Online Inference-Rule Learning from Natural-Language Extractions

Sindhu Raghavan and **Raymond J. Mooney**
Department of Computer Science
The University of Texas at Austin
Austin, TX 78712, U.S.A
{sindhu,mooney}@cs.utexas.edu

Abstract

In this paper, we consider the problem of learning commonsense knowledge in the form of first-order rules from incomplete and noisy natural-language extractions produced by an off-the-shelf information extraction (IE) system. Much of the information conveyed in text must be inferred from what is explicitly stated since easily inferable facts are rarely mentioned. The proposed rule learner accounts for this phenomenon by learning rules in which the body of the rule contains relations that are usually explicitly stated, while the head employs a less-frequently mentioned relation that is easily inferred. The rule learner processes training examples in an online manner to allow it to scale to large text corpora. Furthermore, we propose a novel approach to weighting rules using a curated lexical ontology like WordNet. The learned rules along with their parameters are then used to infer implicit information using a Bayesian Logic Program. Experimental evaluation on a machine reading testbed demonstrates the efficacy of the proposed methods.

Introduction

Much of the information conveyed in text must be inferred from what is explicitly stated since easily inferable facts are rarely mentioned. This was first discussed in detail by Grice (1975), who postulated the maxims of quantity, quality, manner, and relation that characterize natural-language communication. The maxim of quantity refers to the concise nature of natural language that leaves much implicit information unstated. Human readers read between the lines and infer such information from explicitly stated facts using commonsense knowledge. However, automated information extraction (IE) systems (Cowie and Lehnert 1996; Sarawagi 2008), which are trained to extract explicitly-stated information, are limited in their ability to extract implicitly stated facts. These systems do not have access to commonsense knowledge, and hence are incapable of performing deeper inference. However, answering many queries can require such inference. Consider the text "Iker Casilas captains the La Liga club Real Madrid and the Spanish national team." An IE system cannot answer the query, "Iker Casilas is a member of what team?", since the membership information is not explicitly stated. However, a human reader easily infers that the captain of a team is also a member of that team.

In this paper, we consider the problem of learning commonsense knowledge in the form of first-order rules from noisy extractions produced by an off-the-shelf IE system. These rules are then used to infer implicit information from explicitly stated facts (Carlson et al. 2010; Schoenmackers et al. 2010; Doppa et al. 2010; Sorower et al. 2011; Raghavan, Mooney, and Ku 2012). Most existing inference-rule learners(Quinlan 1990; Mccreath and Sharma 1998; Srinivasan 2001; Kersting and Raedt 2008) assume that the training data is largely accurate and complete, and hence are not adept at learning useful rules from noisy and incomplete IE output. Also, most of them do not scale to large corpora. Due to these limitations, we have developed an efficient online rule learner that handles the concise, incomplete nature of natural-language text.

The proposed rule learner learns probabilistic first-order definite-clause rules from IE extractions. The learned rules are such that the body of the rule typically consists of relations that are frequently explicitly stated, while the head is a relation that is more typically inferred. We use the frequency of occurrence of extracted relations as a heuristic for distinguishing those that are typically explicitly stated from the ones that are usually inferred. In order to allow scaling to large corpora, we develop an efficient online rule learner. For each training document, it constructs a *directed* graph of relation extractions, adding directed edges between nodes that share one or more constants. It then traverses this graph to learn first-order rules. Our approach is closest to that of Dinh et al. (2011), which constructs an *undirected* graph of first-order predicates and adds edges between nodes whose predicates share arguments of the same type. In our approach, the directionality of the edges helps discover rules for those relations that can be inferred from others that are usually explicitly stated. Further, by constructing a graph of ground relational literals instead of relational predicates, our learner can be used in domains that do not have a strictly tree-structured ontology. Typically, relations that accept arguments or constants belonging to multiple types are found in ontologies which have the structure of a DAG (directed acyclic graph) rather than a tree. Accommodating such an ontology is critical to handling the machine-reading corpus used in our experiments. The approach by Dinh et al. is not directly applicable to such domains since it relies on a unique type for each predicate argument.

After learning first-order rules, they are used to infer additional information from that explicitly extracted from future documents. Approaches to inference use either purely logical deduction, which fails to account for the uncertainty inherent in such rules, or a probabilistic logic such as Markov Logic Networks (MLNs) (Domingos and Lowd 2009) or Bayesian Logic Programs (BLPs) (Kersting and De Raedt 2007). Based on a recent study by Raghavan et al. (2012) that demonstrated the superior performance of BLPs compared to MLNs and logical deduction, we use BLPs to support inference.

