2013 IEEE 25th International Conference on Tools with Artificial Intelligence

Ontology Learning from Incomplete Semantic Web Data by BelNet

Man Zhu[†], Zhiqiang Gao[†], Jeff Z. Pan[‡], Yuting Zhao[‡], Ying Xu[†], Zhibin Quan[†]
[†]*School of Computer Science & Engineering, Southeast University, P.R. China*
[‡]*Department of Computer Science, The University of Aberdeen, UK*

Abstract—**Recent years have seen a dramatic growth of semantic web on the data level, but unfortunately not on the schema level, which contains mostly concept hierarchies. The shortage of schemas makes the semantic web data difficult to be used in many semantic web applications, so schemas learning from semantic web data becomes an increasingly pressing issue. In this paper we propose a novel schemas learning approach - BelNet, which combines description logics (DLs) with Bayesian networks. In this way BelNet is capable to understand and capture the semantics of the data on the one hand, and to handle incompleteness during the learning procedure on the other hand. The main contributions of this work are: (i) we introduce the architecture of BelNet, and correspondingly propose the ontology learning techniques in it; (ii) we compare the experimental results of our approach with the state-of-the-art ontology learning approaches, and provide discussions from different aspects.**

I. Introduction

Ontologies play an important role [14] in applications of the Semantic Web (SW). Ontology annotated data is growing rapidly; e.g., from May 2009 to March 2010, the number of RDF triples has grown from 4.7 billion to 16 billion [1]. However, the knowledge acquisition bottleneck has resulted in inexpressive schemas on SW [2], which gives rise to the research of ontology learning [20] and query learning [13] from SW data.

Although data mining and machine learning techniques, such as association rule mining [18] and inductive logic programming (ILP) [11], can be applied to SW data, the problem of learning schemas from instance-level data remains challenging. The widely adopted assumption in machine learning domain is the closed-world assumption (CWA) - assuming true of the specified and derivable statements, and false otherwise. However, the SW uses open-world assumption (OWA); i.e., the truth values of unspecified and underived statements are assumed as *unknown* [16], rather than false. In other words, SW data is assumed to be incomplete. The presence of incompleteness can lead to the over-fitting problem if CWA is made. For example, one might learn the axiom that *Grandson* is a *Male* who is not a *Person* from a data set missing statements of individual grandsons are persons.

The approach proposed in this paper adopts a probabilistic point of view to deal with the aggressive 'false' under 'risky' CWA. More precisely, we propose the Bayesian description logic Network, or simply BelNet, a description logic based Bayesian Network [15] for learning schema axioms (TBox) from data axioms (ABox). In addition to the CWA issue, BelNet can deal with two further issues for ontology learning, i.e. (i) only being able to learn one axiom at a time, and (ii) only being able to learn crisp axioms. Moreover, learning one single axiom may reject the 'probable' correct answers or accept the 'likely' incorrect answers. To address this issue, a global target function is used in BelNet for leading the ontology learner out of the local optimum.

We have intensively studied the properties of BelNet, theoretically and practically (cf. Sec 3), after introducing the basic notions of description logics and Bayesian networks (cf. Sec 2). In BelNet, the links normally signify the subsumption relationship. Given the ABox data in the ontology, BelNet firstly learns the structure that best encodes the subsumption dependencies supported by ABox data. From the structure, subsumption axioms, such as $Grandson \sqsubseteq Male$, which means a person who is a *Grandson* is also a *Male*, are extracted directly. In addition, we propose an approach to generate candidate weighted axioms, which are consequently transformed into linear time inferencing in BelNet (cf. Sec 4). We compare the performance of the ontology learning approach using BelNet with the state-of-the art systems DLLearner and GoldMiner (cf. Sec 5). Our experiments show that our proposed approach is able to learn TBox axioms even when the ABox data in the ontology is quite rare (incomplete).

II. Preliminary

A. Bayesian Networks

Bayesian networks (BNs), belonged to the family of probabilistic graphical models. The graphical structures are used to represent knowledge about an uncertain domain. Each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies. BNs enable an effective representation and computation of the *joint probability distribution* (JPD) over a set of random variables [15].

More formally, a BN is defined by a pair $B = \langle \mathcal{G}, \Theta \rangle$, where $\mathcal{G}$ is a DAG whose nodes $V_1, V_2, ..., V_n$ represent random variables, and whose edges represent the direct dependencies between these variables. Θ denotes the set of parameters of the network.

1082-3409/13 $31.00 © 2013 IEEE
DOI 10.1109/ICTAI.2013.117

IEEE computer society

Belief propagation, also known as *sum-product message passing* is a widely used message passing algorithm for performing inference on graphical models, and *will be used in BelNet as well for inference*. There are two main approaches to deal with parameter estimation: one is based on *maximum likelihood estimation*, and the other uses Bayesian approaches. In BelNet, the Bayesian approach will be adopted for the parameter estimation which is better to avoid overfitting.

B. Ontology & Description Logic $\mathcal{ALC}$

An ontology comprises TBox (*terminology*, i.e., the vocabulary of an application domain) and ABox (*assertions*). TBox consists of concepts denoting sets of individuals (we denote the set of concept names by N_C), and roles denoting binary relationships between individuals (we denote the set of role names by N_R). ABox contains assertions about named individuals (we denote the set of individual names by N_I) in terms of the TBox. We further categorise the ABox into two sets. One is the set of concept assertions such as *Grandson(Mathiew)* , and the other is the set of role assertions between individuals such as *hasChild(Paul, Mathiew)*. The assertions in the ABox are also called *facts*.

