$ using the natural language label and the entity type of e_i; $\mathcal{H}$ is a set of threshold values for $\mathcal{S}$ above which two terms are considered to be equivalent; $\mathcal{OP}$ is a set of *operations* over $\overrightarrow{\mathbf{e_i}}$ in $VS^{\mathcal{DSM}}$ and over E and Ω in the $\mathcal{KB}$. The set of operations may include *search*, *query* and *graph navigation* operations using the distance measure $\mathcal{S}$.

The DRM supports a double perspective of semantics, keeping the fine-grained precise semantics of the structured KB but also complementing it with the distributional model. Two main categories of DRMs and associated applications can be distinguished:

Semantic Matching & Commonsense Reasoning: In this category the $\mathcal{RC}$ is unstructured and it is distinct from the $\mathcal{KB}$. The large-scale *unstructured* $\mathcal{RC}$ is used as a *commonsense knowledge base*. Freitas & Curry [1] define a DRM $(\tau - Space)$ for supporting schema-agnostic queries over the structured $\mathcal{KB}$: terms used in the query are projected into the distributional vector space and are semantically matched with terms in the $\mathcal{KB}$ via distributional semantics using commonsense information embedded on large scale unstructured corpora $\mathcal{RC}$. In a different application scenario, Freitas et al. [3] uses the $\tau - Space$ to support selective reasoning over commonsense $\mathcal{KB}s$. Distributional semantics is

used to select the facts which are semantically relevant under a specific reasoning context, allowing the scoping of the reasoning context and also coping with incomplete knowledge of commonsense KBs. Pereira da Silva & Freitas [2] used the $\tau - Space$ to support approximate reasoning on logic programs.
Knowledge Discovery: In this category, the structured $\mathcal{KB}$ is used as a distributional reference corpora (where $\mathcal{RC} = \mathcal{KB}$). Implicit and explicit semantic associations are used to derive new meaning and discover new knowledge. The use of structured data as a distributional corpus is a pattern used for knowledge discovery applications, where knowledge emerging from *similarity patterns in the data* can be used to retrieve similar entities and expose implicit associations. In this context, the ability to represent the $\mathcal{KB}$ entities' attributes in a vector space and the use of vector similarity measures as way to retrieve and compare similar entities can define universal mechanisms for knowledge discovery and semantic approximation. Novacek et al. [5] describe an approach for using web data as a bottom-up phenomena, capturing meaning that is not associated with explicit semantic descriptions, applying it to entity consolidation in the life sciences domain. Speer et al. [8] proposed AnalogySpace, a DRM over a commonsense $\mathcal{KB}$ using Latent Semantic Indexing targeting the creation of the analogical closure of a semantic network using dimensional reduction. AnalogySpace was used to reduce the sparseness of the $\mathcal{KB}$, generalizing its knowledge, allowing users to explore implicit associations. Cohen et al. [6] introduced PSI, a predication-based semantic indexing for biomedical data. PSI was used for similarity-based retrieval and detection of implicit associations.

4 The Distributional Semantic Web Stack

DRMs provide universal mechanisms which have fundamental features for semantic systems: (i) built-in semantic approximation for terminological and instance data; (ii) ability to use large-scale unstructured data as commonsense knowledge, (iii) ability to detect emerging implicit associations in the $\mathcal{KB}$, (iv) simplicity of use supported by the vector space model abstraction, (v) robustness with regard to poorly structured, heterogeneous and incomplete data. These features provide a framework for a robust and easy-to-deploy semantic approximation component grounded on large-scale data. Considering the relevance of these features in the deployment of semantic systems in general, this paper synthesizes its vision by proposing a *Distributional Semantic Web stack* abstraction (Figure 1), complementing the Semantic Web stack. At the bottom of the stack, unstructured and structured data can be used as reference corpora together with the target $\mathcal{KB}$ (RDF(S)). Different elements of the distributional model are included as optional and composable elements of the architecture. The *approximate search and query operations layer* access the *DSM layer*, supporting users with semantically flexible search and query operations. A *graph navigation layer* defines graph navigation algorithms (e.g. such as spreading activation, bi-directional search) using the semantic approximation and the distributional information from the layers below.

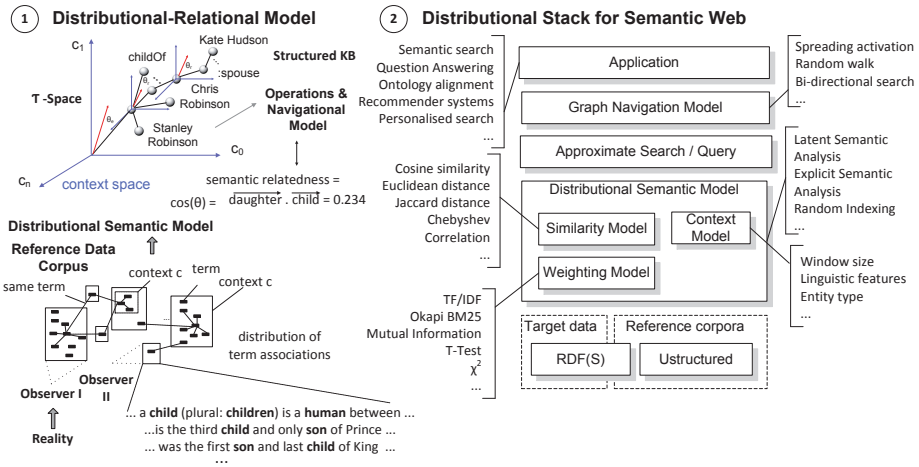


Fig. 1: (A) Depiction of an example DRM ($\tau - Space$) (B) Distributional Semantic Web stack.

Acknowledgment: This publication was supported in part by Science Foundation Ireland (SFI) (Grant Number SFI/12/RC/2289) and by the Irish Research Council.

References

1. Freitas, A., Curry, E., Natural Language Queries over Heterogeneous Linked Data Graphs: A Distributional-Compositional Semantics Approach. *In Proc. of the 19th Intl. Conf. on Intelligent User Interfaces (IUI).* (2014).
2. Pereira da Silva, J.C., Freitas A., Towards An Approximative Ontology-Agnostic Approach for Logic Programs, *In Proc. of the 8th Intl. Symposium on Foundations of Information and Knowledge Systems.* (2014).
3. Freitas, A., Pereira Da Silva, J.C., Curry, E., Buitelaar, P., A Distributional Semantics Approach for Selective Reasoning on Commonsense Graph Knowledge Bases. *In Proc. of the 19th Int .Conf. on Applications of Natural Language to Information Systems (NLDB).* (2014).
4. Speer, R., Havasi, C., Lieberman, H., AnalogySpace: Reducing the Dimensionality of Common Sense Knowledge. *In Proc. of the 23rd Intl. Conf. on Artificial Intelligence*, 548-553. (2008).
5. Novacek, V., Handschuh, S., Decker, S.. Getting the Meaning Right: A Complementary Distributional Layer for the Web Semantics. *In Proc. of the Intl. Semantic Web Conference*, 504-519. (2011).
6. Cohen, T., Schvaneveldt, R.W., Rindflesch, T.C.. Predication-based Semantic Indexing: Permutations as a Means to Encode Predications in Semantic Space. *T. AMIA Annu Symp Proc.*, 114-118. (2009).
7. Turney, P.D., Pantel P., From frequency to meaning: vector space models of semantics. *J. Artif. Int. Res.*, 37(1), 141–188. (2010).
8. Speer, R., Havasi, C., Lieberman, H., AnalogySpace: Reducing the Dimensionality of Common Sense Knowledge. *In Proc. of the 23rd Intl. Conf. on Artificial Intelligence*, 548-553. (2008).

Overview of METHOD 2014: The 3rd International Workshop on Methods for Establishing Trust of (Open) Data

Tom De Nies[1], Davide Ceolin[2], Paul Groth[2], Olaf Hartig[3], and Stephen Marsh[4]

[1] tom.denies@ugent.be
Ghent University – iMinds – Multimedia Lab, Belgium
[2] d.ceolin, p.t.groth@vu.nl
VU University Amsterdam, The Netherlands
[3] ohartig@uwaterloo.ca
University of Waterloo, Canada
[4] stephen.marsh@uoit.ca
University of Ontario Institute of Technology, Canada

1 Introduction

The METHOD workshop aims to bring together researchers working on the problem of trust and quality assessment of (open) data, and all components that contribute to this goal. It provides a forum for researchers from both the Semantic Web and the Trust Management community, with the goal of gaining new insights towards solutions for this complex problem. Due to the relatively low number of submissions, and to maximize the impact of the accepted papers, METHOD 2014 merged with the 10th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW), as a special session. In this short editorial paper, we provide an overview of the topics discussed during this session.