Probabilistic inference in BLPs requires specifying conditional-probability parameters for the first-order rules. Since relations that are easily inferred from stated facts are seldom seen in the training data, learning useful parameters using conventional BLP-parameter-learning approaches like EM (Kersting and Raedt 2008) has had limited success (Raghavan, Mooney, and Ku 2012). Consequently, we propose an alternate approach to specifying parameters for the learned first-order rules using lexical knowledge from a curated ontology like WordNet (Fellbaum 1998). The basic idea behind our approach is that more accurate rules typically have predicates that are closely related to each other in terms of the meanings of the English words used to name them.

The main contributions of this paper are therefore:

- A novel online rule learner that efficiently learns accurate rules from noisy and incomplete natural-language extractions.
- A novel approach to weighting such rules using lexical information from a curated ontology like WordNet (Fellbaum 1998).

Bayesian Logic Programs

Bayesian logic programs (BLPs) (Kersting and De Raedt 2007; Kersting and Raedt 2008) can be viewed as templates for constructing *directed* graphical models (Bayes nets). Formally, a BLP consists of a set of *Bayesian clauses*, definite clauses of the form $a|a_1, a_2, a_3, ...a_n$, where $n \geq 0$ and a, a_1, a_2, a_3,......,a_n are *Bayesian predicates* (defined below), and where a is called the head of the clause (head(c)) and (a_1, a_2, a_3,....,a_n) is the body (body(c)). When $n = 0$, a Bayesian clause is a fact. Each Bayesian clause c is assumed to be universally quantified and range restricted, i.e $variables\{head\} \subseteq variables\{body\}$, and has an associated *conditional probability table* CPT(c) = P(head(c)|body(c)). A *Bayesian predicate* is a predicate with a finite domain, and each ground atom for a Bayesian predicate represents a random variable. Associated with each Bayesian predicate is a combining rule such as *noisy-or* or *noisy-and* that maps a finite set of CPTs into a single CPT.

Given a knowledge base as a BLP, standard logical inference (SLD resolution) is used to automatically construct a Bayes net for a given problem. More specifically, given a set of facts and a query, all possible definite-clause proofs of the query are constructed and used to build a Bayes net for answering that query. The probability of a joint assignment of truth values to the final set of ground propositions is defined as follows:

$$\text{P}(X) = \prod_i P(X_i|Pa(X_i)),$$

where $X = X_1, X_2, ..., X_n$ represents the set of random variables in the network and $Pa(X_i)$ represents the parents of X_i. Once a ground network is constructed, standard probabilistic inference methods can be used to answer various types of queries (Koller and Friedman 2009). BLP parameters can be learned using EM as described by Kersting and De Raedt (2008).

Online Rule Learner

In this section, we describe our online rule learner for inducing probabilistic first-order rules from the output of an off-the-shelf IE system. It involves constructing a *directed graph* of relation extractions for each training example and connecting those relations that share one or more constants with a directed edge. Nodes connected to one another by edges typically represent relations that might be related, and hence might participate in the same rule. The edges are added from relations that are usually explicitly stated in text to those that can be inferred. Since relations that are implicitly stated occur less frequently in the training data, we use the frequency of occurrence of relation predicates as a heuristic to determine if a particular relation is best inferred from other relations. While the frequency of occurrence of relations is a good heuristic, this assumption does not necessarily hold for all relations.

The pseudocode for our Online Rule Learner (ORL) is shown in Algorithm 1. It accepts a set of training examples, where each example consists of a set of facts an IE system has extracted from a single document. The learner processes one example at a time in an online manner as follows. First, it updates counts for the frequency of occurrence for each relational predicate seen in the training example. Then, it builds a directed graph whose nodes represent relation extractions seen in the example. Note that entity types are not added to the graph. The rule learner then adds *directed* edges between every pair of nodes whose relations share one or more constants as arguments. The direction of the edge is determined as follows – for every pair (x,y) of relations that share constants, if the relation predicate of x is seen more frequently than that of y in the training set so far, then the learner adds a directed edge from x to y since y is more likely to be inferred from x.