Description logics (DLs) provide the logical formalism for ontologies and the Semantic Web. Here we briefly introduce DL $\mathcal{ALC}$ ontology, which is used in BelNet. In DLs, interpretations are used to assign a meaning to syntactic constructs. An *interpretation* $\mathcal{I}$ consists of a non-empty set $\Delta^{\mathcal{I}}$. An *interpretation function* $\cdot^{\mathcal{I}}$ assigns to every object $a \in N_{\mathcal{I}}$ an element of $\Delta^{\mathcal{I}}$, to every atomic concept $A \in N_C$ a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and to every atomic role $r \in N_R$ a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. If $\mathcal{I}$ satisfies an axiom (resp. a set of axioms), then we say that it is a *model* of this axiom (resp. set of axioms). The syntax and semantics of $\mathcal{ALC}$ are given in Table I, in which C and D are $\mathcal{ALC}$ concepts. A concept C is subsumed by a concept D if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model $\mathcal{I}$ of this vocabulary $\mathcal{T}$, written as $C \sqsubseteq_{\mathcal{T}} D$ or $\mathcal{T} \models C \sqsubseteq D$ (*subsumption*). Two concepts C and D are disjoint if $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ for every model $\mathcal{I}$ (*disjointness*). Over an ABox $\mathcal{A}$, $\mathcal{A} \models \alpha$ if every model of $\mathcal{A}$ also satisfies α.Please refer to [7] for further details of DLs.

III. Bayesian Description Logic Network

In connection with a DL ontology, the corresponding Bayesian description logic Network (BelNet) is a graph-based knowledge representation showing relationships between concepts. A BelNet contains two components:

The structure of a BelNet is a directed acyclic graph (DAG), where

- vertexes represent DL concepts (expressions).
- links signify the existence of direct influences between the linked vertexes. To be specific, two nodes are linked, if they represent exactly the two concepts in two sides of an inclusion axiom; links can be *conditional*, which means the vertex on one side of the link is completely determined by the other. (c.f. Figure 1).

Table I
Syntax and semantics of DL $\mathcal{ALC}$.

	construct	syntax	semantics
	atomic concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
	atomic role	r	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
	top concept	$\top$	$\Delta^{\mathcal{I}}$
	bottom concept	$\bot$	$\emptyset$
	conjunction	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
	universal restriction	$\forall r.C$	$(\forall r.C)^{\mathcal{I}} = \{a \mid \forall b.(a,b) \in r^{\mathcal{I}}$ implies $b \in C^{\mathcal{I}}\}$
$\mathcal{U}$	*disjunction*	$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\mathcal{C}$	*negation*	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$\mathcal{E}$	*existential restriction*	$\exists r.C$	$(\exists r.C)^{\mathcal{I}} = \{a \mid \exists b.(a,b) \in r^{\mathcal{I}}$ and $b \in C^{\mathcal{I}}\}$

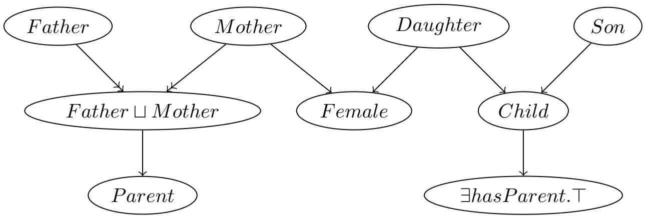
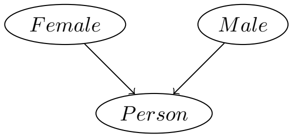


Figure 1. The graphical representation of a BelNet. The links from $Father$ and $Mother$ to $Father \sqcup Mother$ are conditional.

The numerical information relies on statistics approach against the facts in the ABox, and shows how and in which way the ABox is supporting the relations (links) between two concept vertexes.

For a complete example of BelNet please refer to Figure 2. In the following, we will firstly introduce how the graph structure of BelNet is built from an ontology, and then we will illustrate how the information in the ABox is used to calculate the Joint Probability Distribution (JPD) for the BelNet.

$P(Fe^T)$
0.5



$P(Ma^T)$
0.5

Fe	Ma	$P(Pe^T \mid Fe, Ma)$
T	F	1
F	T	1

Figure 2. Motivated BelNet Example. Ma, Fe are short for $Male$ and $Female$. Because $Female$ and $Male$ both equals T or F corresponds to an impossible situation in reality, thus there are only 2 rows in the CPT of node $Person$.

A. Building DAG for BelNet of an Ontology

Given an ontology $\mathcal{O}$, let N_C^+ be all concept expressions appearing in $\mathcal{O}$. For any $C \in N_C^+$, We define its *parents* as $Pa(C) = \{C' \in N_C^+ \mid \mathcal{O} \models C' \sqsubseteq C$, and there is no C'' such that $\mathcal{O} \models C' \sqsubseteq C''$, and $\mathcal{O} \models C'' \sqsubseteq C\}$.

For an ontology $\mathcal{O} = < T, A >$, its *ABox materialisation* is $M_A(\mathcal{O}) = \{A(a) \mid A \in N_C^+, a \in N_I, \mathcal{O} \models A(a)\}$. If $M_A(\mathcal{O}) \subseteq A$, then we say $\mathcal{O}$ is *ABox materialised.*

For convenience in this paper we use the same symbol for both the concept in DL ontology and the corresponding vertex in the graph.

