Modeling Uncertainty in Semantic Web Taxonomies

Markus Holi and Eero Hyvönen

Helsinki University of Technology (TKK), University of Helsinki, and
Helsinki Institute for Information Technology (HIIT)
P.O. Box 5500, FI-02015 TKK, Finland
http://www.seco.tkk.fi/
email: firstname.lastname@tkk.fi

Summary. Information retrieval systems have to deal with uncertain knowledge and query results should reflect this uncertainty in some manner. However, Semantic Web ontologies are based on crisp logic and do not provide well-defined means for expressing uncertainty. We present a new probabilistic method to approach the problem. In our method, degrees of subsumption, i.e., overlap between concepts can be modeled and computed efficiently using Bayesian networks based on RDF(S) ontologies. Degrees of overlap indicate how well an individual data item matches the query concept, which can be used as a well-defined measure of relevance in information retrieval tasks.

1 ONTOLOGIES AND INFORMATION RETRIEVAL

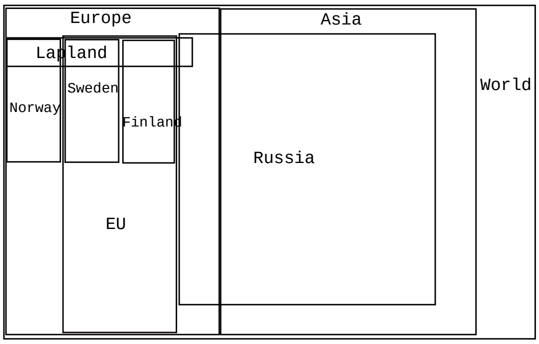
A key reason for using ontologies in information retrieval systems, is that they enable the representation of background knowledge about a domain in a machine understandable format. Humans use background knowledge heavily in information retrieval tasks [7]. For example, if a person is searching for documents about Europe she will use her background knowledge about European countries in the task. She will find a document about Germany relevant even if the word 'Europe' is not mentioned in it. With the help of an appropriate geographical ontology also an information retrieval system could easily make the above inference. Ontologies have in fact been used in a number of information retrieval system in recent years [14, 8, 9].

Ontologies are based on crisp logic. In the real world, however, relations between entities often include subtleties that are difficult to express in crisp ontologies. For example, most of the birds in Antarctica are penguins. Thus, if a document is annotated with the concepts 'Antarctica' and 'Bird', a human will make the inference that the document is related to Penguins. RDF(S) [2] and OWL [1] ontologies do not provide good means to make this kind of inferences.

Also the information system itself is a source of uncertainty. The annotation, i.e. the indexing, of the documents is often inexact or uncertain. For example, if we have a geographical ontology about the countries and areas of North-America, and we want to index photographs using it, then it is most likely that in some point we will encounter a photograph the origin of which we do not know exactly. Typically the photograph will be annotated with the concept North-America. This kind of uncertainty may also be resulted from ontology merging or evolution. When a human information searcher will encounter the annotation, she will infere that there is a high probability that the photograph is taken in the U.S.A, because it is one of the largest countries in North-America.

It would be very useful if also an information retrieval system could make the same kind of inferences and use them when constructing the result sets for queries. Notice that in the above examples the knowledge of degrees of overlap and coverage between concepts is essential for succeeding in the information retrieval task.

2 MODELING UNCERTAINTY IN ONTOLOGIES



```
World 37*23 = 851
Europe 15*23 = 345
Asia 18*23 = 414
EU  8*21 = 168
Sweden 4*9 = 36
Finland 4*9 = 36
Norway 4*9 = 36
Lapland 13*2 = 26  Lapland&(Finland | Sweden | Norway) = 8
Lapland&EU = 16  Lapland&Russia = 2
Russia 18*19 = 342  Russia&Europe = 57  Russia&Asia = 285
```

Fig. 1. A Venn diagram illustrating countries, areas, their overlap, and size in the world.

The Venn diagram of Fig. 1 illustrates some countries and areas in the world. A crisp *partOf* meronymy cannot represent the partial overlap between the geographical area Lapland and the countries Finland, Sweden, Norway, and Russia, for example. A frequently used way to model the above situation would be to represent Lapland as the direct meronym of all the countries it overlaps, as in Fig. 2. This structure, however does not represent the situation of the map correctly, because Lapland is not subsumed by anyone of these countries. In addition, the transitivity of the subsumption relation disappears in this structure. See, for example, the relationship between Lapland and Asia. In the Venn diagram they are disjoint, but according to the taxonomy, Lapland is subsumed by Asia.