Once all documents are processed, the learner traverses the resulting graph to construct rules. For each directed edge (x,y) in the graph, it constructs a rule in which the body contains x and y is the head. It then adds types corresponding to the constants in x. If a constant is associated with multiple types, i.e if the extractor has extracted multiple types for

Algorithm 1 Online Rule Learner

```
Inputs: Training examples $D$, target predicates $T$, and number of rules to output per predicate $n$. Each example $D_i$ consists of a set of extractions.
Output: First-order definite-clause rules $R$ for target predicates $T$.
for each example $D_i$ do
    for each extraction $x$ in $D_i$ do
        Get the predicate $R_x$ for $x$
        if $R_x$ is a relation predicate then
            Increment count for $R_x$
        end if
    end for
    Construct a directed graph $G_i$ in which relation extractions are nodes.
    for each pair of relations $x$ and $y$ that share one or more constants do
        Let $R_x$ be the predicate of $x$ and $R_y$ be the predicate of $y$
        if count of $R_y <$ count of $R_x$ then
            Add an edge from $x$ to $y$
        end if
    end for
    for each relation $x$ do
        for each outgoing edge ($x$,$y$) from $x$ do
            Let $y$ be the head node of the edge
            Create a rule $R_j$ $x \to y$
            for each constant $c_k$ in $x$ do
                Add the type corresponding to $c_k$ to the body of $R_j$
            end for
            Replace all constants in $R_j$ with unique variables to create a first-order rule $FR_j$
            if $FR_j$ is range restricted then
                Add $FR_j$ to $R$ and update the support for $FR_j$
            end if
        end for
    end for
end for
Sort rules in the descending order of their support and output top $n$ rules for each predicate.
```

(a) ***Example text in training***

"Barack Obama is the 44th and the current President of USA. Obama, citizen of USA was born on August 4, 1961 in Hawaii, USA."

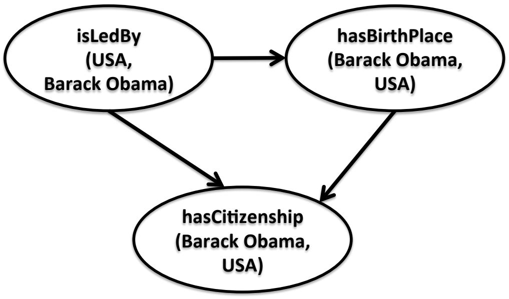
(b) ***IE extractions***

nationState(usa)
person(barack obama)
isLedBy(usa,barack obama)
hasBirthPlace(barack obama,usa)
hasCitizenship(barack obama,usa)

(c)***Frequency counts for relation predicates***

isLedBy: 30, hasBirthPlace: 23, hasCitizenship: 20

(d) ***Directed graph constructed from extracted relations***


isLedBy (USA, Barack Obama)
hasBirthPlace (Barack Obama, USA)
hasCitizenship (Barack Obama, USA)

(e) ***Ground rules constructed by ORL***

isLedBy(usa,barack obama) ∧ person(barack obama) ∧ nationState(usa)
→ hasBirthPlace(barack obama,usa)
isLedBy(usa,barack obama) ∧ person(barack obama) ∧ nationState(usa)
→ hasCitizenship(barack obama,usa)
hasBirthPlace(barack obama,usa) ∧ person(barack obama) ∧ nationState(usa)
→ hasCitizenship(barack obama,usa)

(f) ***First order rules constructed by ORL***

isLedBy(X,Y) ∧ person(Y) ∧ nationState(X) → hasBirthPlace(Y,X)
isLedBy(X,Y) ∧ person(Y) ∧ nationState(X) → hasCitizenship(Y,X)
hasBirthPlace(X,Y) ∧ person(X) ∧ nationState(Y) → hasCitizenship(X,Y)

Figure 1: Example Showing Various Stages of the ORL Algorithm

a constant, then we create a separate rule for each type extracted for the constant. This is needed in domains that have a DAG-structured ontology as described earlier. Finally, it replaces all constants in the rule with unique variables to create a first-order rule. All first-order rules that are range restricted (all variables in the head appear in the body) are retained and the remaining rules are discarded.

The training phase ends when the rule learner has processed all examples in the training set. It then outputs the top n rules per predicate, where n is a value provided by the user. The rules are sorted in descending order of their *support*, which refers to to the number of times the rule is satisfied in the training set (Agrawal, Imieliński, and Swami 1993). Alternately, the rule learner could output only those rules whose support meets a user-specified threshold.