Given a consistent ontology $\mathcal{O} =< T, A >$, its corresponding ***BelNet structure***, denoted as $\mathcal{G} = Bel(\mathcal{O})$, is generated with the following steps:

1) For each $C \in N_C^+$, there is a C vertex in $Bel(\mathcal{O})$;
2) If $C' \in Pa(C)$, then link from vertex C' to C;
3) If $C' \equiv C$ and $C \in V$, then label C with an alias C';

Example 1: Figure 1 shows the graphical representation of the $Bel(\mathcal{O})$, in which the TBox of ontology $\mathcal{O}$ contains:

$$\begin{array}{cc}
Father \sqsubseteq Father \sqcup Mother & Mother \sqsubseteq Female \\
Mother \sqsubseteq Father \sqcup Mother & Daughter \sqsubseteq Female \\
Father \sqcup Mother \sqsubseteq Parent & Daughter \sqsubseteq Child \\
Child \sqsubseteq \exists hasParent.\top & Son \sqsubseteq Child
\end{array}$$

It is not hard to prove Proposition 1, which guarantees the above generated BelNet structure is a DAG, so that we could adapt Bayesian Network approach in it.

Proposition 1: (**DAG**) Given an ontology $\mathcal{O}$, $Bel(\mathcal{O})$ is a Directed Acyclic Graph (DAG).

B. Generating Joint Probability Distribution for BelNet

Along with ontology TBox constructing the links in $Bel(\mathcal{O})$, ontology ABox contributes to the parameters on the vertexes, which reflects the supportiveness from the ABox to the BelNet structure. Here we will introduce how to generate the conditional probability table (CPT) for the vertexes in $Bel(\mathcal{O})$, then for the whole BelNet structure.

It is natural to use a finite ontology domain $\Delta^{\mathcal{I}}$ to restrict all elements in the possible world in the BelNet. For convenience, we assume $\Delta^{\mathcal{I}}$ contains all individual names in the ontology, and an individual name o is always interpreted to itself, *i.e.*, $o^{\mathcal{I}} = o$.

We call all interpretations related to o a *possible observation* $\mathbf{o}$. For example, $C^{\mathcal{I}} = \{a, b\}$, then there are two possible observations, where $C^{\mathbf{o_1}} = \{a\}$, $C^{\mathbf{o_2}} = \{b\}$. A possible observation is an interpretation which assigns at most one element to one concept. We assume that all possible observations are independent.

Firstly for a *marginal node* C, which has no parents in $Bel(\mathcal{O})$, the *marginal probability* is a table of $P(C^\sharp)$, where $\sharp \in \{\texttt{TRUE}, \texttt{FALSE}\}$. Furthermore, $P(C^{\texttt{TRUE}})$ is the probability that a possible observation supports C, *i.e.*, $o \in C^{\mathcal{I}}$. Similarly $P(C^{\texttt{FALSE}})$ is the probability that a possible observation does not support C, *i.e.*, $o \notin C^{\mathcal{I}}$. Actually the values are calculated with the number of individuals satisfiying concept C in the original ontology. For convenience in the following `TRUE` and `FALSE` are shortened to `T` and `F`.

Secondly for nodes with parents, the conditional probability should be calculated.

Definition 1: (Bayesian subsumption axiom) A Bayesian subsumption axiom is in the form of $D|C_1, \ldots, C_n$, where $C_i \sqsubseteq D, i \in \{1, \ldots, n\}$.

The vertexes in $Bel(\mathcal{O})$ are treated as random variables, so the *Conditional Probability Tables* (CPT) of a Bayesian subsumption axiom is calculated as follows:

$$P(D|C_1, \ldots, C_n) = \frac{P(D, C_1, \ldots, C_n)}{P(C_1, \ldots, C_n)} \tag{1}$$

where $P(C_1, \ldots, C_n)$ is a (discrete) joint probability distribution of variables $C_1, \ldots, C_n$, similar for $P(D, C_1, \ldots, C_n)$.

Actually, under a possible observation C_i has two values: `T` and `F`. For a specific observation $\mathbf{o}$, $\mathbf{o}$ supports $C_i^{\texttt{T}}$ if $o \in C_i^{\mathcal{I}}$, and $\mathbf{o}$ supports $C_i^{\texttt{F}}$ if $o \notin C_i^{\mathcal{I}}$. These cases can be abbreviated as $C_i^{\mathbf{o}}$.

Example 2: Given an ontology $\mathcal{O} =< T, A >$, where T includes $\{Male \sqsubseteq Person, Female \sqsubseteq Person\}$. We also have ABox as:

$$Person(a), Person(b), Male(a), Female(b)$$

Figure 2 shows the CPTs for *Female*, *Male*, and *Person*.

Actually the CPT reflects how much degree the ABox supports the subsumption axioms. Obviously we have the following proposition.

Proposition 2: In the BelNet of an ABox materialised ontology $\mathcal{O}$, we have $P(D^T|C^T) = 1$, if $C \in Pa(D)$.

Proof: Follows directly from the steps of transforming ontology into BelNet, nodes C in $Pa(D)$ satisfy $C \sqsubseteq D$. Since $\mathcal{O}$ is ABox materialised, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ means that for any possible observation $\mathbf{o}$, if $o \in C^{\mathcal{I}}$, then we must have $o \in D^{\mathcal{I}}$. So we have: $P(D^T, C^T) = P(C^T)$. From equation (1), $P(D^T|C^T) = \frac{P(D^T, C^T)}{P(C^T)} = 1$. ■

Lemma 1: Given a consistent and ABox materialised ideal ontology $\mathcal{O}$, and the corresponding $Bel(\mathcal{O})$, we have $P(D^T|Pa(D)) = 1$, if for all $C_i \in Pa(D)$ are satisfiable in $\mathcal{O}$, *i.e.*, there exists o_i, *s.t.* $o_i \in C_i^{\mathcal{I}}$.

