# A Bayesian Network Approach to Ontology Mapping

Rong Pan, Zhongli Ding, Yang Yu, and Yun Peng

Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, Maryland, USA
{pan.rong, zding1, yangyu1, ypeng}@umbc.edu

**Abstract.** This paper presents our ongoing effort on developing a principled methodology for automatic ontology mapping based on *BayesOWL*, a probabilistic framework we developed for modeling uncertainty in semantic web. In this approach, the source and target ontologies are first translated into Bayesian networks (BN); the concept mapping between the two ontologies are treated as evidential reasoning between the two translated BNs. Probabilities needed for constructing conditional probability tables (CPT) during translation and for measuring semantic similarity during mapping are learned using text classification techniques where each concept in an ontology is associated with a set of semantically relevant text documents, which are obtained by ontology guided web mining. The basic ideas of this approach are validated by positive results from computer experiments on two small real-world ontologies.

## 1 Introduction

Uncertainty concerns every aspect of semantic web ontologies. In many applications, overlapping between concepts/classes cannot be represented logically by OWL constructs. Even if they can, the degree of overlapping is not represented (e.g., how close a class *A* is to its super class *B*?). A description about an unknown concept or object input to an OWL reasoner may be uncertain (e.g., *x* is an instance of class *A* and is moderately likely to have property *p* related with class *B*). In a previous work, we have developed a Bayesian network based framework *BayesOWL*, to address representation and reasoning with uncertainty within a single ontology ([5], [6]).

Uncertainty becomes more prevalent in concept mapping between two ontologies where it is often the case that a concept defined in one ontology can only find partial matches to one or more concepts in another ontology. Semantic similarities between concepts are difficult, if not impossible to be represented logically, but can easily be represented probabilistically. This has motivated recent development of ontology mapping taking probabilistic approaches (GLUE [7], CAIMAN [11], OntoMapper [19], and OMEN [13]) (See [14] for a survey of existing approaches to ontology mapping, including those based on logical translation, syntactical and linguistic analysis). However, these existing approaches fail to completely address uncertainty in mapping. For example, GLUE captures similarity between two concepts onto1:A and onto2:B by joint probability distribution $P(A, B)$ obtained by text classification of

exemplars (semantically relevant text documents) to each concept. Then onto1:A is mapped to onto2:C whose similarity to onto1:A, measured by, say their Jaccard coefficients [21] (computed from the joint distribution), passes a threshold and is highest among all concepts in onto2. Here, onto1:A is taken as (semantically) equivalent to onto2:C, the degree of similarity between them will not be considered in future reasoning (e.g., subsumption within onto2). Also ignored are the other concepts that are also similar to onto1:A (albeit at smaller degree).

The work reported in this paper extends *BayesOWL* in a number of significant ways so that uncertainty in ontology mapping can be dealt with properly. As depicted in Figure 1 below, this new framework consists of three components: 1) a text classification based *learner* to learn from web data the probabilistic ontological information within individual ontologies and between concepts in two different ontologies; 2) a *BayesOWL* module to translate given ontologies (together with the learned uncertain information) into BNs; and 3) a concept *mapping module* which takes a set of learned raw similarities as input and finds mappings between concepts from two different ontologies based on evidential reasoning across two BNs.

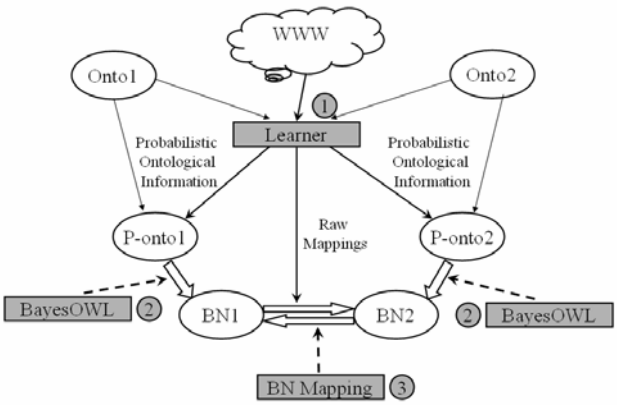



**Fig. 1.** The framework

Before describing the BN Mapping module and the learner in detail (Sections 3 and 4), we first provide some background information in Section 2. This includes a brief summary of *BayesOWL*, and introductions to Jeffrey's rule and iterative proportional fitting procedure (IPFP), two techniques used in this work. Methods and results of computer experiments with two small ontologies are given in Section 5. The paper concludes with discussions and directions of future research in Section 6.

## 2 Background

As background, we briefly introduce Jeffrey's rule, IPFP, and *BayesOWL* here.

### 2.1 Techniques for Updating Probability Distributions

Two techniques for updating a probability distribution by another distribution used in this work are briefly described below.

**Jeffrey's rule**, also known as rule of *probability kinematics* or *J-conditioning*, was proposed by Richard Jeffrey [9] to revise a probability measure (e.g., a joint distribution $P(x)$) by another probability function (e.g., a prior $Q(x_i)$ in another distribution). The rule can be written as follows in this context: if $P(x_i)$, our belief on $X_i \in X$ is changed to $Q(x_i)$, then the beliefs of other variables $X_{j \neq i} \in X$ shall be changed to

$$Q(x_j) = \sum_{x_i} P(x_j \mid X_i = x_i) Q(X_i = x_i) \tag{2.1}$$

if $P(x_j \mid x_i)$ is invariant with respect to $Q(x_i)$.

