The 2nd International Workshop on Inductive Reasoning and Machine Learning for the Semantic Web

Proceedings

edited by Claudia **d'Amato**
Nicola **Fanizzi**
Marko **Grobelnik**
Agnieszka **Ławrynowicz**
Vojtěch **Svátek**

Heraklion, May 31, 2010

(this page is intentionally left blank)

Foreword

Large amounts of data increasingly becoming available and described using real-life ontologies represented in Semantic Web languages, recently opened up the possibility for interesting real-world data mining applications on the Semantic Web. However, exploiting this global resource of data requires new kinds of approaches for data mining and data analysis that would be able to deal at the same time with its scale and with the complexity, expressiveness, and heterogeneity of the representation languages, leverage on availability of ontologies and explicit semantics of the resources, and account for novel assumptions (e.g., "open world") that underlie reasoning services within the Semantic Web.

The workshop tried to address the above issues, in particular focusing on the problems of how machine learning techniques, such as statistical learning methods and inductive forms of reasoning, can work directly on the richly structured Semantic Web data and exploit the Semantic Web technologies, what is the value added of machine learning methods for the Semantic Web, and what are the challenges for developers of machine learning techniques for the Semantic Web data, for example in the area of ontology mining.

The workshop was meant to bring together researchers and practitioners interested in the interdisciplinary research on the intersection of the Semantic Web with Knowledge Discovery and Machine Learning, and provide a meeting point for the related communities to stimulate collaboration and enable cross-fertilization of ideas.

Specifically, the review phase produced a selection of 5 full papers, 1 position paper, and 2 late breaking news abstracts. IRMLeS 2010 program was further enriched by two invited talks from prominent researchers. Dr Melanie Hilario presented in her talk an ongoing research on optimizing the knowledge discovery process through the semantic meta-mining, involving background ontology representing the domain of data mining. Professor Steffen Staab demonstrated in his talk how the enrichment of Web 2.0 data by automatically discovered semantic relationships may improve the user experience. The workshop was also successful in terms of registrations and attendance.

The topics covered by IRMLeS 2010 included: ontology learning, and semantic tagging to expose the semantics of unstructured or semi-structured data as text, or Web 2.0 tags; management, and retrieval of Semantic Web resources, e.g. RDF data; probabilistic approaches; similarity measures for ontological data; inductive reasoning with ontologies; finally using ontologies, and other formal representations as background knowledge to steer whole knowledge discovery process.

In the final wrap-up discussion, a number of open problems and promising directions were elicited. Similarly as the last year, the topic of integration of induction and deduction has been recognized as essential for the Semantic Web, to deal with real, noisy data. Related to this topic, the topics of probabilistic approaches, and uncertain inference over semantic resources were discussed. The need for new metrics for evaluating the output of machine learning methods in the Semantic Web setting was recognized, especially in the context of the open

world assumption. The novel topic of semantic data mining also gained attention during discussion, and a call for gathering the community of people working on ontologies/another KR formats for representing data mining domain has been issued. Some other new tasks have also been identified as an interesting future direction of research on machine learning for the Semantic Web that include: ontology repair, and instance matching (especially in the context of a lack of unique name assumption on the Semantic Web).

Given such open issues and the success of the two first editions, we plan to organize next edition in the near future.

Acknowledgments The workshop chairs are grateful to all the people who contributed to the event from the program committee members to the additional reviewers, the presenters and the participants. A special thank is due to the invited speakers who shared their vision on the topics of the workshop. Finally, we are grateful also to the ESWC 2010 Workshop Chairs, Program Chairs and General Chair for their constant support.

Heraklion, May 31, 2010

Claudia d'Amato
Nicola Fanizzi
Marko Grobelnik
Agnieszka Ławrynowicz
Vojtěch Svátek

Program Committee Members

- Sarabjot S. Anand – University of Warwick
- Bettina Berendt – Katholieke Universiteit Leuven
- Abraham Bernstein – University of Zurich
- Floriana Esposito – University of Bari
- Mohand-Said Hacid – University Lyon 1
- Melanie Hilario – University of Geneva
- Andreas Hotho – University of Kassel
- Jose Iria – IBM Research, Zurich
- Ross D. King – University of Aberystwyth
- Jens Lehmann – University of Leipzig
- Francesca A. Lisi – University of Bari
- Thomas Lukasiewicz – Oxford University
- Matthias Nickles – University of Bath
- Sebastian Rudolph – University of Karlsruhe
- Jetendr Shamdasani – University of the West of England
- Steffen Staab – University of Koblenz-Landau
- Umberto Straccia – ISTI-CNR, Pisa
- Volker Tresp – Siemens, Munich

Additional Reviewers

Jörg-Uwe Kietz – University of Zurich

Workshop Homepage

`http://irmles.di.uniba.it/2010/`

(this page is intentionally left blank)

Invited Talk Abstracts

Optimizing the Knowledge Discovery Process through Semantic Meta-Mining

Melanie Hilario

Computer Science Department
University of Geneva
Geneva, Switzerland

Abstract. I will describe a novel meta-learning approach to optimizing the knowledge discovery or data mining (DM) process. This approach has three features that distinguish it from its predecessors. First, previous meta-learning research has focused exclusively on improving the learning phase of the DM process. More specifically, the goal of meta-learning has typically been to select the most appropriate algorithm and/or parameter settings for a given learning task. We adopt a more process-oriented approach whereby meta-learning is applied to design choices at different stages of the complete data mining process or workflow (hence the term meta-mining). Second, meta-learning for algorithm or model selection has consisted mainly in mapping dataset properties to the observed performance of algorithms viewed as black boxes. While several generations of researchers have worked intensively on characterizing datasets, little has been done to understand the internal mechanisms of the algorithms used. At best, a few have considered perceptible features of algorithms like their ease of implementation or their robustness to noise, or the interpretability of the models they produce. In contrast, our meta-learning approach complements dataset descriptions with an in-depth analysis and characterization of algorithms - their underlying assumptions, optimization goals and strategies, together with the structure and complexity of the models and patterns they generate. Third, previous meta-learning approaches have been strictly (meta) data-driven. To make sense of the intricate relationships between tasks, data and algorithms at different stages of the data mining process, our meta-miner relies on extensive background knowledge concerning knowledge discovery itself. For this reason we have developed a data mining ontology, which defines the essential concepts and relations needed to represent and analyse data mining objects and processes. In addition, a DM knowledge base gathers assertions concerning data preprocessing and machine learning algorithms as well as their implementations in several open-source software packages. The DM ontology and knowledge base are domain-independent; they can be exploited in any application area to build databases describing domain-specific data analysis tasks, datasets and experiments. Aside from their direct utility in their respective target domains, such databases are the indispensable source of training and evaluation data for the meta-miner. These three features together lay the groundwork for semantic meta-mining, the process of mining DM meta-data on the basis of data mining expertise distilled in an ontology and knowledge base.

From Web 2.0 to Web 3.0 using Data Mining

Steffen Staab

Institute WeST - Web Science and Technologies &
Institute for Computer Science
University of Koblenz-Landau
Koblenz, Germany

Abstract. Web 2.0 applications such as Flickr offer a rich set of data with a huge potential for exploitation by the human users. Unfortunately, the sifting through such data is far from easy and rewarding due to a lack of semantics on the one side and a lack of rich data description on the other side. For instance, most photos on Flickr have very little description attached that could be used for retrieving or exploring the photos. In this talk, we demonstrate how the enrichment of Web 2.0 data by automatically discovered (more or less) semantic relationships improves the user experience.

(this page is intentionally left blank)

Full and Position Papers

Structural Similarity in Expressive Description Logics: An Extended Family of Kernels for OWL

Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito

LACAM – Dipartimento di Informatica, Università degli studi di Bari
{fanizzi|claudia.damato|esposito}@di.uniba.it

Abstract. In the context of the Semantic Web many applications of inductive inference ultimately rely on a notion of similarity for the standard knowledge representations of the ontologies. We have tackled the problem of statistical learning with ontologies proposing a family of structural kernels for $\mathcal{ALCN}$ that has been integrated with support vector machines. Here we extend the definition of the kernels to more expressive languages of the family, namely those backing OWL.

1 Concept Similarity in Ontologies

Although *machine learning* techniques may have a great potential for the Semantic Web (SW) applications, *ontology learning* tasks have focused mainly on methods for text [4]. Much less effort has been devoted to the application of machine learning methods to knowledge bases described in formal concept representations of the SW (formal *ontologies*) ultimately based on *Description Logics* (DL) [1].

In order to apply statistical learning some notion of similarity for the specific representation is needed [8]. It is particularly appealing to work with kernel methods that allow for decoupling the final learning algorithm (e.g. perceptrons, support vector machines, etc.) from the notion of similarity which depends on the particular instance space which is encoded in the kernel function. We have been investigating on the definition of kernel functions for instance spaces represented in relational languages such as those based on clauses and lately also on DLs. One of the first works concerns kernels for the *Feature Description Logic* [6] which proved particularly effective for relational structures elicited from text. More recently further proposals for working with more complex DL and ontology languages have been made [10, 2, 13].

In this work, we propose a family of declarative kernel functions that can be applied to DLs representations with different degrees of expressiveness. The kernels encode a notion of similarity of individuals in this representation, based on structural and semantic aspects of the reference representation. Specifically, we extend the definition of a family of kernels for the $\mathcal{ALCN}$ logic [12]. These kernel functions are based on both structural and also semantic aspects, namely a normal form and the extension of the concepts involved in the description, as elicited from the knowledge base.

As such the kernels are designed for comparing concept descriptions. In order to apply them to instance comparison w.r.t. real ontologies, that likely exploit the full extent of the most expressive DL languages, (upper) approximations of the most specific concepts that cover the single individuals had to be computed [5]. This exposes the method to a number of problems. Firstly, a (partially) structural normal form may fail to fully capture the semantic similarity between the individuals. Scaling up to more complex languages wold require specific normal forms. Moreover, the existence of a most specific concept is not guaranteed [1].

Coupling valid kernels with efficient algorithms such as the support vector machines, many tasks based on inductive classification can be tackled. Particularly, we demonstrate how to perform important inferences on semantic knowledge bases, namely concept retrieval and query answering. These tasks are generally grounded on merely deductive procedures which easily fail in case of (partially) inconsistent or incomplete knowledge. The methods has been shown to work comparably well w.r.t. standard reasoners [12, 13], allowing the suggestion of new knowledge that was not previously logically derivable.

Moreover, these kernels naturally induce distance measures which allow for extensions to further metric-based learning methods such as nearest-neighbor classification [7] and conceptual clustering [11]. However these extensions are beyond the scope of this work.

2 Preliminaries on Representation and Inference

The basics of the DLs will be briefly recalled. The reader may refer to the DL handbook [1] for a thorough reference. Such representations provide the basic constructors adopted by the standard ontology languages employed in the SW, such as the Ontology Markup Language OWL. We will focus on the $\mathcal{ALCQ}$ logic which may represent a tradeoff between expressiveness and reasoning efficiency.

2.1 Knowledge Bases in Description Logics

Let us consider a triple $\langle N_C, N_R, N_I \rangle$ made up. respectively, by a set of *primitive concept* names N_C, to be interpreted as sets of objects in a certain domain, a set of *primitive role* names N_R, to be interpreted as binary relationships between the mentioned objects, and a set of individual names N_I for the objects themselves.

The semantics of the descriptions is defined by an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set, the *domain* of the interpretation, and $\cdot^{\mathcal{I}}$ is the *interpretation function* that maps each individual $a \in N_I$ to a domain object $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each primitive concept name $A \in N_C$ to its *extension* $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and for each $R \in N_R$ the extension is a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Complex descriptions can be built in $\mathcal{ALCQ}$ using the language constructors listed in Table 1, along with their semantics derived from the interpretation of atomic concepts and roles [1]. The *top* concept $\top$ is interpreted as the whole domain $\Delta^{\mathcal{I}}$, while the *bottom* concept $\bot$ corresponds to $\emptyset$. Complex descriptions can be built in $\mathcal{ALCQ}$ using the following constructors. The language supports

Table 1. Syntax and semantics of concepts in the $\mathcal{ALCQ}$ logic.

Name	Syntax	Semantics
top concept	$\top$	$\Delta^{\mathcal{I}}$
bottom concept	$\bot$	$\emptyset$
full concept negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
concept conjunction	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
concept disjunction	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
existential restriction	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}((x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}})\}$
universal restriction	$\forall R.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}((x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}})\}$
qual. at least restriction	$\geq nR.C$	$\{x \in \Delta^{\mathcal{I}} \mid \lvert\{y \in \Delta^{\mathcal{I}} : (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\rvert\} \geq n\}$
qual. at most restriction	$\leq nR.C$	$\{x \in \Delta^{\mathcal{I}} \mid \lvert\{y \in \Delta^{\mathcal{I}} : (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\rvert\} \leq n\}$

full negation: a concept negation $\neg C$ has an extension that amounts to the complement of $C^{\mathcal{I}}$ w.r.t. the domain. The *conjunction* and *disjunction* of two concepts are simply interpreted as the intersection and union of their extensions. Concepts can be also defined as restrictions on the roles. The *existential restriction* constrains the elements of its extension to be related via a certain role R to some instances of concept C while the *value* (or *universal*) *restriction* comprises those elements who are related through R only to instances of concept C (if any). Finally, qualified numeric restrictions define concepts by constraining its instances with the minimal or maximal number of instances of concept C related through R.

OWL offers further constructors that extend the expressiveness of this language. Its DL equivalent is $\mathcal{SHOIQ}(\mathbf{D})$ [1], that extends $\mathcal{ALCQ}$ with individual classes, role hierarchies, transitive and inverse roles ($\mathcal{I}$). Besides concrete domains ($\mathbf{D}$), i.e. well-founded external data types (such as numerical types, tuples of the relational calculus, spatial regions, or time intervals), can be dealt with.

The main inference employed with these representations is assessing whether a concept *subsumes* another concept based on their semantics:

Definition 2.1 (subsumption). *Given two descriptions C and D, C* is subsumed *by D, denoted by $C \sqsubseteq D$, iff for every interpretation $\mathcal{I}$ it holds that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. When $C \sqsubseteq D$ and $D \sqsubseteq C$ then they are* equivalent*, denoted with $C \equiv D$.*

Note that this naturally induces a generality relationship on the space of concepts.

Generally subsumption is not assessed in isolation but rather related to the models of a system of axioms representing the background knowledge and the world state:

Definition 2.2 (knowledge base). *A* knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ *contains a* TBox $\mathcal{T}$ *and an* ABox $\mathcal{A}$. *$\mathcal{T}$ is the set of terminological axioms of concept descriptions $C \equiv D$, where C is the concept name and D is its description. $\mathcal{A}$ contains assertions on individuals $C(a)$ and $R(a,b)$.*

An interpretation that satisfies all its axioms is a *model* of the knowledge base.

Note that defined concepts should have a unique definition. However defining concepts through inclusions axioms ($C \sqsubseteq D$) is also generally admitted. Moreover, such definitions are assumed to be unfoldable.

Example 2.1 (royal family). This example shows a knowledge base modeling concepts and roles related to the British royal family:

$\mathcal{T} = \{$ $\texttt{Male} \equiv \neg\texttt{Female}$,
$\texttt{Woman} \equiv \texttt{Human} \sqcap \texttt{Female}$,
$\texttt{Man} \equiv \texttt{Human} \sqcap \texttt{Male}$,
$\texttt{Mother} \equiv \texttt{Woman} \sqcap \exists\texttt{hasChild}.\texttt{Human}$,
$\texttt{Father} \equiv \texttt{Man} \sqcap \exists\texttt{hasChild}.\texttt{Human}$,
$\texttt{Parent} \equiv \texttt{Father} \sqcup \texttt{Mother}$,
$\texttt{Grandmother} \equiv \texttt{Mother} \sqcap \exists\texttt{hasChild}.\texttt{Parent}$,
$\texttt{Mother-w/o-daughter} \equiv \texttt{Mother} \sqcap \forall\texttt{hasChild}.\neg\texttt{Female}$,
$\texttt{Super-mother} \equiv \texttt{Mother} \sqcap \geq 3\,\texttt{hasChild.Human}$ $\}$

$\mathcal{A} = \{$ Woman(elisabeth), Woman(diana), Man(charles), Man(edward),
Man(andrew), Mother-w/o-daughter(diana),
hasChild(elisabeth, charles), hasChild(elisabeth,edward),
hasChild(elisabeth, andrew), hasChild(diana, william),
hasChild(charles, william) $\}$

Note that the *Open World Assumption* (OWA) is made in the underlying semantics, since normally knowledge representation systems are applied in situations where one cannot assume that the knowledge in the base is complete. Although this is quite different from the standard settings considered in machine learning and knowledge discovery, it is convenient for the Semantic Web context where new resources (e.g. Web pages, Web services) may be continuously made available.

2.2 Inference Services

The basic inference services for DL knowledge bases can be viewed as entailments as they amount to verifying whether a generic relationship is a logical consequence of the knowledge bases axioms.

Typical reasoning tasks for TBoxes regards the *satisfiability* of a concept, *subsumption*, *equivalence* or *disjointness* between two concepts. For example in the knowledge base reported in Ex. 2.1, Father is subsumed by Man and is disjoint with Woman. These inferences can be reduced to each of them, hence normally reasoners support only one of them. Reasoning is performed though *tableau* algorithms [9, 1].

Inference problems concerning ABoxes are mainly related to checking its *consistency*, that is, whether it has a model, especially w.r.t. those of the TBox. For example, an ABox like the one in Ex. 2.1 containing also the assertion Father(elisabeth) is not consistent.

Since we aim at crafting inductive methods that manipulate individuals, a prototypical inference is *instance checking* [1], that amounts to checking whether an assertion α is entailed by the knowledge base ($\mathcal{K} \models \alpha$). If α is a role assertion then it is quite easy to perform the inference as normally roles do not have a definition in the TBox. However, we will focus on deciding whether an individual is an instance of a given concept. This can be easily reduced to a consistency problem: $\mathcal{K} \models C(a)$ iff $\mathcal{K} \cup \{\neg C(a)\}$ is inconsistent.

The adopted open world semantics has important consequences on the way queries are answered. While the closed-world semantics identifies a database with a single model an ABox represents possibly infinitely many interpretations. Hence a reasoner might be unable to answer certain queries because it may be actually able to build models for both a positive ($\mathcal{K} \cup \{C(a)\}$) and a negative ($\mathcal{K} \cup \{\neg C(a)\}$) answer.

Example 2.2 (open world semantics).
Given the ABox is Ex. 2.1, while it is explicitly stated that diana is a Mother-w/o-daughter, it cannot be proven for elisabeth because she may have daughters that are simply not known in the knowledge base. Analogously, elisabeth is an instance of Super-mother while this cannot be concluded for diana as only two children are known in the current knowledge base. Although these instance checks fail because of the inherent incompleteness of the knowledge state, this does not imply that the individuals belong to the negated concepts, as could be inferred in case the CWA were made.

Another related inference is *retrieval* which consists in querying the knowledge base to know the individuals that belong to a given concept:

Definition 2.3 (retrieval). *Given an knowledge base $\mathcal{K}$ and a concept C, find all individuals a such that $\mathcal{K} \models C(a)$.*

A straightforward algorithm for a retrieval query can be realized via instance checking on each individual occurring in the ABox, testing whether it is an instance of the concept.

Dually, it may be necessary to find the (most specific) concepts which an individual belongs to. This is called a *realization problem*. One especially seeks for the most specific one (up to equivalence):

Definition 2.4 (most specific concept). *Given an ABox $\mathcal{A}$ and an individual a, the* most specific concept *of a w.r.t. $\mathcal{A}$ is the concept C, denoted $\mathsf{MSC}_{\mathcal{A}}(a)$, such that $\mathcal{A} \models C(a)$ and for any other concept D such that $\mathcal{A} \models D(a)$, it holds that $C \sqsubseteq D$.*

This is a typical way to lift individuals to the conceptual level [5].

For some languages, the MSC may not be expressed by a finite description [1], yet it may be approximated by a more general concept [17]. Generally approximations up to a certain depth k of nested levels are considered, denoted MSC^k. We will generically indicate a maximal depth approximation with MSC^*. For further details see also [12].

2.3 A Normal Form for $\mathcal{ALCQ}$

Many semantically equivalent (yet syntactically different) descriptions can be given for the same concept. Equivalent concepts can be reduced to a normal form by means of rewriting rules that preserve their equivalence [1]. We will adopt a normal form extending one given for $\mathcal{ALC}$ descriptions [3], for concepts that are already in negation normal normal form.

Preliminarily, some notation is necessary for naming the various nested subparts (levels) of a concept description D:

- $\mathsf{prim}(D)$ is the set of all the primitive concepts (or their negations) at the top-level of D;
- $\mathsf{val}_R(D)$ is the[1] concept in $\mathcal{ALCQ}$ normal form in the scope of the value restriction (if any) at the top-level of D (otherwise $\mathsf{val}_R(D) = \top$);
- $\mathsf{ex}_R(D)$ is the set of the descriptions C' appearing in existential restrictions $\exists R.C'$ at the top-level conjunction of D;
- $\mathsf{min}_{R.C}(D) = \max\{n \in \mathbb{N} \mid D \sqsubseteq \geq nR.C\}$ (always a finite number);
- $\mathsf{max}_{R.C}(D) = \min\{n \in \mathbb{N} \mid D \sqsubseteq \leq nR.C\}$ (if unlimited, $\mathsf{max}_{R.C}(D) = \infty$).

A normal form may be recursively defined as follows:

Definition 2.5 ($\mathcal{ALCQ}$ normal form). *A concept description C is in $\mathcal{ALCQ}$* normal form *iff $C = \bot$ or $C = \top$ or if $C = C_1 \sqcup \cdots \sqcup C_n$ with*

$$C_i = \bigsqcap_{P \in \mathsf{prim}(C_i)} P \sqcap \bigsqcap_{R \in N_R} \left\{ \forall R.\mathsf{val}_R(C_i) \sqcap \bigsqcap_{E \in \mathsf{ex}_R(C_i)} \exists R.E \sqcap \bigsqcap_{C \in N_C} \left[\geq m^i_{R.C} R.C \sqcap \leq M^i_{R.C} R.C \right] \right\}$$

where $m^i_{R.C} = \mathsf{min}_{R.C}(C_i)$, $M^i_{R.C} = \mathsf{max}_{R.C}(C_i)$ and, for all $R \in N_R$, $\mathsf{val}_R.C(C_i)$ and every sub-description in $\mathsf{ex}_R.C(C_i)$ are, in their turn, in $\mathcal{ALCQ}$ normal form.