An ideal ontology here means that the ABox contains no noise. Calculations on ideal ontologies are approximations for practical cases. Now we can measure the global supportiveness from evidences in ontology ABox to a BelNet structure $Bel(\mathcal{O})$.

Definition 2: (BelNet Joint Probability Distribution(JPD)) Given an ontology $\mathcal{O}$ and its BelNet structure $\mathcal{G}$, the joint probability distribution is defined as:

$$P(\mathcal{G}) = P(V_1, \ldots, V_n) = \prod_{i=1}^{n} P(V_i|Pa(V_i)) \qquad (2)$$

where $V_1, \ldots, V_n$ are all vertexes in $Bel(\mathcal{O})$.

Proposition 3: Given a consistent ideal ontology $\mathcal{O}$, its BelNet structure $\mathcal{G}$, and an observation $\mathbf{o}$, we have $P(\mathcal{G}^{\mathbf{o}}) = \prod_{Pa(V_i)=\emptyset} P(Vi^{\mathbf{o}})$, if $\mathcal{O}$ is ABox materialised.
Proposition 3 follows from Lemma 1 and Equation (2). As a direct conclusion from the JPD function, we have that

Theorem 1: For an interpretation $\mathcal{I}$ satisfies a consistent ideal ontology $\mathcal{O}$ and its BelNet structure $\mathcal{G}$, we have $P(\mathcal{G}^{\mathcal{I}}) = \prod_{Pa(V_i)=\emptyset} P(V_i^v)^k$ if O is ABox materialised, in where k is the total number of observation $\mathbf{o}$ in $\mathcal{I}$ such that $o \in V_i^{\mathcal{I}}$ for $v = \mathtt{T}$ and $o \notin V_i^{\mathcal{I}}$ for $v = \mathtt{F}$.

Proof:

$$\begin{aligned} P(\mathcal{G}^{\mathcal{I}}) &= \prod_{\mathbf{o}} P(\mathcal{G}^{\mathbf{o}}) \\ &= \prod_{\mathbf{o}} \prod_{Pa(V_i)=\emptyset} P(V_i^{\mathbf{o}}) \\ &= \prod_{Pa(V_i)=\emptyset} P(V_i^v)^k \end{aligned}$$

■

Continuing to Example 2 in Figure 2, we have $P(\mathcal{G}^{\mathcal{I}}) = 0.5 \times 0.5 \times 0.5 \times 0.5$

By now we have introduced the BelNet model for an ontology, and intensively studied the features of BelNet for an ABox materialised ontology which having rich ABox assertions. In the next section we will introduce how to learn the BelNet structure from the evidences in ontology ABox.

IV. Learning with BelNet

The learning approach includes 3 main steps:

1) *Pre-processing*. In pre-processing, given an ontology $\mathcal{O}$, for each $A \in N_C$ and $r \in N_R$, pre-processing creates nodes corresponding to A and $\exists r.\top$ in $Bel(\mathcal{O})$. In addition, because all individuals are belonged to concept $\forall r.A$, we generate $\exists r.A \sqcap \forall r.A$ instead as the approximation for $\forall r.A$. Furthermore, ABox materialization will carry out on each newly created concepts. The result of this step is denoted as *ABox and TBox enriched ontology*.
2) *Structure learning*. The algorithm adopted here is a modified version of the structure learning algorithm in Bayesian networks. Generally speaking, the Bayesian network structure learning algorithm can only recover the structure that is equivalent in terms of representing the independencies among the nodes to the real underlying structure [9]. In this paper, the preference is a single structure that is concise and can directly be used to extract $\mathcal{ALC}$ axioms. To achieve this goal, we incorporate this preference as shown in Algorithm 1.
3) *Post-processing*. After structure learning, a Bayesian network $\mathcal{G}$ is learned, and the parameters of $\mathcal{G}$ are estimated through the Bayesian estimator using $(0.5, 0.5)$ β priors. Using $\mathcal{G}$, in addition to axioms extracted directly from $\mathcal{G}$, more TBox axioms can be extracted through answering probabilisty queries by inferencing in $\mathcal{G}$.

We have discussed the first step in previous sections. In the following, we will further discuss the last two steps.

A. TBox Structure Learning in BelNet

Given an ABox and TBox enriched ontology $\mathcal{O} =< T, A >$, the task of *TBox structure learning* in BelNet is to find a BelNet graph $\mathcal{G} =< V, E >$, such that $\prod_{\mathcal{I} \in A} P(\mathcal{G}^{\mathcal{I}})$ is maximized, under the constraint that each link in $\mathcal{G}$ corresponds to a subsumption dependency relation.

Roughly speaking, the structure learning algorithm starts from an initial structure (with all the vertexes from preprocessing, and the conditional links between vertexes and the nearest parents), and iteratively tries to find the best operation (in terms of adding / deleting) that can be carried out from the current structure. This process iterates until no better structures can be found, or the step reaches the maximum threshold (c.f. Algorithm 1). Two parameters are involved in this procedure. One parameter controls the maximum number of parent nodes a node can have, the other parameter controls the maximum number of iterations for this procedure to exit.

Score function. In the Bayesian network structure learning algorithms, the vital part is evaluating an operation, a.k.a. adding or deleting a link. This is done by score functions. The score functions used in Bayesian network structure learning include maximum likelihood measure, Bayesian score, and extensions of Bayesian score. Likelihood measure suffers from over-fitting problems, and prefers complexer network to a simpler one, which is not always the real preference in practice. Due to the better performance in handling over-fitting problems of Bayesian score [9], we will adopt Bayesian score as our score function.