Jeffrey's rule can be used as a mechanism to update a distribution by soft evidence, represented as a distribution such as $Q(x_i)$. The rule then can be written as

$$P(x_i \mid se) = Q(x_i), \text{ and} \tag{2.2}$$

$$\begin{aligned} Q(x_{j \neq i}) &= P(x_j \mid se) \\ &= \sum_{x_i} P(x_j \mid X_i = x_i) P(X_i = x_i \mid se) \\ &= \sum_{x_i} P(x_j \mid X_i = x_i) Q(X_i = x_i) \end{aligned} \tag{2.3}$$

Pearl ([16], [17]) has shown that the virtual evidence, a method widely adopted in Bayesian network (BN) inference, can be viewed as formally equivalent to the likelihood ratio version of Jeffrey's rule. This is done by adding a virtual node $ve_i$ which has $X_i$ as its only parent in the BN, related by likelihood ratio:

$$L(X_i) = \frac{P(ve_i \mid X_i)}{P(ve_i \mid \overline{X_i})} = \frac{P(X_i) Q(\overline{X_i})}{Q(X_i) P(\overline{X_i})} \tag{2.4}$$

when $X_i$ is binary. Soft evidence update (eqs. 2.2 and 2.3) can be realized by BN belief update with $ve_i$ instantiated to true. It can be shown that $L(X_i)$ for multi-valued variables can also be calculated from $P(x_i)$ and $Q(x_i)$ [17].

As will be seen shortly, we use Jeffrey's rule to propagate probabilistically beliefs on variables between two BNs that are translated from two ontologies during mapping.

**IPFP** (**I**terative **P**roportional **F**itting **P**rocedure) is a computational procedure that updates a given distribution $Q_0(x)$ to satisfy a set of probability constraints $R = \{R_i(y^i)\}$ where each $R_i(y^i)$ is a distribution over $Y^i \subseteq X$ [10]. Roughly speaking, IPFP iterates over constraints in $\{R_i(y^i)\}$ in cycle, at each iteration, the current distribution is updated by one constraint according to

$$Q_k(x) = Q_{k-1}(x) \cdot \frac{R_i(y^i)}{Q_{k-1}(y^i)} \tag{2.5}$$

It has been proved based on *I-divergence* geometry ([4], [22]) that IPFP converges to an unique distribution $Q^*(x)$, which 1) satisfies all $R_i(y^i)$ in $R$, i.e., $Q^*(y^i) = R_i(y^i)$ for $R_i \in R$, and 2) has the smallest Kullback-Leibler distance (or I-divergence) to $Q_0(x)$ among all distributions $Q(x)$ that satisfy all constraints in $R$, i.e.,

$$I(Q^* \| Q_{(0)}) = \sum_x Q^*(x) \log \frac{Q^*(x)}{Q_{(0)}(x)} \quad \textbf{(2.6)}$$

is minimized. $Q^*(x)$ is called $I_1$-projection of $Q_0(x)$ on $R$. Bock [1] and Cramer [2] extended IPFP to conditional IPFP (CIPFP) to allow constraints with the form of conditional probability distributions and proved its convergence.

If we consider $Q(y^i)$ as soft evidence on a collection of variables $Y^i$, then IPFP can be considered as another mechanism of processing soft evidence [20]. The difference between Jeffrey's rule and IPFP in this regard is that the former requires the invariance of domain knowledge (i.e., $P(x_j \mid x_i)$ remains unchanged in $Q(x)$) while the latter requires minimizing I-divergence which in general destroys the invariance in the updated $Q^*(x)$. How to combine these two techniques together when used in ontology to BN translation and in concept mapping will be given in Subsection 2.2 and Section 3.

### 2.2 BayesOWL

*BayesOWL* ([5], [6]) is a framework which augments and supplements OWL for representing and reasoning with uncertainty based on Bayesian networks. This framework provides a set of rules and procedures for direct translation of an OWL ontology into a BN structure (a directed acyclic graph or DAG) and a method based on IPFP that utilizes available probability constraints about classes and interclass relations in constructing the conditional probability tables (CPTs) of the BN. The translated BN, which preserves the semantics of the original ontology and is consistent with the probabilistic constraints, can support ontology reasoning, both within and across ontologies, as Bayesian inferences.

**Structural translation** The general principle underlying the structural translation rules is that all classes (specified as "subjects" and "objects" in RDF triples of the OWL file) are translated into nodes in BN, and an arc is drawn between two nodes in BN if the corresponding two classes are related by a "predicate" in the OWL file, with the direction from the superclass to the subclass.

The model-theoretic semantics of OWL treats the domain as a non-empty collection of individuals. If class $A$ represents a concept, the node it is translated to is treated as a binary random variable of two states $a$ and $\bar{a}$, and we interpret $P(A = a)$ as the prior probability or one's belief that an arbitrary individual belongs to class $A$, and $P(a \mid b)$ as the conditional probability that an individual of class $B$ also belongs to class $A$. Similarly, for $P(\bar{a})$, $P(\bar{a} \mid b)$, $P(a \mid \bar{b})$, and $P(\bar{a} \mid \bar{b})$, we interpret the negation as "not belonging to".

Control nodes are created during the translation to facilitate modeling relations among class nodes that are specified by OWL *logical* operators, and there is a converging connection from each of the concept nodes involved in this logical relation to its specific control node. There are five types of control nodes in total corresponding to the five types of logical relations: "and" (owl:intersectionOf), "or" (owl:unionOf), "not" (owl:complementOf), "disjoint" (owl:disjointWith), and "same as" (owl:equivalentClass).