Example 2.3 ($\mathcal{ALCQ}$ normal form). The concept description
$C \equiv (\neg A_1 \sqcap A_2) \sqcup (\exists R_1.B_1 \sqcap \forall R_2.(\exists R_3.(\neg A_3 \sqcap B_2)))$
is in normal form, whereas the following is not:
$D \equiv A_1 \sqcup B_2 \sqcap \neg(A_3 \sqcap \exists R_3.B_2) \sqcup \forall R_2.B_3 \sqcap \forall R_2.(A_1 \sqcap B_3)$
where A_i's and B_j's are primitive concept names and the R_k's are role names.

This normal form can be obtained by means of repeated applications of equivalence preserving operations, namely replacing defined concepts with their definition as in the TBox and pushing the negation into the nested levels (*negation normal form*). This normal form induces an AND-OR tree structure for the concept descriptions in this language. This structure can be used for comparing different concepts and asses their similarity.

[1] A single one because multiple value restrictions can be gathered using the equivalence $\forall R.C \sqcap \cdots \sqcap \forall R.C_n \equiv \forall R.(C_1 \sqcap \cdots \sqcap C_n)$. Then the nested descriptions can be transposed in normal form using further rewriting rules (distributiveness, etc.. . .) [1].

3 Structural Kernels for $\mathcal{ALCQ}$

A simple way to define a kernel function for concept descriptions in normal form would require to adapt a tree kernel [18] where similarity between trees depends on the number of similar subtrees (or paths unraveled from such trees) which does not fully capture the semantic nature of expressive DLs languages.

The kernel function definition should not be based only on structural elements but also (partly) on their semantics, since different descriptions may have similar extensions and vice-versa, especially with expressive DL languages such as $\mathcal{ALCQ}$.

Normal form descriptions can be decomposed level-wise into sub-descriptions. For each level, there are three possibilities: the upper level is dominated by the disjunction (1) of concepts that, in turn, are made up of a conjunction (2) of complex or primitive (3) concepts. In the following the definition of the $\mathcal{ALCQ}$ kernel is reported.

Definition 3.1 (family of $\mathcal{ALCQ}$ kernels). *Given an interpretation $\mathcal{I}$ of $\mathcal{K}$, the $\mathcal{ALCQ}$* kernel *based on $\mathcal{I}$ is the function*[2] $k_{\mathcal{I}} : \mathcal{X} \times \mathcal{X} \mapsto \mathbf{R}$ *structurally defined as follows: given two disjunctive descriptions $D_1 = \bigsqcup_{i=1}^{n} C_i^1$ and $D_2 = \bigsqcup_{j=1}^{m} C_j^2$ in $\mathcal{ALCQ}$ normal form:*
disjunctive descriptions:

$$k_{\mathcal{I}}^d(D_1, D_2) = \lambda \sum_{i=1}^{n} \sum_{j=1}^{m} k_{\mathcal{I}}^c(C_i^1, C_j^2)$$

with $\lambda \in]0, 1]$
conjunctive descriptions:

$$\begin{aligned} k_{\mathcal{I}}^c(C^1, C^2) = & \prod_{\substack{P_1 \in \mathsf{prim}(C^1) \\ P_2 \in \mathsf{prim}(C^2)}} k_{\mathcal{I}}^p(P_1, P_2) \cdot \\ & \cdot \prod_{R \in N_R} k_{\mathcal{I}}^d(\mathsf{val}_R(C^1), \mathsf{val}_R(C^2)) \cdot \\ & \cdot \prod_{R \in N_R} \sum_{\substack{C_i^1 \in \mathsf{ex}_R(C^1) \\ C_j^2 \in \mathsf{ex}_R(C^2)}} k_{\mathcal{I}}^d(C_i^1, C_j^2) \cdot \\ & \cdot \prod_{R \in N_R} \prod_{C \in N_C} k_{\mathcal{I}}^n((\mathsf{min}_{R.C}(C^1), \mathsf{max}_{R.C}(C^1)), (\mathsf{min}_{R.C}(C^2), \mathsf{max}_{R.C}(C^2))) \end{aligned}$$

numeric restrictions:
if $\min(M_C, M_D) > \max(m_C, m_D)$

$$k_{\mathcal{I}}^n((m_C, M_C), (m_D, M_D)) = \frac{\min(M_C, M_D) - \max(m_C, m_D) + 1}{\max(M_C, M_D) - \min(m_C, m_D) + 1}$$

[2] We use the superscripts $k^{\cdot}$ for more clarity.

otherwise $k^n_{\mathcal{I}}((m_C, M_C), (m_D, M_D)) = 0$.

***primitive concepts**:*

$$k^p_{\mathcal{I}}(P_1, P_2) = k_{\text{set}}(P_1^{\mathcal{I}}, P_2^{\mathcal{I}}) = |P_1^{\mathcal{I}} \cap P_2^{\mathcal{I}}|$$

where k_{set} is the kernel for set structures defined in [14]. This case includes also the negation of primitive concepts: $(\neg P)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus P^{\mathcal{I}}$

Preliminarily, the extension of the concepts can be approximated by the cardinality of their retrieval. Note also that this is a family of kernels parameterized on an interpretation and on a real number λ.

The first kernel function (k^d) computes the similarity between disjunctive descriptions as the sum of the cross-similarities between all couples of disjuncts from the top-level of either description. The term λ is employed to lessen the contribution coming from the similarity of the sub-descriptions (i.e. amount of indirect similarity between concepts that are related to those at this level) on the grounds of the level where they occur.

The conjunctive kernel (k^c) computes the similarity between two input descriptions, distinguishing primitive concepts, those referred in value restrictions and those referred in existential restrictions. These values are multiplied reflecting the fact that all the restrictions have to be satisfied at a conjunctive level.

The similarity of the qualified numeric restrictions is simply computed (by k^n) as a measure of the overlap between the two intervals. Namely it is the ratio of the amounts of individuals in the overlapping interval and those the larger one, whose extremes are minimum and maximum. Note that some intervals may be unlimited above: $\max = \infty$. In this case we may approximate with an upper limit N greater than $|\Delta^{\mathcal{I}}| + 1$.

The similarity between primitive concepts is measured (by k^p) in terms of the intersection of their extension. Since the extension is in principle unknown, we will epistemically approximate it recurring to the notion of retrieval (see Def. 2.3). Making the *unique names assumption* on the names of the individual occurring in the ABox $\mathcal{A}$, one can consider the *canonical interpretation* [1] $\mathcal{I}$, using $\mathsf{Ind}(\mathcal{A})$ as its domain ($\Delta^{\mathcal{I}} := \mathsf{Ind}(\mathcal{A})$). Note that the ABox may be thought of as a (partially complete) graph structure where multiple instances are located accounting for a number of possible worlds.

Besides, the kernel can be normalized as follows: since the kernel for primitive concepts is essentially a set kernel it may be multiplied by a constant $\lambda_p = 1/\Delta^{\mathcal{I}}$ so that the cardinality of the intersection is weighted by the number of individuals occurring in the overall ABox. Alternatively, another choice could be parameterized on the primitive concepts of the kernel definition $\lambda_P = 1/|P_1^{\mathcal{I}} \cup P_2^{\mathcal{I}}|$ which would weight the rate of similarity (the extension intersection) measured by the kernel with the size of the concepts measured in terms of the individuals belonging to their extensions.

Discussion. Being partially based on the concept structure and only ultimately on the extensions of the concepts at the leaves of the tree, it may be objected that

the proposed kernel functions may only roughly capture the semantic similarity of the concepts. This may be well revealed by the case of input concepts that are semantically almost equivalent yet structurally different. However, it must be also pointed out that the rewriting process for putting the concepts in normal form tends to eliminate these differences. More importantly, the ultimate goal for defining a kernel is comparing individuals rather than concepts. This will be performed recurring to the most specific concepts of the individuals w.r.t. the same ABox (see Sect. 4). Hence, it was observed that semantically similar individuals tend to share the same structures as elicited from the same source.

The validity of a kernel depends on the fact that the function is *definite positive*. Yet validity can be also proven exploiting some closure properties of the class of kernel functions w.r.t. several operations [15]. Namely, the multiplication of a kernel by a constant, the addition or multiplication of kernels yields valid kernels. Thus one can demonstrate that the functions introduced above are indeed valid kernels for the given space of hypotheses. Then, exploiting these closure properties it can be proven that:

Proposition 3.1. *Given an interpretation $\mathcal{I}$, the function $k_{\mathcal{I}}$ is a valid kernel for the space $\mathcal{X}$ of $\mathcal{ALCQ}$ descriptions in normal form.*

Observe that the core function is the one on primitive concept extensions. It is essentially a set kernel [14]. The kernel functions for top-level conjunctive and disjunctive descriptions are also positive definite being essentially based on the primitive kernel. Descending through the levels there is an interleaving of the employment of these function up the the basic case of the function for primitive descriptions.

As regards the computational complexity, it is possible to show that the kernel function can be computed in time $O(|N_1||N_2|)$ where $|N_i|$, $i = 1, 2$, is the number of nodes of the concept AND-OR trees. It can computed by means of dynamic programming. It is also worthwhile to note that Knowledge Base Management Systems, especially those dedicated to storing instances [16], generally maintain information regarding concepts and instances which may further speed-up the computation.

4 Extensions

It has been objected that the kernels in this familty would not work for concepts that are equivalent yet syntactically different. However, they are not intended for assessing a concept similarity: ultimately kernel machines employed in inductive tasks need to be applied to instances described in this representation, therefore the most important extension is towards the case of individuals.

Indeed, the kernel function can be extended to the case of individuals $a, b \in \mathsf{Ind}(\mathcal{A})$ by taking into account the approximations of their MSCs (see Sect. 2.2). In this way, we move from a graph representation like the ABox portion containing an individual to an intensional tree-structured representation:

$$k_{\mathcal{I}}(a, b) = k_{\mathcal{I}}(\mathsf{MSC}^*(a), \mathsf{MSC}^*(b))$$

Note that before applying the kernel functions a sort of *completion* of the input descriptions is necessary, substituting the defined concepts with the concept descriptions corresponding to their definitions, so to make explicit the relevant knowledge concerning either individual (example).

The extension of the kernel function to more expressive DL is not trivial. DLs allowing normal form concept definitions can only be considered. Moreover, for each constructor not included in the $\mathcal{ALCQ}$ logic, a kernel definition has to be provided.

Another extension for the kernel function could be made taking into account the similarity between different relationships in a more selective way. This would amount to considering each couple of existential and value restrictions with one element from each description (or equivalently from each related AND-OR tree) and the computing the convolution of the sub-descriptions in the restriction. As previous suggested for λ, this should be weighted by a measure of similarity between the roles measured on the grounds of the available semantics. We propose therefore the following weight: given two roles $R, S \in N_R$: $\lambda_{RS} = |R^{\mathcal{I}} \cap S^{\mathcal{I}}|/|\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}|$.

All of these weighting factors need not to be mere constants. Another possible extension is considering them as functions of the depth of the nodes to be compared: $\lambda : \mathbb{N} \mapsto]0, 1]$ (e.g. $\lambda(n) = 1/n$). In this way one may control the decay of impact of the similarity of related individuals/concepts located ad more deeply nested levels.

As suggested before, the intersection could be measured on the grounds of the relative role extensions with respect to the whole domain of individuals, as follows:

$$\lambda_{RS} = \frac{|R^{\mathcal{I}} \cap S^{\mathcal{I}}|}{|R^{\mathcal{I}} \cup S^{\mathcal{I}}|}$$

It is also worthwhile to recall that some DLs knowledge bases contain also an *R-box* [1] with axioms concerning the roles, one knows beforehand that, for instance, $R \sqsubseteq S$ and compute their similarity consequently.

In order to increase the applicability of precision of the structural kernels and tackle the DL languages supporting the OWL versions, they should be able to work with inverse roles and nominals. The former may be easily accommodated by considering, in the normal form and kernels, a larger set of role names $N_R^* = N_R \cup \{S \mid \exists R \in N_R \colon\ S = R^-\}$. The latter can be dealt with with a set kernel, as in the sub-kernel for primitive concepts. Given the set of individual names O_1 and O_2: $k_{\mathcal{I}}^o(O_1, O_2) = k_{\text{set}}(O_1^{\mathcal{I}}, O_2^{\mathcal{I}}) = |O_1^{\mathcal{I}} \cap O_2^{\mathcal{I}}|$.

Finally, as discussed in [10], related distance measures can also be derived from kernel functions which essentially encode a notion of similarity between concepts and between individuals. This can enable the definition of various distance-based methods for these complex representations spanning from relational clustering [11] to instance-based methods [7].

5 Conclusions and Outlook

Kernel functions have been defined for OWL descriptions which was integrated with a SVM for inducing a statistical classifier working with the complex representations. The resulting classifier could be tested on inductive retrieval and classification problems.

The induced classifier can be exploited for predicting or suggesting missing information about individuals, thus completing large ontologies. Specifically, it can be used to semi-automatize the population of an ABox. Indeed, the new assertions can be suggested to the knowledge engineer that has only to validate their inclusion. This constitutes a new approach in the SW context, since the efficiency of the statistical and numerical approaches and the effectiveness of a symbolic representation have been combined.

The derivation of distance measures from the kernel function may enable a series of further distance-based data mining techniques such as clustering and instance-based classification. Conversely, new kernel functions can be defined transforming newly proposed distance functions for these representations, which are not language dependent and allow the related data mining methods to better scale w.r.t. the number of individuals in the ABox.

References

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

[2] S. Bloehdorn and Y. Sure. Kernel methods for mining instance data in ontologies. In K. Aberer et al., editors, *In Proceedings of the 6th International Semantic Web Conference, ISWC2007*, volume 4825 of *LNCS*, pages 58–71. Springer, 2007.

[3] S. Brandt, R. Küsters, and A.-Y. Turhan. Approximation and difference in description logics. In D. Fensel et al., editors, *Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning, KR02*, pages 203–214. Morgan Kaufmann, 2002.

[4] P. Buitelaar, P. Cimiano, and B. Magnini, editors. *Ontology Learning from Text: Methods, Evaluation And Applications*. IOS Press, 2005.

[5] W. Cohen and H. Hirsh. Learning the CLASSIC description logic. In P. Torasso et al., editors, *Proceedings of the 4th International Conference on the Principles of Knowledge Representation and Reasoning*, pages 121–133. Morgan Kaufmann, 1994.

[6] C. Cumby and D. Roth. On kernel methods for relational learning. In T. Fawcett and N.Mishra, editors, *Proceedings of the 20th International Conference on Machine Learning, ICML2003*, pages 107–114. AAAI Press, 2003.

[7] C. d'Amato, N. Fanizzi, and F. Esposito. Query answering and ontology population: An inductive approach. In S. Bechhofer et al., editors, *Proceedings of the 5th European Semantic Web Conference, ESWC2008*, volume 5021 of *LNCS*, pages 288–302. Springer, 2008.

[8] C. d'Amato, S. Staab, and N. Fanizzi. On the influence of description logics ontologies on conceptual similarity. In A. Gangemi and J. Euzenat, editors, *Proceedings of the 16th EKAW Conference, EKAW2008*, volume 5268 of *LNAI*, pages 48–63. Springer, 2008.

[9] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Deduction in concept languages: From subsumption to instance checking. *Journal of Logic and Computation*, 4(4):423–452, 1994.

[10] N. Fanizzi and C. d'Amato. A declarative kernel for $\mathcal{ALC}$ concept descriptions. In F. Esposito et al., editors, *In Proceedings of the 16th International Symposium on Methodologies for Intelligent Systems, ISMIS2006*, volume 4203 of *Lecture Notes in Computer Science*, pages 322–331. Springer, 2006.

[11] N. Fanizzi, C. d'Amato, and F. Esposito. Randomized metric induction and evolutionary conceptual clustering for semantic knowledge bases. In M. Silva et al., editors, *Proceedings of the ACM International Conference on Knowledge Management, CIKM2007*, pages 51–60, Lisbon, Portugal, 2007. ACM.

[12] N. Fanizzi, C. d'Amato, and F. Esposito. Learning with kernels in Description Logics. In F. Železný and N. Lavrač, editors, *Proceedings of the 18th International Conference on Inductive Logic Programming, ILP2008*, volume 5194 of *LNAI*, pages 210–225. Springer, 2008.

[13] N. Fanizzi, C. d'Amato, and F. Esposito. Statistical learning for inductive query answering on OWL ontologies. In A. Sheth et al., editors, *Proceedings of the 7th International Semantic Web Conference, ISWC2008*, volume 5318 of *LNCS*, pages 195–212. Springer, 2008.

[14] T. Gärtner, J. Lloyd, and P. Flach. Kernels and distances for structured data. *Machine Learning*, 57(3):205–232, 2004.

[15] D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, Department of Computer Science, University of California – Santa Cruz, 1999.

[16] I. R. Horrocks, L. Li, D. Turi, and S. K. Bechhofer. The instance store: DL reasoning with large numbers of individuals. In V. Haarslev and R. Möller, editors, *Proceedings of the 2004 Description Logic Workshop, DL 2004*, volume 104 of *CEUR Workshop Proceedings*, pages 31–40. CEUR, 2004.

[17] T. Mantay. Commonality-based ABox retrieval. Technical Report FBI-HH-M-291/2000, Department of Computer Science, University of Hamburg, Germany, 2000.

[18] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

A Ranking-Based Approach to Discover Semantic Associations Between Linked Data

María-Esther Vidal[1] and Louiqa Rashid[2] and Luis Ibáñez[1] and Jean Carlo Rivera[1] and Héctor Rodríguez[1] and Edna Ruckhaus[1]

[1] Universidad Simón Bolívar
Caracas, Venezuela
{mvidal,libanez,jrivera,hector,ruckhaus}@ldc.usb.ve
[2] University of Maryland
louiqa@umiacs.umd.edu

Abstract. Under the umbrella of the Semantic Web, Linked Data projects have the potential to discover links between datasets and make available a large number of semantically inter-connected data. Particularly, Health Care and Life Sciences have taken advantage of this research area, and publicly hyper-connected data about disorders and disease genes, drugs and clinical trials, are accessible on the Web. In addition, existing health care domain ontologies are usually comprised of large sets of facts, which have been used to annotate scientific data. For instance, annotations of controlled vocabularies such as MeSH or UMLS, describe the topics treated in PubMed publications, and these annotations have been successfully used to discover associations between drugs and diseases in the context of the Literature-Based Discovery area. However, given the size of the linked datasets, users have to spend uncountable hours or days, to traverse the links before identifying a new discovery. In this paper we provide an authority-flow based ranking technique that is able to assign high scores to terms that correspond to potential novel discoveries, and to efficiently identify these highly scored terms. We propose a graph-sampling method that models linked data as a Bayesian network and implements a Direct Sampling reasoning algorithm to approximate the ranking scores of the network. An initial experimental study reveals that our ranking techniques are able to reproduce state-of-the-art discoveries; additionally, the sampling-based approach is able to reduce the exact solution evaluation time.

1 Introduction

During the last decade, emerging technologies such as the Semantic Web, the Semantic Grid, Linked Data projects, and affordable computation and network access, have made available a great number of publicly inter-connected data sources. Life science is a good example of this phenomenon. This domain constantly evolves, and has generated publicly available information resources and services whose number and size, have dramatically increased during the last years. For example, the amount of gene expression data has grown exponentially, and most of the biomedical sources that publish this information have been gaining data at a rate of 300 % per year. The same trend is observed in biomedical literature where the two largest interconnected bibliographic databases in biomedicine, PubMed and BIOISIS, illustrate the extremely large size of

the scientific literature today. PubMed publishes at least 16 million references to journal articles, while BIOSIS makes available more than 18 million abstracts.

On the other hand, a great number of ontologies and controlled vocabularies have become available under the umbrella of the Semantic Web. Ontologies are specified in different standard languages, such as XML, OWL or RDF, and regular requirements are expressed using query languages such as SPARQL. Ontologies play an important role and provide the basis for the definition of concepts and relationships that make global interoperability among available Web resources possible. In the Health Care and Life Sciences domains, large ontologies have been defined; for example, we can mention MesH [15], Disease [1], Galen [16], EHR_RM [2], RxNorm [20], and GO [5]. Ontologies are commonly applied in these domains to annotate publications, documents, and images; also ontologies can be used to distinguish similar concepts, to generalize and specialize concepts, and to derive new properties. To fully take advantage from the linked data sources and their ontology annotations, and to be able to recognize novel discoveries, scientists have to navigate through the inter-connected sources, and compare, correlate and mine some of these annotated data. Nevertheless, because the size and number of available sources and the set of possible annotations are very large, users may have to spend countless hours or days before recognizing relevant findings.

In order to facilitate the specification of scientist's semantic connection needs, we present a ranking technique able to assign high scores to potential novel associations. Furthermore, given the size of the search space and to reduce the effect of the number of available linked data sources and ontology annotations on the performance, we also propose an approximate solution named graph-sampling. This approximate ranking technique samples events in a Bayesian network that models the topology of the data connections; it also estimates ranking scores that measure how important and relevant are the associations between two terms. In addition, the approximate technique exploits information about the topology of the hyperlinks and their ontology annotations, to guide the ranking process into the space of relevant and important terms.

In this paper we describe our ranking techniques and show their effectiveness and efficiency. The paper is composed of five additional sections. In Section 2, we compare existing approaches. Section 3 illustrates techniques proposed in the area of Literature Based Discovery (LBD) by showing the discovery reported in [21] where curcumin longa was associated with retinal diseases. Section 4 describes our proposed sampling technique. Section 5 reports our experimental results. Finally, we give our conclusions and future work in Section 6.

2 Related Work

Under the umbrella of the Semantic Web, Linked Data projects have proposed algorithms to discover links between datasets. Particularly, the Linking Open Drug Data (LODD) task has connected a list of datasets that includes disorders and disease genes [6], clinical trials [9] and drug banks [26]. Some of these link discovery or generation tools apply similarity metrics to detect potential similar concepts and their relationships [25]. However, none of the existing link discovery techniques make use of information about the link structure to identify potential novel associations. Also, the ontology voID [24]

has been proposed to describe interlinked datasets and enable their discovery and usage, and provides the basis for our proposed approach.