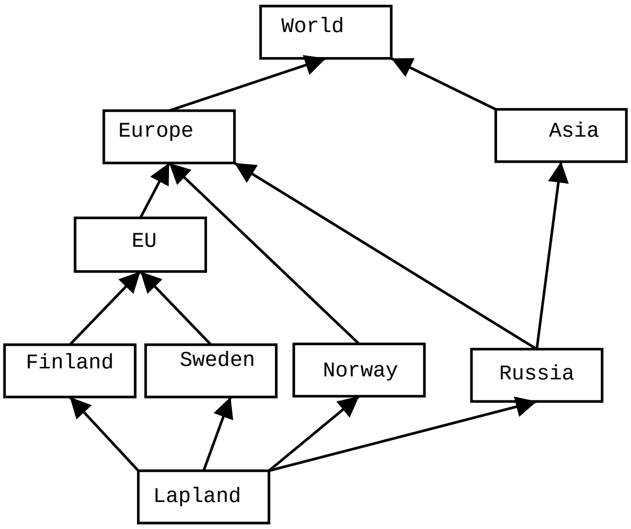


Fig. 2. A standard semantic web taxonomy based on the Venn diagram of Fig. 1.

Another way would be to partition Lapland according to the countries it overlaps, as in Fig. 3. Every part is a direct meronym of both the respective country and Lapland. This structure is correct, in principle, but it too does not contain enough information to make inferences about the degrees of overlap between the areas. It does not say anything about the sizes of the different parts of Lapland, and how much they cover of the whole area of Lapland and the respective countries.

According to Fig. 1, the size of Lapland is 26 units, and the size of Finland is 36 units. The size of the overlapping area between Finland and Lapland is 8 units. Thus, 8/26 of Lapland belongs to Finland, and 8/36 of Finland belongs to Lapland. On the other hand, Lapland and Asia do not have any overlapping area, thus no part (0) of Lapland is part of Asia, and no part of Asia is part of Lapland. If we want a taxonomy to be an accurate representation of the 'map' of Fig. 1, there should be a way to make this kind of inferences based on the taxonomy.

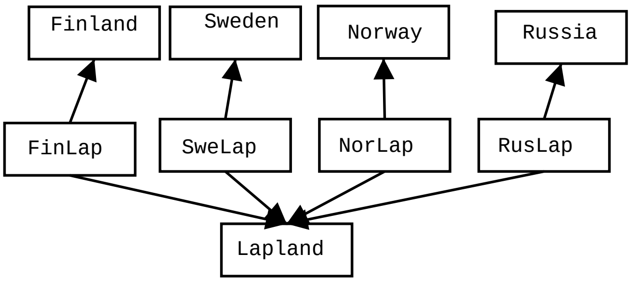


Fig. 3. Representing Lapland's overlaps by partitioning it according to the areas it overlaps. Each part is subsumed by both Lapland and the respective country.

Table 1. The *overlap table* of Lapland according to Fig. 1.

Selected	**Referred**	**Overlap**
Lapland	World	26/851 = 0.0306
	Europe	26/345 = 0.0754
	Asia	0/414 = 0.0
	EU	16/168 = 0.0953
	Norway	8/36 = 0.2222
	Sweden	8/36 = 0.2222
	Finland	8/36 = 0.2222
	Russia	2/342 = 0.0059

Our method enables the representation of overlap in taxonomies, and the computation of overlap between a *selected* concept and every other, i.e. *referred* concept in the taxonomy. Thus, an *overlap table* is created for the selected concept. The overlap table can be created for every concept of a taxonomy. For example, Table 1 present the overlap table of Lapland based on the the Venn diagram of Fig. 1. The Overlap column lists values expressing the mutual overlap of the selected concept and the other - referred - concepts, i.e., $Overlap = \frac{|Selected \cap Referred|}{|Referred|} \in [0, 1]$.

Intuitively, the overlap value has the following meaning: The value is 0 for disjoint concepts (e.g., Lapland and Asia) and 1, if the referred concept is subsumed by the selected one. High values lesser than one imply, that the meaning of the selected concept approaches the meaning of the referred one.

This overlap value can be used in information retrieval tasks. Assume that an ontology contains individual products manufactured in the different countries and areas of Fig. 1. The user is interested in finding objects manufactured in Lapland. The overlap values of Table 1 then tell how well the annotations "Finland", "EU", "Asia", etc., match with the query concept "Lapland" in a well-defined probabilistic sense, and the hit list can be sorted into an order of relevance accordingly.