Trust assessment of content on the Web is a highly complex concept that depends on objective as well as subjective criteria, including the content's provenance, data quality estimation, and also the consumer's background, personality, and context. However, the exact criteria and tolerances differ for each context and for each assessor, requiring detailed knowledge about the data and its uses. This also makes it very challenging to find generic solutions and assessments that are applicable everywhere or transferrable from one context to another. Therefore, stakeholders in this field are continuously investigating new techniques to handle and prepare data in such a way that it becomes easier for machines to process it with the goal of trust and/or quality assessment. The METHOD workshop is a venue for presenting and discussing novel research ideas in this field, as well as technical applications.

2014 is the third year for METHOD. The two previous editions were held in conjunction with the IEEE Annual International Computer Software & Applications Conference (COMPSAC). This year, the workshop is held for the first time at ISWC, since the topics of provenance, data quality, and trust are highly relevant to the Semantic Web community. The diversity in the ways that these topics may be approached is also visible judging from the subject of the submis-

sions we received. Although only three papers were accepted, each submission covers a distinctly different aspect of the general theme of the workshop.

2 Papers and Discussion Topics

This year's submissions cover the following aspects of trust of (open) data: content *rating*, data quality *rewarding*, data *attribution*, and trust *representation*.

The first two aspects are covered by Couto, who proposes a number of guidelines for a system that rates as well as rewards good practices in data sharing, by means of a virtual currency [1]. A large number of unsolved technical challenges are identified by this position paper, and some issues are left open, such as how to uniquely identify and attribute data to its creators. This is exactly what Höfig and Schieferdecker investigate, proposing a new hash function for RDF graphs [3]. The solution proposed in this research paper may contribute to a tamper-resistant way of attributing RDF data to its authors, allowing one to make claims about the data's trustworthiness. Of course, a representation for these trustworthiness assessments is needed. In the final position paper of our workshop, Ceolin et. al. propose an ontology to represent trust of web data, extending existing solutions [2].

Despite the diverse aspects discussed in the submissions, a number of common themes and open questions can be identified, listed below.

- What *incentive* is there for data creators to make their data trustworthy?
- Which *mechanisms* are in place to attribute data to its creators and editors, and do they suffice for our needs?
- How can we estimate *ratings or assessments* of data quality and trustworthiness?
- How would it be possible to allow the *reuse* of such trust and quality estimations?
- How do we *represent* the aforementioned aspects in an *interoperable* way?

These are the questions we hope to see addressed during the discussions at METHOD 2014, and in future editions of the workshop.

References

[1] Couto, F.: Rating, recognizing and rewarding metadata integration and sharing on the semantic web. In: Proceedings of the 10th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW), Riva del Garda, Italy. CEUR-WS.org (2014)

[2] Davide Ceolin, Archana Nottamkandath, W.F., Maccatrozzo, V.: Towards the definition of an ontology for trust in (web) data. In: Proceedings of the 10th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW), Riva del Garda, Italy. CEUR-WS.org (2014)

[3] Hoefig, E., Schieferdecker, I.: Hashing of RDF graphs and a solution to the blank node problem. In: Proceedings of the 10th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW), Riva del Garda, Italy. CEUR-WS.org (2014)

Hashing of RDF Graphs and a Solution to the Blank Node Problem

Edzard Höfig[1] and Ina Schieferdecker[2]

[1] Beuth University of Applied Sciences, Luxemburger Str. 10, 13353 Berlin, Germany
edzard.hoefig@beuth-hochschule.de
[2] Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
ina.schieferdecker@fokus.fraunhofer.de

Abstract. The ability to calculate hash values is fundamental for using cryptographic tools, such as digital signatures, with RDF data. Without hashing it is difficult to implement tamper-resistant attribution or provenance tracking, both important for establishing trust with open data. We propose a novel hash function for RDF graphs, which does not require altering the contents of the graph, does not need to record additional information, and does not depend on a concrete RDF syntax. We are also presenting a solution to the deterministic blank node labeling problem.

1 Introduction

Two years ago we started to discuss requirements for Open Information Spaces (OIS): distributed systems that facilitate the sharing of data, while supporting certain trust-related properties, e.g. attribution, provenance, or non-repudiation [1, Section II]. While working on the topic, we quickly found out that it is necessary to not only record trust-relevant information, but also to make sure that the information cannot easily be tampered with. In closed systems, where access is strictly regulated and monitored, it is possible to record attribution information like "Alice created this data set" in a trustworthy manner. In open systems, where everyone is free to share and re-use any provided data sets, this is harder and usually requires some sort of cryptographic processing.

One of most commonly used methods employed in this context are hash functions. They take a fixed version of a data set (the snapshot) and calculate a smaller, characteristic string for that data (the hash value). Cryptographic hash functions are constructed in a way that even very small changes to the original snapshot result in completely different hash values. Furthermore, it is a runtime expensive operation to construct a data set that yields a given hash value. Thus, by publishing the hash value for a snapshot x, it is possible to verify that some data set y is highly likely to be identical to x, simply because their hash values match. For example, to record the information "Alice created this data set", Alice would create a hash value of the data set and digitally sign it using common cryptographic techniques. The signature could then be published as meta data along with the data set, allowing verification of the attribution information by calculating the hash value of a local copy of the data set and comparing it to the one in the signature.

1.1 Motivation

Attribution is one of the fundamental characteristics that OIS needs to support, as more complex operations, like provenance tracking, build upon such an attribution framework. To engineer an OIS, we needed to start somewhere, so we decided to quickly implement an attribution system. Technology-wise, we use the Resource Description Framework (RDF) [2] to hold our data sets. Thus, our first task was the implementation of a hash function for RDF graphs. This turned out to be a complex undertaking. The problem is that RDF does not have a single, concrete syntax. Although quite rigidly defined in terms of semantics, the specification explicitly states that "A concrete RDF syntax may offer many different ways to encode the same RDF graph or RDF dataset, for example through the use of namespace prefixes, relative IRIs, blank node identifiers, and different ordering of statements." [2, Section 1.8]. Unfortunately, to work properly, hash functions need a single, concrete syntax. Often, RDF data is transmitted not as a document — which would be bound to a concrete syntax — but comes from query interfaces, e.g., SPARQL endpoints [3]. What we really needed was a hash function that can work on an in-memory RDF graph. The hash function itself is not the issue, as there are several implementations available (we are relying on SHA-256 [4, Section 6.2] to calculate the final hash value). The problem is to deterministically construct a single character string that distinctly represents the RDF graph, and the main issue here is the identity of blank nodes contained in the graph.

1.2 Paper Structure

The current section introduces the reader to the general issue and explains our motivation. In Section 2, we are studying both the underlying problems that arise when trying to implement a hash function, as well as the related work in the scientific community. Section 3 contains the description of an algorithm that is able to create a hash value for in-memory RDF graphs with blank nodes and we conclude with a critical discussion of our contribution in Section 4.

2 Problems and Related Work

Initially, we looked at the literature and found a number of articles, dating back about ten years and discussing the issue in great detail. There were even standards that seemed directly applicable, for example XML Signature [5,6]. After studying the literature, we found that none of the articles fully explains a general solution to the problem. All of them need certain constraints, or make assumptions about the data, for example the use of a certain concrete syntax. For our purposes the situation was inadequate, because of our following requirements:

1. No modification of the RDF data is needed for the algorithm to work
2. No additional data needs to be available, apart from the RDF graph
3. The algorithm works in-memory and not on a concrete syntax

During studies of the subject, we found that three issues were of paramount importance, if we wanted to solve the problem. We will explain these first, before investigating the existing work.

Blank Node Identifiers: RDF graphs might contain blank nodes. Such nodes do not have an Internationalized Resource Identifier (IRI) [7] assigned. Once loaded by an RDF implementation, they are assigned local identifiers, which are not transferable to other implementations. When trying to calculate a hash value, this is a major issue as blank nodes cannot be deterministically addressed. One can think of blank nodes as anonymous and having an identity is a necessary pre-condition for calculating a hash value. Solving the blank node issue is an algorithmic challenge.

Order of Statements Calculation of hash values effectively serialises a RDF graph structure into a string. RDF does not imply a certain order of statements, e.g. the order of predicates attached to a single subject. For serialisation purposes we need a deterministic order, or otherwise we might end up with hash values differing between implementations. This issue can be solved by adhering to a sort order when serialising the graph.

Encoding RDF uses literals to store values in the graph. These literals need to follow a common encoding, otherwise different hash values might be calculated. The same goes for namespace prefixes or relative IRIs (both features of the XML syntax for RDF [8]) –they need to be encoded with fully qualified names when stored in memory, or, at the latest, when serialised.