The discovery of associations between data entries implies descriptive and predictive inference tasks based on the link structure [4] and on semantics suggested by relevant ontologies. In general the idea is to perform random walks in the space of possible associations and discover those that satisfy a particular pattern; correspondences between the discovered patterns are measured in terms of similarity functions. In [7], heuristics are used to discover relevant subgraphs within RDF graphs; relationships among the metadata describing nodes is used to discover relevant relationships among entities. To decide if two objects are semantically similar, Jeh et. al. [11] propose a measure that reflects when two objects are similar based on the relationships that they hold with similar objects. Yan et al. [8] propose strategies to efficiently search subgraphs that are similar to a given query graph. Finally, Hu et al. [10] and Kuramochi and Karypis [12] describe efficient algorithms to discover subgraphs (patterns) that occur in graphs and to aggregate them.

Sampling techniques have been successfully applied to a variety of approximation techniques. For example, in the context of query optimization, different sampling-based algorithms have been proposed to estimate the cardinality of a query efficiently [13, 14, 18]. The challenge of these methods is to reach estimates that satisfy the required confidence levels while the size of the sample remains small. A key decision involves when to stop sampling the population and this is determined by the mean and variance of the sample in comparison to the target population. In this paper we propose a technique that samples paths in an acyclic directed graph that models a dataset of linked data. Paths are sampled based on the joint probability which is computed as the multiplication of the authority transfer flow value of the edges that comprise the path. Similarly, we define the stop condition of the sampling, based on an estimate of the metric score mean. Related to the problem of estimating authority flow metrics, Fogaras et. al. [3] implement a Monte-Carlo based method to approximate personalized PageRank scores. They sample paths whose length is determined by a geometric distribution. Paths are sampled from a Web graph based on a probability that represents whether objects in the paths can be visited by a random surfer. This approach may provide a solution to PageRank; however, it is not applicable to our proposed approach because the length of the paths is determined by the number of layers in the results graph and cannot be randomly chosen. In contrast, graph-sampling samples objects layer by layer, until the last layer in the result graph is visited. Objects with higher probability to be visited by a random surfer and links between these objects, will have greater chance to be chosen during the sampling process. Thus, graph-sampling may be able to only traverse relevant paths that correspond to relevant discoveries.

3 Motivating Example

Consider the area of Literature-Based Discovery (LBD) where by traversing scientific literature annotated with the controlled vocabularies like MesH, drugs have been associated with diseases [21, 22]. LBD can perform *Open* or *Closed* discoveries, where a scientific problem is represented by a set of articles that discuss an input problem

(*Topic A*), and the goal is to prove the significance of the associations between *A* and some other *C* topics discussed in the set of publications reachable from the initial set of publications relevant to *A*. Srinivasan et al. [21] followed this idea and improved the *Open* and *Closed* techniques by recognizing that articles in PubMed have been curated and heavily annotated with controlled vocabulary terms from the MeSH (Medical Subject Heading) ontology. Relationships between publications and terms are annotated with weights or scores that represent the relevance of the term in the document. MeSH term weights are a slight modification of the commonly used TF/IDF scores. Figure 1 illustrates a directed graph that represents the terms and publications visited during the evaluation of an *Open* discovery. Topic *A* is used to search on the PubMed site and retrieve relevant publications, named Pub_A. Then, MeSH term annotations are extracted from publications in Pub_A, and filtered by using a given set of semantic types of the ontology Unified Medical Language System (UMLS) [3]; this new set of MeSH terms is named *B* and is used to repeat the search on the PubMed site. Similarly, sets Pub_B, *C* and Pub_C are built.

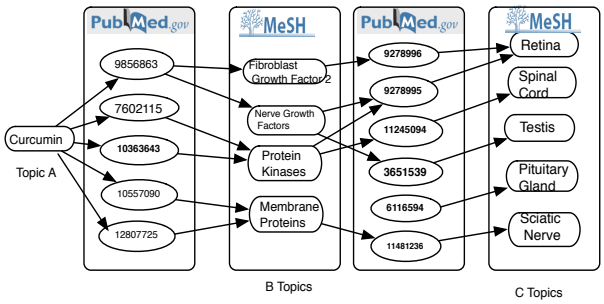


Fig. 1. Open Discovery Graph LBD

The Srinivasan's algorithm considerably reduces the space of intermediate results while identifying novel relationships; however, it still requires human intervention to create the intermediate datasets as well as to rank the terms that may not conduce to potential novel discoveries. We propose a sampling-based ranking technique that is able to estimate which are the nodes that will conduce to novel discoveries, and thus, reduce the discovery evaluation time. We illustrate the usage of this technique in the context of Literature-based Discovery. However, we hypothesize that this technique can be used to efficiently discover associations between the data published in the Cloud of Linked Data.

[3] http://www.nlm.nih.gov/research/umls/

4 A Ranking-based Solution to Discover Semantic Associations

We propose ranking-based solutions to the problem of the semantic association discovery. The proposed techniques take advantage of existing links between data published on the Cloud of Linked Data, or make use of annotations with controlled vocabularies such as MeSH, GO, PO, etc. We present an exact solution, and an approximate technique; both methods have been implemented in BioNav [23].

4.1 An Exact Ranking Technique

The exact ranking technique extends existing authority-flow based metrics like PageRank, ObjectRank or any of their extensions [17]. This ranking approach assumes that the linked data comprise a layered graph, named layered Discovery Graph, where nodes represent published data and edges correspond to hyperlinks.

Formally, a layered Discovery Graph, $lgDG$=(V_{lg}, E_{lg}) is a layered directed acyclic graph, comprised of k layers, $L_1, \ldots, L_k$. Layers are composed of data entries which point to data entries in the next layer of the graph. Data entries in the k-th layer or last layer of the graph, are called target objects. Authority-flow based metrics are used to rank the target objects, and we use these scores to identify relevant associations between objects in the first layer and target objects.

Figure 2 illustrates an example of a layered Discovery Graph that models the Open Discovery Graph in Figure 1. In this example, odd layers are composed of MeSH terms while even layers are sets of publications. Also, an edge from a term b to a publication p indicates that p is retrieved by the PubMed search engine when b is the search term. Finally, an edge from a publication p to a term b represents that p is annotated with b. Each edge $e = (b, p)$(resp., $e = (p, b)$) between the layers l_i and l_{i+1} is annotated with the TF/IDF score; this value either represents how relevant is a term b in the collection of documents in l_{i+1}, or a document relevance regarding to a set of terms. The path of thick edges connects Topic A with $C3$; the value 0.729 corresponds to the authority-flow score and represents the relevance of the association between Topic A and $C3$.

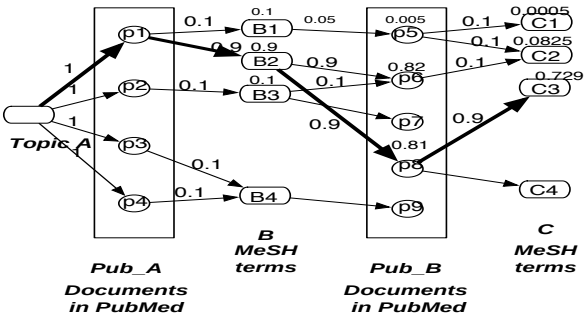


Fig. 2. A Layered Discovery Graph

Given a layered Discovery Graph $lgDG$=(V_{lg}, E_{lg}) of k layers, the authority-flow scores of the target objects are formally defined as a ranking vector R:

$$R = M^{k-1}R_{ini} = (\prod_{l=1}^{k-1} M^{l})\, R_{ini}$$

where, M is a transition matrix and R_{ini} is a vector with the scores of the objects in the first layer of the graph. An entry $M[u, v]$ in the transition matrix M, where u and v are two data objects in $lgDG$, corresponds to $\alpha(u, v)$ or is 0.0. The value $\alpha(u, v)$ is calculated according to the metric used to compute the ranking score of the data.

$$M[u, v] = \begin{cases} \alpha(u, v) \text{ if } (u, v) \in E_{lg}, \\ 0.0 \quad \text{otherwise.} \end{cases}$$

For instance, the *layered graph Weighted Path Count* (lgWP) is an extension of ObjectRank and Path Count and the value of $\alpha(u, v)$ corresponds to the TF/IDF score that denotes how relevant is the object u with respect to the object v. Nodes with high lgWP scores are linked by many nodes or linked by highly scored nodes; for example, in Figure 2, $C3$ is pointed by relevant nodes. In the context of LBD, we use this metric to discover novel associations between a topic A and MeSH terms in the last layer of the $lgDG$, and we have been able to discover the associations identified by Srinivasan et al. [21].

4.2 A Sampling-based Ranking Solution

Although the ranking induced by an authority-flow based metric is able to distinguish relevant associations, the computation of this ranking may be costly. Thus, to speed up this task, we propose a sampling-based technique that traverses only nodes in the layered graph that may conduce to highly ranked target objects.

Given a layered Discovery Graph $lgDG = (V_{lg}, E_{lg})$, the computation of highly ranked target objects is reduced to estimating a subgraph $lgDG'$ of $lgDG$, so that with high confidence (at least δ), the relative error of the distance between the approximate highly ranked target objects in $lgDG'$ and the exact highly ranked target objects, is at least ϵ.

A set SS=$\{lgDG_1, ..., lgDG_m\}$ of independent and identically distributed (i.i.d.) subgraphs of $lgDG$ is generated. Then, $lgDG'$ is computed as the union of the m subgraphs. Each subgraph $lgDG_i$ is generated using a *graph-sampling* technique. This sampling approach is based on a Direct Sampling method for a Bayesian network [19]. This network represents all the navigational information encoded in $lgDG$ and in the transition matrix M of the authority-flow metric. The Direct Sampling technique generates events from a Bayesian network [19].

A Bayesian network $BN = (VB, EB)$ for a layered Discovery Graph $lgDG$, is built as follows:

- BN and $lgDG$ are homomorphically equivalent, i.e., there is a mapping $f : VB \rightarrow V_{lg}$, such that, $(f(u), f(v)) \in E_{lg}$ iff $(u, v) \in EB$.

– Nodes in VB correspond to discrete random variables that represent if a node is visited or not during the discovery process, i.e., $VB = \{X \mid X$ takes the value 1 (true) if the node X is visited and 0 (false), otherwise$\}$.
– Each node X in VB has a conditional probability distribution:

$$Pr(X \mid Parents(X)) = \sum_{j=1}^{n} \alpha(f(Y_j), f(X))$$

where, Y_j is the value of the random variable that represents the j-th parent of the node X in the previous layer of the Bayesian network and n corresponds to the number of parents of X. The value $\alpha(f(Y_j), f(X))$ represents the weight or score of the edge $(f(Y_j), f(X))$ in the layered Discovery Graph and corresponds to an entry in the transition matrix M; it is seen as the probability to move from Y_j to X in the Bayesian network. Furthermore, the conditional probability distribution of a node X represents the collective probability that X is visited by a random surfer starting from the objects in the first layer of the layered Discovery Graph. Finally, the probability of the nodes in the first layer of the Bayesian network corresponds to a score that indicates the relevance of these objects with respect to the discovery process; these values are represented in the R_{ini} vector of the ranking metric.

Given a Bayesian network generated from the layered Discovery Graph $lgDG$, the Direct Sampling generates each subgraph $lgDG_i$. Direct Sampling selects nodes in $lgDG_i$ by sampling the variables from the Bayesian network based on the conditional probability of each random variable or node. Algorithm 1 describes the Direct Sampling algorithm.

Algorithm 1 The Direct Sampling Algorithm

Input: $BN = (VB, EB)$ *A Bayesian network for a layered discovery graph*
Output: A subgraph $lgDG_i$

```
TP ⇐ topologicalOrder(BN);
for X ∈ TP do
    Pr(X | Parents(X)) ⇐ Σ_{j=1}^{n} α(f(Y_j), f(X));
    if (randomNumber >= Pr(X | Parents(X))) then
        X_i ⇐ 1;
    else
        X_i ⇐ 0;
    end if
end for
```

Variables are sampled in turn following a topological order starting from the variables in the first layer of the Bayesian network; this process is repeated until variables in the last layer are reached. The values assigned to the parents of a variable define the probability distribution from which the variable is sampled. The conditional probability of each node in the last layer of $lgDG_i$ corresponds to the approximate value of the implemented metric.

Figure 3 illustrates the behavior of the graph-sampling technique; unmarked nodes correspond to visited notes and comprise a subgraph $lgDG_i$. Direct Sampling is performed as follows: initially, all the nodes in the first layer have the same probability to be visited and all of them are considered. All their children or nodes in the second layer are also visited and the conditional probability is computed; nodes with the highest scores survive, i.e., $n5$ and $n7$. Then, the children of these selected nodes are also visited, and the process is repeated until nodes in the last layer are reached. Note that nodes $n9$ and $n11$ are the target objects with the highest values of the lgWP metric and with the highest conditional probability. These nodes are pointed by nodes with high lgWP scores or pointed by many nodes; thus, they are very likely to be visited when the Direct Sampling algorithm is performed.

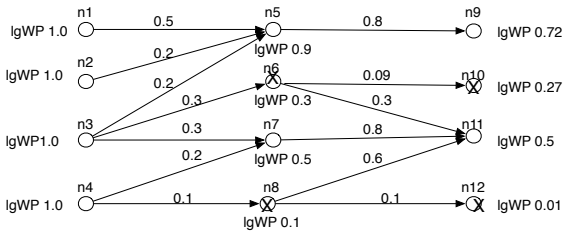


Fig. 3. Graph Sampling

Once an iteration i of the Direct Sampling is finalized, the sampled layered Discovery Graph $lgDG_i = (V_i, E_i)$ is created. Nodes in V_i correspond to the variables sampled during the Direct Sampling process that are connected to a visited variable in the last layer of the Bayesian network. Additionally, for each edge (u, v) in the Bayesian network that connects nodes $f(u)$ and $f(v)$ in V_i, an edge $(f(u), f(v))$ is added to E_i. The conditional probabilities of the target objects of each subgraph $lgDG_i$ correspond to the approximate values of the ranking metric. After all the subgraphs $lgDG_1, \ldots, lgDG_m$ are computed, an estimate $lgDG'$ is obtained as the union of these m subgraphs. The approximation of the ranking metric in the graph $lgDG'$ is computed as the average of the approximate ranking metric values of target objects in the subgraphs $lgDG_1, \ldots, lgDG_m$. A bound of the number of iterations or sampled subgraphs is defined in terms of the Chernoff-Hoeffdings bound.

Theorem: Let $lgDG$ be an exact layered Discovery Graph and $lgDG_i$ be one of the m sampled subgraphs. Let T be a list of the target objects in $lgDG$ ranked with respect to exact values of the ranking metric RM. Let T_i be a list of the target objects in $lgDG_i$ ranked with respect to the approximation of RM. Let $J(lgDG_1, lgDG, \beta)$,..., $J(lgDG_m, lgDG, \beta)$ be independent identically distributed (i.i.d.) random variables with values in the set {0,1}. Each random variable $J(lgDG_i, lgDG, \beta)$ has value=1 if a distance metric value between the ranking list T_i and the list T is at least β; otherwise, value=0. Let S denote the average of these variables, i.e., $X = \frac{1}{m} \sum_{i=1}^{m} J(lgDG_i, lgDG, \beta)$ and $E(S)$ the expectation of S. Then, the size m of the sample has to satisfy the fol-

lowing formula to ensure that the relative error of $E(S)$ is greater than ϵ with some probability:

$$P(|S - E(S)| \geq \epsilon) \leq 2\exp(-2m\epsilon^2)\,.$$

5 Experimental Results

In this section we show the quality of our proposed discovery techniques. First, we compare the results obtained by our ranking technique with respect to the results obtained by the Manjal system [21]. Then, we show the behavior of this technique in the DBLP dataset. Experiments were executed on a Sun Fire V440 equipped with two UltraSPARC IIIi processors running at 1.593 GHZ with 16 GB RAM. The ranking and sampling techniques were implemented in Java 1.6.1.

To conduct the first experiment, we have created a catalog populated with the PubMed publications from the NCBI source[4], all the MeSH terms, and all the links between Mesh terms and PubMed publications. We stored the downloaded data in two tables, *Pub-MeSH* and *MeSH-Pub*. Table *Pub-MeSH* relates a publication p with all the MeSH terms that correspond to annotations of p in PubMed; these annotations are manually done by experts at the National Library of Medicine site. Table *MeSH-Pub* relates a MeSH term m with all publications that are retrieved when the term m is used to search on PubMed. Both tables have an attribute *score* that represents the relevance of the relationships represented in the table. Suppose there is a tuple (p, m, s) in table *Pub-MeSH*, then the score $s = A \times T \times C$, where:

- A: is the augmented document frequency of the publication p, i.e., $A = 0.5 + 0.5 + \frac{tf}{tf_{max}}$, where, tf is the frequency of p in table *Pub-MeSH*, and tf_{max} is the maximum document frequency of any publication in *Pub-MeSH*.
- T: inverse term frequency $log_2(\frac{N}{Np})$, where N is the number of collected MeSH terms, i.e., 20,652, and N_p corresponds to the number of MeSH terms associated with the publication p in the table *Pub-MeSH*.
- C: is a cosine normalization factor.

Similarly, scores in table *MeSH-Pub* were computed. To reproduce the results reported by Srinivasan et al. in [21], we ran the metric lgWP on a layered Discovery Graph *lgDG* comprised of 5 layers, 3,107,901 nodes and 10,261,791 edges. Sets Pub_A, B, Pub_B and C and were built following the criteria proposed by Srinivasan et al., and by selecting data from tables *Pub-MeSH* and *MeSH-Pub*. We ranked the target objects in the graph, and we could observe that our ranking technique was able to produce 4 of the top-5 semantic associations identified by Srinivasan et al. [21]. Table 1 compares the top-5 target objects discovered by [21] and the ones discovered by our ranking technique, i.e., our ranking technique exhibits a precision and recall of 80%.

We have also studied the benefits of performing the graph-sampling technique, and we ran the sampling process for 5 iterations, i.e., 5 sampled subgraphs were computed. Table 2 reports on the top-10 MeSH terms identified by graph-sampling. We can observe that 4 of the top-5 MeSH terms identified by the Srinivasan's algorithm [21],

[4] http://www.ncbi.nlm.nih.gov/

k	Srinivasan's Ranking [21]	lgWP
1	Retina	Testis
2	Spinal Cord	Retina
3	Testis	Spinal Cord
4	Pituitary Gland	Obesity
5	Sciatic Nerve	Pituitary Gland

Table 1. Top-5 MeSH terms

are also identified. We note that iterations do not improve the quality of the discovery process.

k	i=1	i=2	i=3	i=4	i=5
1	Spinal Cord	Spinal Cord	Spinal Cord	Spinal Cord	Spinal Cord
2	Pituitary Gland	Pituitary Gland	Pituitary Gland	Pituitary Gland	Pituitary Gland
3	Celiac Disease	Celiac Disease	Celiac Disease	Celiac Disease	Disease
4	Hepatic Encepha.	Hepatic Encepha.	Hepatic Encepha.	Hepatic Encepha.	Hepatic Encepha.
5	Uremia	Uremia	Uremia	Uremia	Uremia
6	Retina	Anemia	Anemia	Anemia	Anemia
7	Obesity	Retina	Retina	Retina	Retina
8	Testis	Obesity	Phenylketonurias	Phenylketonurias	Phenylketonurias
9	Hypothalamus	Testis	Obesity	Obesity	Obesity
10	Osteoporosis	Hypothalamus	Testis	Testis	Testis

Table 2. Effectiveness of Graph Sampling Techniques

Finally, we report on the number of target MeSH terms produced by the Srinivasan's algorithm and the ones produced during each iteration of graph-sampling (Table 3). We can observe that graph-sampling is able to discover 80% of the top novel MeSH terms, while the number of target terms is reduced by up to one order of magnitude.

# Srinivasan's target MeSH Terms [21]	i=1	i=2	i=3	i=4	i=5
570	24	38	49	61	71

Table 3. Performance of Graph-Sampling Techniques

In the second experiment, we downloaded the DBLP file in a relational database. We ran the graph-sampling technique to discover associations between a given author and the most relevant conferences where this author has published at least one paper. We ran 3 sets of 30 queries and compared the ranking produced by the exact solution and the one produced by graph-sampling; layered Discovery Graphs were comprised

of 5 layers and at most 876,110 nodes and 4,166,626 edges. Author's names with high, medium and low selectivity were considered, where high selectivity means that the author has few publications while low selectivity represents that the author is very productive. The top-5 conferences associated with each author were computed by using the exact ranking and the approximation produced by graph-sampling during 6 iterations. Table 4 reports the average precision of the approximate top-5 conferences with respect to the exact top-5. We can observe that graph-sampling is able to identify almost 65% of the top-5 conferences after iteration 3. The time required to execute the graph-sampling technique was reduced at least by half. These results suggest that the proposed discovery techniques provide an effective and efficient solution to the problem of identifying associations between terms.

Author's Name Selectivity	i=1	i=2	i=3	i=4	i=5	i=6
high	0.390	0.4874	0.635	0.813	0.823	0.871
medium	0.341	0.562	0.681	0.724	0.872	0.890
low	0.64	0.660	0.749	0.803	0.806	0.815

Table 4. Effectiveness of Graph Sampling Techniques DBLP- Average Precision

6 Conclusions and Future Work

In this paper we have presented a sampling-based technique that supports the discovery of semantic associations between linked data. We have reported the results of an empirical study where we have observed that our proposed techniques are able to efficiently reproduce the behavior of existing LBD techniques. This observed property of our discovery technique may be particularly important in the context of large datasets as the ones published in the Cloud of Linked Data. In the future we plan to extend this study to identify potential associations between other sources of the Cloud of Linked Data.

References

1. Disease Ontology. http://diseaseontology.sourceforge.net.
2. EHR Ontology. http://trajano.us.es./ isabel/EHR/EHRRM.owl.
3. D. Fogaras, B. Rácz, K. Csalogány, and T. Sarlós. Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments. *Internet Mathematics*, 2(3), 2005.
4. L. Getoor and C. P. Diehl. Introduction to the special issue on link mining. *SIGKDD Explorations*, 7(2), 2005.
5. The Gene Ontology. http://www.geneontology.org/.
6. K.-I. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A.-L. Barabasi. The human disease network. *Proceedings of the National Academy of Sciences of the United States of America*, 104:8685–8690, 2007.
7. C. Halaschek-Wiener, B. Aleman-Meza, I. B. Arpinar, and A. P. Sheth. Discovering and ranking semantic associations over a large rdf metabase. In *VLDB*, pages 1317–1320, 2004.