The overlap value between the selected concept (e.g. Lapland) and the referred concept (e.g. Finland) can in fact be written as the conditional probability $P(Finland'|Lapland')$ whose interpretation is the following: If a person is interested in data records about Lapland, what is the probability that the annotation "Finland" matches her query? X' is a binary random variable such that $X' = true$ means that the annotation "X" matches the query, and $X' = false$ means that "X" is not a match. This conditional probability interpretation of overlap values will be used in section 4 of this paper. Notice that the modeling of overlap between geographical concepts, as in our example, is truly uncertain, because the exact amount of overlap is never known.

It is mathematically easy to compute the overlap tables, if a Venn diagram (the sets) is known. In practice, the Venn diagram may be difficult to create from the modeling view point, and computing with explicit sets is computationally complicated and inefficient. For these reasons our method calculates the overlap values from a taxonomic representation of the Venn diagram.

Our method consists of two parts:

1. A graphical notation by which partial subsumption and concepts can be represented in a quantified form. The notation can be represented easily in RDF(S).
2. A method for computing degrees of overlap between the concepts of a taxonomy. Overlap is quantified by transforming the taxonomy first into a Bayesian network [4].

3 REPRESENTING OVERLAP

In RDFS and OWL a concept, i.e. class refers to a set of individuals. Subsumption reduces essentially into the subset relationship between the sets corresponding to classes [1]. A taxonomy is therefore a set of sets and can be represented, e.g., by a Venn diagram.

If A and B are sets, then A must be in one of the following relationships to B.

1. A is a subset of B, i.e. $A \subseteq B$.
2. A partially overlaps B, i.e. $\exists x, y : (x \in A \wedge x \in B) \wedge (y \in A \wedge y \notin B)$.
3. A is disjoint from B, i.e. $A \cap B = \emptyset$.

Based on these relations, we have developed a simple graph notation for representing uncertainty and overlap in a taxonomy as an acyclic *overlap graph*. Here concepts are nodes, and a number called *mass* is attached to each node. The mass of concept A is a measure of the size of the set corresponding to A, i.e. $m(A) = |s(A)|$, where $s(A)$ is the set corresponding to A. A solid directed arc from concept A to B denotes crisp subsumption $s(A) \subseteq s(B)$, a dashed arrow denotes disjointness $s(A) \cap s(B) = \emptyset$, and a dotted arrow represents quantified partial subsumption between concepts, which means that the concepts partially overlap in the Venn diagram. The amount of overlap is represented by the *partial overlap value* $p = \frac{|s(A) \cap s(B)|}{|s(A)|}$.

In addition to the quantities attached to the dotted arrows, also the other arrow types have implicit overlap values. The overlap value of a solid arc is 1 (crisp subsumption) and the value of a dashed arc is 0 (disjointness). The quantities of the arcs emerging from a concept must sum up to 1. This means that either only one solid arc can emerge from a node or several dotted arcs (partial overlap). In both cases, additional dashed arcs can be used (disjointness). Intuitively, the outgoing arcs constitute a quantified partition of the concept. Thus, the dotted arrows emerging from a concept must always point to concepts that are mutually disjoint with each other.

Notice that if two concepts overlap, there must be a directed (solid or dotted) path between them. If the path includes dotted arrows, then (possible) disjointness between the concepts must be expressed explicitly using the disjointness relation. If the directed path is solid, then the concepts necessarily overlap.

For example, Fig. 4 depicts the meronymy of Fig. 1 as an overlap graph. The geographic sizes of the areas are used as masses and the partial overlap values are determined based on the Venn diagram. This graph notation is complete in the sense that any Venn diagram can be represented by it. However, sometimes the accurate representation of a Venn diagram requires the use of auxiliary concepts, which represent results of set operations over named sets, for example $s(A) \setminus s(B)$, where A and B are ordinary concepts.

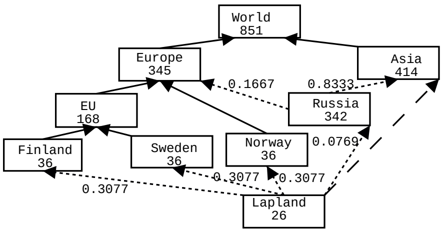


Fig. 4. The taxonomy corresponding to the Venn diagram of Fig. 1.