**Constructing CPTs** The nodes in the DAG obtained from the structural translation step can be divided into two disjoint groups: $X_R$, regular nodes representing concepts in ontology, and $X_C$, control nodes for bridging logical relations. The CPT for a control node in $X_C$ can be determined by the logical relation it represents so that when its state is "True", the corresponding logical relation holds among its parent nodes. When all the control nodes' states are set to "True" (denote this situation as *CT*), all the logical relations defined in the original ontology are held in the translated BN. The remaining issue is then to construct the CPTs for node in $X_R$ so that $P(X_R|CT)$, the joint distribution of all regular nodes in the subspace of *CT*, is consistent with all the given probabilistic constraints about classes and relations between classes. These constraints include, most likely, priors for classes $P(C)$, conditionals $P(C|D)$ for relations between classes *C* and *D*. Several suggestions have been made to encode probability constraints in semantic web languages (e.g., [6] with OWL, and [8] with RDF). These constraints can be obtained from the ontology designers or learned from data (an approach that learns these constraints from web is described in Section 4).

In principle, IPFP can be applied to construct CPTs to satisfy all the given probabilistic constraints. Two difficulties exist. First, as we mentioned earlier, direct application of IPFP may destroy the existing interdependencies between variables (i.e., the given DAG becomes invalid). Secondly, IPFP is computationally very expensive since every entry in the joint distribution of the BN must be updated at each iteration. To overcome these difficulties, we developed an algorithm named D-IPFP that decomposes IPFP so that each iteration only updates a small portion of the BN that are directly involved with the chosen constraint, and the update is done only to CPTs while keeping the DAG of the network intact [18]. In particular, when each of the given constraints involves only one variable $C_i$ and a set of zero or more of its parents $L_i$, (2.5) of IPFP becomes [5]

$$\begin{cases} Q_k(c_i \mid \pi_i) = Q_{k-1}(c_i \mid \pi_i) \cdot \dfrac{Q(c_i \mid L_i)}{Q_{k-1}(c_i \mid L_i)} \\ Q_k(c_j \mid \pi_j) = Q_{k-1}(c_j \mid \pi_j) \end{cases} \quad \forall j \neq i \tag{2.7}$$

The *BayesOWL* framework can support common ontology reasoning tasks as probabilistic inferences in the translated BN. For example, given a concept description *e*, it can answer queries about concept satisfiability (whether $P(e|CT) = 0$), about concept overlapping (how close *e* is to a concept *C* as $P(e|C,CT)$), and about concept subsumption (find the concept which is most similar to *e*) by defining some similarity measures such as Jaccard coefficient [21].

## 3 Concept Mapping Between Ontologies Using BN Mapping

It is often the case when attempting to map concept *A* defined in Ontology 1 to Ontology 2, there is no concept in Ontology 2 that is semantically identical to *A*. Instead, *A* is similar to several concepts in Ontology 2 with different degree of similarity. A solution to this so-called one-to-many problem, as suggested by [19] and [7], is to map *A* to the target concept *B* which is most similar to *A* by some measure. This simple approach would not work well because 1) the degree of similarity between *A* and *B* is not reflected in *B* and thus will not be considered in reasoning after the mapping; 2) potential information loss because other similar concepts are ignored in the mapping; 3) it cannot handle the situation where *A* itself is uncertain; and 4) it does not work well when more than one concepts need to be mapped. To see the last point, consider a situation where concept *x* defined as intersection of *A* and *B* in onto1 is to be mapped to onto2. Suppose the most similar concepts to *A* in onto2 are *C* and *D*, and those to are *B* are *E* and *D*, it would be difficult to determine which of the three (*C*, *D*, and *E*) *x* should be mapped to.

These difficulties in ontology mapping can be dealt with properly in our framework. We assume that pair-wise similarity measures are available between any concepts in two ontologies onto1 and onto2 (or between variables in BN1 and BN2, respectively). We take mapping as update on probability distribution of variables in BN2 by distributions of variables in BN1 in accordance to the similarity measures between these variables. Further inferences (e.g., finding the most probable subsumer in onto2 for a concept defined in onto1) can be drawn by Bayesian inference with the updated distribution of BN2. We present our approach starting with the basis: 1) a notion of *probabilistic semantic linkage* between a pair of concepts/variables; 2) the "1 to n" mapping (one variable in BN1 mapped to multiple similar ones in BN2); and 3) the "m to n" mappings where multiple variables in BN1 need to be mapped.

### 3.1 Pair-wise Probabilistic Semantic Linkage

We assume the similarity information between variable *A* in BN1 and *B* in BN2 is captured by the joint distribution *P*(*A*, *B*). This distribution is in a probability space, denoted as $PS^{1,2}$, which is related but different from the spaces for *A* and *B*, denoted as $PS^1$ and $PS^2$, respectively. Moreover, since this measure is based on the semantic similarity intrinsic to the meanings of these two variables, *P*(*A*, *B*) is assumed invariant with respect to changes in $PS^1$ and $PS^2$. That is, beliefs on variables in *A* and *B* may change when evidence is presented but not that of *P*(*A*, *B*) in $PS^{1,2}$.