8. J. Han, X. Yan, and P. S. Yu. Mining, indexing, and similarity search in graphs and complex structures. In *ICDE*, page 106, 2006.
9. O. Hassanzadeh, A. Kementsietsidis, L. Lim, R. J. Miller, and M. Wang. Linkedct: A linked data space for clinical trials. In *Proceedings of the WWW2009 workshop on Linked Data on the Web (LDOW2009)*, 2009.
10. H. Hu, X. Yan, Y. H. 0003, J. Han, and X. J. Zhou. Mining coherent dense subgraphs across massive biological networks for functional discovery. In *ISMB (Supplement of Bioinformatics)*, pages 213–221, 2005.
11. G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *KDD*, pages 538–543, 2002.
12. M. Kuramochi and G. Karypis. Finding frequent patterns in a large sparse graph*. *Data Min. Knowl. Discov.*, 11(3):243–271, 2005.
13. Y. Ling and W. Sun. A supplement to sampling-based methods for query size estimation in a database system. SIGMOD Record, 21(4):12–15, 1992.
14. R. Lipton and J. Naughton. Query size estimation by adaptive sampling (extended abstract). In PODS '90: Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, pages 40–46. New York, NY, USA: ACM Press, 1990.
15. Medical Subject Heading (MeSH). http://www.nlm.nih/gov/mesh.
16. O. C. Organization. GALEN common reference model.
17. L. Raschid, Y. Wu, W. Lee, M. Vidal, P.Tsaparas, P. Srinivasan, and A. Sehgal. Ranking target objects of navigational queries. In *WIDM*, pages 27–34, 2006.
18. E. Ruckhaus, E. Ruiz, and M. Vidal. Query optimization in the semantic web. In Theory and Practice of Logic Programming. Special issue on Logic Programming and the Web, 2008.
19. S. Russell and P.Norvig. *Artificial Intelligence: A Modern Approach. Second Edition*. Princeton Hall, 2003.
20. An Overview to RxNorm. http://www.nlm.nih.gov/research/umls/rxnorm/.
21. P. Srinivasan, b. Libbus, and A. Kumar. Mining medline: Postulating a beneficial role for curcumin longa in retinal diseases. In L. Hirschman and J. Pustejovsky, editors, *LT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases*, pages 33–40, 2004.
22. D. Swanson. Migraine and magnesium: Eleven neglected connections. In *Perspective in Biology and Medicine*, 1988.
23. M.-E. Vidal, E. Ruckhaus, and N. Marquez. BioNav: A System to Discover Semantic Web Associations in the Life Sciences. In *ESWC 09-Poster Session*, 2009.
24. voiD Guide - Using the Vocabulary of Interlinked Datasets. http://rdfs.org/ns/void-guide.
25. J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data. In *International Semantic Web Conference (ISWC)*, 2009.
26. D. S. Wishart, C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang, and J. Woolsey. DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Research*, 34, 2006.

The BAY-HIST Prediction Model for RDF Documents

Edna Ruckhaus and María-Esther Vidal

Universidad Simón Bolívar
Caracas, Venezuela
{ruckhaus, mvidal}@ldc.usb.ve

Abstract. In real-world RDF documents, property subject and object values are often correlated. The identification of these relationships is of significant relevance to many applications, e.g., query evaluation planning and linking analysis. In this paper we present the BAY-HIST Prediction Model, a combination of Bayesian networks and multidimensional histograms which is able to identify the probability of these dependencies. In general, Bayesian networks assume a small number of discrete values for each of the variables considered in the network. However, in the context of the Semantic Web, variables that represent the concepts in large-sized RDF documents may contain a very large number of values; thus, BAY-HIST implements multidimensional histograms in order to aggregate the data associated with each node in the network. We illustrate the benefits of applying BAY-HIST to the problem of query selectivity estimation as part of cost-based query optimization. We report initial experimental results on the predictive capability of this model and the effectiveness of our optimization techniques when used together with BAY-HIST. The results suggest that the quality of the optimal evaluation plan has improved over the plan identified by existing cost models that assume independence and uniform distribution of the data values.

1 Introduction

The number of controlled vocabularies and annotated data sources in the Web has exploded in the last few years. Individually, many of these documents contain a large number of concepts and instances, and additionally their growth rate is very high. Thus, in order to be capable of scaling up, Web architectures have to be tailored for query processing on large number of resources and instances. We apply BAY-HIST to the problem of query selectivity estimation as part of cost-based query optimization.

The Prediction Model BAY-HIST is a framework that combines Bayesian networks and multidimensional histograms with the purpose of determining dependencies between properties in RDF documents and the distribution of their values. Bayesian Networks are probabilistic models that allow a compact representation of the joint distribution of the concepts defined in an RDF document. In general, Bayesian networks assume a small number of discrete values for each of the variables considered in the network. However, in the context of RDF documents in the Semantic Web, variables that represent the concepts in large-sized RDF documents may contain a very large number of values; thus, BAY-HIST implements multidimensional histograms in order to aggregate the data associated with each node in the Bayesian network that represents the RDF document.

BAY-HIST has been included as a component of the OneQL System, an Ontology System that provides optimization and query evaluation techniques that scale up to large RDF/RDF(S) documents [4, 10]. We report initial experimental results on the predictive capability of this model and the effectiveness of our optimization techniques when used together with BAY-HIST. The results suggest that the quality of the optimal evaluation plan has improved compared to the plan identified by existing cost models that assume independence and uniform distribution of the data values, by up to two orders of magnitude.

The structure of this paper is as follows: first, we will give a motivating example. Following this, we will present the syntax and semantics of BAY-HIST. Next, we will explain the architecture of the BAY-HIST Prediction Model and its application to cost-based query optimization. Then, the experimental study will be described, and finally, the conclusions and future work will be presented.

2 A Motivating Example

The example that follows shows a query to the RDF repository published at `http://www.govtrack.us/`. In this example, besides information concerning the U.S. congress bills voting process, we consider information of the census such as religion and gender, and political information such as the party and the state that is represented by each representative that participates in the voting process. Consider the relationships between party, gender, religion, state and the way a representative votes. To discover if there is any correlation among the values of these five properties, we will try to determine if for different instantiations of the following query, different number of tuples are obtained: *Names of all the representatives of state* ?*S , that belong to party* ?*P, are of gender* ?*G, are of religion* ?*R and have voted for the winning option in the voting process of Bill* ?*B.* The SPARQL representation of this query is illustrated in Figure 1.

```
PREFIX pol:<tag:http://www.rdfabout.com/rdf/schema/politico/>
PREFIX vote:<tag:http://www.rdfabout.com/rdf/schema/vote/>
PREFIX foaf:<tag:http://xmlns.com/foaf/0.1/>
SELECT ?X
FROM <tag:http://www.examples.org/votesdataset/>
WHERE
        {?X pol:forOffice ?S . ?X pol:party ?P . ?Z pol:hasRole ?X . ?Z foaf:gender ?G .
        ?Z foaf:religion ?R . ?O vote:votedBy ?X . ?B vote:winner ?O}
```

Fig. 1. A SPARQL query

This query may have different subject and object instantiations (constants). For instance, we may want to explore for a certain Bill, the different combinations of instantiations for party, religion, gender and state. While for a certain set of instantiations the query has 18 answers, for another one it has no answers. This behavior is due to the lack of uniformity in the property value distribution and the dependency between properties. For example, the probability that a representative has voted for the winning option in

the voting process of Bill 1998-173 if he is Catholic, male, belongs to the Democratic party and represents the state of Massachussets is much higher than the probability that a representative has voted for the winning option in the voting process of Bill 1998-173 if he is Jewish, male, Republican and represents Oklahoma. The identification of these relationships is of significant relevance to many applications. For instance, in query evaluation planning, this information may provide the basis for the optimizer to discriminate between bad or good query plans.

3 The BAY-HIST Prediction Model

Consider the RDF repository presented in the previous example. Let us assume that there are certain causal relationships between the subjects and objects of properties that are represented as an RDF Bayesian Network (RBN), as shown in Figure 2. In this

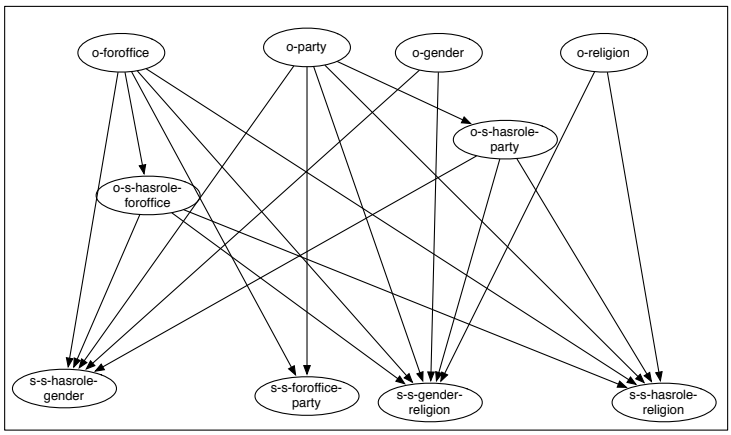


Fig. 2. RBN Votes

RBN, there are nodes that represent property subjects or objects. For example, node `o-religion` represents the values (objects) of property `religion`. We also represent the event of a combination between subjects or objects of related properties. Such is the case of node `s-s-foroffice-party` that represents the event that a subject that is representing a certain state, belongs to a certain party. The arcs in this network represent dependencies between nodes. In this network we model that the combination of voter and gender is conditioned not only by the gender itself, but also by the state he represents and the party to which he belongs to; thus, the probability that a person's gender is 'male', the state is 'Oklahoma' and that he belongs to the 'Republican' party is 0.033. This probability is related to the probabilities of all the rest of combinations of gender, state and party. Tables 1(a) and 1(b) show a portion of the conditional probability tables (CPT) of this RBN. An RBN represents all the conditional dependencies among prop-

Table 1. CPT's Votes

(a) CPT o-party

o-party	prob(o-party)
Democratic	0.51
Independent	0.007
Republican	0.47

(b) CPT s-s-foroffice-party

s-s-foroffice-party	o-foroffice	o-party	prob(s-s-foroffice-party)
true	Democratic	ak	0
false	Democratic	ak	1
true	Independent	ak	0
false	Independent	ak	1
true	Republican	ak	0.03
false	Republican	ak	0.97
true	Democratic	ma	0.038
false	Democratic	ma	0.962
...	...	...	...

erty subjects and objects in an RDF document. Next, we will formally define an RDF Bayesian Network:

Definition 1 (RDF Bayesian Network) *Given an RDF directed graph $O_R = (V_R, E_R)$ where V_R and E_R are the nodes and arcs in the RDF graph. An* **RDF Bayesian Network** *R_B for O_R, is a pair $R_B = \langle O_B, CPT_B \rangle$, where $O_B = (V_B, E_B)$ is a DAG. V_B are the nodes in O_B and E_B are the arcs in O_B. CPT_B are the Conditional Probability Tables for each node. The homomorphism $f : \mathbb{P}(E_R) \rightarrow \mathbb{P}(V_B)$ establishes mappings between O_R and O_B:*

$$f(\{(sub, pro, obj)\}) = \{s\text{-}pro, o\text{-}pro\} \quad \text{(Mapping 1)}$$

$$f(\{(sub_1, pro_1, obj), (sub_2, pro_2, obj)\}) = \{o\text{-}o\text{-}pro_1\text{-}pro_2, o\text{-}o\text{-}pro_2\text{-}pro_1\} \quad \text{(Mapping 2)}$$

$$f(\{(sub, pro_1, obj_1), (sub, pro_2, obj_2)\}) = \{s\text{-}s\text{-}pro_1\text{-}pro_2, s\text{-}s\text{-}pro_2\text{-}pro_1\} \quad \text{(Mapping 3)}$$

$$f(\{(sub, pro_1, obj_1), (sub_2, pro_2, sub)\}) = \{s\text{-}o\text{-}pro_1\text{-}pro_2, o\text{-}s\text{-}pro_2\text{-}pro_1\} \quad \text{(Mapping 4)}$$

$V_C \subseteq V_B$, where V_C is the union of the sets of nodes established by mappings 2 to 4, and it is comprised of all the nodes that represent property combinations.

$E_B \subseteq V_B \times V_C$ is the set of arcs. An arc $(v_1, v_2) \in E_B$ iff there exist two sets of nodes in the RBN, $V_1 \subseteq V_B$ and $V_2 \subseteq V_C$ such that, $v_1 \in V_1$ and $v_2 \in V_2$ and when f^{-1} is applied to these sets, a subset of arcs in the RDF graph is obtained.

CPT_B is the probability $Pr(v/predecessors(v))$ for each node $v \in V_B$, i.e., the distribution on the values of v for each possible value assignment of its predecessors. The CPT_B are multidimensional histograms ordered by value. If a node v is a source node, the histogram will be one-dimensional, because in this case the CPT_B only represents the distribution of values taken up by the variable represented by the node. For each node v, according to the properties of the distribution of the values of v, CPT_B can be represented as an *equi-width* histogram or as an *equi-height* histogram.

Example 1 *Next, we illustrate the use of the homomorphism f. Figure 3 shows a portion of an RDF graph (O_R) and its corresponding RBN graph (O_B). Mapping 1 is applied to the sets of RDF arcs* {(rep1,foroffice,va)} *and* {(rep2,party,democratic)}*:*

$$f(\{(\texttt{rep1,foroffice,va})\}) = \{\texttt{s-foroffice,o-foroffice}\}$$

$$f(\{(\texttt{rep2,party,democratic})\}) = \{\texttt{s-party,o-party}\}$$

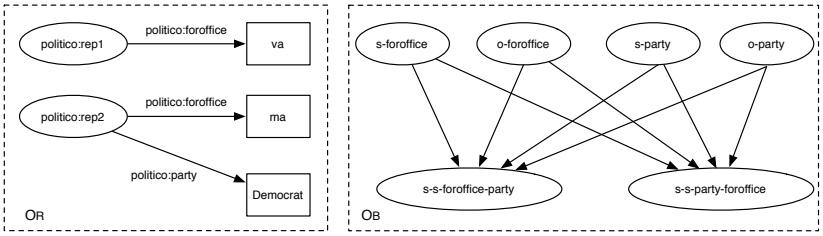


Fig. 3. Example Mapping RDF Graph - RBN Graph

Then, Mapping 3 is applied to the set of RDF arcs {(rep2,foroffice,ma), (rep2,party,democratic)}

$$f(\{(\texttt{rep2,foroffice,ma}),(\texttt{rep2,party,democratic})\})=\{\texttt{s-s-foroffice-party,s-s-party-foroffice}\}$$

The arc (o-foroffice,s-s-foroffice-party) *belongs to* E_B *because the arcs obtained by applying the inverse of* f *are subsets of* E_R*:*

$$f^{-1}(\{\texttt{s-foroffice,o-foroffice}\})\cup f^{-1}(\{\texttt{s-s-foroffice-party,s-s-party-foroffice}\})=$$
$$\{(\texttt{rep1,foroffice,va}),(\texttt{rep2,foroffice,ma}),(\texttt{rep2,party,democratic})\}$$

Intuitively, an RBN is semantically valid if its arcs have been established between nodes that map to properties whose subjects and objects are of the same type, i.e., have some type of matching instantiations, subject-subject, subject-object or object-object. For example, an arc from node `o-s-hasrole-party` to node `s-s-gender-religion` is semantically valid because there are matching subject-subject instantiations between triples of property *hasrole* and triples of *religion*, i.e., both are "persons".

Given the symmetry property of the combinations between triple patterns, the set V_B may contain only one of the nodes in the sets defined with mappings 2, 3 and 4 in Definition 1; thus, the resulting RBN is minimal:

Definition 2 (Minimal RBN) *Given an RBN* $R_B = \langle O_B, CPT_B \rangle$. R_B *is a* **Minimal RBN** *if the set* V_B *contains exactly one node in sets* {*s-s-pro*$_1$*-pro*$_2$*, s-s-pro*$_2$*-pro*$_1$}*,* {*s-o-pro*$_1$*-pro*$_2$*, o-s-pro*$_2$*-pro*$_1$} *and* {*o-o-pro*$_1$*-pro*$_2$*, o-o-pro*$_2$*-pro*$_1$}.

4 Architecture

Figure 4 shows the architecture of the BAY-HIST Prediction Model System. BAY-HIST has two main components that generate and query the RBN: the RBN Analyzer and the RBN Inference Engine. Both components make use of the *SamIam* Bayesian Inference Tool [1].

The analyzer receives an RDF document and creates the RBN structure using the mappings presented in Definition 1 to establish the correspondence between the RDF graph and the nodes and arcs of the RBN structure. Once the RBN structure has been defined, the RDF data is loaded into relational tables, and a multi-dimensional histogram

is generated for each node in the RBN structure through the stored procedures and the histogram option implemented by the DBMS Oracle [8]. Both, the RBN structure and CPT's are fed to the *SamIam* network editor, and a Bayesian network is generated in one of the internal formats recognized by the *SamIam* tool.

When a query is received, the RBN Inference Engine constructs the corresponding probability query (e.g., marginal probability and posterior marginal probability) and passes this query on to the *SamIam* inference engine which then returns an answer.

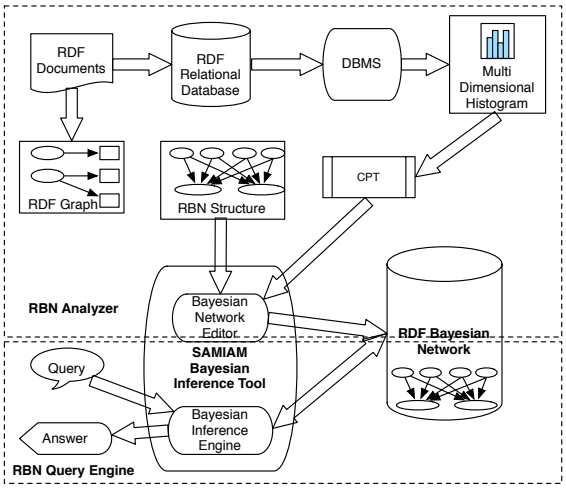


Fig. 4. Architecture of the BAY-HIST System

5 Application of BAY-HIST to Query Optimization

The BAY-HIST Prediction Model is applied to query selectivity estimation. These estimates are used within the cost model of a cost-based query optimizer as part of the formulas that compute the cost and cardinality of query sub-plans. We have developed a randomized optimization strategy based on the Simulated Annealing algorithm [7]. This algorithm explores execution plans of any shape (bushy trees) in contrast to other optimization algorithms that explore a smaller portion, e.g., left-linear plans. Random walks are performed in stages that consist of an initial random *plan generation step* followed by one or more *plan transformation steps*. An equilibrium condition or a number of iterations determines the number of transformation steps in each stage.

The probability of transforming a current plan p into a new plan p' is specified by an acceptance probability function $P(p, p', T)$ that depends on a global time-varying parameter T called the *temperature* which reflects the number of stages to be executed. Function P may be nonzero when $cost(p') > cost(p)$, meaning that the optimizer can

produce a new plan even when it has a higher cost than the current one. This feature prevents the optimizer from becoming stuck in a local minimum. Temperature T is decreased during each stage, and the optimizer concludes when $T = 0$. Transformations applied to the plan during the random walks correspond to SPARQL axioms, e.g., commutativity and associativity of the '.' operator. The optimizer is able to identify near optimal solutions because of the precision of estimates that take into account correlations of values and non uniform distribution.

Using BAY-HIST, the selectivity of an RDF query execution plan that joins A and B over join arguments $\mathcal{J}$ ($A \bowtie_{\mathcal{J}} B$) is expressed in terms of a probability query against the corresponding RBN:

$$fs(A \bowtie_{\mathcal{J}} B) = \prod_{J \in \mathcal{J}} Pr(JoinEvent_J / (JoinEvid_{\mathcal{J}_A} \wedge JoinEvid_{\mathcal{J}_B} \wedge instEvid_{I_A} \wedge instEvid_{I_B}))$$

This is a posterior marginal probability query, i.e., the probability that two pattern instantiations are combined, given the evidence of the instantiations and the joins in its left and right sub-trees.

The probability queries associated with an RDF pattern (the base case) correspond to marginal probabilities, i.e., to the probability that the value of subjects or objects of the property in the pattern is equal to the instantiation in the pattern: Pr(o-pro=obj), Pr(s-pro=sub) or Pr(s-pro=sub $\wedge$ o-pro=obj).

An estimate of the selectivity of an RDF pattern A, carried out by using a probability query on the RBN is more precise than an estimate carried out by using the traditional cost model. The traditional cost model defines the following selectivity formula:

$$fs(A, \mathcal{J}) = \prod_{J \in \mathcal{J}} 1/nKeys(A, J) \tag{1}$$

where $nKeys(A, J)$ is the number of different values taken up by J in pattern A. Likewise, an estimate of the selectivity of a sub-plan $A \bowtie_{\mathcal{J}} B$ carried out through a probability query on the RBN is more precise than an estimate carried out through the traditional cost model. The selectivity formula in the traditional cost model is as follows:

$$fs(A, B, \mathcal{J}) = \prod_{J \in \mathcal{J}} 1/max(nKeys(A, J), nKeys(B, J)) \tag{2}$$

These traditional formulas do not compute a precise estimate of the query evaluation costs because they are based on the following assumptions: (a) the values of the subjects and objects in a triple pattern are uniformly distributed, (b) the values of the subjects and objects in a pattern are independent, and (c) the values of the subjects and objects in properties of the patterns that are combined in a query, are independent.