4 SOLID PATH STRUCTURE

Our method creates an overlap table (cf. Fig. 1) for each concept in the taxonomy. Computing the overlaps is easiest when there are only solid arcs, i.e., complete subsumption relation, between concepts. If there is a directed solid path from A (selected) to B (referred), then overlap $o = \frac{|s(A) \cap s(B)|}{|s(B)|} = \frac{m(A)}{m(B)}$. If the solid path is directed from B to A, then $o = \frac{|s(A) \cap s(B)|}{|s(B)|} = \frac{m(B)}{m(B)} = 1$. If there is not a directed path between A and B, then $o = \frac{|s(A) \cap s(B)|}{|s(B)|} = \frac{|\emptyset|}{m(B)} = 0$.

If there is a mixed path of solid and dotted arcs between A and B, then the calculation is not as simple. Consider, for example, the relation between *Lapland* and *EU* in Fig. 4. To compute the overlap, we have to follow all the paths emerging from *Lapland*, take into account the disjoint relation between *Lapland* and *Asia*, and sum up the partial subsumption values somehow.

To exploit the simple solid arc case, a taxonomy with partial overlaps is first transformed into a *solid path structure*, in which crisp subsumption is the only relation between the concepts. The transformation is done by using to the following principle:

Transformation Principle 1 *Let A be the direct partial subconcept of B with overlap value o. In the solid path structure the partial subsumption is replaced by an additional* middle concept*, that represents $s(A) \cap s(B)$. It is marked to be the complete subconcept of both A and B, and its mass is $o{\cdot}m(A)$.*

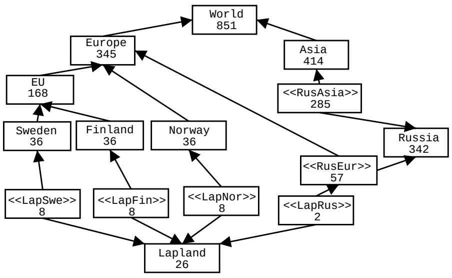


Fig. 5. The taxonomy of Fig. 4 as a solid path structure.

For example, the taxonomy of Fig. 4 is transformed into the solid path structure of Fig. 5. The original partial overlaps of Lapland and Russia are transformed into crisp subsumption by using middle concepts.

The transformation is specified in algorithm 1. The algorithm processes the overlap graph T in a breadth-first manner starting from the root concept. A concept c is processed only after all of its super concepts (partial or complete) are processed. Because the graph is acyclic, all the concept will eventually be processed.

Each processed concept c is written to the solid path structure SPS. Then each arrow emerging from c is processed in the following way. If the arrow is solid, indicating subsumption, then it is written into the solid path structure as such. If the arrow is dotted, indicating partial subsumption, then a middle concept $newMc$ is added into the solid path structure. It is marked to be the complete subconcept of both c and the concept p to which the dotted arrow points in T. The mass of $newMc$ is $m(newMc) = |s(c) \cap s(p)| = o \cdot m(c)$, where o is the overlap value attached to the dotted arrow.

However, if p is connected to its superconcepts (partial or complete) with a middle concept structure, then the processing is not as simple. In that case c

```
Data: OverlapGraph T
Result: SolidPathStructure SPS
SPS := empty;
foreach concept c in T do
    foreach complete or partial direct superconcept p of c in T do
        if p connected to its superconcepts through middle concepts in SPS then
            mc := the middle concept that c overlaps;
            if c complete subconcept of p then
                mark c to be complete subconcept of mc in SPS;
            else
                newMc := middle concept representing
                $s(c) \cap s(p)$;
                mark newMC to be complete subconcept of c and mc in SPS;
            end
        else
            if c complete subconcept of p then
                mark c as complete subconcept of p in SPS;
            else
                newMc .= middle concept representing
                $s(c) \cap s(p)$;
                mark newMc to be complete subconcept of c and p in SPS;
            end
        end
    end
end
```

Algorithm 1: Creating the solid path structure

has to be connected to one of those middle concepts. The right middle concept is found by using the information conveyed in the dashed arcs emerging from c. The right middle concept mc is the one that is not subsumed by a concept that is marked to be disjoint from c in the overlap graph. This is the middle concept that c overlaps. Notice, that if the overlap graph is an accurate representation of the underlying Venn diagram, then mc is the only middle concept that fulfils the condition.

If c is a complete subconcept of p in the overlap graph T, then c is marked to be the complete subconcept of mc in SPS. If c is a partial subconcept of p in T, then it is connected to mc with a middle concept structure.