In the basic algorithm, we have considered rules in which the body of the rule has a *single* relational literal. However, we can extend the algorithm in several ways to search for rules that have several relational literals in their body. For instance, given two rules $A \to B$ and $C \to B$, the rule learner can propose a new rule $A \wedge C \to B$. An alternate approach is to follow the directed edges for a given path length and add all relations except that corresponding to the end node in the path to the rule body and make the relation corresponding to the end node the head. The basic algorithm worked best for our application domain and hence we learned rules with just a single relation in the rule body.

In some ways, the rules learned by this approach are similar to the typed entailment rules considered by Berant et al. (2011). However, as described above, unlike their approach, our method is not limited to learning rules with a single relation in the body. Furthermore, approaches that learn typed entailment rules like Berant et al.'s do not handle DAG ontologies in which relations can take arguments of multiple types. On the other hand, our method explicitly handles this situation.

Consider the example shown in the Figure 1. Figure 1a shows a sample training document and Figure 1b shows the corresponding IE output. Given these extractions and the frequency counts for relational predicates seen so far in training (Figure 1c), ORL constructs a directed graph with relation extractions *isLedBy(usa,barack obama)*, *hasBirthPlace(barack obama,usa)*, and *hasCitizenship(barack obama,usa)* as nodes (Line 8 in Algorithm 1). It then adds directed edges between nodes that share constants *barack obama* and *usa*. The direction is determined by the frequency counts of the relations *isLedBy*, *hasCitizenship*, and *hasBirthPlace* as described in Lines 9–14 in Algorithm 1. After constructing the graph, ORL constructs rules as described in Lines 16–21. Finally, it replaces the constants *barack obama* and *usa* with variables to construct first-order rules (Lines 22). Since all three rules are range restricted, they are all kept.

Weighting Rules with WordNet

We now discuss our new approach to determining parameters for the learned first-order rules. Our approach learns weights between 0 and 1, where a higher weight represents higher confidence. These weights are used as noisy-or parameters when performing probabilistic inference in the resulting BLP (Kersting and Raedt 2008; Raghavan, Mooney, and Ku 2012). Since the predicate names in most ontologies employ ordinary English words, we hypothesized that more confident rules have predicates whose words are more semantically related. We use the lexical information in WordNet (Fellbaum 1998) to measure word similarity.

WordNet is a lexical knowledge base covering around 130,000 English words in which nouns, verbs, adverbs, and adjectives are organized into synonym sets, also called *synsets*. We used the *wup* measure by Wu and Palmer (1994) as implemented in WordNet::Similarity (Pedersen, Patwardhan, and Michelizzi 2004) to measure semantic distances between words. This measure computes the depth of the least common subsumer (LCS) of the given words and then scales it by the sum of the depths of the given words. We used the wup similarity score since it computes a (scaled) similarity scores between 0 and 1, which are easily used as weights for our rules.

We compute the *wup* similarity for every pair of words (w_i,w_j) in a given rule, where w_i is a word in the body and w_j is a word in the head. The words in a given rule are the predicate names of relations and entity types, which are usually English words. However, for predicate names like hasCitizenship or hasMember that are not single English words, we segment the name into English words such as *has*, *citizenship*, and *member*, and then remove stop words. The final weight for a rule is the average similarity between all pairs (w_i,w_j), which basically measures how closely predicates in the body are related to the predicate in the head. Using the highest similarity score between all word pairs or between words from relation predicates only as the rule weight did not yield meaningful results.

employs	eventLocation
eventLocationGPE	hasMember
hasMemberPerson	isLedBy
mediatingAgent	thingPhysicallyDamaged
hasMemberHumanAgent	killingHumanAgent
hasBirthPlace	thingPhysicallyDestroyed
hasCitizenship	attendedSchool

Table 1: Target Relations Selected for Evaluation

Experimental Evaluation

Data

We evaluated our approaches to rule learning and weighting on DARPA's machine-reading intelligence-community (IC) data set, which consists of news articles on terrorist events around the world. Our specific data set consists of $10,000$ documents, each containing an average of 93.14 facts extracted by SIRE (Florian et al. 2004), an IE system developed by IBM.

The ontology provided by DARPA for the IC domain consists of 57 entity types and 79 relations. The entity types include Agent, PhysicalThing, Event, TimeLocation, Gender, and Group, each with several subtypes. The type hierarchy is a DAG rather than a tree, and several types have multiple super-classes. For instance, a *GeopoliticalEntity* can be a *HumanAgent* as well as a *Location*.