The most influential paper on the subject of RDF graph hashing was written by Carroll in 2003 [9]. Building on earlier work [10], Carroll explains that the runtime of any algorithm for generic hashing of RDF data is equivalent to the graph isomorphism problem, which is not known to be $\mathcal{NP}$-complete nor known to be in $\mathcal{P}$. Carroll then refrains from finding a generic solution to the problem and details his algorithm, which runs in $\mathcal{O}(n\ log(n))$, but re-writes RDF graphs to a canonical format. The proposed algorithm works on the N-Triples format (a concrete document syntax). As far as solving the blank node identity problem, the article states: "Since the level of determinism is crucial to the workings of the canonicalization algorithm, we start by defining a deterministic blank node labelling algorithm. This suffers from the defect of not necessarily labeling all the blank nodes." [9, Section 4]. Carroll continued to work on the subject, for example by publishing applications based on digitally signing graphs, together with Bizer, Hayes, and Stickler [11], but did not seem to have designed a general algorithm for hashing RDF graphs.

Sayers and Karp, colleagues of Carroll, published two technical reports at Hewlett-Packard that explains RDF hashing and applications thereof [12,13]. They identify four different ways to tackle the blank node problem [13, Section 3 ff.], namely:

Limit operations on the graph The idea is to maintain blank node identifiers across implementations, which is not possible in an open world scenario.

Limit the graph itself Avoid the use of blank nodes. This is clearly not the way for us, as we strive for a general solution to the problem.

Modify the graph Work around the problem by adding information about blank node identity within the graph. Not possible for us, as we don't want to change the RDF graph.

Change the RDF specification Generally assign globally unique identifiers to blank nodes. This will most likely never happen.

Apart from changing the RDF specification, none of these methods actually solves the problem and they can be seen as workarounds.

Other authors also investigated RDF graph hashing, e.g. Giereth [14] for the purpose of encrypting fragments of a graph. Giereth works around the blank node issue by modifying the graph. A more current approach by Kuhn and Dumontier [15] proposes an encoding of hash values in URIs. The approach replaces blank nodes with arbitrarily generated identifiers and thus needs to modify the RDF data to work.

Although Carroll did already provide pointers in the right direction, the final idea for solving the issue can be attributed to Tummarello et al., who introduce the concept of a "Minimum Self-Contained Graph" (MSG) [16]. A MSG is a partitioning of a graph, so that each MSG contains at most one transitively connected sub-graph of blank nodes. We apply this idea to construct a blank node identity, as explained in the next section.

3 The Algorithm

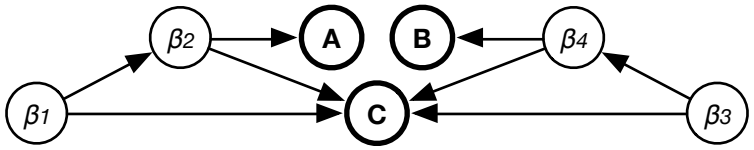


Fig. 1. An example structure with blank nodes

Our approach to the blank node labeling problem relies on constructing the identity of a blank node though its context. This is similar to the MSG principle of Tummarello et al. [16, Section 2] and a logical continuation of the thoughts of Carroll [9, Section 7.1]. For an example, see Fig. 1. The diagram shows three nodes with IRIs (**A**, **B**, **C**) and four blank nodes (β_1 ... β_4). If we go on to define the identity of a node as determined by its direct subjects, we can distinguish β_2 and β_4, but not yet the two other ones: there are both blank nodes pointing to another blank node and the **C** node. We can only discern every node by taking more of the context into account: not only direct neighbours, but neighbouring nodes one hop away. Consequentially, the identity of a blank node can only be constructed when following all of the transitive blank nodes, reachable from the original one. Using this scheme, we are able to establish an identity for blank

nodes. Having identity for both: blank nodes, and IRI nodes, we can generate a characteristic, implementation-independent string for both node types. As RDF only allows these two types of nodes as subjects, we are now able to create a list of so-called "subject strings". To solve the issue of implementation-dependent statement order, it is necessary to establish an overall ordering criterium over this list. We are using a simple lexicographical ordering based on the unicode value of each character in a string. As we are solely relying on subject nodes to calculate the hash value, we need to also encode the predicates and objects of the RDF graph into the subject strings, as well. This way of encoding graph structure into the overall data used to calculate the hash value is a major difference to other algorithms striving for the same goal. Usually, only flat triple lists are processed and the hash function calculates a value for each triple. To encode the graph structure we are using special delimiter symbols. Without these symbols, differing graph topology might lead to the same string representation and thus, the same hash value. For the final cryptographic calculation of the hash value, we are using SHA-256 on UTF-8 [17] encoded data. SHA-256 was chosen as the amount of characters in the overall data string seems to be moderate. It is recommended to use SHA-256 for less than 2^{64} bits of input [4, Section 1].

3.1 Preconditions and Remarks

To calculate the digest of a single RDF graph g, the graph has to reside in memory first, as we are not concerned with any network or file-based representations of the graph. We require that the graph's content is accessible as a set of $< S, P, O >$ triples [3]. It is beneficial to have fast access to all subject nodes of the graph, and to all properties of a node and we are using matching patterns to express this type of access, e.g. $<?n_s, ?, ?>$ to denote any node n_s that appears in the role of a subject in the RDF triple data[4]. Literals and IRI identifiers need to be stored as unicode characters. There are no restrictions on the blank node identifiers, as we do not use them for calculation of the digest. For sake of clarity, we present our algorithm broken down in four separate sub-aglorithms: A function that calculates the hash value for a given graph (Algorithm 1), a procedure that calculates the string representation for a given subject node (Algorithm 2), another one for the string representation of the properties of a given subject node (Algorithm 3), and a last one for calculation of the string representation of an object node (Algorithm 4). These sub-algorithms call each other and thus, could be combined in a single operation. It should be noted, that the algorithm uses reentrance to establish the transitive relationship needed to assign identities to the blank nodes.

Our algorithm uses a number of different delimiter symbols with strictly defined, constant values, which we assigned greek letters to. Table 1 gives an overview of these symbols.

[3] Triples with a *S*ubject, *P*redicate, and *O*bject

[4] The question mark notation is inspired by the SPARQL query syntax [3]

Symbol	Symbol Name	Value (Unicode)	Value Name
α_s	Subject start symbol	{ (U+007B)	Left curly bracket
ω_s	Subject end symbol	} (U+007D)	Right curly bracket
α_p	Property start symbol	((U+0028)	Left parentheses
ω_p	Property end symbol	) (U+0029)	Right parentheses
α_o	Object start symbol	[(U+005B)	Left square bracket
ω_o	Object end symbol	] (U+005D)	Right square bracket
β	Blank node symbol	$*$ (U+002A)	Asterisk

Table 1. Guide to delimiters and symbols used in the algorithms

Use of these symbols is unproblematic in regard to their appearance as part of the RDF content. As we use two symbols to delimit a scope in the string, we can clearly distinguish between use as delimiter and use as content. As our goal is the creation of a string representation for each subject node, the algorithm makes heavy use of string concatenation and we are using the $\oplus$ symbol to denote this operation. In several places, strings are nested between start and stop delimiters, like this: $\alpha \oplus string \oplus \omega$.

Some final remarks regarding the implementation before delving into the specifics of the algorithm: We are using some variables (*visitedNodes*, g) as parameters for functions and procedures. Of course, these should be better put away as shared variables (e.g. attributes in an object). The variable *result* is always local and needs to be empty at the start of each function. Furthermore, there are some functions that dependent on a concrete implementation and are quite trivial to use. We skip an in-depth discussion of those, e.g., *predicates*(...).

3.2 Calculating the Hash Value

Algorithm 1 Calculating a hash value for a RDF graph g

```
1: function CALCULATEHASHVALUE(g)
2:     for all n_s ∈ g that match <?n_s, ?, ? > do          ▷ All subject nodes
3:         visitedNodes ← ∅
4:         subjectStrings ← subjectStrings ∪ encodeSubject(n_s, visitedNodes, g)
5:     end for
6:     sort subjectStrings in unicode order
7:     for all s ∈ subjectStrings do
8:         result ← result ⊕ α_s ⊕ s ⊕ ω_s
9:     end for
10:    return hash(result)                                  ▷ Using SHA-256 and UTF-8
11: end function
```

To calculate the hash value for a given RDF graph, Algorithm 1 is used. The function takes a single parameter: the RDF graph g to use for calculation of

its hash value. At first the algorithm iterates over all of the subject nodes n_s that exist in g (lines 2–5). A subject node is any node that appears in the role of a subject in a RDF triple contained in g. For each of the subject nodes we create a data structure, called $visitedNodes$, which is used to record if we already processed some blank node. This is necessary for termination of the construction of the blank node identities. $visitedNodes$ needs to be empty before calculating the string representation for n_s by calling the procedure $encodeSubject(...)$ in line 4, which is explained in the next section. After all subject nodes are encoded, the resulting list needs to be sorted. Any sorting order could be used and as we do not require specific semantics for this step, we are establishing an ordering simply by comparing the unicode numbering of letters. The sorting operation in line 6 is key to deterministically create a hash value, as the result would otherwise build upon the (implementation-dependent) order of nodes in g. All of the sorted subject-strings are then concatenated to form an overall result string, while each single subject string is enclosed with the subject delimiter symbols (line 8). Finally, the result string is subjected to a cryptographic hash function and returned (line 10).