Notice, that if c was connected directly to p, instead of mc, then the information conveyed in the dashed arrows, indicating disjointness between concepts would have been lost. For example, in Fig. 5 *Lapland* was connected directly to *Russia*, then the information about the disjointness of *Lapland* and *Asia* would have been lost.

5 COMPUTING THE OVERLAPS

Based on the solid path structure, the overlap table values o for a selected concept A and a referred concept B could be calculated by the algorithm 2, where notation X_s denotes the set of (sub)concepts subsumed by the concept X.

The overlap table for A could be implemented by going through all the concepts of the graph and calculating the overlap value according to the above algorithm. However, because the overlap values between concepts can be interpreted as conditional probabilities, we chose to use the solid path structure as a Bayesian network topology. In the Bayesian network the boolean random variable X' replaces the concept X of the solid path structure. The efficient evidence propagation algorithms developed for Bayesian networks [4] to take care of the overlap computations. Furthermore, we saw a Bayesian representation of the taxonomy valuable as such. The Bayesian network could be used for example in user modelling [13].

```
if $A$ subsumes $B$ then
    $o := 1$
else
    $C = A_s \cap B_s$
    if $C = \emptyset$ then
        $o := 0$
    else
        $o := \frac{\sum_{c \in C} m(c)}{m(B)}$
    end
end
```

Algorithm 2: Computing the overlap

Recall from section 2 that if A is the selected concept and B is the referred one, then the overlap value o can be interpreted as the conditional probability

$$P(B' = true|A' = true) = \frac{|s(A) \cap s(B)|}{|s(B)|} = o, \qquad (1)$$

where $s(A)$ and $s(B)$ are the sets corresponding to the concepts A and B. A' and B' are boolean random variables such that the value *true* means that the corresponding concept is a match to the query, i.e, the concept in question is of interest to the user. $P(B'|A')$ tells what is the probability that concept B matches the query if we know that A is a match. Notice that the Venn diagram from which $s(A)$ and $s(B)$ are taken is not interpreted as a probability space, and the elements of the sets are not interpreted as elementary outcomes of some random phenomenon. The overlap value between $s(A)$ and $s(B)$ is used merely as a means for determining the conditional probability defined above.

The joint probability distribution of the Bayesian network is defined by conditional probability tables (CPT) $P(A'|B'_1, B'_2, \ldots B'_n)$ for nodes with parents $B'_i, i = 1 \ldots n$, and by prior marginal probabilities set for nodes without parents. The CPT $P(A'|B'_1, B'_2, \ldots B'_n)$ for a node A' can be constructed by enumerating the value combinations (true/false) of the parents $B'_i, i = 1 \ldots n$, and by assigning:

$$P(A' = true|B'_1 = b_1, \ldots B'_n = b_n) = \frac{\sum\limits_{i \in \{i:b_i = true\}} m(B_i)}{m(A)} \qquad (2)$$

The value for the complementary case $P(A' = false|B'_1 = b_1, \ldots B'_n = b_n)$ is obtained simply by subtracting from 1. The above formula is based on the above definition of conditional probability, and algorithm 2. The intuition behind the formula is the following. If a user is interested in Sweden and in Finland, then she is interested both in data records about Finland and in data records about Sweden. The set corresponding to this is $s(Finland) \cup s(Sweden)$. In terms of the OG this is written as $m(Finland) + m(Sweden)$. In the Bayesian network both Finland and Sweden will be set "true". Thus, the bigger the number of European countries that the user is interested in, the bigger the probability that the annotation "Europe" matches her query, i.e., $P(Europe' = true|Sweden' = true, Finland' = true) > P(Europe' = true|Finland' = true)$.

If A' has no parents, then $P(A' = true) = \lambda$, where λ is a very small nonzero probability, because we want the posterior probabilities to result from conditional probabilities only, i.e., from the overlap information.

The whole overlap table of a concept can now be determined efficiently by using the Bayesian network with its conditional and prior probabilities. By instantiating the nodes corresponding to the selected concept and the concepts subsumed by it as evidence (their values are set "true"), the propagation algorithm returns the overlap values as posterior probabilities of nodes. The query results can then be ranked according to these posterior probabilities.

Notice that when using the Bayesian network in the above way, a small inaccuracy is attached to each value as the result of the λ prior probability that was given to the parentless variables. This error approaches zero as λ approaches zero. Despite this small inaccuracy we decided to define the Bayesian network in the above manner for the following reasons.