BLP Parameters and Inference

We used a deterministic *logical-and* model to encode the CPT entries for the Bayesian clauses and a *noisy-or* model to combine evidence from multiple rules that have the same head (Pearl 1988). As shown later, given the limited amount of training data available for each relation, learning noisy-or parameters using the EM algorithm developed for BLPs (Kersting and Raedt 2008) did not give very good results. Consequently, we manually set the noisy-or parameters for all rules to 0.9, since this approach has been shown to work well in the IC domain (Raghavan, Mooney, and Ku 2012). To infer implicit relations, we performed BLP inference using SampleSearch (Gogate and Dechter 2007) to compute marginal probabilities.

Evaluation Metric

We evaluated our approach by performing 10-fold cross validation on this data. We learned first-order rules using 14 target relations given in Table 1 that had an appreciable amount of data. To measure performance, we randomly sampled 4 documents from each test set, 40 documents in total. We manually evaluated the inferences for these test documents since there is no ground truth available for this data set. We ranked all inferences in descending order of their marginal probability and computed precision for the top n inferences. Precision measures the fraction of inferences that were judged correct. Our evaluation is similar to that used in previous related work (Carlson et al. 2010; Schoenmackers et al. 2010; Raghavan, Mooney, and Ku 2012).

Evaluation of Online Rule Learner

The systems compared in our experiments are:

isLedBy(B,A) ∧ person(A) ∧ nationState(B) → hasBirthPlace(A,B) (0.62) *If person A leads a nation B, then A is born in B*
thingPhysicallyDamaged(A,B) ∧ bombing(A) ∧ nationState(B) → eventLocation(A,B) (0.71) *If a nation B is physically damaged in a bombing event A,* *then the event location of A is B*
employs(A,B) ∧ humanOrganization(A) ∧ personGroup(B) → hasMemberHumanAgent(A,B) (0.57) *If a human organization A employs B, then B is a member of A*
isLedBy(B,A) ∧ nationState(B) → hasCitizenship(A,B) (0.48) *If a nation B is led by A, then A is a citizen of B*

Table 2: Sample Rules Learned by ORL

- ORL: This approach learns rules using the online rule learner described earlier. For each target relation, we specify the number of rules to output to be 10. Table 2 gives some sample rules learned by ORL along with rule weights computed using the approach described earlier.
- LIME: Raghavan et al. (2012) demonstrated that LIME (Mccreath and Sharma 1998), an existing Inductive Logic Programming (ILP) system, could learn accurate rules for the IC data set, and hence we use it as a baseline. Like Raghavan et al., we learned rules using only positive instances and using both positive and negative instances for each target relation, where negative instances were generated using the closed world assumption. The final BLP included rules learned from both settings.[1]
- COMBINED: This approach combines rules from both ORL and LIME into a single BLP.

We observed that all methods learn inaccurate rules for certain target relations like *mediatingAgent* and *attendedSchool* since they are less easily inferred compared to other relations such as *hasMember* that are more easily inferred. Therefore, we removed 4 relations – *mediatingAgent*, *attendedSchool*, *thingPhysicallyDamaged*, and *thingPhysicallyDestroyed* from the original set and also report results for the remaining 10 target relations. We refer to the original set of target relations as “Full-set” and the reduced set as “Subset”.

Figure 2 gives the precision at top-n for inferences made by rules learned using ORL, LIME, and COMBINED on both the Full-set and Subset of target relations. On the Full-set, LIME outperforms ORL at the initial points on the curve, while ORL outperforms LIME by a significant margin at later points on the curve. However, on the Subset, ORL outperforms LIME at all points on the curve. A closer examination of the rules learned by both approaches revealed that ORL learned rules that were more specific than LIME's. As a result, ORL makes fewer but more accurate inferences. On both Full-set and Subset, COMBINED outperforms both ORL and LIME for both target sets, indicating a definite advantage to combining rules from both LIME and ORL. Therefore, for the rest of the paper, we use the COMBINED model only. In general, performance of all models on the Subset is better than that on the Full-set since the difficult relations have been removed.