In an ontological knowledge base, $C \sqsubseteq D$ and $D \sqsubseteq E$ induces $C \sqsubseteq E$, in another word, $C \sqsubseteq E$ is a redundant axiom if it occurs together with the other two axioms. In the context of ontology learning, a learner is also expected to generate less redundancies. When using Bayesian score to add links, it prefers to link two nodes that are much closer to each other [9]. Generally speaking, in our approach, the links in the network structure correspond to subsumption relationships. Thus, with Bayesian score it is less likely to generate this kind of redundancies.

```
Algorithm 1: structure learning in BelNet
input : BelNet graph $\mathcal{G} =< V, E_{conditional} >$,
        $\mathcal{M} =< C, Inst_C >$, $max\_iter$
output: $\mathcal{G}'$
begin
    Initialize $best\_score$ with the score of $\mathcal{G}$;
    for each pair of nodes do
        cache the score for adding/deleting the link
        between them;
    while $max\_iter$ not reached do
        while best operation not found and cache not
        fully visited do
            $o \leftarrow$ the best operation from the cache;
            if $o$ satisfies the selection criteria then
                best operation found;
        if best operation found and
        $new\_score \geq best\_score$ then do operation $o$,
        and label the network as $\mathcal{G}'$;
        $best\_score \leftarrow$ score of $\mathcal{G}'$;
        else return $\mathcal{G}'$;
end
```

```
Algorithm 2: Post-processing in BelNet
input : BelNet graph $\mathcal{G} =< V, E >$, JPD,
        $threshold_{disjoint}$, $\mathcal{O}$
output: $\mathcal{O}'$
begin
    Initialize an empty $axiomlist$ and an ontology $\mathcal{O}'$;
    for each node who has more than one parent do
        for any combination of two parent nodes $V_i$, $V_j$
        do
            if $P(V_i^T, V_j^T) < threshold_{disjoint}$ then
                add $(< V_i,$ disjointWith $V_j >$
                $, P(V_i^T, V_j^T)) \rightarrow axiomlist$;
    sort $axiomlist$ ASC according to the probability;
    for each element in $axiomlist$ do
        if adding $axiom$ to $\mathcal{O}'$ not causing
        inconsistency then
            add $axiom \rightarrow \mathcal{O}'$;
    return $\mathcal{O}'$;
end
```

Selection criteria. After an operation is selected by the score function, in order to meet the demand of BelNet, to be specific, the preference over structures whose links signifying the special dependency called 'subsumption', the operations not satisfying the demands are filtered out by the selection criteria.

We denote the candidate operation as O, where O_{head} is the node to which the link points to, and O_{tail} represents the node from which the link starts. Further, we denote the count of instances that belongs to both concepts corresponding to O_{tail} and O_{head} as $M[O_{head}, O_{tail}]$, the count of instances belonging to concept O_{head} as $M[O_{head}]$, similar for $M[O_{tail}]$. Then, operation O will be selected iff $M[O_{head}, O_{tail}] = M[O_{tail}]$ and $M[O_{tail}] > threshold_{parent}$. In this paper, the $threshold_{parent}$ is selected to be 0.

Table II
PROBABILITIES IN BELNET AND THE CORRESPONDING DL AXIOMS.

	probabilities of DL axioms
conjunction	$P(\sqcap_{i=1}^{n} C_i) = P(C_1^T, \ldots, C_n^T)$
disjunction	$P(\sqcup_{i=1}^{n} C_i) = 1 - P(C_1^F, \ldots, C_n^F)$
disjointness	$P(\sqcap_{i=1}^{n} C_i \sqsubseteq \bot) = 1 - P(C_1^T, \ldots, C_n^T)$
subsumption	$P(C \sqsubseteq D) = P(D^T \vert C^T)$

B. Post-processing

After the structure of BelNet is learnt, we can extract various kinds of axioms from BelNet by inferencing in it. (refer to Table II for the details of this translation). The result probabilities of CPD query are the weights of the corresponding DL axioms. In practice, in order to select the axioms from the axioms with probabilities, we choose different threshold for this selection. For example, to select the disjointness axioms, we choose the axioms with probability greater than $1 - threshold_{disjoint}$. After the BelNet has been learned, post-processing extracts subsumptions and disjointness from the BelNet by the following procedure:

(1) For each non-conditional link $C \rightarrow D$ in the BelNet, generate an axiom $C \sqsubseteq D$. If there are more than one nodes $D_1 \ldots D_n$, such that $C \rightarrow D_i, i \in \{1, \ldots, n\}$, generate an axiom $C \sqsubseteq \sqcap_{i \in \{1,\ldots,n\}} D_i$.
(2) If there are more than one nodes $C_1 \ldots C_n$, such that $C_i \rightarrow D, i \in \{1, \ldots, n\}$, generate an axiom $\sqcup_{i \in \{1,\ldots,n\}} C_i \sqsubseteq D$.
(3) Generate disjointness axioms by Algorithm 2.

V. EXPERIMENTS

We designed and carried out some experiments to evaluate our proposed ontology learning approach in the following aspects: 1) We evaluate the proposed approach by checking the correctness of the axioms learnt by BelNet, and whether the axioms in the input ontology can be learned. 2) We analyse the performance of BelNet under the existence of incomplete semantic web data.