**Probabilistic semantic linkage** between *A* and *B*, which serves as a basis mapping mechanism between similar variables, is defined as

$$SL^{1,2}_{A,B} = < PS^1, PS^2, A, B, P(A, B)>,$$

where $A \in PS^1$, and $B \in PS^2$, and *P*(*A*, *B*) measures the semantic similarity between *A* and *B*. Then the influence to *B* by *A* via the single linkage $SL^{1,2}_{A,B}$ changes *P*(*B*) to *Q*(*B*) by *P*(*A*). This update can be viewed as twice applications of Jeffrey's

rule across these three spaces, first from $PS^1$ to $PS^{1,2}$, then $PS^{1,2}$ to $PS^2$, as depicted in Figure 2 below. Since $A$ in $PS^1$ is identical to $A$ in $PS^{1,2}$, P($A$) in $PS^1$ becomes soft evidence $Q(A)$ to $PS^{1,2}$ by (2.2), the distribution of $B$ in $PS^{1,2}$ is updated by (2.3) to

$$Q(B) = \sum_A P(B \mid A)Q(A), \tag{3.1}$$

$Q(B)$ is then applied as soft evidence from $PS^{1,2}$ to node $B$ in $PS^2$, updating distribution of other variables $C$ in $PS^2$ by (2.3) as

$$Q(C) = \sum_B P(C \mid B)Q(B) = \sum_B P(C \mid B)\sum_A P(B \mid A)P(A). \tag{3.2}$$

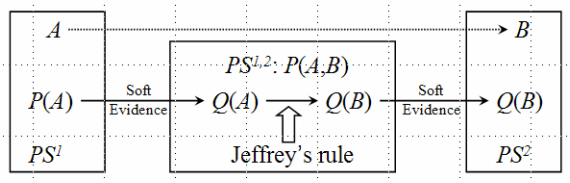



**Fig. 2.** Mapping concept $A$ to $B$ via semantic linkage $SL^{1,2}_{A,B}$

### 3.2 Multiple Semantic Linkages

Usually, $A$ in onto1 may be semantically similar to more than one concept in onto2. For, example, if $A$ is fairly similar to $B$ in onto2, it would also be similar to all super concepts and also some sub-concepts of $B$, possibly with different similarity measures. In other words, mapping $A$ to BN2 amounts mapping it through all semantic linkages that initiate from $A$ and end at each similar concept $B^J$ in BN2. Probabilistically, BN2 can be seen as receiving $n$ soft evidences, one for a linkage from A to $B^J$ for each concept $B^J$ in BN2. This requires 1) all similarity measures $P(A, B^J)$ remain invariant, and 2) conditional dependencies among variables in BN2 also remain invariant. This "1 to n" mapping can be carried out by a process that combines both Jeffrey's rule and IPFP. Like IPFP, this process is iterative over these linkages in a cycle until convergence.

This process can be realized by generalizing Pearl's virtual evidence approach for soft evidence update [15]. In this method of ours, each node $B^J$ is attached a virtual evidence node. At iteration step $k$, if linkage from A to $B^J$ is chosen, then we first calculate likelihood $L_K(B^J)$ for virtual evidence node $ve^J$ that will be used to simulate soft evidence $Q(B^J)$ by

$$L_K(B^J) = \frac{Q_{K-1}(B^J)Q(\overline{B}^J)}{Q(B^J)Q_{K-1}(\overline{B}^J)}, \tag{3.3}$$

and then apply Jeffrey's rule of (3.1) and (3.2) with the modified likelihood to update variable beliefs in BN2. Note that (3.3) is the same as (2.4) except for $Q_{k-1}(B^J)$, the new distribution obtained at step $k$-1 is used rather than the initial $P(B^J)$. Also note that this process does not explicitly modify the joint distribution of BN2 as the standard IPFP would do, instead, it modifies the likelihood associated with each virtual evidence node $ve^J$ while keep the joint distributions $P(A, B^J)$ and CPT's in BN2

unchanged. It can be shown that when the process converges, beliefs on variables in BN2 are consistent with all similarity measures $P(A, B^J)$ and $P(A)$, the belief of A in BN1.

**Mapping Reduction** Using all n linkages in "1 to n" type of mapping, as described above, is computationally very expensive because the IPFP process takes a number of iterations to converge, and each iteration involves belief update of BN2, which itself is exponential to the size of BN2. The problem gets worse for "m to n" type of mapping where what needs to be mapped is a composite concept that is defined as a conjunction (intersection) of several variables or their negations in BN1.

Fortunately, satisfying a given probabilistic relation $P(A, B)$ does not always require the use of a linkage from $A$ to $B$ or even know what the linkage looks like. Several probabilistic relations may be satisfied by one linkage. Consider a simple example in Figure 3 with variables $A$ and $B$ in $BN_1$, $C$ and $D$ in $BN_2$, and similarity (joint probabilities) between every pair as below:

$$P(C, A) = \begin{pmatrix} 0.3 & 0 \\ 0.1 & 0.6 \end{pmatrix},\ P(D, A) = \begin{pmatrix} 0.33 & 0.18 \\ 0.07 & 0.42 \end{pmatrix},$$

$$P(C, B) = \begin{pmatrix} 0.3 & 0 \\ 0.16 & 0.54 \end{pmatrix},\ P(D, B) = \begin{pmatrix} 0.348 & 0.162 \\ 0.112 & 0.378 \end{pmatrix}$$

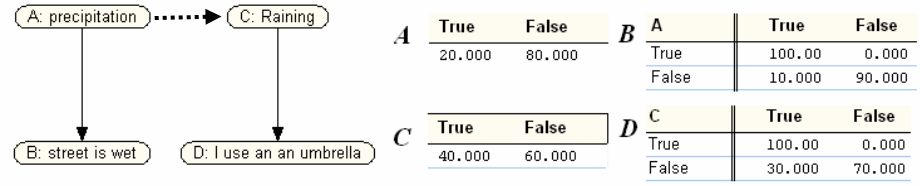



**Fig. 3.** Mapping Reduction Example

However, we do not need to set up linkages for all these relations. As Figure 3 depicts, when we have a linkage from $A$ to $C$, all these relations are satisfied (the other three linkages are thus redundant). This is because not only beliefs on $C$, but also beliefs on $D$ are properly updated by the mapping $A$ to $C$.