3.3 Encoding the Subject Nodes

In RDF, subject nodes can be of two types: they can either be blank nodes or IRIs [2, Section 3.1]. The procedure $encodeSubject(...)$, shown in Algorithm 2 and used by Algorithm 1, needs to take care of this. The procedure has three arguments: n_s – the subject node to encode as a string, $visitedNodes$ – our data structure for tracking already visited nodes, and g – the RDF graph.

Algorithm 2 Encoding a subject node n_s

```
 1: procedure ENCODESUBJECT(n_s, visitedNodes, g)
 2:     if n_s is a blank node then
 3:         if n_s ∈ visitedNodes then
 4:             return ∅                                   ▷ This path terminates
 5:         else
 6:             visitedNodes ← visitedNodes ∪ n_s   ▷ Record that we visited this node
 7:             result ← β
 8:         end if
 9:     else
10:         result ← IRI of n_s                              ▷ n_s has to be a IRI
11:     end if
12:     result ← result ⊕ encodeProperties(n_s, visitedNodes, g)
13:     return result
14: end procedure
```

The discrimination of types comes first: lines 2–8 process blank nodes and lines 9–11 take care of IRIs. For the blank nodes, we have to distinguish between the case where we already met a blank subject node (line 3–4), and the case where

we didn't (line 5–7). In the case that the subject node was already encountered, the graph traversal ends and we return an empty string. If the blank subject node is hitherto unknown, then *result* is set to the blank node symbol β (see Table 1) and the node is recorded in *visitedNodes* as being processed. If the subject is not a blank node, but an IRI, we simply set *result* to be the IRI itself. Only encoding the subject node itself is not sufficient for our purposes, as we need to establish an identity based on the context of the current subject node. In line 12 this process is triggered by calling the *encodeProperties*(...) procedure (see Algorithm 3) and concatenating the returned string with the existing *result*. The *result* itself is returned in line 13.

3.4 Encoding the Properties of a Subject Node

Algorithm 3 is responsible for encoding all of the properties of a given subject node n_s into a single string representation. We understand properties as the predicates (p) and objects (o) that fullfil $< n_s, ?p, ?o >$, where n_s is a given subject. Apart from n_s, the procedure *encodeProperties*(...) needs *visitedNodes* as a second, and g as a third argument.

Algorithm 3 Encode properties for a subject node

```
 1: procedure encodeProperties(n_s, visitedNodes, g)
 2:     p ← predicates(n_s, g)        ▷ Retrieve all predicate IRIs that have n_s as subject
 3:     sort p in unicode order
 4:     for all iri ∈ p do
 5:         result ← result ⊕ α_p ⊕ iri
 6:         for all n_o ∈ g that match < n_s, iri, ?n_o > do   ▷ All objects for n_s and iri
 7:             objectStrings ← objectStrings ∪ encodeObject(n_o, visitedNodes, g)
 8:         end for
 9:         sort objectStrings in unicode order
10:         for all o ∈ objectStrings do
11:             result ← result ⊕ α_o ⊕ o ⊕ ω_o
12:         end for
13:         result ← result ⊕ ω_p
14:     end for
15:     return result
16: end procedure
```

The algorithmic structure reflects the complexity of graph composition using RDF predicates: a subject node can be associated with multiple predicates, and the predicates are allowed to be similar, if associated with different objects. We use a two stage process to encode properties: First, all unique predicate IRIs of the given subject node n_s are retrieved and ordered (lines 2–3). We are postulating a function *predicates*(...) that returns all predicate IRIs for a given subject node n_s by searching all triples for matches to $< n_s, ?p_x, ? >$, extracting the IRI of the identified predicate p_x, and eliminating double entries. In a second step,

we encode each predicate IRI and the set of objects associated with it (line 4–14). The property encoding starts in line 5, where the *result* is concatenated with the property start symbol α_p and the predicate IRI. Subsequently, we retrieve all object nodes (nodes that appear in triples with n_s as subject and *iri* as predicate), encode each object node using the procedure *encodeObject*(...), and store their respective string representations in a *objectStrings* list (line 6–8). The *encodeObject*(...) procedure is detailed in Algorithm 4. After collecting all the encoded object strings, the resulting list has to be sorted (line 9). In line 10–12 each object string is appended to *result*, while taking care to enclose the string in delimiter symbols. Encoding of a single property (one pass of the loop started in line 4) ends with appending the property stop symbol ω_p to *result*. Once the procedure has encoded all properties it returns with the complete *result* string in line 15.

3.5 Encoding the Object Nodes

Processing the object nodes itself is trivial when compared to the property encoding. Objects in RDF triples can be three things: an IRI, a literal, or a blank node [2, Section 3.1]. The *encodeObject*(...) procedure needs to return an appropriate string representation for each of these three cases. It takes three arguments: an object node n_o, *visitedNodes*, and the RDF graph g.

Algorithm 4 Encode an object node

```
1: procedure ENCODEOBJECT(n_o, visitedNodes, g)
2:     if n_o is a blank node then
3:         return encodeSubject(n_o, visitedNodes, g)        ▷ Re-enter Algorithm 2
4:     else if n_o is a literal then
5:         return literal representation of n_o              ▷ Consider language and type
6:     else
7:         return IRI of n_o                                 ▷ n_o has to be a IRI
8:     end if
9: end procedure
```

The three aforementioned cases are treated as follows. If n_o is a blank node, we continue with re-entering Algorithm 2 (see line 3). The re-entrance allows us to construct a path through neighbouring blank nodes. Together with having potentially many object nodes associated with a single subject, this yields a connected graph, similar to the MSGs of Tummarello et al. If n_o is a literal, it is returned in a format according to [2, Section 3.3] in line 5, including any language and type information. If n_o is neither a blank node, nor a literal, it has to be an IRI and we return it verbatim. After all objects, properties, and subjects have been encoded, all sub-algorithms have returned and Algorithm 1 terminates.

3.6 An Example

Consider the RDF graph shown in Figure 1 at the beginning of Section 3. When applying the algorithm to the given graph, we end up with the following string before calculating the SHA-256 hash (see Algorithm 1, line 10):

$$\{*(-[*(-[\mathbf{A}][\mathbf{C}])][\mathbf{C}])\}\ \{*(-[*(-[\mathbf{B}][\mathbf{C}])][\mathbf{C}])\}\ \{*(-[\mathbf{A}][\mathbf{C}])\}\ \{*(-[\mathbf{B}][\mathbf{C}])\}$$

Please note that this string is not valid RDF, as scheme and path information are missing from the employed IRIs[5] used by **A**, **B**, and **C**. Also the symbol "−" is used to indicate an arbitrary IRI employed for all predicates.
Thanks to the delimiters, it is quite easy to understand the string's structure. The four subject node strings for β_1, β_3, β_2, and β_4 (in this order) are encapsulated between curly brackets each. **A**, **B**, and **C** only appear as object nodes and thus do not trigger the creation of additional subject strings. Instead, they are encoded as part of the blank node subject strings. Let's take a look at the first subject string for node β_1: $\{*(-[*(-[\mathbf{A}][\mathbf{C}])][\mathbf{C}])\}$. Apart from the curly brackets, the string starts with the blank node symbol "$*$", followed by the properties of that node, delimited in parentheses. The node has only a single predicate, used with two different objects: $[*(-[\mathbf{A}][\mathbf{C}])]$ and $[\mathbf{C}]$. If we would have additional predicate types, there would also be further parentheses blocks. Objects are delimited by square brackets and due to the re-entrant nature of the algorithm object strings follow the same syntax as just discussed.

4 Conclusion and Future Work

The presented algorithm calculates a hash value for RDF graphs including blank nodes. It is not necessary to alter the RDF data or to record additional information. It is not dependent on any concrete syntax. It solves the blank node labeling problem by encoding the complete context of a blank node, including the RDF graph structure, into the data used to calculate the hash value. The algorithm has a runtime complexity of $\mathcal{O}(n^n)$, which is consistent with current research on algorithms for solving the graph isomorphism problem [9]. The worst-case scenario is a fully meshed RDF graph of blank nodes, which does not seem to make any sense whatsoever — usually, we would expect the amount of blank nodes in a graph to be far smaller, thus the real execution speed to be less catastrophic. We concentrated on solving the primary problem, not on runtime optimisations and we are certain, that there is room for improvement in the given algorithm. There are some obvious starting points for doing this. For example, there are redundancies in the string representations for transitive blank node paths (the string representations for β_2 and β_4 appear twice in the example given in Section 3.6). One could cache already computed subject-strings, pulling them from the cache when needed. Also, the interplay between the SHA-256 digest computation and the subject string calculations has not been researched in sufficient detail.