A. Experiment Setup

1) Evaluation Metrics: Given $\mathcal{O}$ the original ontology (as the input of a target ontology learner), $\mathcal{O}'$ the output of the ontology learner with input $\mathcal{O}$, and $\mathcal{O}^{\mathcal{S}}$ the gold-standard ontology, *precision* and *recall* can be calculated as follows:

$$Precision(\mathcal{O}^{\mathcal{S}}, \mathcal{O}') = \frac{|\{\alpha | \alpha \in \mathcal{O}' \text{ and } \mathcal{O}^{\mathcal{S}} \models \alpha\}|}{|\{\alpha | \alpha \in \mathcal{O}'\}|}$$

Table III
STATISTICS OF THE DATA SETS FOR EVALUATION. THE DATASET NAME END WITH ' IS THE GOLD-STANDARD DATASET.

ontology	# c	# op	# $\sqsubseteq$ / $\equiv$ / $\perp$	# ind	DL expressivity
Family	19	4	27 / 0 / 0	202	$\mathcal{AL}$
Family'	19	4	27 / 17 / 14	202	$\mathcal{ALC}$
Semantic Bible	49	29	51 / 0 / 5	724	$\mathcal{SHOIN}(\mathcal{D})$
Semantic Bible'	49	29	52 / 6 / 34	724	$\mathcal{SHOIN}(\mathcal{D})$
LUBM	43	25	36 / 6 / 0	1555	$\mathcal{ALEHI}(\mathcal{D})$
LUBM'	43	25	36 / 6 / 52	1555	$\mathcal{SHI}(\mathcal{D})$
Financial	60	16	55 / 0 / 113	17941	$\mathcal{ALCOIF}$

$$Recall(\mathcal{O}^{\mathcal{S}}, \mathcal{O}') = \frac{|\{\alpha|\alpha \in \mathcal{O}^{\mathcal{S}} \text{ and } \mathcal{O}' \models \alpha\}|}{|\{\alpha|\alpha \in \mathcal{O}^{\mathcal{S}}\}|}$$

where α is a subsumption or disjointness axiom. *F1-measure* is the harmonic mean of precision and recall.

2) Datasets: The test ontologies include: family[1], semantic bible[2], LUBM[3], and financial[4]. The evaluation metrics are calculated with the manually constructed gold-standard ontologies[5] and the original financial ontology, which is already complete enough for the evaluations. The related statistics of the datasets are shown in Table III where we calculate the number of concepts (c for short), object properties (op for short), number of subclassOf, equivalentClass, disjointWith axioms, number of individuals and the DL expressivity. The DL expressivity of the ontologies chosen are not restricted to $\mathcal{ALC}$. In the proposed approach, all concept expressions in the original ontology are treated as concepts, and if they exceed the expressivity of $\mathcal{ALC}$, they will be treated the same way as a named concept.

The experiments were performed on a computer with 4 core 2.27GHz CPU, and 4G RAM. We evaluated the performance of our approach under the existence of incompleteness by randomly partitioning the dataset into 10 parts. Each time of the experiment, we randomly selected one part from the partitions, and to which we add another randomly selected partition at the second time. At last, we get the whole dataset, which is the complete one. This procedure was carried out 10 times in order to demonstrate the objectiveness of the evaluation.

B. Results

In the structure learning algorithm, we selected the maximum number of parents to be 5, and the maximum number of iterations is 100. We ran experiments varying the $threshold_{disjoint}$ parameter in the range of $[0.01 - 0.20]$ and by checking the evaluation metrics we fix this parameter $threshold_{disjoint}$. Due to limited space, we only show the $F1$ results versus parameter and the size of the partition on dataset family (c.f. Figure 3, page 7).

We can observe from Figure 3 that the overall performance goes higher when the dataset gets larger. Interestingly, we can still see a minor drop after we get 30 percent of the whole dataset. Figure 4 (page 7) explains the reason: as the dataset is getting larger, the learner starts to extract disjointness axioms more carefully; thus, the recall gets lower. We will select the $threshold_{disjoint}$ by the following criterion: 1) we expect a large area of better performance in terms of $F1$, and 2) we expect a stable performance in the whole range of evaluations. Thus, in the experiments, we set the $threshold_{disjoint}$ to be 0.16.

Figure 4 (page 7) presents the performance of the proposed approach in terms of precision, recall, and F1-measure compared with GoldMiner and DLLearner. GoldMiner consists of 4 tunable parameters, namely supportness and confidence in learning subsumptions and disjointness seperately. We tried parameters in the scope of $[0 - 1]$ for GoldMiner, and finally we chose the support threshold to be 0, and confidence threshold to be 0.9 in learning subsumptions, and 0.1 (supportness), 0.8 (confidence) in learning disjointness, which is also the setting recommended in [4], in order to get a higher $F1$. From the figure, we conclude that 1) for most of the dataset, our method is better than DLLearner and GoldMiner in terms of precision and F1-measure. 2) the recall is not high compared with precision.

Table IV (page 7) shows a comparison of a snippet of the results of BelNet and DLLearner when the size of the dataset changes. It shows the over-fitting problem of DLLearner when the dataset is not complete enough. Most of the presented axioms (except the 6th and 10th) learnt by DLLearner are incorrect; e.g., when DLLearner runs on 10 percent of the whole dataset, the concept description of $Grandson$ is a $Male$ who has a Parent that is not a $Person$. The axioms that BelNet learns are all correct.

VI. RELATED WORK

Learning schemas from instance-level data has attracted attentions since the fast development of the semantic web. Interested readers can refer to [3] for a thorough survey [3]. In this section, we only notice a subset of works that focus on learning a broader sense of axioms from ABox data here. Due to the relationship between BelNet and statistical relational learning, important and closely related works on SRL models are also briefly reviewed in this section.