The example that follows shows the motivating example query with two different sets of instantiations:

- *Names of all the male representatives of the state of Massachussets that belong to the Democratic party, are Catholic and have voted for the winning option in the voting process of Bill 1998-173.*
- *Names of all the male representatives of the state of Oklahoma that belong to the Republican party, are Jewish and have voted for the winning option in the voting process of Bill 1998-173.*

```
PREFIX pol:<tag:http://www.rdfabout.com/politico/>
PREFIX vote:<tag:http://www.rdfabout.com/vote/>
PREFIX foaf:<tag:http://xmlns.com/foaf/0.1/>
SELECT ?X
FROM <tag:http://www.examples.org/votesdataset/>
WHERE
        {?X pol:forOffice senate:ma .
        ?X pol:party 'Democratic' .
        ?Z foaf:gender 'male' .
        ?Z pol:hasRole ?X .
        ?Z foaf:religion 'Catholic' .
        ?O vote:votedBy ?X .
        '1998-173' vote:winner ?O}
```

(a) SPARQL Query 1

```
PREFIX pol:<tag:http://www.rdfabout.com/politico/>
PREFIX vote:<tag:http://www.rdfabout.com/vote/>
PREFIX foaf:<tag:http://xmlns.com/foaf/0.1/>
SELECT ?X
FROM <tag:http://www.examples.org/votesdataset/>
WHERE
        {?X pol:forOffice senate:ok .
        ?X pol:party 'Republican' .
        ?Z foaf:gender 'male' .
        ?Z pol:hasRole ?X .
        ?Z foaf:religion 'Jewish' .
        ?O vote:votedBy ?X .
        '1998-173' vote:winner ?O}
```

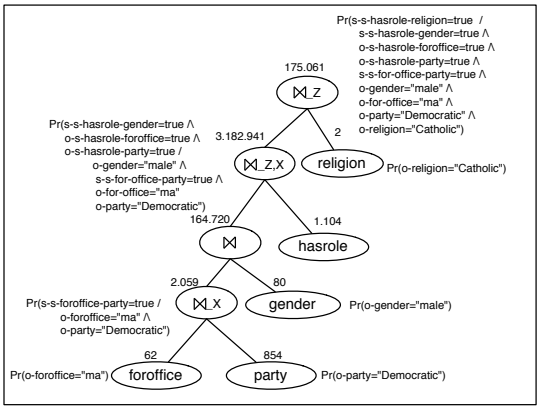
(b) SPARQL Query 2

Fig. 5. Two Queries with Different Instantiations

The SPARQL representation of these two queries is illustrated in Figure 5. Query 1 and Query 2 differ in their subject and object instantiations (constants), and their answers are different: while the first query has 18 answers, the second one has no answers. This behavior is due to the lack of uniformity in the property value distribution and the dependencies between properties. Based on this observation, we use an RBN to differentiate the selectivity of the sub-plans of each query execution plan taking into account the existing correlation between the various RDF properties. To estimate the selectivity of the sub-plan shown in Figure 6(a), a posterior marginal probability query is carried out in the RBN and the result of this probability query is 0.0275.

```
{?X pol:forOffice senate:ma .
?X pol:party 'Democratic' .
?Z foaf:gender 'male' .
?Z pol:hasRole ?X .
?Z foaf:religion 'Catholic'}
```

(a) Query Sub-Plan



(b) Sub-Plan Tree

Fig. 6. Probability Queries on an Execution Sub-plan (`http://www.govtrack.us/`)

For the corresponding sub-plan in the second query, i.e., the same sub-plan with different instantiations, the result of the inference on the RBN is 0, which is consistent

with the expectation that the cardinality of the first query is higher than the cardinality of the second query. Figure 6(b) shows the tree representation of the sub-plan in Figure 6(a). Each node is annotated with the probability query corresponding to the sub-plan (sub-tree) selectivity estimate, and with its cardinality. The cost estimate of the sub-plan, is equivalent to the total number of intermediate results that must be estimated to obtain the answer:
$cost(P) = 62 + 854 + 2.059 + 80 + 164.720 + 1.104 + 3.182.941 + 2 = 3.351.822$

6 Related Work

In [6], Bayesian networks are applied to the problem of imprecise estimates of the selectivity of a relational query; this framework is known as the Probabilistic Relational Model (PRM). This imprecision stems from the assumption of uniform distribution of values for attributes in a table, attribute independence in one table, and attribute independence in tables that are semantically related. The proposed solution uses a probabilistic model to represent the distribution of values of each attribute and the correlations between attributes. Thus, instead of computing the query selectivity in terms of the number of different values of each attribute in the *select* condition of the query, the selectivity is computed using the result of a probability query to the model. In [5], Statistical Relational Models (SRM) were developed. They are different from PRM because they represent a statistical model of a particular database state instead of representing any state. Thus, Conditional Probability Table (CPT) construction in SRMs is done through queries to the database whereas the structure and CPT construction in PRMs is conducted by using machine-learning techniques.

The difference between the solution proposed by Getoor, et. al. [5, 6] and the solution presented in our paper, is the scalability to large-sized RDF repositories by means of multidimensional histograms. The SRM, developed in [5] assume a low number of values for each variable in the model. On the other hand, although in our work, an RDF document is modeled similarly to an SRM, its nodes and arcs have a particular semantics based on the RDF graph semantics, i.e., subject, property and object triples. Besides this, in our proposed RBN model, there are also *Join* variables, but restricted to the possible combinations between subjects and objects. Additionally, the purpose of the Bayesian network proposed by Getoor, et. al., is the estimation of query selectivity. In our work, Bayesian networks are applied to RDF documents in order to estimate the selectivity of query evaluation plans and sub-plans.

The work described in [9, 11, 12] extends the Ontology Web Language (OWL) with constructs that allow the annotation of an ontology with probabilities and causal relationships. These annotations are done with the purpose of reasoning on uncertainty in ontologies. Once an ontology is annotated, it is translated to a Bayesian network, and Bayesian inference queries may be answered. The main difference between these models and our research is that since the information on subject an object values are kept in an aggregated form, our combinated approach of Bayesian networks and multidimensional histograms scales up to large RDF documents. Besides this, in our work we define random variables that represent the event that a property may be combined (*Join*) with another property; these type of variables are not considered in these approaches.

7 Experimental Study

The goal of the experimental study was to analyze the benefits of the proposed predictive model when applied to the problem of query optimization. First, the predictive capacity of the model was studied and then, the quality of the optimal query was compared to the original query and to the optimal plan identified by a cost model that assumes independence between properties and uniform distribution of values.

We used the real-world dataset on the US Congress bills voting process for the years 1998, 1999 and 2000 published at `http://www.govtrack.us/`. Besides the election results, we also consider census information about representatives such as religion and gender, and political information such as the party and their state. The number of triples in the dataset for years 1998, 1998-1999 and 1998-1999-2000 is 50,860, 94,590 and 128,852, respectively.

The query benchmark is comprised of 112 queries with five instantiated patterns. The properties in the patterns and their ordering are the same for all queries, but the instantiations are different. Previously, we determined that these properties are correlated and thus, queries with different instantiations will have different selectivity.

We use the Bayesian inference tool, *SamIam* [1], to build the RBN based on the graph structure, and the CPT which is represented as a multidimensional histogram. Currently, the graph structure is built by hand, but this could be done semi-automatically. The graph in the RBN was built according to the properties represented in the ontology. Then, the CPT were developed using multidimensional histograms to aggregate the node values. The structure of this RBN was illustrated before as Figure 2. Each CPT for a target node is a multidimensional histogram, where the first dimension corresponds to a node itself, and the rest of the dimensions correspond to the predecessors of the node. The algorithm for multidimensional histogram generation constructs a histogram for the first dimension, and then for each bucket, it generates a histogram for the second dimension, and so on, until all dimensions are completed. These histograms were generated through the histogram options provided by the Oracle DBMS [8]. The default histogram option generates equal-width or equal-height histograms according to the number of different values of an attribute and its distribution.

In order to exploit the DBMS histogram mechanisms, we loaded a relational table for each property in the ontology. For each target node, we created a relational table that is a combination of the subject or object of the property that is represented by the node, with the subjects or objects of all its predecessors. We used methods in the Oracle package DBMS_STATS to generate an histogram on the column that represents the target node in the "combination" table. Then, for each bucket we created a table and again used DBMS_STATS to generate an histogram on the second dimension, and so on until all the dimensions had been covered. The histogram was completed with the computation of the frequency of each value of the target node given the different sets of values of its predecessors.

Bayesian inference queries are posed to the network through the *SamIam* tool in order to estimate the selectivity of each query based on the instantiations of its patterns. We use one of the algorithms implemented by *SamIam*, the Shenoy-Shafer exact inference algorithm [2]. Each query was also evaluated and we obtained the number of results. Thus, we compared the estimate of the selectivity with the actual number

of answers. The correlation value is 0.95. This result indicates that there exists a linear relationship between the estimates and the actual values, so we may assert that the BAY-HIST model is capable of predicting the selectivity of a query plan or sub-plan, and therefore, we can have a precise estimate of this plan's evaluation cost.

The purpose of our next experiment was to study the effectiveness of our optimization techniques when used with the BAY-HIST prediction model. Given that the BAY-HIST model is capable of considering dependencies between properties and its distribution of values, the quality of the optimal plan identified by the optimizer using BAY-HIST should be better than the quality of a plan identified by an optimizer that uses a cost model that does not consider dependencies between properties and non-uniform distribution of values. We report on runtime performance, which corresponds to the *user time* produced by the *time* command of the Unix operation system.

We used the same dataset and RBN as the previous experiment. We also used the same query benchmark, but we shuffled the queries, evaluated them and chose the 21 queries that had the worst evaluation time. The experiment was performed using these 21 queries. The Simulated Annealing optimization algorithm was configured with an initial temperature of 700, and 20 iterations in the initial stage.

We compared the performance of the original query, the optimal plan identified by the optimizer with the model that assumes property independence and uniform distribution, and the optimal plan identified by the optimizer with the BAY-HIST model. These plans were evaluated with and without index structures[1].

The average evaluation time is reported in Figure 7. We can observe that the performance of the optimal plans without index structures exceeds the performance of the original queries by up to one order of magnitude. The improvement with the use of the index strucures with respect to the original plans is up to two orders of magnitude, but the improvement is even greater when the optimizer uses the BAY-HIST model. We also observed that this difference is proportional to the incremental size of the datasets.

These results indicate that the quality of the plan identified by the optimizer and the BAY-HIST model, is better than the quality of the optimal plan identified by the optimizer with the traditional prediction model and the benefits are even greater when index structures are used.

8 Conclusions and Future Work

We present the BAY-HIST Prediction Model, a combination of Bayesian networks and multidimensional histograms, which is able to estimate correlations between data values in an RDF document as well as their distribution. We study the benefits of applying BAY-HIST to the problem of query selectivity estimation as part of cost-based query optimization; also, we report initial experimental results that suggest that the quality of the optimal evaluation plans can be improved when selectivity is estimated using the BAY-HIST Prediction Model.

In the future we plan to use BAY-HIST on the RDF(S) and OWL formalisms; also, we will study the benefits of this prediction model when it is used to discover links be-

[1] Denoted as Bhyper according to the hypergraph RDF model that these index structures implement [3].

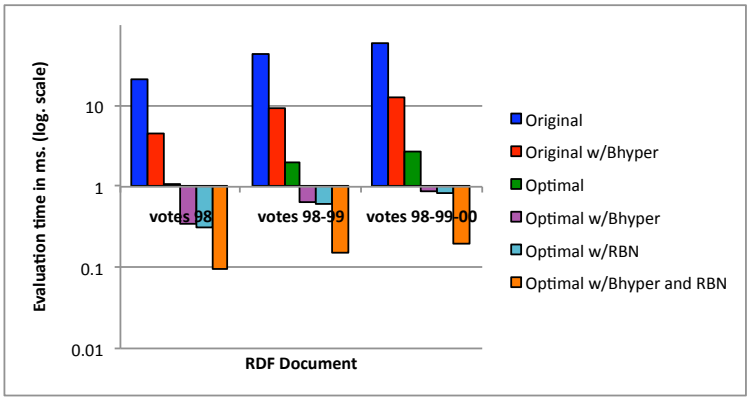


Fig. 7. Quality of the Optimal Plan

tween data terms. Currently, the optimization algorithm queries the RBN for the selectivity of all the sub-plans in each execution plan. Future work will also include keeping track of probability queries posed against an RBN in each execution plan, in order to improve the efficiency of the cost model.

References

1. *SamIam* - Sensitivity Analysis Modeling Inference and More. Automated Reasoning Group, University of California, Los Angeles. http://reasoning.cs.ucla.edu/samiam/.
2. Darwiche A. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.
3. Martinez A. and Vidal M. A Directed Hypergraph Model for RDF. In *KWEPSY*, 2007.
4. Ruckhaus E., Ruiz E., and Vidal M. Query evaluation and optimization in the semantic web. *Theory and Practice of Logic Programming - TPLP*, 8(3):393–409, 2008.
5. Getoor L. Learning statistical models from relational data, 2001.
6. Getoor L., Taskar B., and Koller D. Selectivity estimation using probabilistic models. In *SIGMOD Conference*, pages 461–472, 2001.
7. Vidal M., Ruckhaus E., Lampo T., Martinez A., Sierra J., and Polleres A. Efficiently joining group patterns in SPARQL queries. In *Proceedings ESWC*, 2010.
8. *ORACLE*. Oracle Database Management System. http://www.oracle.com/.
9. Da Costa P., Laskey K., and Laskey K. PR-OWL: A bayesian ontology language for the semantic web. In *ISWC-URSW*, pages 23–33, 2005.
10. Lampo T., Ruckhaus E., Sierra J., Vidal M., and Martinez A. OnEQL: An Ontology-based Architecture to Efficiently Query Resources on the Semantic web. In *Proceedings of SSWS, collocated with ISWC*, 2009.
11. Yi Yang and Jacques Calmet. Ontobayes: An ontology-driven uncertainty model. In *Proceedings CIMCA '05*, 2005.
12. Ding Z., Peng Y., and Pan R. A Bayesian Approach to Uncertainty Modeling in OWL Ontology. In *Proceedings of the International Conference on Advances in Intelligent Systems - Theory and Applications*, 2004.

Extending Bayesian Classifier with Ontological Attributes

Tomasz Łukaszewski

Institute of Computing Science, Poznan University of Technology,
ul. Piotrowo 2, 60-965 Poznan, Poland
tlukaszewski@cs.put.poznan.pl

1 Introduction

The goal of inductive learning classification is to form generalizations from a set of training examples such that the classification accuracy on previously unobserved examples is maximized. Given a specific learning algorithm, it is obvious that its classification accuracy depends on the quality of training data. In learning from examples, *noise* is anything which obscures correlations between attributes and the class [1]. There are many possible solutions to deal with the existence of noise. Data cleaning or detection and elimination of noisy examples constitutes the first approach. Due to the risk of data cleaning, when noisy examples are retained while good examples are removed, efforts have been taken to construct noise tolerant classifiers. Although both these approaches seem very different, they try to somehow 'clean' this noisy training data.

In this paper, we propose an approach to 'admit and utilize' noisy data by enabling to model different *levels of knowledge granularity* both in *training* and *testing* examples. The proposed knowledge representation use hierarchies of sets of attribute values, derived from subsumption hierarchies of concepts from an ontology represented in description logic. The main contributions of the paper are: (i) we propose a novel extension of the naïve Bayesian classifier by hierarchical, ontology based attributes (*ontological attributes*), (ii) we propose an inference scheme that handles ontological attributes.

2 Description-noise and Levels of Knowledge Granularity

There are three major sources of noise: (i) insufficiency of the description for attributes or the class (or both), (ii) corruption of attribute values in the training examples, (iii) erroneous classification of training examples [1]. The second and third source of noise can lead to so-called *attribute-noise* and *class-noise* respectively. Attribute-noise is represented by: (i) erroneous attribute values, (ii) missing or "don't care" attribute values, (iii) incomplete attributes or "don't care" values. The class-noise is represented by: (i) contradictory examples, or (ii) misclassification [2]. However, the first major source of noise, although not easily quantifiable, is important. This insufficiency of the description can lead to

both erroneous attribute values and erroneous classification. Let us call this resulting noise as *description-noise*. Following for example [3] the main reason for description-noise may be in a language used to represent attribute values, which is not expressive enough to model different *levels of knowledge granularity*. In such a case, erroneous or missing attribute values may be introduced by users of a system that are required to provide very specific values, but the level of their knowledge of the domain is too general to precisely describe the observation by the appropriate value of an attribute. Even if the person is an expert of the domain, erroneous or missing attribute values can be observed as a consequence of lack of time, or other resources to make detailed observations (ie. a more complete description). However, if the language enabled modeling different levels of knowledge granularity (very precise or more general descriptions), we would be able to decrease a level of this description-noise.

In order to model different levels of knowledge granularity, each testing and training example would be described by *a set of values* for any attribute. These sets of values should reflect the domain knowledge and could not be constructed arbitrarily. Let us notice, that in some domains, hierarchical or taxonomical relationships between sets of values, represented by so called *concepts*, may be observed and this knowledge could be explored. Such knowledge is currently often available in the form of *ontologies*. The most widely used language to represent ontologies, suitable in particular to model taxonomical knowledge, is *Web Ontology Language (OWL)* [1]. The theoretical counterpart of OWL, from which its semantics is drawn, is constituted by a family of languages called *description logics (DLs)* [4]. A description logic *knowledge base*, KB, is typically divided into *intensional* part (*terminological* one, a $TBox$), and *extensional* part (*assertional* one, an $ABox$).

3 An Ontological Attribute

Given is an attribute A and the set $V = \{V_1, V_2, ..., V_n\}$, where $n > 1$, of nominal values of this attribute. Let us assume that given is a TBox, which specifies domain knowledge relevant to a given classification task. In particular, it expresses a multilevel subsumption ("is-a") *hierarchy of concepts*. Each concept is described by a subset of the set V for every attribute A. Then we can formulate a definition of an *ontological attribute* as follows.

Ontological attribute An ontological attribute $\mathcal{A}$ is defined by a tuple $\langle \mathcal{H}, V \rangle$, where:

- by $\mathcal{H}$ is denoted a multilevel subsumption hierarchy of concepts, derived from a DL knowledge base. This hierarchy of concepts consists of the set of nodes $N^H = \{root, N^C, N^T\}$. This hierarchy defines a *root-node*, denoted by *root*, a set N^C of *complex-nodes* and a set N^T of *terminal-nodes*.

[1] www.w3.org/TR/owl-features/

– by V is denoted a finite set $V = \{V_1, V_2, ..., V_n\}$, where $n > 1$ of nominal values of A.
– each node $N_k \in N^T \cup N^C$ represents a subset of the set V, denoted as $val(N_k)$; the root-node represents the set V

To model actual training examples, an ABox would be used.

3.1 Using Ontological Attributes in the Naïve Bayesian Classifier

In order to apply the proposed ontological attributes in the naïve Bayesian classifier, we further specify the general definition of an ontological attribute given in the former section. Please note, that by making the assumptions presented in the following paragraphs, we will implicitly switch from the usual open world assumption used to reason with a DL knowledge base to produce a concept hierarchy, to the closed world assumption, more appropriate to the case of inference with naïve Bayesian classifier. In particular we will assume that a hierarchy of concepts would represent such *hierarchical partitioning* of the set V of attribute values, such that each concept would correspond to a non-empty subset of V.

Properties of nodes Each complex-node represents a concept from the KB, described by a proper, non-empty subset of V. Each terminal-node represents a concept from the KB, described by a unique value V_i from the set V.

Relations between nodes For a given ontological attribute $\mathcal{A}$, the hierarchy $\mathcal{H}$ is a tree, i.e. each node $N_k \in \{N^C \cup N^T\}$ has exactly one parent, denoted as $pa(N_k)$, such that $val(N_k) \subset val(pa(N_k))$. Moreover, each node $N_k \in \{root \cup N^C\}$ specifies a set $ch(N_k)$ of his children. To model different levels of knowledge granularity, we assume that for each $N_k \in \{root \cup N^C\}$ all his children are pairwise disjoint and this node N_k is a union of his children. Finally, for each node $N_k \in \{root \cup N^C\}$ we define a set $de(N_k)$ of descendants of this node, as a set of its children or children of his descendants.

The role of complex-nodes In the setting of learning with description-noise, each training and testing example can be described in general by a set Z_l of values for each attribute A, where $Z_l \subseteq V$. We can divide training examples into *no-noisy examples* ($|Z_l| = 1$) and *noisy examples* ($|Z_l| > 1$). In order to represent noisy (training and testing) examples, the ontological attribute $\mathcal{A}$ uses complex-nodes. We will call such a hierarchy a *complex-hierarchy.*

Algorithm 1 (Populating a complex-hierarchy). For each ontological attribute $\mathcal{A}$ we proceed as follows:
We associate each training example t described by a set Z_l of values of A and a class label C_j ($t : A = Z_l \wedge C = C_j$) to a node N_k. When $|Z_l| = 1$, Z_l is associated to a terminal-node N_k, such that $Z_l = val(N_k)$. Otherwise, we associate the training example to a complex-node N_k, such that $Z_l \subseteq val(N_k)$, at the *lowest* possible level of the complex-hierarchy.

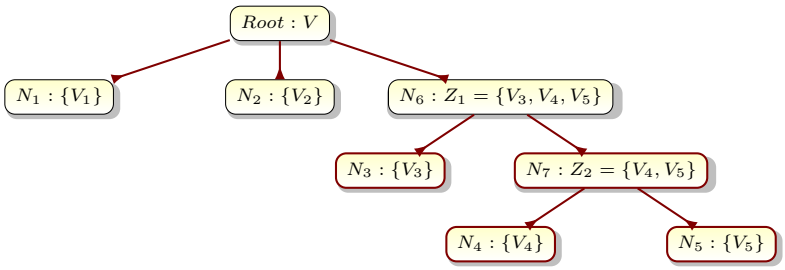


Fig. 1: A complex-hierarchy in setting with description-noise

Example Given is an attribute A such that V=$\{V_1$,V_2,V_3,V_4,$V_5\}$ and given is a class variable C such that it takes values from the set $\{C_1, C_2\}$. Let us assume, that the description-noise is modeled by sets $Z_1 = \{V_3, V_4, V_5\}$ and $Z_2 = \{V_4, V_5\}$. Let us assume a sample scenario in which the single values of the attribute A are determined by conducting three medical tests. The first test is able to partition the set V into the following disjoint subsets: $\{V_1\}$, $\{V_2\}$ and $Z_1 = \{V_3, V_4, V_5\}$. If the result of the first test is Z_1, then in some cases it is conducted a second test, that partitions the set Z_1 into the following disjoint subsets: $\{V_3\}$ and $Z_2 = \{V_4, V_5\}$. Only in critical cases it is conducted the last test, that can partition the set Z_2 into disjoint subsets: $\{V_4\}$ and $\{V_5\}$. Following this domain-knowledge, we have introduced two complex-nodes N_6 and N_7, such that they represent the sets Z_1 and Z_2 respectively. Terminal-nodes N_1, N_2, N_3, N_4, N_5 represent single values from the set V. The root-node represents the set V. The resulting complex-hierarchy is presented in Figure 1.