Several experiments with large BNs have shown that only a very small portion of all $n_1 \cdot n_2$ linkages are needed in satisfying all probability constraints. This, we suspect, is due to the fact that some of these constraints can be derived from others based on the probabilistic interdependencies among variables in the two BNs. We are currently actively working on developing a set of rules that examine the BN structures and CPTs so that redundant linkages can be identified and removed.

## 4 Learning Probabilities from Web Data

In this work, we use prior probability distributions $P(C)$ to capture the uncertainty about concepts (i.e., how likely an arbitrary individual belongs to class $C$), condi-

tional distributions $P(C|D)$ for relations between $C$ and $D$ in the same ontology (e.g., how likely an arbitrary individual in class $D$ is also in $D$'s subclass $C$), and joint probability distributions $P(A,B)$ for semantic similarity between concepts $C$ and $D$ from different ontologies. Often these kinds of probabilistic information are not available and are difficult to obtain from domain experts. Our solution is to learn these probabilities using text classification technique ([3], [12]) by associating a concept with a group of sample text documents called *exemplars*. The idea is inspired by those machine learning based semantic integration approaches such as [7], [11], and [19] where the meaning of a concept is implicitly represented by a set of exemplars that are relevant to it.

Learning the probabilities for semantic similarity between concepts in two ontologies is straightforward, assuming we have sufficient exemplars of good quality associated with each concept. First, we can build a model (classifier) for each concept in Ontology 1 according to the statistical information in that concept's exemplars using a text classifier such as Rainbow[1] or Bayesian text classifier dbacl[2]. Then concepts in Ontology 2 are classified into classes of Ontology 1 by feeding their respective exemplars into the models of Ontology 1 to obtain a set of probabilistic scores. These scores showing the inter-concept similarity in a probability form. Concepts in Ontology 1 can be classified in the same way into classes of Ontology 2. This cross-classification process (Figure 4) helps find a set of raw mappings between Ontology 1 and Ontology 2. Similarly, we can obtain prior or conditional probabilities related to concepts in a single ontology through self-classification with the models learned for that ontology.

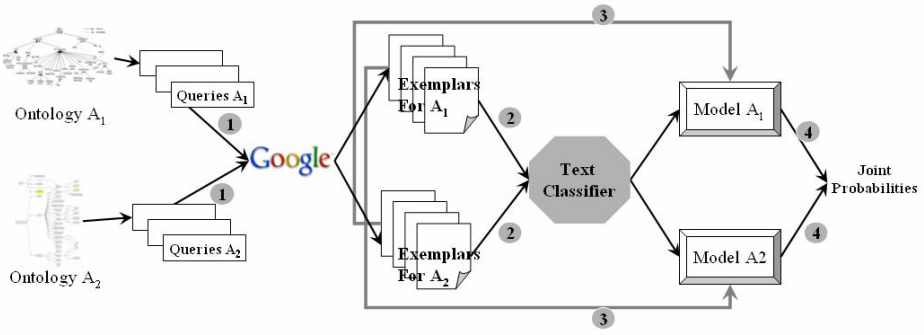



**Fig. 4.** Cross-classification using Text Classifiers on Web Data

The quality of these text classification based methods is highly dependent on the quality of text exemplars to each concept, which together should well capture the meaning of the concept. Two criteria are seen to be crucial in assessing the quality of exemplars: each exemplar (at least most of them) should be *relevant* to the meaning of the concept, and that these exemplars together should well *cover* all aspects of that concept. For example, articles on computer games are very relevant to the concept of "computer applications", but they alone hardly cover all computer applications.

[1] http://www-2.cs.cmu.edu/~mccallum/bow/rainbow
[2] http://www.lbreyer.com/

The need to find sufficiently many relevant exemplars for a large number of concepts greatly reduces the attractiveness and applicability of these machine learning based approaches. It would be a very time-consuming task for knowledge workers to find high quality text exemplars manually, as apparently the case for GLUE [7]. Our approach is to use search engines such as Google[3] to retrieve text exemplars for each concept node automatically from WWW, the richest information resource available nowadays. The goal is to search for documents in which the concept is used in its intended semantics. The rationale is that the meaning of a concept can be described or understood by the way it is used.

To find out what documents are relevant to a term, one cannot simply use the words in the name of the term as keywords to query the search engine. This because a word may have multiple meanings (word senses) and a query using only the name of the term in attention may return documents related to a meaning different from the intended semantics of the term. For example, in an ontology for "food", a concept named "apple" is a subconcept of "fruit". If one only uses "apple" as the keyword for query, documents showing how to make an apple pie and how to use an iPod may both be returned. Clearly, the documents using "apple" for its meaning in computer field is irrelevant to "apple" as a fruit. Fortunately, since we are dealing with concepts in well defined ontologies, the semantics of a term is to a great extent specified by the other terms used in defining this concept in the ontology, including names of its super and subconcept classes and the properties of this concept and its super classes. This semantic information can thus be used to guide the web search with increased relevancy. There are a number of ways the semantic information can be used to help search. The simplest one, and the one we have experimented so far is to form search query for one concept by combining all the terms on the path from root to that concept node in the taxonomy. In the "apple" example, the query would then become "food fruit apple", and documents about iPod and Apple computers would not be returned.