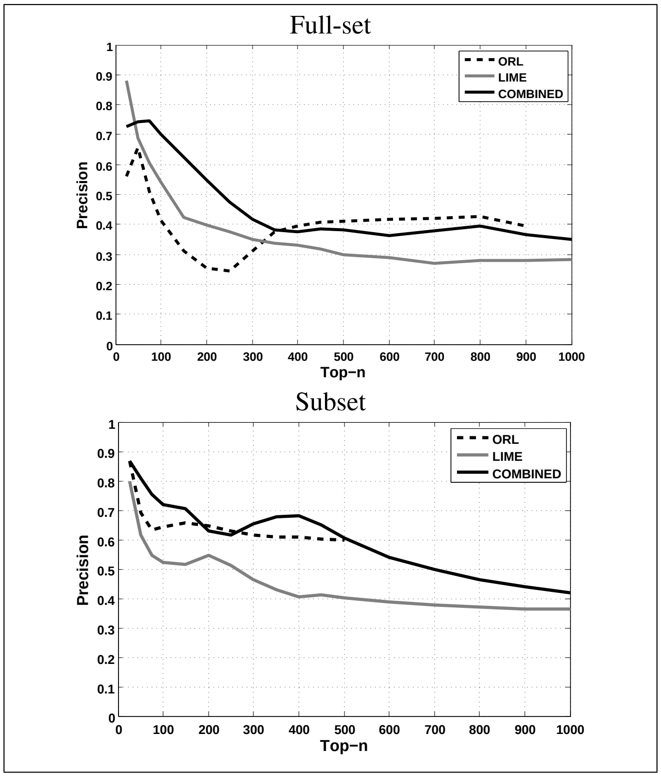


Figure 2: Precision at Top-n on the Full-set (top) and Subset (bottom) of Target Relations.

ORL takes 3.8 minutes per fold on an average to learn rules, while LIME takes 11.23 hours to learn rules. As discussed by Raghavan et al. (2012), since LIME does not scale to large data sets, we ran LIME on smaller subsets of the training data and combined the results in order to process the full IC data. The runtime includes the total time taken to produce the final set of rules. Unlike ORL, LIME learns first-order rules for each target predicate separately, further increasing its running time. On the other hand, ORL learns rules for all target predicates during one pass through the training data. As a result, ORL trains two orders of magnitude faster than LIME. The timing information empirically demonstrates ORL's ability to effectively scale to large data sets.

Evaluation of Weighting Rules with WordNet

We also computed noisy-or parameters using the lexical approach described earlier for all rules in the COMBINED model. For the baseline, we set all noisy-or parameters to 0.9 and performed inference in the BLP framework. We refer to the former as “WUP” and the latter as “Default”. For the Subset of target relations, we also compare to learning noisy-or parameters using EM (Kersting and Raedt 2008). We were unable to learn noisy-or parameters on the Full-set due to the large size of the networks, which made both

[1] Note that our results can not be directly compared to that of Raghavan et al.'s due to differences in the exact set of target predicates selected.

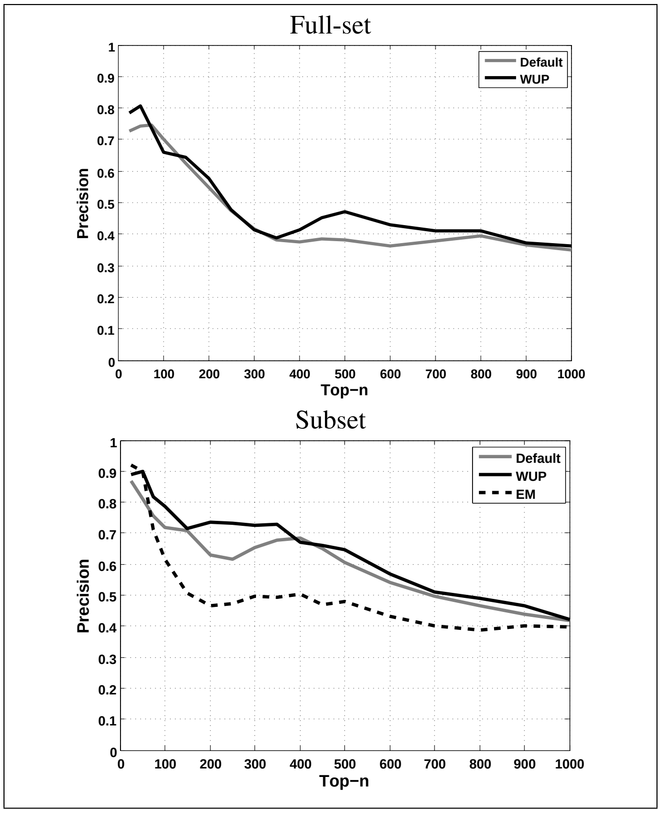


Figure 3: Precision at Top-n for COMBINED using Different Weights on the Full-set (top) and Subset (bottom) of Target Relations.

logical and probabilistic inference intractable on several examples.