3.2 Inference with Ontological Attributes

We can approximate the required probability distribution for a noisy *testing* example described by a set $Z_l = val(N_k)$, following principles of the probabilistic theory, by *collecting* frequencies of training examples T, described by sets $Z_m \subseteq Z_l$, as follows:

$$P(Z_l|C_j) = \frac{\sum_{Z_m \subseteq Z_l} |T : A = Z_m \wedge C = C_j|}{|T : C = C_j|} \tag{1}$$

Let us remind, that a set Z_l is assigned to the node N_k, such that $Z_l = val(N_k)$. The key property of an ontological attribute $\mathcal{A}$, is that for the node N_k all its children are *pairwise disjoint*. Since then, all training examples described by sets $Z_m \subseteq Z_l$, are represented by the node N_k or its descendants, and the probability distribution for a noisy *testing* example described by a set Z_l we can define as follows:

$$P(Z_l|C_j) = \frac{|T : A \subseteq val(N_k) \wedge C = C_j| + \sum_{N_d \in de(N_k)} |T : A \subseteq val(N_d) \wedge C = C_j|}{|T : C = C_j|} \tag{2}$$

In this way we are able to *classify a new noisy example using other less noisy and no-noisy training examples*. For example, we can classify a testing example, described by the set Z_1, and associated to the node N_6 using all training examples described by all subsets of the set Z_1. These training examples would be associated to the complex-node N_6 or his descendants.

4 Conclusions

The topic of learning with ontologies is relatively new, and so far there are few approaches in this line of research, for the classification task see for example [5]. The simple use of ontology (Attribute Value Taxonomies) in the naïve Bayesian classifier (AVT-NBL) is presented in [6]. This approach, to the best of our knowledge, is the only one existing approach for learning the naïve Bayesian classifier from noisy (partially specified) data. Both in our approach and in AVT-NBL, noisy (partially specified) data is represented using hierarchical structures and similar aggregation procedures are used. Let us notice, that AVT-NBL requires a *static*, predefined, taxonomy of attribute values. In our approach, the hierarchy of sets of attribute values can be constructed *dynamically* driven by observations and hypotheses to prove. Moreover, our aggregation procedure allows to construct the complex-hierarchy from all possible subsets of attribute values. In this way we would be able to model any noisy training and testing example in order to achieve the highest classification accuracy, that is not possible using an Attribute Value Taxonomy. Due to limitations of the presentation, this generalization is not discussed in the paper. Let us point out, that AVT-NBL uses a propagation procedure, that does not follow principles of the probabilistic theory. Moreover, to the best of our knowldge, AVT-NBL does not classify noisy instances, which is the main goal of our approach.

In the future, we will concentrate on the problem of the optimality of the complex-hierarchy, derived from a knowledge domain of the form of subsumption hierarchies of concepts.

References

1. Hickey, R.J.: Noise Modelling and Evaluating Learning from Examples. Artif. Intell. **82**(1-2) (1996) 157–179
2. Zhu, X., Wu, X.: Class Noise vs. Attribute Noise: A Quantitative Study. Artif. Intell. Rev. **22**(3) (2004) 177–210
3. Clark, P., Niblett, T.: Induction in Noisy Domains. In: EWSL. (1987) 11–30
4. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook. Cambridge University Press (2003)
5. d'Amato, C., Fanizzi, N., Esposito, F.: Distance-Based Classification in Owl Ontologies. In Lovrek, I., Howlett, R.J., Jain, L.C., eds.: KES (2). Volume 5178 of Lecture Notes in Computer Science., Springer (2008) 656–661
6. Zhang, J., Honavar, V.: AVT-NBL: An Algorithm for Learning Compact and Accurate Naïve Bayes Classifiers from Attribute Value Taxonomies and Data. In: ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining, Washington, DC, USA, IEEE Computer Society (2004) 289–296

Random indexing spaces for bridging the Human and Data Webs

Jose Quesada, Ralph Brandao-Vidal, Lael schooler

Max Planck Institute, Adaptive Behavior and Cognition, Berlin
Lentzeallee 94, 14195 Berlin
{quesada, rbrandao, schooler}@mpi-berlin.mpg.de

Abstract. There exists a wide gap between the information that people and computers respectively can operate with online. Because most of the web is in plain text and the Semantic Web requires structured information (RDF), bridging the two worlds is an important current research topic. Here we propose a web service that uses a Random Indexing (RI) semantic space trained on the plain text of the one million most central Wikipedia concepts. The space provides us with vectors for each of the equivalent DBpedia concepts and vectors for any text or webpage. It can also provide a hashed version of the RI vector that works as unique handler like URIs do, but with the additional advantage that it represents text meaning. As a result, any page (previously readable only for humans) is now integrated with the Semantic Web graph using links to one of its most central parts, DBpedia.

Keywords: text mining, statistical semantics, structured information, identifiers, resources, literals, RDF

1 Introduction

Most of the existing knowledge on the Web is in plain, unstructured text[1]. That is, most web pages contain data expressed in a way that is easily understandable for humans but hard to interpret for machines. The Semantic Web promises large interoperability gains, but it all depends on how well we can integrate two separate worlds. On the one hand we have rich structured datasets following linked data principles [1] with the ultimate goal of being able to use the Web like a single global

[1] Governments, enterprises and almost any dynamic website all have large bodies of knowledge already in structured form (relational databases) but not following linked data principles. Converting it into RDF is an interesting problem, but not directly related to the problem we discuss here (how to find the closest DBpedia concepts to any text passage) so we will not elaborate further.

database. But while the Semantic Web graph is growing at a very healthy rate[2], it is still a marginal part of the entire Web. On the other hand we have the flat, messy (but abundant) plain text Web pages. Traditional information retrieval and machine learning techniques that work on plain text have been making steady progress for some time now. Some of these techniques use structured data too [2].

The problem we aim to solve in this paper is simply converting literals into resources. This problem is trivial if the only requirement is a unique ID (a random one would suffice). But giving unique IDs is not a solution; to integrate the new resource we need to generate outwards links to the rest of the Semantic Web graph. That is, we enhance the meaning of the new node by generating new connections. We achieve this thanks to statistical semantics on a corpus that has parallel representations of both worlds: Wikipedia/DBpedia [3]. Random indexing (RI) [4] offers a highly scalable way of assigning semantic vectors to Wikipedia concepts. We then compress the vectors using MD5 hashing and use these hashes as meaningful identifiers that become part of the RDF graph.

For clarity, we will refer to the Semantic Web and linked data initiative as the data Web. The human Web is simply the current Web made of pages and unnamed links.

1.1 An overview of URIs, Resources and literals

We build on three basic notions: URI, resources, and literals. In summary, URIs are unique identifiers, and resources differ from literals in that they have URIs and can link to other nodes in the graph. We will describe these three concepts next.

1.1.1 Uniform Resource Identifier (URI)

A Uniform Resource Identifier (URI), according to the specification [5], is a compact sequence of characters that identifies an abstract or physical resource. Valid URIs take the following form:

Scheme ":" ["//" *authority* "/"] [*path*] ["?" *query*] ["#" *fragment*]

Uniform Resource Locators (URLs) are a subclass of URI, subject to the same grammar. The main difference is that a URL must point to specific information, usually a file that can be displayed on a browser or downloaded, whereas URIs do not need to[3]. People who have been on the internet for years now are completely used to this grammar. Note that none of the parts are particularly informative to describe the resource they point to. *Authority* is perhaps informative because it could carry the name of the entity (company, person, association, etc) that hosts the page. In recent years RESTful services [6] make *Paths* describe the actions they perform (e.g. read, delete, etc). The title of the page can also be part of *Path*, and some popular software such as *WordPress* implements this policy by default. However, these are all usage

[2] New datasets are added constantly to the W3C site 'Linking Open Data' http://esw.w3.org/TaskForces/CommunityProjects/LinkingOpenData/DataSets and the existing ones keep growing.

[3] But it is a good practice to make URIs point at some description of what they are.

conventions, but not enforced by the URI design. There is nothing in the scheme that says URIs should be meaningful for humans or machines[4].

The main role of a URI (and only requirement) is to provide a unique identifier for a resource. In this paper we will propose that it is desirable to make identifiers meaningful for machines, in a way that uses human similarity judgments.

1.1.2 Resource

The first explicit definition of resource is found in RFC 2396 [7] and states that *A resource can be anything that has identity. Familiar examples include an electronic document, an image, a service (e.g., "today's weather report for Los Angeles"), and a collection of other resources. Not all resources are network "retrievable"; e.g., human beings, corporations, and bound books in a library can also be considered resources.*

The concept of resource is primitive in the Web architecture, and is used in the definition of its fundamental elements. The term was first introduced to refer to targets of URLs, but its definition has been further extended to include the referent of any Uniform Resource Identifier in RFC 3986 [5]. That is, the concept started in the human Web, and grew to be used in the data Web. A resource is simply anything that can be identified with a URI. Note that the concept of URI contains the URL as a special case.

Resources can have properties. For example, the resource 'FidoTheDog' may have the Name property 'Fido'. That is, resources can link to other resources and to literals.

1.1.3 Literals

Literals are values that do not have a unique identifier. They are usually a string that contains some human-readable text, for example names, dates and other types of values about a subject. In the previous example, the string 'Fido' is a literal. They optionally have a language (e.g., English, Japanese) or a type (e.g., integer, Boolean, string), but this is about all that can be said about literals. They cannot have properties like resources. Unlike resources, literals cannot link to the rest of the graph. They are second-class citizens on the Semantic Web. In terms of graphs, literals are one-way streets: since they cannot be the subject of a triple, there can be no outgoing links to other nodes.

[4] In practice, URLs do have some meaning for humans, but mostly due to cues acquired after years of using them. Short, meaningful names are better, and of course more expensive, so they hint that the owner must have made a serious investment and thus be committed to the content.

1.2 Why turning literals into resources is useful

Consider that human Web nodes (pages) are literals once they merge with the data Web. The number of literals in the joint graph will be enormous, considering that the human Web is several orders of magnitude larger than the current data Web. But the number of new nodes is not necessarily just the number of webpages: a selection of text, say a paragraph, can also become a literal.

We offer a method to transform literals into resources. This solution we propose here is one of many possible: for example, there are efficient tools, such as openCalais, that link entities to semantic Web concepts using named-entity recognition. The key difference is that named-entity recognition links individual words to existing resources, where as we create a URI for larger chunks of text, such as a sentence, a paragraph or entire webpage. Like openCalais, we link the resulting URI's to DBpedia [3], one of the most central datasets in the Semantic Web.

In the next subsections, we show advantages to turning literals into resources from a graph machine learning point of view.

1.2.1 Increased integration of the human and data Webs

The current state is that even though the two Webs are essentially separated, there is some integration in at least two fronts. First, semantic Web URIs resolve into something a human with a browser can see (e.g., plain text description of an object). This is a good practice, but not enforced. Second, recently more and more parts of the human Web carry snippets of structured information (RDFa). Only recently have webmasters started using RDFa. Search engines such as Yahoo and Google are indexing RDFa too.

Integration is challenging because the two webs are structurally very different. The semantic Web is a directed labeled graph, whereas the 'human' Web is a directed unlabeled graph[5]. To merge them, we would need to produce labels for unlabeled links. But this is a problem because links in the human Web, by design, do not have labels. We could use a homogeneous label name (something like 'links-to') but then 'links-to' would become the most frequent label, eclipsing every other one and making the resulting graph harder to do reasoning with. An example of this generalist label is the 'wikilink' predicate in DBpedia. Wikilinks are the simplest links from one Wikipedia article to another. They are parsed from Wikipedia articles bodies for DBpedia as simple "source page" and "destination page" pairs. Compared to the other kinds of RDF triples in DBpedia, they are the most general, in the sense that they cover the most kind of relations, yet are the least precise, because they don't have a relation property, only using a generic "wikilink" relation type. There are 70 millions Wikilink triples, compared to 30 million Infobox dataset triples or only 7 millions Wikipedia Categories dataset triples. In such a large proportion, unnamed links would

[5] Directed labeled graphs are a lot harder to work with than unlabeled graphs, and the algorithms that work on directed labeled graphs are but a portion of all graph algorithms.

overpower the named ones for some tasks (e. g. [8]) and their addition would be detrimental.

1.2.2 Dangling nodes

One important feature of RDF is that a literal may be the object of an RDF statement, but not the subject or the predicate. Because a resource offers richer possibilities to Semantic Web practitioners compared to a literal, the joint graph would be better served having as many resource nodes as possible. From the point of view of graph theory, literals are 'one-way street' nodes that can be problematic. A node that receives connections but never links outwards is called a 'dangling node'. So literals are dangling nodes. Operating on a graph with a high proportion of dangling nodes makes some useful algorithms slower (e.g., finding shortest paths), and some other harder to use or impractical. For example, straight pagerank has problems with dangling nodes, even though in practice they can be solved [9], but other algorithms such as singular value decomposition require a matrix with no all-zero rows (a dangling node produces an all-zero row).

One alternative is to remove dangling nodes. Some studies that look for shortest paths remove literals because dangling nodes would add one-way-streets and search would take longer [10]. But this has unintended secondary effects. Removing the dangling nodes somewhat skews the results on the non-dangling nodes since the outdegrees from the non-dangling nodes are adjusted to reflect the lack of links to dangling nodes.

The Semantic Web graph has a large proportion of dangling nodes. According to data reported in the landing page of the Linked-Data Semantic Repository (LDSR, [11] including DBpedia, Freebase, Geonames, UMBEL, Wordnet, CIA World Factbook, Lingvoj, MusicBrainz and others), 39% of the nodes are literals (see table 1). This is the proportion of literals over the total number of entities. LDSR is not the Semantic Web's entire graph, but we would expect to find a similar distribution of URIs vs. literals if we could access equivalent statistics for every subcomponent.

Table 1. Statistics from the linked-data semantic repository (LDSR, [11], retrieved 3-4-2010)

Number of URI:	126,875,974
Number of Literals:	227,758,535
Total number of entities:	354,635,159

Reducing the proportion of literals compared to resources on the Semantic Web graph may open the door to better machine learning algorithms. We will explore this idea in the next section.

2 How to create Identifiers that are not only unique, but meaningful

Here we use statistical semantics to create meaningful identifiers for literals. We term these meaningful unique identifier (MUID, pronounced "mood"). We propose that algorithm for generating MUID's should have the following properties:

1. A MUID should have some (primitive) form of compositionality. If we generate a MUID for part of a page, the part's MUID should be similar to that of the full page.

2. If two pages get similar MUIDs, they should be perceived as similar by human observers.

3. Changes that are perceived as incremental by people (e.g., a blog post getting comments), should result in incremental changes to the corresponding MUID's corresponding to before and after the changes.

To understand how our proposal implements these requirements, we next describe statistical semantics, focusing on Random Indexing (RI) [15].

2.1 Statistical semantics

Statistical semantics is a general category of machine learning algorithms that exploits statistical patterns of human word usage to figure out word meaning. These algorithms come from cognitive science and information retrieval. A typical task for statistical semantics is to measure the semantic similarity of two passages. The answer is given as a number, usually the cosine between the vectors that represent the passages in some high dimensional space. The vector for a passage is usually the average of the vectors for all the words in it. The vectors for each passage, when averaged together, form the document vector. This implements compositionality (property 1 above) and addresses incremental changes (property 3), because recomputing a vector when the text is only slightly different will produce only a slightly different vector.

Most statistical semantics methods start with a frequency matrix of word by documents [12], and many apply different transformations to these matrix (example, truncated singular value decomposition). The vector space model [12] was the first of these methods. It improves over Boolean information retrieval (IR) in that it allows computing a continuous degree of similarity between queries and documents, and this makes ranking possible. It also moved IR from set theory to linear algebra, which facilitated the explosion of newer models. These newer models such as LSA and random indexing extend the approach by adding generalization, that is, these models are able to tell when two words are synonyms. Table 2 shows an example. For an exact matching algorithm based on a Boolean vector space, the similarity between pairs of words is all-or-nothing. In contrast, newer models such as LSA and RI capture the similarity of doctor to physician and surgeon.

Table 2. Generalization. How LSA solves synonymy. Cosine values from lsa.colorado.edu

	Boolean/vector space	LSA
doctor – doctor	1	1
doctor – physician	0	.8
doctor – surgeon	0	.7

Statistical semantic models are trained on a large corpus of text that is representative of the domain of interest. Once this space has been created, it can be used to compare not only passages from the training corpus, but novel passages as well. For general knowledge, the corpus used to be an encyclopedia or a sample of textbooks representative of what a college student would have read [13]. A corpus composed of traditional encyclopedias or textbooks have significant limitations. First many recently-coined terms common on the web, such as iPad, are not in those datasets. And second, there's no direct mapping between these corpora and resources in the semantic web. These limitations lead us to use Wikipedia as a training corpus. In the combination of Wikipedia and DBpedia we have exactly what we need, a parallel corpus that exists in both the human and data Webs.

Starting with the full text of a recent (March 2008) Wikipedia dump, we selected the most central concepts by dropping those with fewer than five in links or fewer than five out links. We applied other basic preprocessing steps described in [14]. The initial parsing produced close to a million types; as expected from natural, unedited text, most of these were typos. We then dropped types that occurred less than 10 times, and those that appeared in less than 10 documents. Approximately half of the types went away. After parsing the Wikipedia XML dump, we obtained 2.7 Gb of text in 1,279,989 articles.

3 Random indexing

Our application of Random Indexing [4] starts with the same words by documents matrix described above, taking a document to be a Wikipedia article. Then each word and each context is first assigned a random high-dimensional sparse vector: they are seeded with a small proportion of ones and negative ones with all other elements set to zero.

Once the sparse binary index vectors are constructed, a word's vector becomes the sum of the vectors for the contexts in which it appears throughout the text corpus. Conversely, a document space can also be constructed as the sum of the index vectors for words appearing in each document. Random Indexing depends on the term-document matrix computed from a corpus being sufficiently sparse that vector representations can be projected onto a basis comprising a smaller number of randomly allocated vectors. Due to the sparseness condition, the basis of random vectors has, in general, a high probability of being orthonormal. That is, every random vector will be orthogonal to any other random vector. The most exhaustive

description of RI is [15]. The main advantage of RI compared to LSA [16] is its scalability. The SVD is a computationally expensive operation. It needs to place large matrices in memory, and it may take days to compute for a dataset the size of LDSR, if at all possible. RI does not have large memory requirements and the linear algebra operations are simpler and faster.

We used the semantic vectors library [17]. This library has been proved to scale well: Cohen et al. [18] use it in an experiment with 15M documents from MedLine.

We manipulated the following parameters (see first two rows in table 3): (1) Number of dimensions. This is simply the size of the random vector that represents a word. It has the largest influence on how long it takes to compile a space, and how much storage it needs. Surprisingly, there is little published on how to select the optimal dimensionality. We manipulated dimensions from 800 to 1200. (2) Nonzero seed values. This parameter is not commonly reported in the literature. However, we found that it does change results, so we manipulate it here systematically.

Our assumption is that the parameters that work best on traditional psychological tasks will also work well for our current task of getting the most meaningful neighbors on a Wikipedia space. In the next section we try to obtain the best parameters for our web service using four well-known human similarity datasets.

3.1 Results

We used the following datasets for word pair similarity judgments: Rubenstein and Goodenough (1965) [19], Miller and Charles (1991) [20], Resnik [21] (1995; this is a replication of Miller and Charles) and Finkelstein et al. (2002) [22]. An example of the materials on these tasks would be 'How similar are gem and jewel?' Participants produce ratings going from zero (not related at all) to four (perfect synonymy).

Table 3 shows how models, based on different parameterizations of RI, correlate with the human judgments in these four datasets for word-word comparisons.

Since average human agreement in tasks like these is around .6, our results are acceptable, even though they are below some other published results [23]. For the web service, we kept the space with 1000 dimensions and 700 seed values, which seems to do well across datasets.

4 The Web service

The interface to the Web service is described using WSDL [24]. The Web service takes either a URL or plain text. When taking a URL, it parses the page and extracts the plain text. The text is then transformed into a RI vector by retrieving vectors for all its terms and averaging these together. What we provide here is a prototype that

takes about 2 min to process a request. Fortunately, the algorithm is parallelizable. An interface for testing can be reached at: http://mpi-ldsr.ontotext.com/webservice6.

Table 3. RI correlation to the human gold standard in four datasets for word-word comparisons. The average human agreement is around .6. Best results are bold.

dims	seed	Miller	Resnik	Rubenstein	wordsim
800	500	0.39	0.46	0.42	0.35
	600	0.61	0.54	0.52	0.4
	700	0.57	0.56	0.48	0.35
	750	0.37	0.44	0.39	0.34
1000	300	0.5	0.46	0.42	**0.4**
	700	**0.55**	**0.6**	0.5	0.39
	800	0.42	0.47	0.46	0.36
	900	0.48	0.5	0.42	0.37
	950	0.6	0.55	**0.53**	0.37
	980	0.49	0.43	0.4	0.36
1200	500	0.53	0.55	0.53	0.38
	900	0.5	0.56	0.47	0.36
	1000	0.47	0.51	0.46	0.36
	1050	0.34	0.5	0.41	0.37
	1100	0.42	0.45	0.38	0.37
1800	1500	0.43	0.43	0.5	0.37
2000	1900		0.57		

The web service returns both a list of nearest neighbors in the space and a unique, meaningful ID (MUID). Table 4 shows an example of the concepts that the Web service produces for an interview with Shane Simonsen[8], a UK computer science professor who abandoned the University system. The ten closest neighbors are related to education in different parts of the world, which reflects the gist of the text.

What follows is the N3-formatted RDF that this text would return for the first item in the example. The kind of links we generate are essentially unlabeled (as discussed in the introduction), but right now we use `skos:related` to express the fact that the input text is related to the DBpedia concept listed.

```
  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
  @prefix mpib <http://mpi-ldsr.ontotext.com/mpib#'> .
  @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
  @prefix DBpedia: <http://en.wikipedia.org/wiki#> .
  mpib:39f2ea57cf982d7eedccf28f92ebf13f            skos:related
dbpebia:Education_in_the_People's_Republic_of_China> .
```

[6] Alternatively http://93.123.21.85:8087/ri-webservice/
[8] http://www.lambdassociates.org/blog/interview.htm

Since 1000-dimensional vectors are too long to be used as metadata, we return a hashed version of the vector compacted with the hashing function MD5 digest.