[5] For example **A** instead of http://a/

It might be possible to reduce processing and storage overhead by combining these two operations, calling the digest operation on each subject string and combining the resulting values in order of the final sorted list. The sensibility of such optimizations should largely depend on the susceptibility of the digest algorithm implementation to the length of the given input strings. This, in turn, depends on the usage of blank nodes in the input RDF data. More blank nodes in the input data and more references between blank nodes means longer subject strings. Consequentially, to come to a more substantial assessment of the presented algorithm, we will need to study its performance on a number of real (and larger) data sets.

While we trust the general approach for solving the blank node labeling problem through an assignment of identity based on the surrounding context of the node, we did not proof that the algorithm works correctly. To assure that it works properly, we did test it: on the one hand in regard to its ability to process all possible RDF constructs using tests from W3C's RDF test cases recommendation [18], on the other hand in regard to the correctness of the blank node labeling approach using manually constructed graphs. These graphs range from simple, non cyclic ones with a single blank node to all possible permutations of a fully meshed graph of blank nodes with variations on the attachment of IRI nodes and predicate types.

Acknowledgments. We would like to thank Thomas Pilger and Abdul Saboor for their collection of material documenting the current state of the art and for tireless implementation work.

References

1. Höfig, E. Supporting Trust via Open Information Spaces. Proc IEEE 36th Annual Computer Software and Applications Conference, pp. 87–88, (2012)
2. World Wide Web Consortium.: RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation, `http://www.w3.org/TR/rdf11-concepts/` (2014)
3. World Wide Web Consortium.: SPARQL 1.1 Query Language, W3C Recommendation, `http://www.w3.org/TR/sparql11-query/` (2013)
4. National Institute of Standards and Technology.: Secure Hash Standard (SHS). Federal Information Processing Standards Publication 180–4 (2012)
5. World Wide Web Consortium.: XML Signature Syntax and Processing (Second Edition), W3C Recommendation, `http://www.w3.org/TR/xmldsig-core/` (2008)
6. Cloran, R., Irwin B.: XML Digital Signature and RDF. Proc. Information Security South Africa (2005)
7. Duerst, M., Suignard, M.: Internationalized Resource Identifiers (IRIs). Internet Engineering Task Force – Request for Comments 3987 (2005)
8. World Wide Web Consortium.: RDF 1.1 XML Syntax, W3C Recommendation, `http://www.w3.org/TR/rdf-syntax-grammar/` (2014)
9. Carroll, J. J.: Signing RDF Graphs. Proc. of the Second International Semantic Web Conference, LNCS 2870, pp. 369–384 (2003)
10. Carroll, J. J.: Matching RDF Graphs. Proc. of the First International Semantic Web Conference, LNCS 2342, pp. 5–15 (2002)

11. Carroll, J. J., Bizer, C., Hayes, P., Stickler, P.: Named Graphs, Provenance and Trust. Proc. International World Wide Web Conference, pp. 613–622 (2005)
12. Sayers, C., Karp. A. H.: Computing the Digest of an RDF Graph. Hewlett-Packard Labs Technical Report HPL-2003-235 (2003)
13. Sayers, C., Karp. A. H.: RDF Graph Digest Techniques and Potential Applications. Hewlett-Packard Labs Technical Report HPL-2004-95 (2004)
14. Giereth, M.: On Partial Encryption of RDF-Graphs. Proc. of the Fourth International Semantic Web Conference, LNCS 3729, pp. 308–322 (2005)
15. Kuhn, T., Dumontier, M.: Trusty URIs: Verifiable, Immutable, and Permanent Digital Artifacts for Linked Data. Proc. Eleventh European Semantic Web Conference, LNCS 8465, pp. 395–410 (2014)
16. Tummarello, G., Morbidoni, C., Puliti, P., Piazza, F.: Signing Individual Fragments of an RDF Graph. International World Wide Web Conference, pp. 1020–1021 (2005)
17. Yergeau, F.: UTF-8, a Transformation Format of ISO 10646. Internet Engineering Task Force – Request for Comments 3629 (2003)
18. World Wide Web Consortium.: RDF Test Cases, W3C Recommendation, `http://www.w3.org/TR/2004/REC-rdf-testcases-20040210/` (2004)

Rating, recognizing and rewarding metadata integration and sharing on the semantic web

Francisco M. Couto

LASIGE, Dept. de Informática, Faculdade de Ciências, Universidade de Lisboa, Portugal
fcouto@di.fc.ul.pt

Abstract. Research is increasingly becoming a data-intensive science, however proper data integration and sharing is more than storing the datasets in a public repository, it requires the data to be organized, characterized and updated continuously. This article assumes that by rewarding and recognizing metadata sharing and integration on the semantic web using ontologies, we are promoting and intensifying the trust and quality in data sharing and integration. So, the proposed approach aims at measuring the knowledge rating of a dataset according to the specificity and distinctiveness of its mappings to ontology concepts.
The knowledge ratings will then be used as the basis of a novel reward and recognition mechanism that will rely on a virtual currency, dubbed KnowledgeCoin (KC). Its implementation could explore some of the solutions provided by current cryptocurrencies, but KC will not be a cryptocurrency since it will not rely on a cryptographic proof but on a central authority whose trust depends on the knowledge rating measures proposed by this article. The idea is that every time a scientific article is published, KCs are distributed according to the knowledge rating of the datasets supporting the article.

Keywords: Data Integration, Data Sharing, Linked Data, Metadata, Ontologies

1 Introduction

Research is increasingly becoming a data-intensive science in several areas, where prodigious amounts of data can be collected from disparate resources at any time [6]. However, the real value of data can only be leveraged through its trust and quality, which ultimately results in the acquisition of knowledge through its analysis. Since multiple types of data are involved, often from different sources and in heterogeneous formats, data integration and sharing are key requirements for an efficient data analysis. The need for data integration and sharing has a long-standing history, and besides the big technological advances it still remains an open issue. For example, in 1985 the Committee on Models for Biomedical Research proposed a structured and integrated view of biology to cope with the available data [8]. Nowadays, the BioMedBridges [1] initiative aims at constructing the data and service bridges needed to connect the emerging biomedical sciences research infrastructures (BMSRI), which are on the roadmap of the European Strategy Forum on Research Infrastructures (ESFRI). One common theme to

[1] www.biomedbridges.eu

all BMSRIs is the definition of the principles of data management and sharing [3]. The Linked Data initiative [2] already proposed a well-defined set of recommendations for exposing, sharing and integrating data, information and knowledge using semantic web technologies. In this paradigm data integration and sharing is achieved in the form of links connecting the data elements themselves and adding semantics to them. Following and understanding the links between data elements in publicly available Data Linked stores (Linked Data Cloud) enables us to access the data and knowledge shared by others. The Linked Data Cloud offers an effective solution to break down data silos; however the systematic usage of these technologies requires a strong commitment from the research community.

Promoting the trust and quality of data through their proper integration and sharing is essential to avoid the creation of silos that store raw data that cannot be reused by others, or even by the owners themselves. For example, the current lack of incentive to share and preserve data is sometimes so problematic, that there are even cases of authors that cannot recover the data associated with their own published works [5]. However, the problem is how to obtain a proactive involvement of the research community in data integration and sharing. In 2009, Tim Berners-Lee gave a TED talk[3], where he said: "you have no idea the number of excuses people come up with to hang onto their data and not give it to you, even though you've paid for it as a taxpayer." Public funding agencies and journals may enforce the data-sharing policies, but the adherence to them is most of the times inconsistent and scarce [1]. Besides all the technological advances that we may deliver to make data integration and sharing tasks easier, researchers need to be motivated to do it correctly. For example, due to the Galileos strong commitment to the advance of Science, he integrated the direct results of his observations of Jupiter with careful and clear descriptions of how they were performed, which he shared in Sidereus Nuncius [4]. These descriptions enabled other researchers not only to be aware of Galileos findings but also to understand, analyze and replicate his methodology. This is another situation that we could characterize with the famous phrase "That's one small step for a man, one giant leap for mankind." Now let us imagine if we could extend Galileos commitment to all the research community, the giant leap that it could bring to the advance of science.

Thus the commitment of the research community to data integration and sharing is currently a major concern, and this explains why BMSRIs have recently included in their definition of the principles of data management and sharing the following challenge: "to encourage data sharing, systematic reward and recognition mechanisms are necessary". They suggest studying not only measurements of citation impact, but also highlighting the importance to investigate other mechanisms as well. Systematic reward and recognition mechanisms should motivate the researchers in a way that they become strongly committed in sharing data, so others can easily understand and reuse it. By doing so, we encourage the research community to improve previous results by replicating the experiments and testing new solutions. However, before developing a reward and recognition mechanism we must formally define: i) what needs to be rewarded and recognized; ii) and measure its value in a quantitative and objective way.