First, to be able to easily use the the solid path structure as the topology of the Bayesian network. The CPTs can be calculated directly based on the masses of the concepts. Second, with this definition the Bayesian evidence propagation algorithm returns the overlap values readily as posterior probabilities. We experimented with various ways to construct a Bayesian network according to probabilistic interpretations of the Venn diagram. However, none of these constructions answered to our needs as well as the construction described above.

Third, in the solid path structure d-separation indicates disjointness between concepts. We see this as a useful characteristic, because it makes the simultaneous selection of two or more disjointed concepts possible.

6 IMPLEMENTATION

The presented method has been implemented as a proof-of-concept.

6.1 Overlap Graph

Overlap graphs are represented as RDF(S) ontologies in the following way. Concepts are represented as RDFS classes[1] The concept masses are represented using a special *Mass* class. It has two properties, subject and mass that tell the concept resource in question and mass as a numeric value, respectively. The subsumption relation can be implemented with a property of the users choice. Partial subsumption is implemented by a special *PartialSubsumption* class with three properties: subject, object and overlap. The subject property points to the direct partial subclass, the object to the direct partial superclass, and overlap is the partial overlap value. The disjointness arc is implemented by the *disjointFrom* property used in OWL.

6.2 Overlap Computations

The architecture of the implementation can be seen in Fig. 6. The input of the implementation is an RDF(S) ontology, the URI of the root node of the overlap graph, and the URI of the subsumption property used in the ontology. Additionally, also an RDF data file that contains data records annotated according to the ontology may be given. The output is the overlap tables for every concept in the taxonomy extracted from the input RDF(S) ontology. Next, each submodule in the system is discussed briefly.

The *preprocessing* module transforms the taxonomy into a predefined standard form. If an RDF data file that contains data records annotated according to the ontology is given as optional input, then the preprocessing module determines the mass of each concept in the taxonomy based on these annotations. The mass is the number of data records annotated to the concept directly or indirectly. The quantification principle is illustrated in Fig. 7.

The *transformation* module implements the transformation algorithm, and defines the CPTs of the resulting Bayesian network. In addition to the Bayesian network, it creates an RDF graph with an identical topology, where nodes are classes and the arcs are represented by the *rdf:subClassOf* property. This graph will be used by the *selection* module that expands the selection to

[1] Actually, any resources including instances could be used to represent concepts.

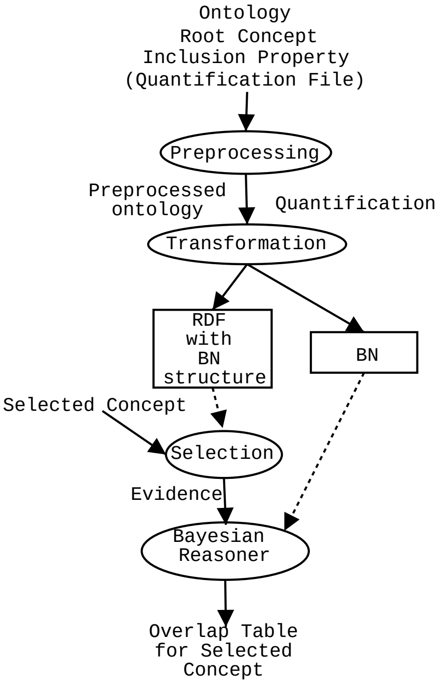


Fig. 6. The architecture of the implementation.

include the concepts subsumed by the selected one, when using the Bayesian network.

The *Bayesian reasoner* does the evidence propagation based on the selection and the Bayesian network. The selection and Bayesian reasoner modules are operated in a loop, where each concept in the taxonomy is selected one after the other, and the overlap table is created.

The *preprocessing*, *transformation*, and selection modules are implemented with SWI-Prolog[2]. The Semantic Web package is used. The *Bayesian reasoner* module is implemented in Java, and it uses the Hugin Lite 6.3[3] through its Java API.

7 DISCUSSION

7.1 Related Work

The problem of representing uncertain or vague inclusion in ontologies and taxonomies has been tackled also by using methods of fuzzy logic [21] and

[2] http://www.swi-prolog.org/

[3] http://www.hugin.com/

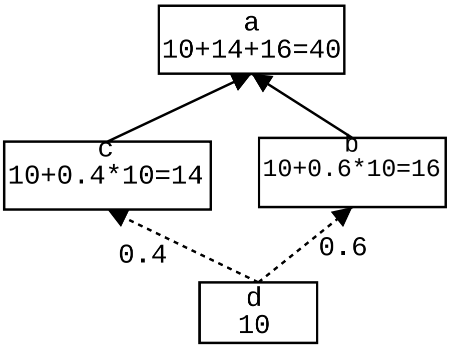


Fig. 7. Quantification of concepts. The number of direct instances of each concept is 10. In the case of partial subsumption, only a part of the mass of the subconcept is taken as the mass of the superconcept

rough sets [19, 10]. With the rough sets approach only a rough, egg-yolk representation of the concepts can be created [19]. Fuzzy logic, allows for a more realistic representation of the world.