Table 4. A sample paragraph from submitted text (left) and top 10 DBpedia concepts that the web service produced.

Your article "Why I am not a Professor", outlining your departure from academia in 1999 over declining standards and conditions, was written in 2007. Can you shed light on any further changes of the state of the tertiary education system since then? There is a recognition at government level that the standards have dropped at university and that degree inflation is rife. The UK government has abandoned its target of 50% of the population in higher education. The public sector deficit has caused the university budget to be cut by £500 million in 2010 and we shall see further cuts. However all the mechanisms of assessment discussed in that essay are still in place.	Education in the People's Republic of China Education in the United States Community college Tertiary education in Australia Education in South Korea Business-education partnerships Unemployment Secondary education in Japan Education in Thailand Education in England

5 Discussion and conclusions

We have presented a method, reachable as a web service, to attach meaning to a resource that locates it in a semantic space. Using statistical semantics we integrate any plain text literal (a paragraph or an entire page) with DBpedia, one of the central components of the Semantic Web. When literals are passed to our web service they receive a Random Indexing vector and a list of links to the 10 closest DBpedia concepts. This Random indexing vector is taken as a meaningful, unique ID (MUID) that can be used to refer to this newly-created resource. These MUIDs serve not only as unique identifiers, but as well add functionality. In the same way that in the physical world coordinates enabled location-aware applications, the semantic annotation of literals enables new functionality, such as defining the similarity of pairs of objects, and finding the most similar resources. But there is a critical difference between semantic and physical spaces. Whereas the physical world has 3 dimensions, the semantic world, as we have proposed here, may have thousands.

6 References

[1] C. Bizer, T. Heath, and T. Berners-Lee, “Linked data–the story so far,” *International Journal on Semantic Web and Information Systems*, vol. 5, 2009, pp. 1-22.

[2] Renaud Delbru, Nickolai Toupikov, Michele Catasta, Robert Fuller, and Giovanni Tummarello, “SIREn: Efficient search on semi-structured

documents," *Lucene in Action*, Manning Publications, 2004.
[3] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia - A crystallization point for the web of data," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, 2009, pp. 154-165.
[4] M. Sahlgren, "An introduction to random indexing," *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*, Citeseer, 2005.
[5] T. Berners-Lee, R. Fielding, and L. Masinter, "RFC 3986: Uniform resource identifier (uri): Generic syntax," *The Internet Society*, 2005.
[6] L. Richardson and S. Ruby, *Restful Web Services*, O'Reilly Media, 2007.
[7] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform resource identifiers (URI): generic syntax," 1998.
[8] O. Liu, "Relation Discovery on the DBpedia Semantic Web," 2009.
[9] I.C.F. Ipsen and T.M. Selee, "PageRank computation, with special attention to dangling nodes," *SIAM J. Matrix Anal. Appl*, vol. 29, 2007, pp. 1281–1296.
[10] J. Lehmann, J. Schüppel, and S. Auer, "Discovering unknown connections–the DBpedia relationship finder," *1st SABRE Conference on Social Semantic Web (CSSW)*, 2007.
[11] A. Kiryakov, D. Ognyanoff, R. Velkov, Z. Tashev, and I. Peikov, "LDSR: a Reason-able View to the Web of Linked Data," *International Semantic Web Conference (ISWC)*, 2009.
[12] G. Salton, A. Wong, and C.S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, 1975, p. 620.
[13] S.M. Zeno, S.H. Ivens, R.T. Millard, and R. Duvvuri, *The educator's word frequency guide*, Brewster, NY: Touchstone Applied Science Associates, 1995.
[14] E. Gabrilovich and S. Markovitch, "Wikipedia-based semantic interpretation for natural language processing," *Journal of Artificial Intelligence Research*, vol. 34, 2009, pp. 443-498.
[15] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive Computation*, vol. 1, 2009, pp. 139-159.
[16] T.K. Landauer and S.T. Dumais, "A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge," *Psychological Review*, vol. 104, 1997, pp. 211-240.
[17] D. Widdows and K. Ferraro, "Semantic vectors: a scalable open source package and online technology management application," *Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, 2008.
[18] T. Cohen, R. Schvaneveldt, and D. Widdows, "Reflective Random Indexing and indirect inference: A scalable method for discovery of implicit connections," *Journal of Biomedical Informatics*, 2009.
[19] H. Rubenstein and J.B. Goodenough, "Contextual correlates of synonymy," *Communications of the As- sociation for Computing Machinery*, vol. 8, 1965, pp. 627-633.
[20] G.A. Miller and W.G. Charles, "Contextual Correlates of Semantic Similarity.," *Language and cognitive processes*, vol. 6, 1991, pp. 1-28.
[21] P. Resnik, "Using information content to evaluate semantic similarity in a

taxonomy," *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 1, 1995, pp. 448-453.

[22] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, and G. Wolfman, "Placing search in context: The concept revisited," *ACM Transactions on Information Systems*, vol. 20, 2002, pp. 116–131.

[23] M.N. Jones and G. Recchia, "Scalable Techniques for Creating Semantic Vector Representations," *under review*, 2010.

[24] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, *Web services description language (WSDL) 1.1*, 2001.

XML Schema and Topic Map Ontology for Background Knowledge in Data Mining

Tomáš Kliegr[1], Vojtěch Svátek[1], Milan Šimůnek[1], Daniel Šťastný, Andrej Hazucha

University of Economics, Prague, Dept. Information and Knowledge Engineering, Nám. Winstona Churchilla 4, 130 67 Praha 3, Czech Republic, {tomas.kliegr,svatek,simunek,xstad17,xhaza00}@vse.cz

Abstract. Background (or sometimes referred to as domain) knowledge is extensively used in data mining for data pre-processing and for nugget-oriented data mining tasks: it is essential for constraining the search space and pruning the results. Despite the costs of eliciting background knowledge from domain experts, there has been so far little effort to devise a common exchange standard for its representation. This paper proposes the Background Knowledge Exchange Format (BKEF), a lightweight XML Schema for storing information on features and patterns, and the Background Knowledge Ontology (BKOn), as its semantic abstraction. The purpose of BKOn is to allow reasoning over and integration of analysed data with existing domain ontologies. We show an elicitation interface producing BKEF and discuss the possibilities for integration of such background knowledge with domain ontologies.

1 Introduction

Elicitation of knowledge from experts has long been known as a crucial research topic in the field of expert systems, and its importance is now starting to rise in data mining applications, too. Background (or sometimes referred to as domain) knowledge is extensively used in preprocessing of data for most mining algorithms. It has special importance in association rule mining, where it is used to separate the nuggets from rules conveying uninteresting information.

Despite the potential of expert-provided background knowledge for improving the quality of data mining results, there has been so far little research effort onselecting pieces of information that should be collected and little standardization efforts on devising a common format for representation of background knowledge. This paper presents one of the first attempts to address these problems by introducing the Background Knowledge Exchange Format (BKEF) XML Schema. Simultaneously, to allow reasoning and integration of analysed data with existing domain ontologies, we propose a semantic abstraction over BKEF – the Background Knowledge Ontology (BKOn).

This paper is organized as follows. Section 2 gives an account of the proposed design objectives of a background knowledge specification. Section 3 introduces

its elementary building blocks and section 4 gives account of specificalities for association rules. The proposed BK specification consisting of BKEF XML Schema and the BKOn ontology is described in Sections 5 and 6 respectively. The new possibilities that BKEF and BKOn open in the areas of automating data mining tasks and result postprocessing are sketched in Section 7. The conclusion presents an outlook for future work.

2 Design Objectives

The work presented here reacts to the pressing need for an industry standard that would provide a common way of conveying pieces of background knowledge that express expertise related to features and patterns relevant to datasets in a given domain. Hence, although in the common case the knowledge acquistion is driven by the need for knowledge pertaining to a specific mining task and specific dataset, the standard should impose such principles that would foster reuse of the knowledge in a different task-dataset scenario. While the work presented here has experimental character, it follows some of the design guidelines that, we believe, should be addressed by any serious attempt on an industry standard specification.

We will use the term *background knowledge producer* to denote a computer program, such as a specialized elicitation interface, used by the domain expert to input his/her background knowledge related to the data mining task.

The *background knowledge consumer*, in turn, denotes a computer program that uses background knowledge (BK). We consider the following types of BK consumers: data preprocessing algorithms, data mining algorithms, postprocessing algorithms and semantic knowledge bases.

2.1 One size does not fit all

The standard should be constituted by an XML Schema and an ontology to accommodate for the different needs of background knowledge producers and consumers.

It may seem natural that the language in which the specification is defined is selected so that its expressivity is at least such as required by the most demanding consumer type, which is the semantic knowledge base. The semantic knowledge base [11] interlinks mining models, background knowledge and domain ontologies, and as such it would take advantage of background knowledge comming directly in a semantic format such as RDF/OWL [2] or the Topic Maps' XTM [7]. However, there are reasons for not using a semantic format as the primary standard used by data mining and knowledge elicitation software. The main ones include:

- poor readability due to structural complexity
- verbosity
- the need for specialized, not widely available APIs

Therefore, we propose using an XML Schema as an interchange format between background knowledge consumers and background knowledge producers. To foster the interoperability on the semantic level, the specification should also define a semantic version of the XML Schema (an ontology) and a transformation between the schema and the ontology. This transformation is to be executed on the side of the BK consumer.

2.2 Background Knowledge Consumer Requirements

The primary goal of the specification is to provide pieces of information that can be automatically processed by background knowledge consumers and doing so can enhance their functioning.

#	Consumer Type	Information	Utilization
1	Data Preprocessing	Similar value grouping	Decreasing the granularity
2	Data Mining	Search space constraints	Localizing the search
3	Postprocessing	Known patterns	Pruning
4	Semantic KBs	Annotations	Search

Table 1. Frequent use cases for background knowledge

An overview of requirements on the specification posed by the individual consumers is given in Table 1. This table was constructed based on the analysis of requirements of the LISp-Miner mining suite[1] and the SEWEBAR framework[2] as Semantic KB for association rules, but the authors conjecture that the table should be, with some changes, applicable to other mining tasks and algorithms.

Requirements on storing the types of information of types 1–3 require inherently no semantics and can be met by the XML Schema specification. Since indisputably one of the consumers of background knowledge is the human data analyst, the specification should also provide the domain expert with the possibility to complement the machine-readable values with a free-text annotation.

The requirements of the Semantic KB consumer type are addressed in subsection 2.3. While closely linked to background knowledge and essential for the Semantic KB, machine-readable annotations fall out of the scope of the background knowledge specification.

2.3 Integration with Other Specifications

The background knowledge specification discussed here has strong links with PMML, the widely adopted standard for data mining model interchange [3]. The

[1] `http://lispminer.vse.cz`
[2] `http://sewebar.vse.cz/`
[3] `http://dmg.org`

proposed specification plays the same role for background knowledge as PMML does for mining models. For background knowledge consumers to be able to apply this knowledge together with knowledge gained from PMML, the need for alignment with PMML arises.

While one of the key design objectives is independence of the BK specification of a specific dataset/task scenario, the bond between the BK specification and a concrete dataset or mining model should be established in a separate mapping specification. Further, we briefly introduce an attempt for such a specification dubbed FML (Field Mapping Language).

PMML is backed by an XML Schema, which eases the design of the mapping. A more complex problem arises with the requirements imposed by the Semantic KB consumer type. The purpose of Semantic KBs is to perform reasoning, integration and search over the data. From this arises the necessity to annotate the entities that emerged during the background knowledge elicitation process (such as features, values and patterns) with an association to relevant concepts in other ontologies or with unstructured sources. Since this annotation information transcedes the scope of a single dataset, we suggest to support it with a standalone specification (an XML Schema or an ontology) so that it is not a direct part of BKEF, but is only linked with it. Since the only BKEF consumer in our framework that has direct use for this kind of information is the Semantic KB, a semantic format such as RDF/OWL could be more convenient for storing the annotations than XML Schema. Additionaly, this annotation can aid the process of automatic mapping of BKEF onto a specific dataset resulting into an FML specification.

3 Basic Concepts

3.1 Metaattribute

The basic building block of a background knowledge specification is a *metaattribute* [14], which is an abstraction representing the underlying property of a data-field. There is a hierarchical structure between metaattributes. The metaattribute on the finest granularity level is referred to as *atomic metaattribute*. Other attributes are called *group metaattributes*.

Since a property can be sometimes measured in different ways, most commonly using different units, each metaattribute has multiple *formats*. Actually, most pieces of information relating to a metaattribute are format-dependent. Specifically, a format can contain:

- *a value range*,
- *standard value binning(s)*,
- *a collation*.

Since the specification is intended to be used in conjunction with a dataset, where a datafield always conforms to one metaattribute format, it is advantageous to introduce a common term *Meta-field* for an atomic metaattribute-format pair.

Similarly *Meta-field Value* is an abstraction of a possible 'value' of a metafield – value or interval falling within the scope given in the value range or one of the groupings.

3.2 Patterns

Known relationships between metaattributes are captured using *patterns*. Since often the pattern only applies to a specific format or involves a value, the notion of meta-field and meta-field value is central for their definition.

The purpose of patterns is to be used in conjunction with the data mining algorithm, most commonly either in the algorithm itself or in the further processing of results. As such, it is difficult to introduce a unified framework for pattern representation that would be equally usable for all types of data mining tasks and algorithms. Therefore the specification should propose suitable types of patterns for the main data mining algorithms (such as classification, clustering or association rule mining).

4 Background Knowledge for Association Rule Mining

We introduce two types of patterns that were designed to aid the association mining algorithms; their prospective utilization for other types of mining algorithms is a matter for further research. These two types are *Mutual Influences* and *Background Knowledge Association Rules*.

A Background Association Rule (BAR) has the form of

$$\kappa \ \approx_{[\iota]} \ \lambda \ [/\chi] \tag{1}$$

.

Here the Antecedent κ, Consequent λ and Condition χ are Boolean Meta-attributes and $\approx$ is a type of 4ft-quantifier. The optional ι explicitly corresponds to value(s) of Interest Measures associated with the 4ft-quantifier. The BAR is Conditional if the Condition χ is present.

4ft-quantifier corresponds to a set of conditions (*interest measures*) defined on the four-field contingency table, which is a quadruple of natural numbers $\langle a, b, c, d\rangle$ so that: a is the number of objects(rows) from the data matrix satisfying φ and ψ, b satisfying φ and $\neg\psi$, c satisfying $\neg\varphi$ and ψ and d the number of objects satisfying $\neg\varphi$ and $\neg\psi$. A *Boolean Meta-attribute* is a recursive structure comprising conjunctions, disjunctions and negations of combinations of individual items (Metafield-Value pairs). A Boolean Meta-attribute is *Basic* or *Derived*. A *Basic Boolean Meta-Attribute* has the form of $b(\sigma)$, where the *Coefficient* σ is a subset of possible Values of Meta-Field b. A *Derived Boolean Attribute* is a conjunction or disjunction of Boolean Meta-attributes, or a negation of a Boolean Meta-attribute.

The Background Association Rule can be input independently into the Pattern component of a BKEF document, or as an Atomic Consequences element

within a Mutual Influences element. The notion of *Mutual Influence* comes out of research by Rauch & Šimůnek [14], who proposed to use it as a knowledge elicitation aid.

5 Background Knowledge Exchange Format

The Background Knowledge Exchange Format (BKEF) is defined by an XML Schema and used for storing mining models of a particular knowledge domain. The BKEF XML Schema consists of two main building blocks: definitions of **meta-attributes** and definitions of patterns. A metaattribute is understood as an abstraction of the ultimate property of the mining model [14] with all characteristics explained so far, hence metaattributes are simultaneously comprised in the BKEF XML Schema. Mutual influences among the metaattributes together form a *pattern*. A simplified schema is shown in Fig. 1.

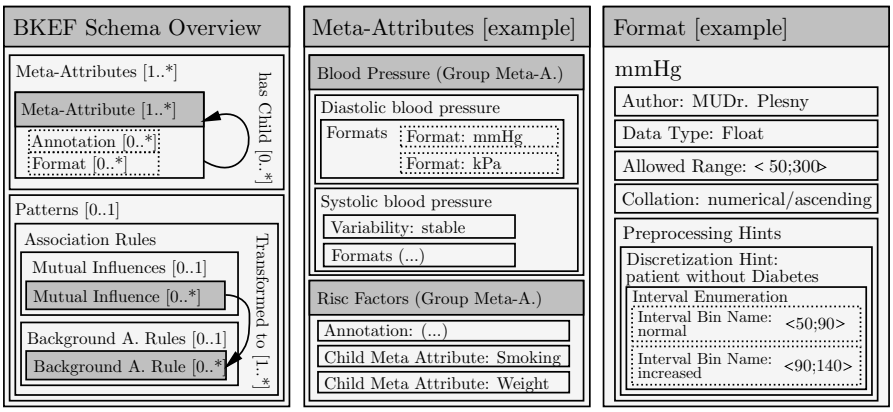


Fig. 1. Schema of BKEF

5.1 Metaattribute Definitions in BKEF

The XML Schema restricts meta-attributes to a two-level hierarchy. The base level encompasses indivisible `MetaAttributes`[4] (level = 0) - basic layer, evenly *atomic metaattribute*. The upper level comprises groups of the `MetaAttribute` elements (level = 1); each group contains an unlimited number of the `MetaAttribute`.

[4] *Typewriter* text labels on particular elements of the BKEF XML Schema where it is necessary to refer about XML elements for the proper understanding.

Groups of meta-attributes A general collection of `MetaAttribute` elements. The group should have a name, unique identification and at least one link to the `MetaAttribute` of level = 0 (which is called `ChildMetaAttribute` from this point of view).

Meta-Attribute The main focus of the `MetaAttribute` is the multiple definition of the `Format` as the property could be expressed in different ways of measurement. The `Annotation` together with the author's name are used for additional information on different authors. See an example:

```
<Annotation>
<Text>Measured in 2009</Text>
  <Author>MUDr. Plesny</Author>
</Annotation>
```

The `Variability` of the `MetaAttribute` is expressed either as *stable* or *actionable* whereas the unchangeable properties in the mining model are stable. E.g. the date of birth cannot be changed, thus this metaattribute is referred to as *stable*. If we for example expect that the systolic blood pressure can be influenced by some other property, we refer to the `Variability` as actionable [17], otherwise it can also be a stable `MetaAttribute`; this depends on the mining model and its research targets. An atomic `MetaAttribute` element contains at least one `Format`.

Format The `Format` is identified by a unique name (within the collection) and encompasses the following elements: `Author`, `Annotations` (which is a collection of particular annotations), `DateType`, `ValueType`, `ValueAnnotations`, `AllowedRange`, `Collation`, `PreprocessingHints` and `ValueDescriptions`.

Each `Annotation` consists of the name of an author and the commentary - each format could be commented through the `Annotations` (collection of `Annotation` elements). The `Author` of the `Format` is self-explanatory, as a value of the `DataType` is used some of the common data type readable by the intended consumer BK (string, integer, boolean etc.). The `ValueType` content distinguishes between cardinal, nominal, ordinal and a real number. Commonly used are values as nominal and ordinal for qualitative meta-attribute and cardinal (which means an interval or a rational number) for quantitative metaattributes [13].

The `ValueAnnotations` element is defined for the commentary to particular values: each value can be commented separately more than once. The particular annotation has the same format as the `Annotation`.

The `AllowedRange` element denotes a value boundary of the particular format of the `MetaAttribute`. Thus the formats of the same values can differ. The range can be defined by `Interval` for quantitative values (maximum and minimum) or by `Enumeration` for qualitative values. See an example of allowed range defined by an interval:

```
<Interval>
<LeftBound type="closed" value="2"/>
<RightBound type="closed" value="15"/>
</Interval>
```

The `Collation` expresses a commonly accepted arrangement of the *greater than* relation between format values, if such an arrangement exists. This is essential for interpretation of the be *greater than* relationship between values [14]. The BKEF XML Schema differentiates between easily sortable numerical values and qualitative values whose sequence is expressed by the enumeration as depicted on the following example:

```
<Collation type="Numerical" sense="Ascending" />
```

respectively

```
<Collation type="Enumeration" sense="Ascending">
  <Value>elementary</Value>
  <Value>secondary</Value>
<Value>university</Value>
</Collation>
```

The `PreprocessingHints` element conveys to a BK Consumer the information on how to prepare data. The current version of the BKEF XML Schema allows one or more `DiscretizationHint` elements as the only possible child elements of the `Preprocessing Hint`. The values of the `DiscretizationHint` are assorted into discreet counterparts. There can be more than one preprocessing hint, for example depending on the desired granularity of the metaattribute values. The way of discretization is set up by `ExhaustiveEnumeration` or `IntervalEnumeration`. It reflects all intended values of the metaattribute designated for the BK consumer and consecutive mining tasks. The element `IntervalEnumeration` is used for numerical values, as seen from an example:

```
 <IntervalEnumeration>
  <IntervalBin name="normal">
   <Annotation>...</Annotation>
<Interval>
<LeftBound type="closed" value="60"/>
<RightBound type="closed" value="88"/>
</Interval>
</IntervalBin>
  <IntervalBin name="overweight indicator">
   <Annotation>...</Annotation>
<Interval>
<LeftBound type="closed" value="88"/>
<RightBound type="closed" value="140"/>
</Interval>
```

```
  </IntervalBin>
</IntervalEnumeration>
```

An example of `ExhaustiveEnumeration` for non-numerical values is:

```
<ExhaustiveEnumeration>
<Bin name="yes">
   <Annotation>...</Annotation>
   <Value>yes</Value>
  </Bin>
  <Bin name="no">
   <Annotation>...</Annotation>
   <Value>no</Value>
  </Bin>
</ExhaustiveEnumeration>
```

The exhaustive enumeration corresponds with the Map Values (where the values are defined as a table) of PMML 3.2 [4].
There are another two variations of interval enumeration: `Equifrequent` (the number of intervals is given and the interval boundaries are determined automatically so that the frequency of values falling into each interval is roughly identical) and `Equidistant` (given exact lenght of an interval). The *Discretization Hint* element does not include the value sets aggregation (known from PMML[4]), otherwise the clear and expressive discretization hint structure is one of the strengths of the BKEF XML Schema.