In [11], the authors developed *DLLearner* to learn $\mathcal{ALC}$ *cocnept descriptions* from ontologies based on ILP techniques, where the candidate concept descriptions are generated by a downward refinement operator.In addition, in [6], the particular focused is handling larger datasets, such as DBpedia. DLLearner generates concept descriptions quite well when the data quality is relatively high. However, under the existence of incompleteness, which is the main focus of this paper, DLLearner would drop into local optimum descriptions for concepts due to the incorrect 'false' values

[1] https://github.com/fresheye/belnet/blob/master/ontology/family-benchmark_rich_background.owl

[2] http://www.semanticbible.com

[3] http://swat.cse.lehigh.edu/projects/lubm/

[4] http://www.cs.put.poznan.pl/alawrynowicz/financial.owl

[5] https://github.com/fresheye/belnet/blob/master/ontology/

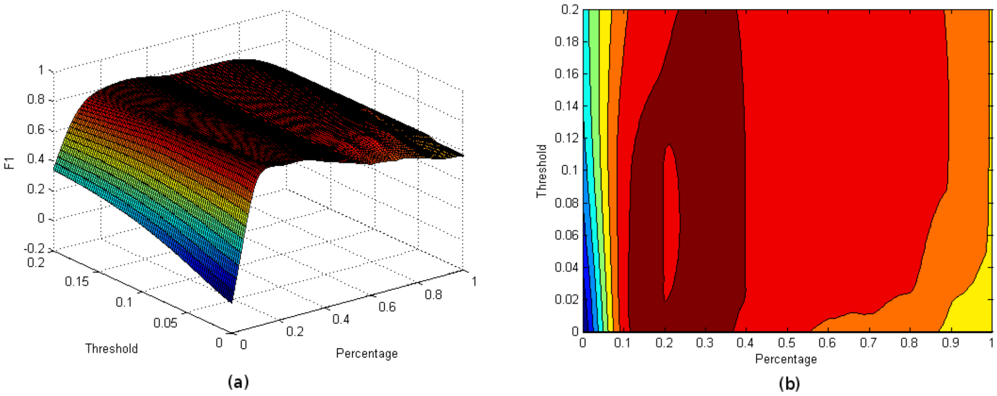


Figure 3. The F1 versus $threshold_{disjoint}$ and partition size on dataset Family. (a) demonstrates F1 in terms of $threshold_{disjoint}$ and the size of the dataset. (b) represents the contour of subgraph (a).

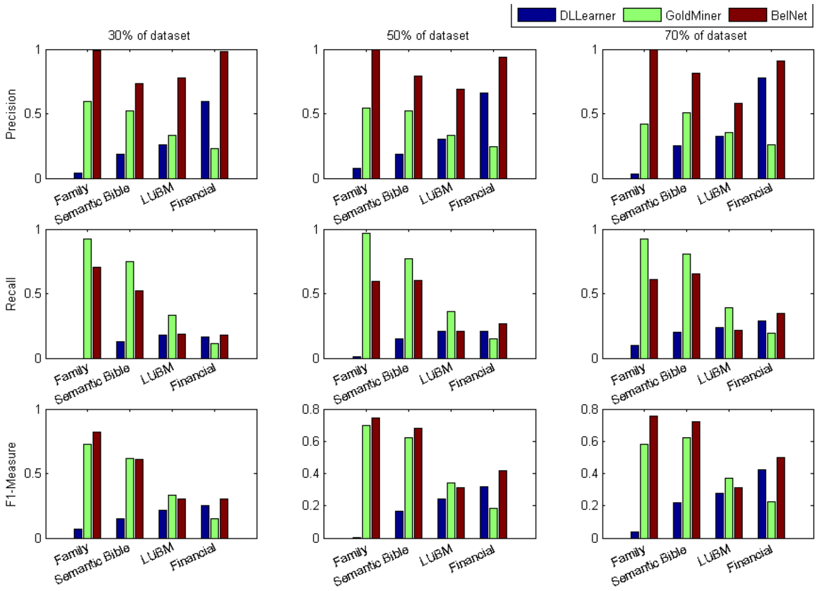


Figure 4. The Precision, Recall, and F1-Measure of DLLearner, GoldMiner, and BelNet, in terms of the size of the data.

Table IV
AXIOMS LEARNED FOR CONCEPT $Grandson$ ON FAMILY DATASET. THE FIRST COLUMN IS THE SIZE OF THE PARTITION IN THE EVALUATION.

%	BelNet ($Grandson \sqsubseteq$)	DLLearner ($Grandson \equiv$)
10	$Male \sqcap Grandchild \sqcap$ $\exists hasParent.\top \sqcap Child \sqcap \neg GrandDaughter$	$Male \sqcap \exists hasParent.\neg Person$
20	$Male \sqcap Grandchild \sqcap \neg GrandDaughter$	$(Male \sqcap \neg Parent) \sqcup \neg Person$
30	$Male \sqcap Grandchild \sqcap \neg GrandDaughter$	$(\neg Female \sqcap \neg Parent) \sqcap \forall hasChild.Mother$
40	$Male \sqcap Grandchild \sqcap \neg GrandDaughter$	$\neg Female \sqcap \neg Grandparent \sqcap \forall hasSibling.Child$
50	$Male \sqcap Grandchild \sqcap \neg GrandDaughter$	$\neg Female \sqcap \forall hasChild.(Child \sqcap \neg Parent)$
60	$Male \sqcap Grandchild \sqcap \neg GrandDaughter$	$\neg Female \sqcap \forall married.\forall married.Son$
70	$Male \sqcap Grandchild \sqcap \neg GrandDaughter$	$Person \sqcap \neg Female \sqcap \forall married.\forall hasParent.Sister$
80	$Grandchild \sqcap \neg GrandDaughter$	$\neg Female \sqcap \forall married.\forall hasParent.Brother$
90	$Male \sqcap Grandchild \sqcap \neg GrandDaughter$	$\neg Female \sqcap \exists hasParent. \leq 1 hasChild.GrandParent$
100	$Male \sqcap Grandchild \sqcap Son \sqcap \neg GrandDaughter$	$Son \sqcap \exists hasParent.Child$

generated by making CWA. *Gold-Miner* [18] tries to learn $\mathcal{EL}$ axioms from ontologies based on association rule mining method, and in [4], this approach is further extended to learn disjointness axioms. However, Gold-Miner tends to learn a large number of irrelevant results, which put an extra burden to end-users of ontology learning applications. In addition, Galárraga et al. [5] proposed a *rule* mining model supporting OWA scenario by introducing a new confidence measure in association rule mining.