Straccia [18] presents a fuzzy extension to the description logic SHOIN(D)corresponding to the ontology description language OWL DL. It enables the representation of fuzzy subsumption for example.

Widyantoro and Yen [20] have created a domain-specific search engine called PASS. The system includes an interactive query refinement mechanism to help to find the most appropriate query terms. The system uses a fuzzy ontology of term associations as one of the sources of its knowledge to suggest alternative query terms. The ontology is organised according to narrower-term relations. The ontology is automatically built using information obtained from the system's document collections.

The fuzzy ontology of Widyantoro and Yen is based on a set of documents, and works on that document set. However, our focus is on building taxonomies that can be used, in principle, with any data record set. The automatic creation of ontologies is an interesting issue by itself, but it is not considered in this paper. At the moment, better and richer ontologies can be built by domain specialists than by automated methods.

The fuzzy logic approach is criticised because of the arbitrariness in finding the numeric values needed and mathematical indefiniteness [19]. In addition, the representation of disjointness between concepts of a taxonomy seems to be difficult with the tools of fuzzy logic. For example, the relationships between Lapland, Russia, Europe, and Asia are very easily handled probabilistically, but in a fuzzy logic based taxonomy, this situation seems complicated. There is not a readily available fuzzy logic operation that could determine that if Lapland partly overlaps Russia, and is disjoint from Asia, then the fuzzy inclusion value between Europe and $Lapland \cap Russia$ is 1 even though Russia is only a fuzzy part of Europe.

We chose to use crisp set theory and Bayesian networks, because of the sound mathematical foundations they offer. The set theoretic approach also gives us means to overcome to a large degree the problem of arbitrariness. The calculations are simple, but still enable the representation of overlap and vague subsumption between concepts. The Bayesian network representation of a taxonomy is useful not only for the matching problem we discussed, but can also be used for other reasoning tasks [13].

Ng [17, 16] presents methods to combine probabilistic information with logic programming. This is called probabilistic logic programming. In principle we could have also created a probabilistic logic database for the taxonomy with algorithm 2. However, this would be inefficient with large ontologies, because all the possible concept combinations would have to be taken into account and encoded in the database.

Ding and Peng [3] present principles and methods to convert an OWL ontology into a Bayesian network. Their methods are based on probabilistic extensions to description logics. For more information on these extensions, see [12, 5]. The approach has some differences to ours. First, their aim is to create a method to transform any OWL ontology into a Bayesian network. Our goal is not to transform existing ontologies into Bayesian networks, but to create a method by which overlap between concepts could be represented and computed from a taxonomic structure. However, we designed the overlap graph and its RDF(S) implementation so, that it is possible, quite easily, to convert an existing crisp taxonomy to our extended notation. Second, in the approach of Ding and Peng, probabilistic information must be added to the ontology by the human modeller that needs to know probability theory. In our approach, the taxonomies can be constructed without virtually any knowledge of probability theory or Bayesian networks.

Also other approaches for combining Bayesian networks and ontologies exist. Gu [6] present a Bayesian approach for dealing with uncertain contexts. In this approach probabilistic information is represented using OWL. Probabilities and conditional probabilities are represented using classes constructed for these purposes. Mitra [15] presents a probabilistic ontology mapping tool. In this approach the nodes of the Bayesian network represents matches between pairs of classes in the two ontologies to be mapped. The arrows of the BN are dependencies between matches.

Kauppinen and Hyvönen [11] present a method for modeling partial overlap between versions of a concept that changes over long periods of time. The approach differs from ours in that we are interested in modelling degrees of overlap between different concepts in a single point of time.

7.2 Lessons Learned

Overlap graphs are simple and can be represented in RDF(S) easily. Using the notation does not require knowledge of probability or set theory. The concepts can be quantified automatically, based on data records annotated according to the ontology, for example. The notation enables the representation of any Venn diagram, but there are set structures, which lead to complicated representations.