[2] http://linkeddata.org/

[3] http://www.ted.com/talks/tim_berners_lee_on_the_next_web

2 Metadata Quality

Proper data integration and sharing is more than storing the datasets in a public repository, it requires the data to be organized and characterized in a way that others can find it and reuse it effectively. In an interview[4] to Nature, Steven Wiley emphasized that sharing data "is time-consuming to do properly, the reward systems aren't there and neither is the stick". Not adding links to external resources hampers the efficient retrieval and analysis of data, and therefore its expansion and update. Making a dataset easier to find and access is also a way to improve its initial trust and quality, as more studies analyze, expand and update it. Like the careful and clear descriptions provided by Galileo, semantic characterizations in the form of metadata must also be present so others can easily find the raw data and understand how it can be retrieved and explored.

Metadata is a machine-readable description of the contents of a resource made through linking the resource to the concepts that describe it. However, to fully understand such diverse and large collections of raw data being produced, their metadata need to be integrated in a non-ambiguous and computational amenable way [9, 13]. Ontologies can be loosely defined as "a vocabulary of terms and some specification of their meaning" [7, 14]. If an ontology is accepted as a reference by the community (e.g., the Gene Ontology), then its representation of its domain becomes a standard, and data integration and sharing facilitated. The complex process of enriching a resource with metadata by means of semantically defined properties pointing to other resources often requires human input and domain expertise. Thus, the proposed approach assumes that by rewarding and recognizing metadata sharing and integration on the semantic web using standard and controlled vocabularies, we are promoting and intensifying scientific collaboration and progress.

Figure 1 illustrates the Semantic Web in action with two datasets annotated with its respective metadata using a hypothetical Metal Ontology. A dataset including Gold Market Stats contains an ontology mapping (e.g., an RDF triple) to the concept Gold, and another dataset Silver Market Stats contains an ontology mapping to the concept Silver. Given that Gold and Silver are both coinage metals, a semantic search engine is capable of identifying as relevant both datasets when asked for market stats of coinage metals.

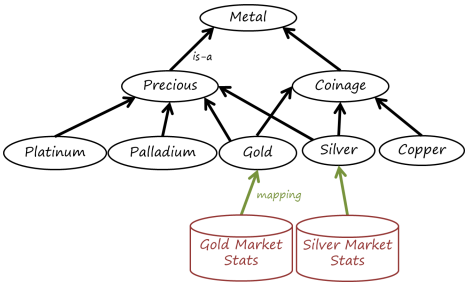


Fig. 1. An hypothetical metal ontology and dataset mappings.

Now, we need to define the value of metadata in terms of knowledge it provides about a given dataset. Semantic interoperability is a key requirement in the realization

[4] http://www.nature.com/news/2011/110914/full/news.2011.536.html

of the semantic web and it is mainly achieved through mappings between resources. For example, all dataset mappings to ontology concepts are to some extent important to enhance the retrieval of that dataset, but the level of importance varies across mappings. The proposed approach assumes that metadata can be considered as a set of links where all the links are equal, but some links are more equal than others (adaption of George Orwells quote). Thus, the proposed approach aims at measuring the knowledge rating of any given dataset through its mappings to concepts specified in an ontology.

3 Knowledge rating

The proposed approach assumes that the metadata integration and sharing value of a dataset, dubbed as **knowledge rating**, is proportional to the **specificity** and **distinctiveness** of its mappings to ontology concepts in relation to all the others datasets in the Linked Data Cloud.

The specificity of a set of ontology concepts can be defined by the information content (IC) of each concept, which was introduced by [11]. For example, intuitively the concept dog is more specific than the concept animal. This can be explained because the concept animal can refer to many distinct ideas, and, as such, carries a small amount of information content when compared to the concept dog, which has a more informative definition. The distinctiveness of a set of ontology concepts can be defined by its conceptual similarity [2,12] to all the others sets of ontology concepts, i.e. a distinctiveness of a dataset is high if there are no other semantically similar datasets available. Conceptual similarity explores ontologies and the relationships they contain to compare their concepts and, therefore, the entities they represent. Conceptual similarity enables us to identify that arm and leg are more similar than arm and head, because an arm is a limb and a leg is also a limb. Likewise, because an airplane contains wings, the two concepts are more related to each other than wings is to boat.

Most implementations of IC and conceptual similarity only span a single domain specified by an ontology [10]. However, realistic datasets frequently use concepts from distinct domains of knowledge, since reality is rarely unidisciplinary. So, the scientific challenge is to propose innovative algorithms to calculate the IC and conceptual similarity using multiple-domain ontologies to measure the specificity and distinctiveness of a dataset. Similarity in a multiple ontology context will have to explore the links between different ontologies. Such correspondences already exist for some ontologies that provide cross-reference resources. When these resources are unavailable, ontology matching techniques can be used to automatically create them.

4 Reward and recognition mechanism

The reward and recognition mechanism can rely on the implementation of a new virtual currency, dubbed KnowledgeCoin (KC), that will be specifically designed to promote and intensify the usage of semantic web technologies for scientific data integration and sharing. The idea is that every time a scientific article is published, KCs are distributed according to the knowledge rating of the datasets supporting that article. Note that KCs

should by no means be a new kind of money and the design of KC transactions will focus on the exchange of scientific data and knowledge.

After developing the knowledge rating measures, they can be used to implement the supply algorithm of a new virtual currency, KC. This will not only aim at validating the usefulness of the proposed knowledge ratings but also deliver an efficient reward and recognition mechanism to promote and intensify the usage of semantic web technologies for scientific data integration and sharing. Unlike conventional cryptocurrencies, the KCs will rely on a trusted central authority and not on a cryptographically proof. But even without being a cryptocurrency, the KC will take advantage of the technical solutions provided by existing cryptocurrencies, such as bitcoin[5].

The scientific challenge is to create a trusted central authority that issue new KCs when new knowledge is created in the form of a scientific article, as long as it references a supporting dataset properly integrated in the Linked Data Cloud. If there is no reference to the dataset in the Linked Data Cloud no KCs will be issued. This way, researchers will be incentivized to publically share the dataset, including the raw data or at least a description of the raw data, in the Linked Data Cloud. If a dataset is shared through the Linked Data Cloud then its level of integration will be measured by its knowledge rating. This way, researchers will be encouraged to properly integrate their data. The success of this mining process will rely on the trustworthiness of the knowledge ratings, and therefore will further validate the developed measures.

From recognition researchers may get reputation, and from reputation they may get a reward. For example, researchers recognize the relevance of a research's work by citing it, and by having a high number of citations the researcher obtains a strong reputation, which may in the end help him to be rewarded with a project grant. Thus, KCs can be interpreted as a form of reputation that in the end can result in a reward. However, we can also design and implement direct reward mechanisms through KCs transactions as a way to establish a virtual marketplace of scientific data and knowledge exchanges. The main scenario of a KCs transaction is to represent the exchange of datasets identified by an URI from the data provider to the data consumer, which may include recognition statements.

5 Future Directions

The design of the approach is ongoing work and its direction depends on a more detailed analysis of many social and technical challenges that its implementation poses. For example, some of the issues that need to be further studied and discussed: i) knowledge ratings implementation, i.e. their validation, aggregation, performance, exceptions, and extension to any mappings besides the ontological ones; ii) potential abuses, such as the creation of spam mappings and other security threats; iii) central trusted authority for the KC vs. the peer-to-peer mechanisms used by bitcoin; iv) use case scenarios for the KC, e.g. exchange of datasets and their characterization based on KC transactions.

In a nutshell, this paper presents the guidelines for delivering sound knowledge rating measures to serve as the basis of a systematic reward and recognition mechanism

[5] `http://bitcoin.org/`

based on KCs for improving the trust and quality of data through proper data integration and sharing on the semantic web. The proposed idea aims to be the first step in providing an effective solution towards data silos extinction.

Acknowledgments

The anonymous reviewers for their valuable comments and suggestions. Work funded by the Portuguese FCT through the LASIGE Strategic Project (PEst-OE/EEI/UI0408/2014) and SOMER project (PTDC/EIA-EIA/119119/2010).