In the experiments, for each concept *A*, we search the web to obtain two sets of exemplars: $U^{A+}$ containing exemplars that support (or positively related to) *A*; and $U^{A-}$, containing exemplars that support the negation of (or negatively related to ) *A*. Exemplars in $U^{A+}$ are obtained by searching the web for pages that contain *A* and all names of *A*'s ancestors on the taxonomy, while that for $U^{A-}$ are obtained by search pages that contain all names of *A*'s ancestors but not *A*.

With all these documents, we can obtain joint probabilities of *A* and *B* by text classification, similar to what is done in GLUE [7]: applying the classifiers of concepts *A* and *B* to all text documents in *U*, and classify them into four categories: $U^{A+B+}$, $U^{A+B-}$, $U^{A-B+}$, and $U^{A-B-}$. Then the joint probabilities can be obtained by counting the items in each category, e.g., $P(A, B)= |U^{A+B+}| / |U|$. If we only search for positive exemplars $U^{A+}$ and $U^{B+}$, then only conditional probability $P(B|A)$ can be obtained (by applying *B*'s classifier to *A*'s supportive exemplars to obtain $U^{A+B+}$ and compute $P(B|A) = |U^{A+B+}| / |U^{A+}|$). The first approach is the one that works for our purpose.

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

## 5 Experiments

We have performed computer experiments on two small-scale real-world ontologies. Our goals are to find how good the learning can be with the exemplars mined from the web, and how the uncertainty inference across multiple Bayesian networks could help ontology mapping.

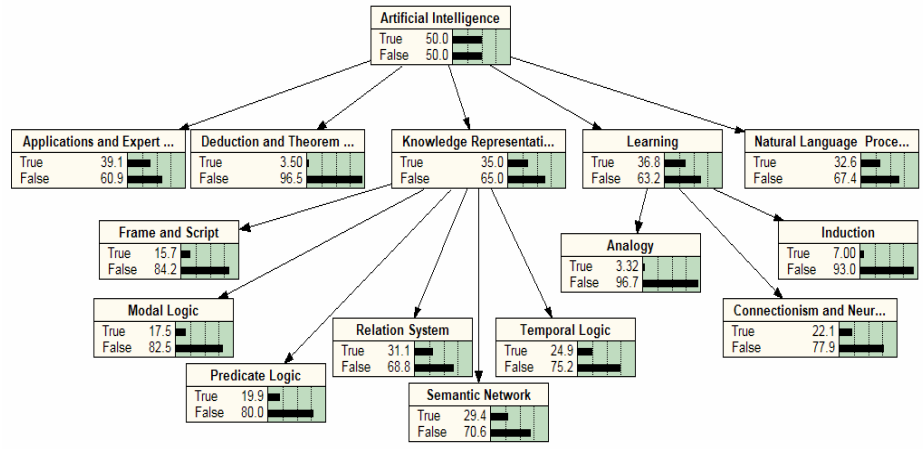
**Translating Taxonomies to BNs** We took the Artificial Intelligence sub-domain from ACM Topic Taxonomy[4] and DMOZ[5] (Open Directory) hierarchies and pruned some concepts to form two ontologies, both of which have a single root node *Artificial Intelligence*. All other concepts in the hierarchies are sub categories of AI. These two hierarchies differ in both terminologies and modeling methods. DMOZ categorizes concepts by popularities of web pages to facilitate people's easy access to these pages, while ACM topic hierarchy categorizes concepts from super to sub to structure a classification primarily for academics.

**Table 1.** Statistics of the expirements

| Taxonomies | # Nodes | Depth | Total Exemplar size | Avg. Exemplar Size | # Exemplar | Avg. # Exp./node |
|---|---|---|---|---|---|---|
| ACM AI | 15 | 3 | 19.7 MB | 698 KB | 24533 | 1636 |
| DMOZ AI | 25 | 3 | 29.2 MB | 612 KB | 35148 | 1406 |

For every concept, except the root, we obtained exemplars by querying Google as described in the previous section. The statistics of these web pages is listed in Table 1. We used Bayesian text classifier dbacl to create a model for each non-root concept $X$ and obtained the pair-wise conditional probability $P(X \mid Parent(X))$. The root nodes were assigned a prior probability as (0.5, 0.5).

Then, using *BayesOWL*'s translation rules, the two ontologies were translated into two BNs as shown in Figure 5.