We briefly review the SRL methods closely related to BelNet. Koller et al. extended DL CLASSIC with nodes in a BN represent probabilistic information of the individuals in a specific class [10], which is closely related to the representation in BelNet. However, in BelNet, the edges correspond to the specific type of dependency (subsumption). BLP [8] unifies definite logic programs with Bayesian networks. In BLP, ground atoms are mapped to random variables. BelNet differs from BLP in that 1) the representation languages are different; 2) BelNet models concepts with random variables; 3) BelNet is suitable for schema level ontology learning. OntoBayes [19] extends OWL with annotating RDF triples with probabilities and dependencies. In [12], $\mathcal{EL}^{++}$-LL was proposed to extend crisp ontological axioms with weights. Using $\mathcal{EL}^{++}$-LL, a subset of coherent axioms can be learned from a set of *weighted* $\mathcal{EL}^{++}$ axioms.

VII. Conclusion and Future Work

In this paper, we proposed Bayesian description logic Network (BelNet) to deal with the problem of schema learning from incomplete semantic web data. In BelNet, DL concept expressions correspond to probabilistic nodes, and subsumption relationships between DL concept descriptions are represented as links. The problem of learning schema is transformed into structure learning and query answering in BelNet, which, from our experiment, has been shown to be effective for learning from incomplete semantic data.

In the future, we will explore 1) learning equivalence axioms with BelNet; 2) learning axioms in other DL species, and 3) scalable solutions of BelNet on larger datasets. We also plan to investigate the combination of learning algorithms with reasoning engines such as TrOWL [17].

VIII. Acknowledgements

This work is partially funded by the National Science Foundation of China under grant 61170165, the EU IAPP K-Drive project (286348) and the EPSRC WhatIf project (EP/J014354/1).

References

[1] C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.

[2] P. Cimiano. *Ontology learning and population from text: algorithms, evaluation and applications*. 2006.

[3] C. d'Amato, N. Fanizzi, and F. Esposito. Inductive learning for the semantic web: What does it buy? *Semantic Web*, 1(1-2):53–59, 2010.

[4] D. Fleischhacker and J. Völker. Inductive learning of disjointness axioms. In *On the Move to Meaningful Internet Systems: OTM 2011*, pages 680–697. Springer, 2011.

[5] L. Galárraga, C. Teflioudi, K. Hose, and F. M. Suchanek. Amie: Association rule mining under incomplete evidence in ontological knowledge bases. In *Proc. of WWW2013*, 2013.

[6] S. Hellmann, J. Lehmann, and S. Auer. Learning of owl class descriptions on very large knowledge bases. *IJSWIS*, 5(2):25–48, 2009.

[7] I. Horrocks and P. F. Patel-Schneider. KR and reasoning on the semantic web: OWL. In *Handbook of Semantic Web Technologies*, pages 365–398. Springer, 2011.

[8] K. Kersting and L. De Raedt. Towards combining inductive logic programming with bayesian networks. In *Inductive Logic Programming*, pages 118–131. Springer, 2001.

[9] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

[10] D. Koller, A. Levy, and A. Pfeffer. P-classic: A tractable probablistic description logic. In *Proceedings of the National Conference on Artificial Intelligence*, pages 390–397, 1997.

[11] J. Lehmann and P. Hitzler. Concept learning in description logics using refinement operators. *Machine Learning*, 78:203–250, 2010.

[12] M. Niepert, J. Noessner, and H. Stuckenschmidt. Log-linear description logics. In *IJCAI*, pages 2153–2158, 2011.

[13] J. Z. Pan, Y. Ren, H. Wu, and M. Zhu. Query generation for semantic datasets. In *Proc. of KCAP 2013*, 2013.

[14] J. Z. Pan, E. Thomas, Y. Ren, and S. Taylor. Tractable Fuzzy and Crisp Reasoning in Ontology Applications. In *IEEE Computational Intelligence Magazine*, 2012.

[15] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[16] A. Rettinger, U. Lösch, V. Tresp, C. d'Amato, and N. Fanizzi. Mining the semantic web. *Data Mining and Knowledge Discovery*, pages 1–50, 2012.

[17] E. Thomas, J. Z. Pan, and Y. Ren. TrOWL: Tractable OWL 2 Reasoning Infrastructure. In *the Proc. of the Extended Semantic Web Conference (ESWC2010)*, 2010.

[18] J. Völker and M. Niepert. Statistical schema induction. In *Proc. of ESWC'2011*, pages 124–138, 2011.

[19] Y. Yang and J. Calmet. Ontobayes: An ontology-driven uncertainty model. In *Proc. of CIMCA2005*, volume 1, pages 457–463, 2005.

[20] M. Zhu. Dc proposal: ontology learning from noisy linked data. *The Semantic Web–ISWC 2011*, pages 373–380, 2011.