References

1. Alsheikh-Ali, A.A., Qureshi, W., Al-Mallah, M.H., Ioannidis, J.P.: Public availability of published research data in high-impact journals. PloS one 6(9), e24357 (2011)
2. Couto, F.M., Pinto, H.S.: The next generation of similarity measures that fully explore the semantics in biomedical ontologies. Journal of bioinformatics and computational biology 11(05) (2013)
3. ELIXIR, EU-OPENSCREEN, BBMRI, EATRIS, ECRIN, INFRAFRONTIER, INSTRUCT, ERINHA, EMBRC, Euro-BioImaging, LifeWatch, AnaEE, ISBE, MIRRI: Principles of data management and sharing at European Research Infrastructures (DOI:105281/zenodo8304, Feb 2014)
4. Galilei, G.: Sidereus Nuncius, or The Sidereal Messenger. University of Chicago Press (1989)
5. Goodman, A., Pepe, A., Blocker, A.W., Borgman, C.L., Cranmer, K., Crosas, M., Di Stefano, R., Gil, Y., Groth, P., Hedstrom, M., et al.: Ten simple rules for the care and feeding of scientific data. PLoS computational biology 10(4), e1003542 (2014)
6. Hey, A.J., Tansley, S., Tolle, K.M.: The fourth paradigm: data-intensive scientific discovery (2009)
7. Jasper, R., Uschold, M., et al.: A framework for understanding and classifying ontology applications. In: Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW. vol. 99, pp. 16–21 (1999)
8. National Research Council (US). Committee on Models for Biomedical Research: Models for biomedical research: a new perspective. National Academies (1985)
9. Noy, N.F.: Semantic integration: a survey of ontology-based approaches. ACM Sigmod Record 33(4), 65–70 (2004)
10. Pedersen, T., Pakhomov, S.V., Patwardhan, S., Chute, C.G.: Measures of semantic similarity and relatedness in the biomedical domain. Journal of biomedical informatics 40(3), 288–299 (2007)
11. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th international joint conference on Artificial intelligence-Volume 1. pp. 448–453. Morgan Kaufmann Publishers Inc. (1995)
12. Rodríguez, M.A., Egenhofer, M.J.: Determining semantic similarity among entity classes from different ontologies. Knowledge and Data Engineering, IEEE Transactions on 15(2), 442–456 (2003)
13. Uschold, M., Gruninger, M.: Ontologies and semantics for seamless connectivity. ACM SIGMod Record 33(4), 58–64 (2004)
14. Uschold, M., Gruninger, M.: Ontologies: Principles, methods and applications. The knowledge engineering review 11(02), 93–136 (1996)

Towards the Definition of an Ontology for Trust in (Web) Data

Davide Ceolin, Archana Nottamkandath,

Wan Fokkink, and Valentina Maccatrozzo

VU University, Amsterdam, The Netherlands
{d.ceolin,a.nottamkandath,w.j.fokkink,v.maccatrozzo}@vu.nl

Abstract. This paper introduces an ontology for representing trust that extends existing ones by integrating them with recent trust theories. Then, we propose an extension of such an ontology, tailored for representing trust assessments of data, and we outline its specificity and its relevance.

Keywords: Trust, Ontology, Web Data, Resource Definition Framework (RDF)

1 Introduction

In this paper we tackle the problem of modeling and representing trust assertions, in particular about (Web) data. This is an important issue for a variety of reasons. First, trust is an important aspect both of everyday life and of many computational approaches, for similar reasons. In fact, trust is a "leap of faith"[1] that is necessary to be taken whenever we need to rely on third party agents or information. We decide whether or not to take this leap of faith based on the evaluation of the trustworthiness of the agent or information. In general, when trusting, a risk is involved, i.e., the risk of relying on uncertain and possibly unpredictable actions or information. We can soften such a risk, and one way to achieve this result is to share trust and trustworthiness values, along with their provenance, to allow their reuse and increase the probability to correctly place trust thanks to the availability of this information. Therefore, an ontology for trust assessments, in particular of Web data, can indicate the basic elements that are necessary to define a trust value.

This paper aims at introducing an ontology for trust representation, starting from existing ones and extending them to cover aspects indicated by recent trust theories. In Section 2 we present related work, in Section 3 we provide a summary of the trust theory of O'Hara that we use in the rest of the paper, in Section 4 we propose an extended ontology for representing trust and in Section 5 we expose our vision about the issues of trusting (Web) data. Lastly, we conclude in Section 6.

[1] Stephen Marsh, "Trust: Really, Really Trust", IFIP Trust Management Conference 2014 Tutorial

2 Related Work

Trust is a widely explored topic within a variety of computer science areas. Here, we focus on those works directly touching upon the intersection of trust, reputation and the Web. We refer the reader to the work of Sabater and Sierra [19], Artz and Gil [2], and Golbeck [12] for comprehensive reviews about trust in artificial intelligence, Semantic Web and Web respectively. Trust has also been widely addressed in the agent systems community. Pinyol and Sabater-Mir provide an up-to-date review of the literature in this area [18].

We extend the ontology proposed by Alnemr et al. [1]. We choose it because: (1) it focuses on the computational part of trust, rather than on social and agent aspects that are marginal to our scope, and (2) it already presents elements that are useful to represent computational trust elements. Nevertheless, we propose to extend it to cover at least the main elements of the trust theory of O'Hara, that are missing from their original ontology, and we highlight how these extensions can be beneficial to model trust in (Web) data. Viljanen [21] envisions the possibility to define an ontology for trust, but puts a particular emphasis on trust between people or agents. Heath and Motta [14] propose an ontology for representing expertise, thus allowing us to represent an important aspect of trust, but again posing more focus on the agents rather than on the data. A different point of view is taken by Sherchan et al. [20], who propose an ontology for modeling trust in services.

Goldbeck at al. [13], Cesare et al. [5] and Huang et al. [15] propose ontologies for modeling trust in agents. Although these could be combined with the ontology we propose (e.g., to model the trust in the author of a piece of data), for the moment their focus falls outside of the scope of our work, that is trust in data.

Trust has been modeled also in previous works of ours [7,6,8,9] using generic models (e.g., the Open Annotation Model [4] or the RDF Data Cube Vocabulary [10]). Here we aim at providing a specific model for representing trust.

3 A Definition of Trust in Short

We recall here the main elements of "A Definition of Trust" by O'Hara [17], that provide the elements of trust we use to extend the ontology of Alnemr et al.

Tw<Y,Z,R(A),C> (Trustworthiness) agent Y is willing, able and motivated to behave in such a way as to conform to behaviour R, to the benefit of members of audience A, in context C, as claimed by agent Z.

Tr<X,Y,Z,I(R(A),c),Deg,Warr> (Trust attitude) X believes, with confidence Deg on the basis of warrant Warr, that Y's intentions, capacities and motivations conform to I(R[A],c), which X also believes is entailed by R(A), a claim about how Y will pursue the interests of members of A, made about Y by a suitably authorised Z.

X places trust in Y (Trust Action) X performs some action which introduces a vulnerability for X, and which is inexplicable without the truth of **Trust attitude**.

4 Extending a Trust Ontology

We are interested in enabling the sharing of the trust values regarding both trust attitude and actions, along with their provenance. The ontology proposed by Alnemr et al. [1] captures the basic computational aspects of these trust values. However we believe that it lacks some peculiar trust elements that are present in the theory of O'Hara, and thus we extend that ontology as shown in Figure 1[2]. Compared with the ontology of Alnemr et al., we provide some important additions. We clearly identify the parts involved in the trust relation:

Trustor (source): every trust assessment is made by an agent (human or not), that takes his decision based on his policy and on the evidence at his disposal;

Trustee (target): the agent or piece of information that is actually being trusted or not trusted. This class replaces the generic "Entity" class, as it emphasizes its role in the trust relation.

We also distinguish between the attitude and the act of trusting.

Trust Attitude Object: it represents the graded belief held by the trustor in the trustworthiness of the trustee and it is treated as a quality attribute when deciding if to place trust in the trustee or not. It replaces the reputation object defined by Alnemr et al. because it has a similar function to it (quantifying the trust in something), but implements the trust attitude relation defined by O'Hara that is more precise and complete (e.g. warranties are not explicitly modeled by the reputation object);

Trust Action Object: the result of the action of placing trust. Placing trust is an instantaneous action based on an "outright" belief. Therefore the trust value is likely to be a Boolean value.

Role, Context and Warranty: in the original ontology, the criterion is a generic class that contextualizes the trust value. We specialize it, to be able to model the role and the context indicated in the theory of O'Hara, as well as the evidence on which the trust value is based, by means of the warranty.

The trustworthiness relation is not explicitly modeled, since it falls outside our current focus. We discuss this further in the following section. The remaining elements of the model shown in Figure 1 are part of the original trust ontology. These include a criterion for the trust value, and an algorithm that allows combining observations (warranties) into a trust value (an algorithm is used also to determine the value of the trust action). The trust attitude value corresponds to the Deg element of the theory of O'Hara. We model the action both when it is performed and when it is not. Both trust values are modeled uniformly.

5 Modeling Trust in Data

In the previous section we provided an extended ontology that aims at capturing the basic elements that are involved in the process of taking trust decisions. Here we focus on the specificity of trusting (Web) data.