Figure 3 gives the precision for inferences made with the COMBINED model using different weights on both the Full-set and Subset of target relations. On both sets, WUP outperforms Default indicating the efficacy of our approach for learning weights using WordNet. On the Subset, EM weights slightly outperform both Default and WUP weights for the top 25 inferences; however, the precision for EM weights drops significantly for higher values of n, indicating that the learned weights are not very good, possibly due to the lack of sufficient training data. When selecting an average of 1 inference per document, WUP gives a precision of 80.6% and 90% on the Full-set and Subset respectively.

Overall, our online rule learner produces accurate rules very efficiently. Even though it does not always outperform LIME, combining rules from both approaches results in a model that gives the best performance on the IC data set. Though weight learning using conventional parameter learning techniques like EM did not yield reasonable results, our approach to weighting rules using WordNet similarity shows significant promise.

Related Work

Approaches proposed in the literature for learning first-order rules generally fall into one of three categories. The first category consists of rule learners (Quinlan 1990; Mccreath and Sharma 1998; Kersting and Raedt 2008; Dinh, Exbrayat, and Vrain 2011; Kok and Domingos 2009; 2010) that expect the training data to be complete and accurate, due to which they are less suited for learning effective inference rules from noisy, incomplete IE extractions. The second category consists of rule learners (Schoenmackers et al. 2010; Sorower et al. 2011) that are specifically developed for learning probabilistic rules from IE extractions using the MLN framework. These approaches perform an exhaustive search, which becomes computationally intractable for larger datasets. The third category consists of approaches (Lin and Pantel 2001; Yates and Etzioni 2007; Berant, Dagan, and Goldberger 2011) that learn entailment rules that are typically restricted to having antecedents with a single literal, and have utilized binary logic rather than a well-founded probabilistic logic such as MLNs or BLPs, and therefore are unable to systematically integrate evidence from multiple rules to produce confidences in their inferences.

Basu et al. (2001) use WordNet distances to estimate the novelty of rules discovered by data-mining systems. Feldman and Dagan (1995) and Han and Fu (1995) use domain-specific concept hierarchies to weight and filter redundant rules. Garrette et al. (2011) use distributional lexical semantics to set weights for rules in an MLN.

Conclusions and Future Work

We have introduced a novel online learner for inducing first-order inference rules from noisy and incomplete extractions from an off-the-shelf IE system. Experimental comparisons to LIME, an existing rule-learner on the machine reading task has demonstrated that our approach generally learns rules of equal or greater accuracy in significantly less time. However, the best predictive accuracy was obtained when rules from both approaches were combined. We have also presented a novel approach to weighting inference rules using WordNet lexical similarity, demonstrating superior predictive accuracy compared to default parameters or those learned using EM.

Future work includes developing better techniques for learning BLP parameters in the presence of limited, noisy data and performing large scale evaluation of our approach on larger test sets by employing crowdsourcing via Amazon Mechanical Turk. Using lexical knowledge to initialize the weights and then using EM to improve the weights is another direction for future work. Finally, we would like to explore using distributional lexical semantics to compute word similarities for weighting rules (Garrette, Erk, and Mooney 2011).

Acknowledgements

This research was funded by MURI ARO grant W911NF-08-1-0242, Air Force Contract FA8750-09-C-0172 under the DARPA Machine Reading Program, and DARPA DEFT program under AFRL grant FA8750-13-2-0026. Experiments were run on the Mastodon Cluster provided by NSF grant EIA-0303609.

References

Agrawal, R.; Imieliński, T.; and Swami, A. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the SIGMOD 1993*, 207–216.

Basu, S.; Mooney, R. J.; Pasupuleti, K. V.; and Ghosh, J. 2001. Evaluating the novelty of text-mined rules using lexical knowledge. In *Proceedings KDD 2001*, 233–239.

Berant, J.; Dagan, I.; and Goldberger, J. 2011. Global learning of typed entailment rules. In *Proceedings of ACl-HLT 2011*, 610–619.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Jr., E. H.; and Mitchell, T. 2010. Toward an architecture for never-ending language learning. In *Proceedings of AAAI 2010*, 1306–1313. AAAI Press.

Cowie, J., and Lehnert, W. 1996. Information extraction. *CACM* 39(1):80–91.