[4] http://www.acm.org/class/1998/

[5] http://dmoz.org/

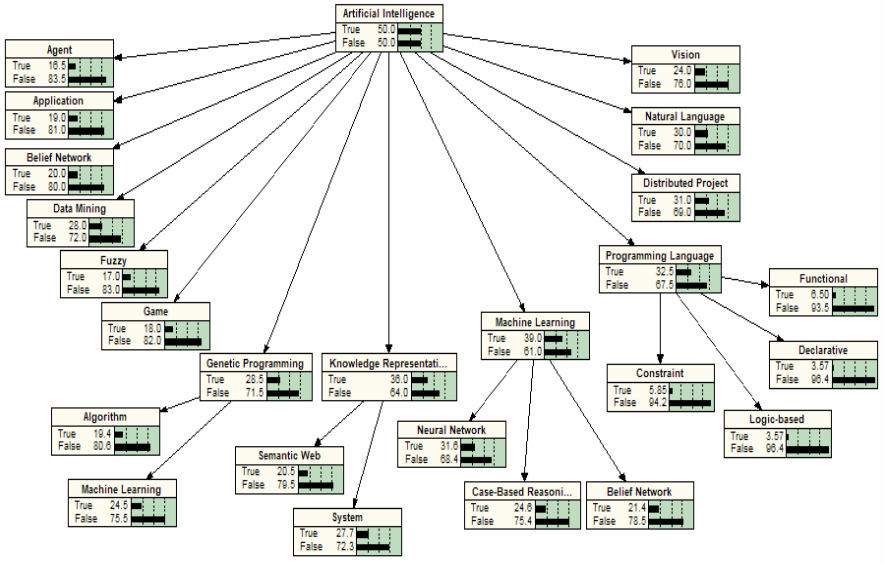



**Fig. 5.** Bayesian network for ACM topics' AI sub-domain and DMOZ's AI sub-domain

**Learning uncertainty mappings** Raw mappings $P(A, B)$ were computed for each pair of concepts of the two BNs. The similarity between A and B were measured by their Jaccard coefficient, computed from the joint probability. Table 2 lists the five most similar concepts and five most different concepts in the learning result. The top three most similar concepts are actually identical concepts. However, besides these three, another pair of identical concepts is not measured as highly related. They are /*Learning*/*Connectionism & Neural Net* in ACM topic and /*Machine Learning*/*Neural Network* in DMOZ. Their similarity is only 0.61. We speculate this is because the term "connectionism" is not as popular as when ACM topic hierarchy was constructed, and thus is not used along with "*Neural Network*" in most web pages.

**Table 2.** Five most similar concepts and most different concepts in the learning result. The root concept's name is omitted.

| ACM topic | DMOZ | Similarity |
|---|---|---|
| /Knowledge Representation & Formalism Method | /Knowledge Representation | 0.96 |
| /Natural Language Processing | /Natural Language | 0.90 |
| /Learning | /Machine Learning | 0.88 |
| /Learning | /Knowledge Representation | 0.81 |
| /Applications & Expert System | /Knowledge Representation | 0.79 |
| …… | | |
| /Fuzzy | /Learning/Analog | 0.03 |
| /Learning/Induction | /Learning/Game | 0.02 |
| /Deduction & Theorem Proving | /Programming Language/Declarative | 0.02 |
| /Learning/Induction | /Application | 0.01 |
| /Learning/Analogy | Agent | 0.01 |

**Inference with BN Mappings** Treating ontology mapping as Bayesian network mapping as described here allows us to conduct probabilistic reasoning far beyond finding the best concept match. We are currently actively investigating this issue and developing related algorithms. To illustrate our point, consider the example of finding a description of DMOZ's */Knowledge Representation/Semantic Web* (*dmoz.sw*) in ACM topic. There is no ACM concept that is identical to *dmoz.sw*, it must be described by a composite expression involving multiple ACM concepts. The two most semantically similar concepts to *dmoz.sw* in ACM are */Knowledge Representation and Formalism Method/Relation System* (*acm.rs*) and */Knowledge Representation and Formalism Method/Semantic Network* (*acm.sn*) with the joint distributions

$$P(dmoz.sw, acm.rs) = \begin{pmatrix} 0.60 & 0.12 \\ 0.21 & 0.07 \end{pmatrix} \text{ and } P(dmoz.sw, acm.sn) = \begin{pmatrix} 0.58 & 0.13 \\ 0.25 & 0.04 \end{pmatrix},$$

and respective Jaccard coefficients $J(dmoz.sw, acm.rs) = 0.64$, and $J(dmoz.sw, acm.sn) = 0.61$.

From the two joint probabilities, we can see that *dmoz.sw* is not a subconcept of either *acm.rs* or *acm.sn*, but had a sizable overlap with each of them. From the following joint probabilities

$$P(acm.rs, acm.sn) = \begin{pmatrix} 0.2612 & 0.0498 \\ 0.0323 & 0.6557 \end{pmatrix},$$

we can see that *acm.rs* and *acm.sn* also overlap with each other. Figure 6 illustrates the overlap of these three concepts.

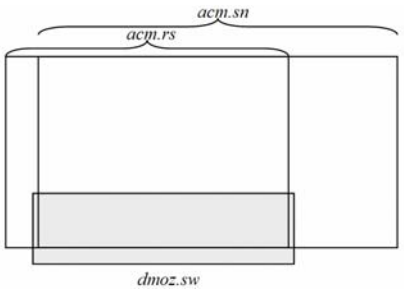



**Fig. 6.** The Venn diagram for *dmoz.sw*, *acm.rs*, and *acm.sn*

This leads to a conjecture that *dmoz.sw* may be described in terms of *acm.rs* and *acm.sn*. To validate this conjecture, we need to have the conditional probability $P(acm.rs = true, acm.sn = \text{true} \mid dmoz.sw = true)$. This can be obtained as follows.

1. Using learned probabilities $P(dmoz.sw, acm.rs)$ and $P(dmoz.sw, acm.sn)$, two semantic linkage were created, from *dmoz.sw* to *acm.rs* and to *acm.sn*, respectively.
2. Instantiate *dmoz.sw* as *true*, and compute the likelihoods for the two virtual evidence nodes associated with *acm.rs* and *acm.sn*.
3. Compute $P(acm.rs = true, acm.sn = \text{true} \mid dmoz.sw = true)$ by any Bayesian network inference algorithm with the two virtual evidence nodes set to true.