[2] The ontology is available at `http://trustingwebdata.org/ontology`

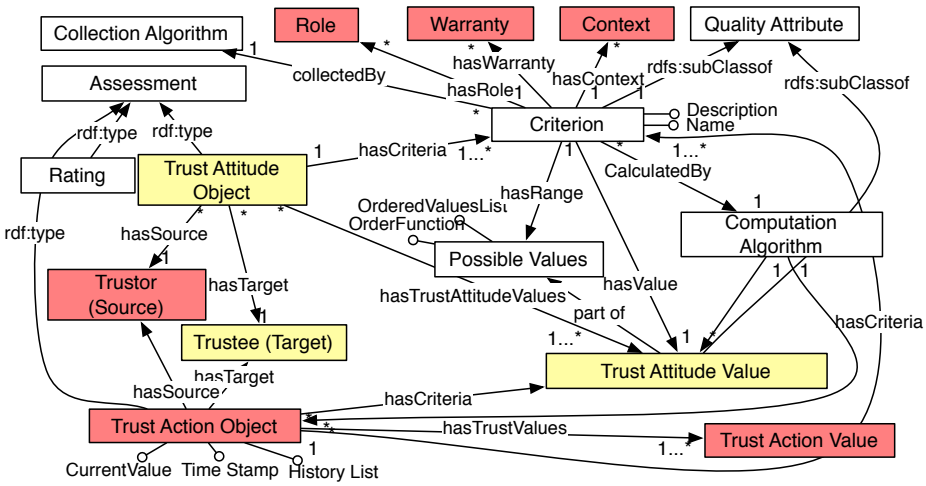


Fig. 1. Extended Trust Ontology. We highlight in red the added elements and in yellow the updated ones.

Data are used as information carriers, so actually trusting data does not mean to place trust in a sequence of symbols. It rather means to place trust in the interpretation of such a sequence of symbols and on the basis of its trustworthiness. For instance, consider a painting reproducing the city "Paris" and its annotation "Paris". To trust the annotation, we must have evidence that the painting actually corresponds to the city Paris. But, to do so, we must: (1) give the right interpretation to the word "Paris" (e.g., there are 26 US cities and towns named "Paris"), and (2) check if one of the possible interpretations is correct. Both in the case the picture represents another city or in the case the picture represents a town named Paris which existence we ignored, we would not place trust in the data, but for completely different reasons. One possible RDF representation of the above example is: `exMuseum:ParisPainting ex:depicts dbpedia:Paris`, where we take for granted the correctness of the subject and of the property and, if we accept the triple, we do so because we believe in the correctness of the object in that context (represented by the subject), and role (represented by the property). We make use of the semantics of RDF 1.1 [22], from which we recall the elements of a simple interpretation I of an RDF graph:

1. A non-empty set IR of resources, called the domain or universe of I.
2. A set IP, called the set of properties of I.
3. A mapping IEXT : IP $\rightarrow \mathcal{P}$ (IR $\times$ IR).
4. A mapping IS: IRIs $\rightarrow$ (IR $\cup$ IP). An IRI (Internationalized Resource Identifier [11]) is a generalization of a URI [3].
5. A partial mapping IL from literals into IR

Also, the following semantic conditions for ground graphs hold:

a. if E is a literal then I(E) = IL(E)
b. if E is an IRI then I(E) = IS(E)
c. if E is a ground triple s p o. then I(E) = true if I(p) ∈ IP and the pair <I(s),I(o)> ∈ IEXT(I(p)) otherwise I(E) = false.
d. if E is a ground RDF graph then I(E) = false if I(E') = false for some triple E' ∈ E, otherwise I(E) = true.

Items 1, 2, 4 and 5 map the URIs, literals and the RDF triples to real-world objects. We are particularly interested in Item 3, that maps the property of an RDF triple to the corresponding real-world relation between subject and object. Trusting a piece of data means to place trust in the information it carries, in a given context. The trust context can be represented by means of the subject and object of an RDF triple, so their semantic interpretation is assumed to be known by the trustor. If the trustor trusts the triple, he believes that the interpretation of the object *o* makes the interpretation of the triple *s p o* true:

$$TrustAttitude_{trustor}(o|s,p) = Belief_{trustor}(\exists I(o) : <\text{I(s)},\text{I(o)}> \in IEXT(I(p))$$

Belief is an operator that maps logical propositions to values that quantify their believed truth, e.g., by means of subjective opinions [16] quantified in the Deg value of the theory of O'Hara and based on evidence expressed by Warranty.

By virtue of items *c* and *d*, we do not model explicitly the trustworthiness relation defined by O'Hara: we consider an object *o* to be trustworthy by virtue of the fact that it is part of an RDF triple that is asserted.

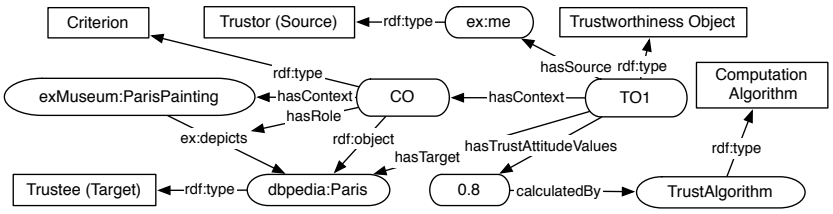


Fig. 2. Snapshot of the trust ontology, specialized for modeling data trustworthiness.

Figure 2 presents a snapshot of the trust ontology modeling the example above and adding a trust attitude value computed with a sample trust algorithm.

6 Conclusion

In this paper we introduce an ontology for trust representation that extends an existing model with recent trust theories. We specialize it in order to model data-related trust aspects, and we motivate our design choices based on standard RDF 1.1 semantics. This model is still at a very early stage, but it emerges from previous research and from standard trust theories. In the future, it will be extended, and evaluated in depth, also by means of concrete applications.

References

1. R. Alnemr, A. Paschke, and C. Meinel. Enabling reputation interoperability through semantic technologies. In *I-SEMANTICS*, pages 1–9. ACM, 2010.
2. D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Journal of Semantic Web*, 2007.
3. T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax (RFC 3986). Technical report, IETF, 2005. `http://www.ietf.org/rfc/rfc3986.txt`.
4. S. Bradshaw, D. Brickley, L. J. G. Castro, T. Clark, T. Cole, P. Desenne, A. Gerber, A. Isaac, J. Jett, T. Habing, B. Haslhofer, S. Hellmann, J. Hunter, R. Leeds, A. Magliozzi, B. Morris, P. Morris, J. van Ossenbruggen, S. Soiland-Reyes, J. Smith, and D. Whaley. Open Annotation Core Data Model. `http://www.openannotation.org/spec/core`, 2012. W3C Community Draft.
5. S. J. Casare and J. S. Sichman. Towards a functional ontology of reputation. In *AAMAS*, pages 505–511. ACM, 2005.
6. D. Ceolin, A. Nottamkandath, and W. Fokkink. Automated Evaluation of Annotators for Museum Collections using Subjective Logic. In *IFIPTM 2012*, pages 232–239. Springer, May 2012.
7. D. Ceolin, A. Nottamkandath, and W. Fokkink. Semi-automated Assessment of Annotation Trustworthiness. In *PST 2013*. IEEE, 2013.
8. D. Ceolin, A. Nottamkandath, and W. Fokkink. Efficient semi-automated assessment of annotations trustworthiness. *Journal of Trust Management*, 1(1):3, 2014.
9. D. Ceolin, W. R. van Hage, and W. Fokkink. A Trust Model to Estimate the Quality of Annotations using the Web. In *WebSci 2010*. Web Science Trust, 2010.
10. R. Cyganiak, D. Reynolds, and J. Tennison. The rdf data cube vocabulary. Technical report, W3C, 2014.
11. M. Dürst and M. Suignard. Internationalized Resource Identifiers (IRIs) (RFC 3987). Technical report, IETF, 2005. `http://www.ietf.org/rfc/rfc3987.txt`.
12. J. Golbeck. Trust on the World Wide Web: A Survey. *Foundations and Trends in Web Science*, 1(2):131–197, 2006.
13. J. Golbeck, B. Parsia, and J. A. Hendler. Trust networks on the semantic web. In *CIA*, pages 238–249. Springer, 2003.
14. T. Heath and E. Motta. The Hoonoh Ontology for describing Trust Relationships in Information Seeking. In *PICKME 2008*. CEUR-WS.org, 2008.
15. J. Huang and M. S. Fox. An ontology of trust: Formal semantics and transitivity. In *ICEC '06*, pages 259–270. ACM, 2006.
16. A. Jøsang. A logic for uncertain probabilities. *Intl. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–212, 2001.
17. K. O'Hara. A General Definition of Trust. Technical report, University of Southampton, 2012.
18. I. Pinyol and J. Sabater-Mir. Computational trust and reputation models for open multi-agent systems: A review. *Artif. Intell. Rev.*, 40(1):1–25, June 2013.
19. J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24:33–60, 2005.
20. W. Sherchan, S. Nepal, J. Hunklinger, and A. Bouguettaya. A trust ontology for semantic services. In *IEEE SCC*, pages 313–320. IEEE Computer Society, 2010.
21. L. Viljanen. Towards an ontology of trust. In *Trust, Privacy, and Security in Digital Business*, pages 175–184. Springer, 2005.
22. W3C. RDF 1.1 Semantics. `http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/`, 2014.