Such a situation arises, for example, when three or more concepts mutually partially overlap each other. In these situations some auxiliary concepts have to be used. We are considering to extend the notation so that this kind of situations could be represented better. On the other hand, we do not think such situations are frequent in real-world taxonomies.

The Bayesian network structure that is created with the presented method is only one of the many possibilities. This one was chosen, because it can be used for computing the overlap tables in a most direct manner. However, it is possible that in some situations a different Bayesian network structure would be better.

7.3 Future Work

We intend to apply the overlap calculation in various realistic information retrieval situations. Also the refinement of the taxonomy language is considered to enhance its usability. The transformation of the taxonomy to alternative Bayesian network structures is an issue of future work, as well as trying the Bayesian network as a basis for personalisation.

8 ACKNOWLEDGEMENTS

Our research was funded mainly by the National Technology Agency Tekes.

References

1. Smith MK, Welty C, McGuinnes DL (2003) OWL web ontology language guide. http://www.w3.org/TR/2003/CR-owl-guide-20030818/
2. Brickley D, Guha RV (2004) RDF vocabulary description language 1.0: RDF Schema. http://www.w3.org/TR/rdf-schema/
3. Ding Z, Peng Y (2004) A probabilistic extension to ontology language owl. In: Proceedings of the 37th Hawaii international conference on system sciences. Big Island, Hawaii
4. Jensen FV (2001) Bayesian networks and decision graphs. Springer-Verlag, New York Berlin Heidelberg
5. Giugno R, Lukasiewicz T (2002) P-shoq(d): A probabilistic extension of shoq(d) for probabilistic ontologies in the semantic web. INFSYS research report 1843-02-06, Technische Universität Wien
6. Gu T, Pung HK, Zhang DQ (2004) A bayesian approach for dealing with uncertain contexts. In: Advances in pervasive computing. Austrian computer society, Vienna, Austria

7. Guarino N (1998) Formal ontology and information systems. In: Proceedings of FOIS'98. IOS Press, Amsterdam
8. Hyvönen E, Junnila M, Kettula S, Mäkelä E, Saarela S, Salminen M, Syreeni A, Valo A, Viljanen K (2004) Finnish museums on the semantic web. User's perspective on museumfinland. In: Proceedings of museums and the web 2004 (MW2004). Arlington, Virginia, USA
9. Hyvönen E, Valo A, Viljanen K, Holi M (2003) Publishing semantic web content as semantically linked HTML pages. In: Proceedings of XML Finland 2003. Kuopio, Finland
10. Pawlak J (1982) Rough sets. International Journal of Information and Computers 11:341–356
11. Kauppinen T, Hyvönen E (2005) Geo-spatial reasoning over ontology changes in time. In: Proceedings of IJCAI-2005 workshop on spatial and temporal reasoning. Edinburgh, Scotland
12. Koller D, Levy A, Pfeffer A (1997) P-classic: A tractable probabilistic description logic. In: Proceedings of AAAI-97
13. Kuenzer A, Schlick C, Ohmann F, Schmidt L, Luczak H (2001) An empirical study of dynamic bayesian networks for user modeling. In: Proc. of the UM'2001 workshop on machine learning for user modeling. Sonthofen, Germany
14. Mahalingam K, Huhns MN (1999) Ontology tools for semantic reconciliation in distributed heterogeneous information environments. Intelligent automation and soft computing (special issue on distributed intelligent systems)
15. Mitra P, Noy N, Jaiswal AR (2004) Omen: A probabilistic ontology mapping tool. In: Working notes of the ISWC-04 workshop on meaning coordination and negotiation. Hiroshima, Japan
16. Ng RT, Subrahmanian VS (1993) A semantical framework for supporting subjective and conditional probabilities in deductive databases. Automated reasoning journal 10(2):191–235
17. Ng RT, Tian X (1997) Semantics, consistence and query processing of empirical deductive databases. IEEE transactions on knowledge and data engineering 9(1):32–49
18. Straccia U (2005) Towards a fuzzy description logic for the semantic web. In: Proceedings of the Second european semantic web conference. Herakleion, Crete, Greece
19. Stuckenschmidt H, Visser U. (2000) Semantic translation based on approximate re-classification. In: Proceedings of the Semantic approximation, granularity and vagueness workshop. Breckenridge, Colorado, USA
20. Widyantoro DH, Yen J (2002) A fuzzy ontology-based abstract seawrch engine and its user studies. In: Proceedings of the 10th IEEE international conference on fuzzy systems. Melbourne, Australia
21. Zadeh L (1965) Fuzzy sets. Information and control 8:338-353