In our experiment, this probability was computed to be 0.851. From this we could conclude that intersection of *acm.rs* and *acm.sn* is the highly probable subsumer of

*dmoz.sw*. More detailed analysis may require having the joint distribution of the three concept nodes (in two ontologies/BNs) or distribution involving additional relevant ACM concepts (with similarity measure lower than those of *acm.rs* and *acm.sn*). These distributions can be computed in the similar fashion.

## 6 Discussion and Future Work

This paper describes our ongoing research on developing a probabilistic framework for automatic ontology mapping. In this framework, ontologies (or parts of them) are first translated into Bayesian networks, and then the concept mapping is realized as evidential reasoning between the two BNs by Jeffrey's rule. The probabilities needed in both translation and mapping can be obtained by using text classification programs, supported by associating to individual concepts relevant text exemplars retrieved from the web.

We are currently actively working on each of these components. In searching for relevant exemplars, we are attempting to develop a measure of relevancy so that less relevant documents can be removed. We are also investigating how semantic information can be utilized to post-process text documents mined from the web so that less relevant ones can be identified and excluded. We are expanding the ontology to BN translation from taxonomies to include properties, and develop algorithms to support common ontology-related reasoning tasks. As for a general BN mapping framework, our current focus is on linkage reduction. We are also working on the semantics of BN mapping and examining its scalability and applicability. Future work also includes developing methods to properly deal with inconsistent probability constraints in IPFP process.

## Acknowledgement

This work was supported in part by DARPA contract F30602-97-1-0215 and NSF award IIS-0326460.

## References

1. Bock, H. H. 1989. A Conditional Iterative Proportional Fitting (CIPF) Algorithm with Applications in the Statistical Analysis of Discrete Spatial Data. *Bull. ISI, Contributed Papers of 47th Session in Paris*, 1: 141-142.
2. Cramer, E. 2000. Probability Measures with Given Marginals and Conditionals: *I*-projections and Conditional Iterative Proportional Fitting. *Statistics and Decisions*, 18: 311-329.
3. Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; and Slattery, S. 2000. Learning to Construct Knowledge Bases from the World Wide Web. *Artificial Intelligence*, 118(1-2): 69-114.
4. Csiszar, I. February 1975. *I*-divergence Geometry of Probability Distributions and Minimization Problems. *The Annuals of Probability*, 3(1): 146-158.

5. Ding, Z.; Peng, Y.; and Pan, R. November 2004. A Bayesian Approach to Uncertainty Modeling in OWL Ontology. In *Proceedings of 2004 International Conference on Advances in Intelligent Systems - Theory and Applications (AISTA2004)*. Luxembourg-Kirchberg, Luxembourg.
6. Ding, Z.; and Peng, Y. January 2004. A Probabilistic Extension to Ontology Language OWL. In *Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS-37)*. Big Island, Hawaii.
7. Doan, A.; Madhavan, J.; Domingos, P.; and Halevy, A. 2004. Ontology Matching: A Machine Learning Approach. *Handbook on Ontologies in Information Systems*, S. Staab and R. Studer (eds.), Springer-Velag, 2004. Invited paper. Pages 397-416.
8. Fukushige, Y. October 2004. Representing Probabilistic Knowledge in the Semantic Web. Position paper for *the W3C Workshop on Semantic Web for Life Sciences*. Cambridge, MA, USA.
9. Jeffery, R. 1983. *The logic of Decisions 2nd Edition*, University of Chicago Press.
10. Kruithof, R. Telefoonverkeersrekening, *De Ingenieur* 52, E15-E25, 1937.
11. Lacher, M.; and Groh, G. May 2001. Facilitating the Exchange of Explicit Knowledge through Ontology Mappings. In *Proceedings of the 14th International FLAIRS Conference*. Key West, FL, USA.
12. McCallum, A.; and Nigam, K. 1998. A Comparison of Event Models for Naive Bayes Text Classification. *AAAI-98 Workshop on "Learning for Text Categorization"*.
13. Mitra, P.; Noy, N. F.; and Jaiswal, A. R. 2004. OMEN: A Probabilistic Ontology Mapping Tool. In *Workshop on Meaning Coordination and Negotiation at the Third International Conference on the Semantic Web (ISWC-2004)*. Hisroshima, Japan.
14. Noy, N. 2004. Semantic integration: A survey of ontology-based approaches. *SIGMOD Record*.
15. Pan, R.; and Peng, Y.;. 2005. A Framework for Bayesian Network Mapping. (Extend Abstract). Accepted by *AAAI-05*.
16. Pearl, J. 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufman, San Mateo, CA.
17. Pearl, J. 1990. Jeffery's rule, passage of experience, and neo-Bayesianism. In H.E. et al. Kyburg, Jr., editor, *Knowledge Representation and Defeasible Reasoning*, pages 245-265.
18. Peng, Y.; and Ding, Z. July 2005. Modifying Bayesian Networks by Probability Constraints. *Proceedings of the 24th Conference on Uncertainty in AI (UAI 2005)*. Edinburgh, Scotland.
19. Prasad, S.; Peng, Y.; and Finin, T. 2002. A Tool For Mapping Between Two Ontologies (Poster), *International Semantic Web Conference* (*ISWC02*).
20. Valtorta, M.; Kim, Y.; and Vomlel, J. 2002. Soft Evidential Update for Probabilistic Multiagent Systems. *International Journal of Approximate Reasoning*, 29(1): 71-106.
21. van Rijsbergen, C. J. 1979. *Information Retrieval*. London: Butterworths. Second Edition.
22. Vomlel J. 1999. Methods of Probabilistic Knowledge Integration. *PhD thesis*, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University.