The `Value Descriptions` element is used for characteristics of particular values. It uses the `Interval` or `Value` elements for numerical and non-numerical values, respectively.

```
<ValueDescriptions>
<ValueDescription type="Significant">
   <Annotation>...</Annotation>
    <Interval>
      <LeftBound type="closed" value="100"/>
      <RightBound type="closed" value="150"/>
    </Interval>
</ValueDescription>
</ValueDescriptions>
```

In general, setting of the `Collation`, `PreprocessingHints` and `ValueDescriptions` is not a question of an exact method, as their determination is fully dependent on the domain expert and a particular mining task.

5.2 Patterns in BKEF

The current BKEF XML Schema allows to define `MutualInfluences`, which are a base for the BAR.

A `MutualInfluences` contains at least one `MutualInfluence`, which forms a relation between two metaattributes $A \rightarrow B$.

```
<Influence type="Positive-bool-growth" id="20" arity="2">
<KnowledgeValidity>Unknown</KnowledgeValidity>
<MetaAttribute role="A" name="weight">
<RestrictedTo><Format name="kg"/></RestrictedTo>
</MetaAttribute>
<MetaAttribute role="B" name="Hyperlipoproteinemy">
<RestrictedTo>
<Format name="boolean value">
<Value format="boolean value">yes</Value>
</Format>
</RestrictedTo>
</MetaAttribute>
</Influence>
```

`KnowledgeValidity` can have two values – *Unknown*, *Proven* or *Rejected* – regarding the mining task result. The metaattribute appearing in the influence might be restricted to the `Format` or even particular value (which should be linked with the corresponding `Format` of the atomic `MetaAttribute`).

6 Background Knowledge Ontology

The *Background Knowledge Ontology* is a semantic abstraction of the BKEF XML Schema introduced in section 5. The purpose of the BKEF XML Schema is to rigidly enumerate what types of background knowledge are acceptable and in what format. To this, BKOn adds information on relations between the pieces of background knowledge by explicitly linking them through typed associations, thus adding machine-readable semantics for background knowledge consumers. The most prominent consumer is the Semantic KB, which utilizes these relations for reasoning.

Adding semantics to the BKOn results in reshuffling of the BKEF content. The design guidelines that were followed when translating BKEF nodes to BKOn ontology topics are the same that were followed when creating the Association Rule Mining Ontology from PMML as described in [10]. Reenumerating the guidelines is out of the scope of this paper, nevertheless the main principle is simple – allow for automatic transformation of BKEF XML documents into instances of the ontology concepts while making the resulting ontology as clean as possible.

To achieve this, the following prominent changes in BKOn compared to BKEF were made

- some concepts that were only implicitly present in the BKEF XML Schema are explicitly present in BKOn,

– some BKEF XML nodes do not have a corresponding concept in the ontology as they are contained in the newly created concepts,
– explicit superclasses for closely related topics are introduced.

Some of the concrete examples of these changes are as follows: *Metafield* becomes an explicit ontology concept and a concept directly corresponding to the `Format` BKEF element is no longer explicitly present in the ontology. One instance of the `Metafield` concept is created from each pair of `Format` element and its containing `Metaattribute` element.

The *Metafield Binned Content* is used as a superclass for `EnumerationBin` and `IntervalBin`, and *Metafield Raw Content* as a superclass for `Interval` and `Value`. Both these newly introduced concepts have the *Metafield Content* superclass.

We make a reference transformation implemented as an XSLT stylesheet available[5]. The gist of BKOn is depicted on Figure 2.

7 Exploiting BKEF and BKOn in the Data Mining Loop

This section demonstrates a possible use case of BKEF and BKOn, in conjunction with the academic data mining system LISp-Miner and the SEWEBAR framework. LISp-Miner is an academic system for KDD developed at University of Economics, Prague [1] for teaching and research in the area of KDD. It consists of several procedures covering the entire process of KDD as described in the CRISP-DM methodology.[6]. The SEWEBAR (for: Semantic Web – Analytical Reports) framework involves a content management system and a semantic knowledge base for creating and sharing knowledge relating to data mining tasks. It is based on the Joomla! CMS and the Ontopia Topic-Map-based Knowledge Base.[7]

This section goes through elicitation of background knowledge within SEWEBAR-CMS, its linking with the mined data using the FML, using it to localize search and prune results within the LISp-Miner system, and finally through its semantic postprocessing, again in SEWEBAR-SKB. The description of the workflow is illustrated in a data mining task whose purpose is to find novel knowledge in a cardiological dataset.

7.1 Background Knowledge Elicitation

The first implementation of background knowledge elicitation was integrated into the LM KnowledgeSource and LM DataSource modules [19] of the LISp-Miner system. However, it emerged later that it is more suitable for domain experts to use a web-based system. This prompted the development of the BKEF Editor (see [5]), as one of the modules of SEWEBAR-CMS.

[5] At `http://sewebar.vse.cz`
[6] `www.crisp-dm.org`
[7] See `ontopia.net` and `joomla.org` for more info

Example Starting the aforementioned data mining use case, consider a medical expert, a cardiologist, who initiates the data mining process. The cardiologist uses the BKEF editor to convey her knowledge of the characteristics that are recorded about cardiological patients and indicates known and interesting relationships appearing in these characteristics.

7.2 Linking Background Knowledge with Mined Data

The main challenge faced is how to properly match data fields that are used in the current data mining task with the semantically equivalent metaattributes. This problem can be divided into two steps: choosing the right BKEF file for the domain being mined and matching metaattributes and their values with data fields and data field values. While this problem is a unique one, it bears significant resemblance with problems that are addressed in ontology alignment and schema mapping research [6]. Since fully automated construction of a reliable mapping seems to be unfeasible given the state of the art in ontology matching and schema mapping, a semi-automated mapping approach is proposed. There is an ongoing work on a web-based system that would propose such a mapping based on a mixture of schema mapping and ontology alignment techniques, which would then have the user confirm the proposed mappings. The result of this mapping is a Field Mapping Language (FML) document. The data mining system will use a web service to locate and retrieve correct FML and BKEF files.

Example The data analyst working with the cardiological dataset searches for BKEF files related to the dataset. Two such files are found. The first one is a BKEF file created by the cardiologist; the second is from a different domain, but it contains general medical fields such as Age or Blood pressure. Once the metaattributes are mapped to datafields though the semiautomatic process highlighted above, the data mining software can use the Preprocessing hints associated with mapped metaattributes to automatically perform discretization and outlier treatment.

7.3 Background Knowledge for Localizing Search

In LISp-Miner, the first implemented use of background knowledge was to guide users in the process of defining Local Analytical Questions (LAQs). That is to properly define what kind of patterns in the analyzed data we are looking for. LAQs are based on pre-defined patterns that lead to different types of questions asked and therefore to different data mining procedures used for answering them. LAQs were first proposed in [18].

Based on actual background knowledge the first type of LAQ pattern could be to mine for yet unknown influences between two groups of attributes (e.g. social status attributes and health status attributes). Or, another LAQ pattern could be used to pinpoint some condition under which some relationship stored into ontology does not hold (e.g. Concerning men above 50 living in Prague it IS

NOT TRUE that..."). Solving such a LAQ could lead to updates of background knowledge.

Example The data analyst is looking for guides to help him/her design the parameters of the data mining task. Based on the information contained in the BKEF pattern section, the data mining system shows that it is already known by the experts that high waist-hip ratio is associated with hypertension. Based on this piece of information, the data analyst instructs the system to look for exceptions to this rule – i.e. to find subsets of data (circumstances) where the high waist-hip ratio is NOT associated with hypertension.

7.4 Background Knowledge for Result Pruning

Another prospective use of background knowledge is pruning of the results of data mining that are of no value for experts (e.g. of patients giving birth to child, at least 99 % are women). If such a relationship is stored in BKEF, no implicational[8] association rule with the attribute concerning ability to give birth to a child on the left side (antecedent) and gender on the right side (succedent) will be placed into results.

Even more useful is pruning in case of a function-like dependency between two attributes, e.g. Age and Height. In general, there is a clear dependency between the age of people and their height. When described by association rules many specific rules will emerge in results, which is undesirable. Instead, a better-suited procedure of the KL-Miner (see e.g. [16] could be (automatically) used and many association rules related to this dependency could be pruned from the results and represented by a single KxL-fold contingency table to describe this function like dependency as a single pattern.

Example The cardiologist is not interest in obvious facts in the results. So all patterns expressing already known relationship between the high waist-hip ratio and hypertension are automatically pruned from the results (if not explicitly overruled by the data analyst). This covers all the derived patterns, i.e. even pruning of extended patterns that logically follow from the simple implication of the form waist-hip ratio(high) =¿ hypertension(true).

7.5 Background Knowledge for Postprocessing

SEWEBAR-CMS [11] accepts mining models in PMML sent through a web service by the data mining system. The BKEF XML files are already present in the system as they originate there. Combining these pieces of information, the analyst conveys the results to the domain expert through a textual analytical report using special report-authoring tools within the CMS [20]. PMML and BKEF documents are semantized according to the Data Mining Ontology [10]

[8] A subclass of association rules [12].

and the BKOn ontology. They are interlinked and stored in the SEWEBAR-SKB, which answers queries issued from the CMS. The queries are issued in the tolog query language, which is a combination of Prolog and SQL. The results of the queries are returned by the Semantic KB in XML, using an XSLT transformation converted to HTML and returned to the user.

Example To communicate the results to medical specialists, the data analyst creates a textual analytical report summarizing his/her findings. In the report s/he also includes the semantic query against the Semantic KB for related association rules that were found in previous tasks, including those executed over different datasets.

8 Conclusions

The main purpose of this paper was to discuss the requirements on a standard for exchange of background knowledge in data mining. The paper also details an attempt for such a specification consisting of the BKEF Schema and BKOn ontology. Practical experience with these formats has already been described in [11], including the interlinking of BKOn with a data mining ontology for association rules introduced in [10] and examples of semantic queries over the merged ontologies.

Future work will primarily address the issue of 'smart' interlinking to domain ontologies, presumably using ontology patterns[9]. This will allow to explicitly disambiguate vague notions, e.g. that of hypertension, which can equally be a summarization of several measurements or a permanent characteristic of a patient. In relation to that, a version of BKOn based on the RDF/OWL formalism (in addition to the Topic Map one) will be built.

9 Acknowledgment

This work has been partly supported from grant no IGA 15/2010 of UEP and by grant GAR 201/08/0802 of Czech Grant Agency.

References

1. LISp-Miner: academic system for KDD [online]. [cit. 2010-03-20], available from WWW: ¡http://lispminer.vse.cz¿
2. OWL Web Ontology Language Overview. W3C Recommendation, 10 February 2004. `http://www.w3.org/TR/owl-features/`
3. W3C: XSL Transformation. Online: `www.w3.org/TR/xslt`. 1999
4. DMG: PMML 3.2 Specification, Online: `http://www.dmg.org/pmml-v3-2.html`
5. Balhar, J., Kliegr, T., Stastny D., Vojir S.: Elicitation of Background Knowledge for Data Mining. In: Znalosti 2010, Czech Republic, February 2010.

[9] `http://www.ontologydesignpatterns.org`

6. Euzenat J. and Shvaiko P.: *Ontology matching*. Springer-Verlag. 2007. ISBN 3-540-49611-4.
7. Garshol L. M., Moore G.: Topic Maps i?1 XML Syntax. ISO/IEC JTC1/SC34, `http://www.isotopicmaps.org/sam/sam-xtm/`.
8. Garshol, L.M.: TMRAP -i?1 Topic Maps Remote Access Protocol. In: Maicher, L., Sigel, A., Garshol, L.M. (eds.) TMRA 2006. LNCS (LNAI), vol. 4438. Springer, Heidelberg (2007)
9. Garshol, L.M.: Towards a Methodology for Developing Topic Maps Ontologies. In: Maicher, L., Sigel, A., Garshol, L.M. (eds.) TMRA 2006. LNCS (LNAI), vol. 4438. Springer, Heidelberg (2007)
10. Kliegr, T., Ovecka M., Zemanek, J.: Topic Maps for Association Rule Mining. In: Proc. TMRA 2009. University of Leipzig 2009.
11. Kliegr M., Ralbovský M., Svátek, V, Šimůnek M., Jirkovský V., Nemrava J., Zemánek J.: Semantic Analytical Reports: A Framework for Post-Processing Data Mining Results. In: Foundations of Intelligent Systems (ISMIS'09). Springer Verlag, LNCS, 2009, 88i?198.
12. Rauch, J.: Classes of Association Rules: An Overview. In: Studies In Computational Intelligence. Springer 2008.
13. Rauch J.: Considerations on Logical Calculi for Dealing with Knowledge in Data Mining. In: Advances in Data Management. Studies in Computational Intelligence, Volume 223/2009, Springer 2009.
14. Rauch J., Šimůnek M.: Dealing with Background Knowledge in the SEWEBAR Project. In: Knowledge Discovery Enhanced with Semantic and Social Information. Studies in Computational Intelligence, Volume 220/2009, Springer 2009.
15. Rauch J., Šimůnek M.: Alternative Approach to Mining Association Rules. In Lin T Y, Ohsuga S, Liau C J, and Tsumoto S (eds): Data Mining: Foundations, Methods, and Applications, Springer-Verlag, 2005.
16. Rauch, J., Šimůnek, M., Lín, V.: Mining for Patterns Based on Contingency Tables by KL-Miner First Experience. In: Foundations and Novel Approaches in Data Mining. Berlin : Springer-Verlag, 2005, s. 155167. ISBN 3-540-28315-3. ISSN 1860-949X.
17. Rauch, J., Šimůnek, M.: Action Rules and the GUHA Method: Preliminary Considerations and Results. ISMIS 2009: 76-87
18. Rauch, J., Šimůnek, M.: LAREDAM Considerations on System of Local Analytical Reports from Data Mining. Toronto 20.05.2008 – 23.05.2008. In: Foundations of Intelligent Systems. Berlin : Springer-Verlag, 2008, pp. 143–149.
19. Šimůnek, M.: Academic KDD Project LISp-Miner. In: Advances in Soft Computing - Intelligent Systems Desing and Applications. Heidelberg : Springer-Verlag, 2003, s. 263272. ISBN 3-540-40426-0.
20. Vojir S.: SEWEBAR - gInclude - Analytical Report Design using gInclude. In: Znalosti 2010, Czech Republic, in Czech, February 2010.

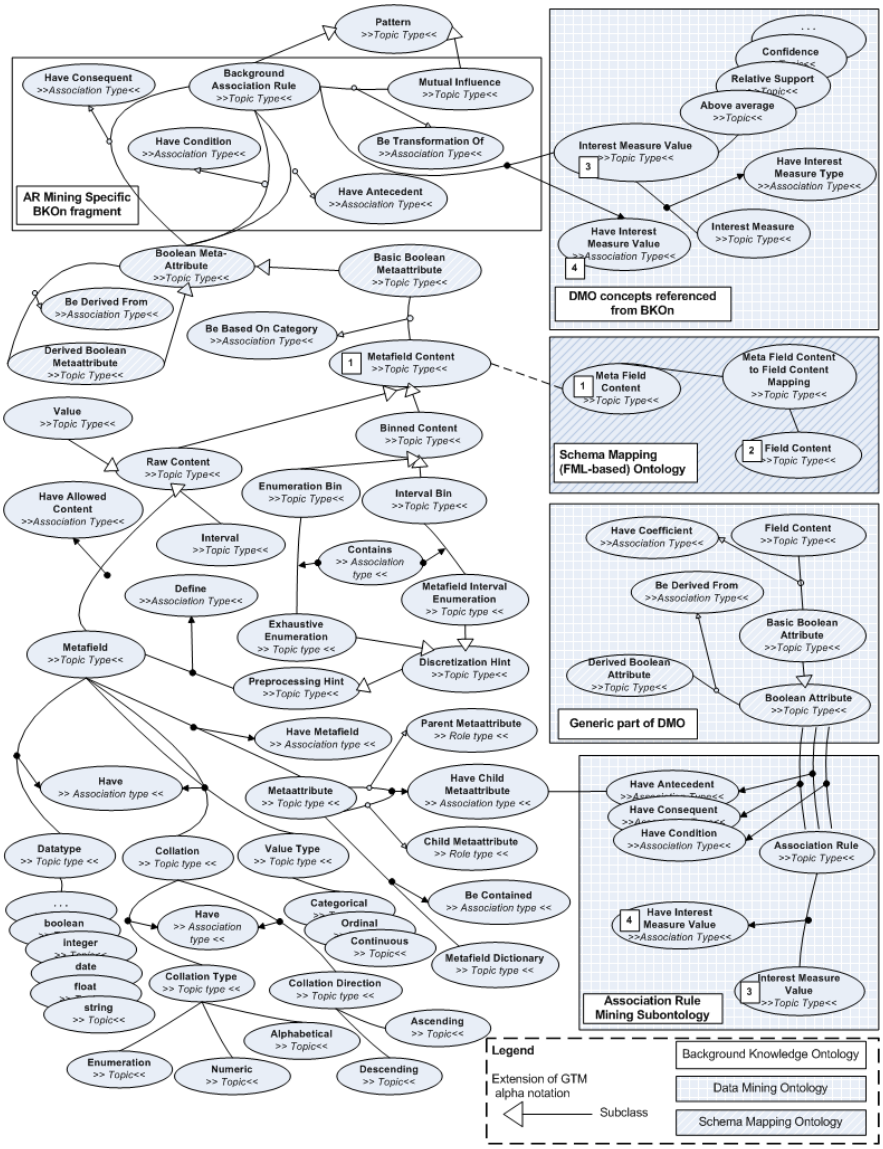


Fig. 2. Background Knowledge Ontology Overview

Late Breaking News

Importing Knowledge Fragments to CMS-Enabled Data Mining Analytical Reports

Andrej Hazucha, Tomáš Kliegr, and Vojtěch Svátek

University of Economics, Prague,
{xhaza00,tomas.kliegr,svatek}@vse.cz

Descriptive data mining only brings its fruits when the results are provided to the end user in a palatable form. The vehicle for end-user delivery of mining results (and associated information such as data schema, task settings, and domain background knowledge) are so-called *analytical reports*. In order to manage a huge number of reports referring to different mining sessions, we designed a *data mining web portal* based on a *content management system*, together called SEWEBAR-CMS.[1] One of the requirements on the CMS was the ability to interact with *semantic knowledge sources* and other structured data, see [1].

The data analyst who authors an analytical report in the CMS has different possibilities of (semi-)automatically entering structured data into the text.

First, for *locally stored* data such as mining task/result/data descriptions exported from mining tools in PMML (Predictive Model Mark-Up Language), a CMS plugin can pick marked segments of HTML code, produced from PMML using XSLT, and insert them into the report as indicated by the analyst.

Second, sophisticated support for *remote* data/knowledge has been newly added. The infrastructure for this functionality allows to persistently specify

- Links to queriable *resources*
- Template *queries* for these resources (which can be paramatrized by the end-user at runtime)
- *XSLT transformations* allowing to insert the results of queries as HTML fragments, either static or *dynamically updated* from the resources.

Currently we experiment with queriable resources in the form of *native XML database* (Berkeley, queried via XQuery), which stores PMML data, and semantic knowledge bases both in the form of *SPARQL endpoint* and *Ontopia Knowledge Suite* (a Topic Maps tool, queried via a Prolog-like language called tolog). Inclusion of further types of resources such as Lucene indices is in progress.

This work has been partially supported by the CSF project no.201/08/0802, and by Grant F4/15/2010 of the University of Economics, Prague.

References

1. Kliegr M., Ralbovský M., Svátek, V, Šimůnek M., Jirkovský V., Nemrava J., Zemánek J.: Semantic Analytical Reports: A Framework for Post-Processing Data Mining Results. In: Proc. ISMIS'09, Springer Verlag, LNCS, 2009, 8898.

[1] SEWEBAR stands for SEmantic WEB and Analytical Reports. More details in http://sewebar.vse.cz.

Towards a Semantic Foundation for Bioinformatics

Ross D. King

Department of Computer Science, Aberystwyth University, UK, rdk@aber.ac.uk

1 Abstract

With a two and half thousand year tradition logic is the best understood way of representing scientific knowledge. Only logic provides the semantic clarity necessary to ensure the comprehensibility, reproducibility, and free exchange of knowledge. The use of logic is also necessary to enable computers to play a full part in science [1]. The semantic web is transforming the dissemination of science by making for the first time making a large amount of scientific knowledge available expressed in logic.

Bioinformatics is one of the undoubted successes stories of the semantic web, with bioinformatic knowledge making up a large percentage of the scientific semantic web. Many of the problems that make semantic web reasoning difficult don't apply to bioinformatics: a ground truth of scientific knowledge exists, top level ontologies have been agreed (BFO), many other ontological standards exist, and the bioinformatic semantic web is large but not too large.

The use of bioinformatic software is essential to modern biology. However, there is a clear mismatch between the increasing use of the semantic web and logic, and the way bioinformatic systems utlilise and make inferences with this knowledge. This is because almost all computer based bioinformatic reasoning is done using *ad hoc* programs. From a formal point of view these programs are invariably making logical inferences: deductions, abductions, inductions, with perhaps a probabilistic element. However, what exactly these inferences exactly are is generally unclear.

The aim of my research is to make these inferences clear and to express them in logic, and make them executable across the semantic web.

For example, we argue that abductive inference is central to modern evolutionary based phylogenetics - clustering. This can be seen in evolutionary definition of a taxon (grouping of organisms): “that all members of a taxon are descendants of the nearest common ancestor (monophyly sensu stricto)” [2]. We express this in logic as:

$$\forall A \,.\, A \in taxon1 \Rightarrow (\exists Ancestor \,.\, \forall B \,.\, B \notin taxon1 \wedge ancestor(Ancestor, A) \wedge \neg ancestor(Ancestor, B)).$$

This clustering is based on the abductive inference of the existence of an ancestor organism not shared by any other taxon.

References

1. King, R.D., Rowland, J., Oliver, S.G., Young, M., Aubrey, W., Byrne, E., Liakata, M., Markham, M., Pir, P., Soldatova, L.N., Sparkes, A., Whelan, K.E., Clare, C. (2009) The Automation of Science. *Science*. **324**, 85-89.
2. Mayr, E. (1982). The Growth of Biological Thought: Diversity, Evolution, and Inheritance. Cambridge, Mass: Belknap Press.

(this page is intentionally left blank)