Dinh, Q.-T.; Exbrayat, M.; and Vrain, C. 2011. Generative structure learning for Markov logic networks based on graph of predicates. In *Proceedings of IJCAI 2011*, IJCAI 2011, 1249–1254. AAAI Press.

Domingos, P., and Lowd, D. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. San Rafael, CA: Morgan & Claypool.

Doppa, J. R.; NasrEsfahani, M.; Sorower, M. S.; Dietterich, T. G.; Fern, X.; and Tadepalli, P. 2010. Towards learning rules from natural texts. In *Proceedings of FAM-LbR 2010*, 70–77. Stroudsburg, PA, USA: Association for Computational Linguistics.

Feldman, R., and Dagan, I. 1995. Knowledge discovery in textual databases (kdt). In *Proceedings of KDD 1995*, 112–117. AAAI Press.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: The MIT Press.

Florian, R.; Hassan, H.; Ittycheriah, A.; Jing, H.; Kambhatla, N.; Luo, X.; Nicolov, N.; and Roukos, S. 2004. A statistical model for multilingual entity detection and tracking. In *Proceedings of NAACL-HLT 2004*, 1–8.

Garrette, D.; Erk, K.; and Mooney, R. 2011. Integrating logical representations with probabilistic information using Markov logic. In *Proceedings of ICCS 2011*, 105–114.

Gogate, V., and Dechter, R. 2007. Samplesearch: A scheme that searches for consistent samples. In *Proceedings of AISTATS 2007*.

Grice, H. P. 1975. Logic and conversation. In Cole, P., and Morgan, J. L., eds., *Syntax and Semantics: Vol. 3: Speech Acts*. San Diego, CA: Academic Press. 41–58.

Han, J., and Fu, Y. 1995. Discovery of multiple-level association rules from large databases. In *Proceedings of VLDB 1995*, 420–431.

Kersting, K., and De Raedt, L. 2007. Bayesian Logic Programming: Theory and tool. In Getoor, L., and Taskar, B., eds., *Introduction to Statistical Relational Learning*. Cambridge, MA: MIT Press.

Kersting, K., and Raedt, L. D. 2008. *Basic principles of learning Bayesian Logic Programs*. Berlin, Heidelberg: Springer-Verlag.

Kok, S., and Domingos, P. 2009. Learning Markov logic network structure via hypergraph lifting. In *Proceedings of ICML 2009*, 505–512. ACM.

Kok, S., and Domingos, P. 2010. Learning Markov logic networks using structural motifs. In *Proceedings of ICML 2010*, 551–558.

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Lin, D., and Pantel, P. 2001. Discovery of inference rules for question answering. *Natural Language Engineering* 7(4):343–360.

Mccreath, E., and Sharma, A. 1998. Lime: A system for learning relations. In *Ninth International Workshop on Algorithmic Learning Theory*, 336–374. Springer-Verlag.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo,CA: Morgan Kaufmann.

Pedersen, T.; Patwardhan, S.; and Michelizzi, J. 2004. Wordnet:: Similarity - measuring the relatedness of concepts. In *Proceedings of AAAI 2004*, 1024–1025.

Quinlan, J. R. 1990. Learning logical definitions from relations. *Machine Learning* 5(3):239–266.

Raghavan, S.; Mooney, R. J.; and Ku, H. 2012. Learning to read between the lines using Bayesian Logic Programs. In *Proceedings of ACL 2012*.

Sarawagi, S. 2008. Information extraction. *Foundations and Trends in Databases* 1(3):261–377.

Schoenmackers, S.; Etzioni, O.; Weld, D. S.; and Davis, J. 2010. Learning first-order Horn clauses from web text. In *Proceedings of EMNLP 2010*, 1088–1098. Stroudsburg, PA, USA: Association for Computational Linguistics.

Sorower, M. S.; Dietterich, T. G.; Doppa, J. R.; Walker, O.; Tadepalli, P.; and Fern, X. 2011. Inverting Grice's maxims to learn rules from natural language extractions. In *Proceedings of NIPS 2011*.

Srinivasan, A. 2001. *The Aleph manual*. `http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/`.

Wu, Z., and Palmer, M. 1994. Verb semantics and lexical selection. In *Proceedings of ACL 1994*, 133–138.

Yates, A., and Etzioni, O. 2007. Unsupervised resolution of objects and relations on the web. In *Proceedings of NAACL-HLT 2007*.