Learning to “Read Between the Lines” using Bayesian Logic Programs

Sindhu Raghavan Raymond J. Mooney Hyeonseo Ku
Department of Computer Science
The University of Texas at Austin
1616 Guadalupe, Suite 2.408
Austin, TX 78701, USA
{sindhu,mooney,yorq}@cs.utexas.edu

Abstract

Most information extraction (IE) systems identify facts that are explicitly stated in text. However, in natural language, some facts are implicit, and identifying them requires “reading between the lines”. Human readers naturally use common sense knowledge to *infer* such implicit information from the explicitly stated facts. We propose an approach that uses Bayesian Logic Programs (BLPs), a statistical relational model combining first-order logic and Bayesian networks, to infer additional implicit information from extracted facts. It involves learning uncertain commonsense knowledge (in the form of probabilistic first-order rules) from natural language text by mining a large corpus of automatically extracted facts. These rules are then used to derive additional facts from extracted information using BLP inference. Experimental evaluation on a benchmark data set for machine reading demonstrates the efficacy of our approach.

1 Introduction

The task of information extraction (IE) involves automatic extraction of typed entities and relations from unstructured text. IE systems (Cowie and Lehnert, 1996; Sarawagi, 2008) are trained to extract facts that are stated explicitly in text. However, some facts are implicit, and human readers naturally “read between the lines” and *infer* them from the stated facts using commonsense knowledge. Answering many queries can require inferring such implicitly stated facts. Consider the text “Barack Obama is the president of the United States of America.” Given the query “Barack Obama is a citizen of what country?”, standard IE systems cannot identify the answer since citizenship is not explicitly stated in the text. However, a human reader possesses the commonsense knowledge that the president of a country is almost always a citizen of that country, and easily infers the correct answer.

The standard approach to inferring implicit information involves using commonsense knowledge in the form of logical rules to deduce additional information from the extracted facts. Since manually developing such a knowledge base is difficult and arduous, an effective alternative is to automatically *learn* such rules by mining a substantial database of facts that an IE system has already automatically extracted from a large corpus of text (Nahm and Mooney, 2000). Most existing rule learners assume that the training data is largely accurate and complete. However, the facts extracted by an IE system are always quite noisy and incomplete. Consequently, a purely logical approach to learning and inference is unlikely to be effective. Consequently, we propose using *statistical relational learning* (SRL) (Getoor and Taskar, 2007), specifically, Bayesian Logic Programs (BLPs) (Kersting and De Raedt, 2007), to learn probabilistic rules in first-order logic from a large corpus of extracted facts and then use the resulting BLP to make effective probabilistic inferences when interpreting new documents.

We have implemented this approach by using an off-the-shelf IE system and developing novel adaptations of existing learning methods to efficiently construct fast and effective BLPs for “reading be-

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 349–358,
Jeju, Republic of Korea, 8-14 July 2012.

tween the lines." We present an experimental evaluation of our resulting system on a realistic test corpus from DARPA's Machine Reading project, and demonstrate improved performance compared to a purely logical approach based on Inductive Logic Programming (ILP) (Lavrač and Džeroski, 1994), and an alternative SRL approach based on Markov Logic Networks (MLNs) (Domingos and Lowd, 2009).

To the best of our knowledge, this is the first paper that employs BLPs for inferring implicit information from natural language text. We demonstrate that it is possible to learn the structure and the parameters of BLPs automatically using only noisy extractions from natural language text, which we then use to infer additional facts from text.

The rest of the paper is organized as follows. Section 2 discusses related work and highlights key differences between our approach and existing work. Section 3 provides a brief background on BLPs. Section 4 describes our BLP-based approach to learning to infer implicit facts. Section 5 describes our experimental methodology and discusses the results of our evaluation. Finally, Section 6 discusses potential future work and Section 7 presents our final conclusions.

2 Related Work

Several previous projects (Nahm and Mooney, 2000; Carlson et al., 2010; Schoenmackers et al., 2010; Doppa et al., 2010; Sorower et al., 2011) have mined inference rules from data automatically extracted from text by an IE system. Similar to our approach, these systems use the learned rules to infer additional information from facts directly extracted from a document. Nahm and Mooney (2000) learn *propositional rules* using C4.5 (Quinlan, 1993) from data extracted from computer-related job-postings, and therefore cannot learn multi-relational rules with quantified variables. Other systems (Carlson et al., 2010; Schoenmackers et al., 2010; Doppa et al., 2010; Sorower et al., 2011) learn *first-order rules* (i.e. Horn clauses in first-order logic).

Carlson et al. (2010) modify an ILP system similar to FOIL (Quinlan, 1990) to learn rules with probabilistic conclusions. They use purely logical deduction (forward-chaining) to infer additional facts. Unlike BLPs, this approach does not use a well-founded probabilistic graphical model to compute coherent probabilities for inferred facts. Further, Carlson et al. (2010) used a human judge to manually evaluate the quality of the learned rules before using them to infer additional facts. Our approach, on the other hand, is completely automated and learns fully parameterized rules in a well-defined probabilistic logic.

Schoenmackers et al. (2010) develop a system called SHERLOCK that uses statistical relevance to learn first-order rules. Unlike our system and others (Carlson et al., 2010; Doppa et al., 2010; Sorower et al., 2011) that use a pre-defined ontology, they automatically identify a set of entity types and relations using "open IE." They use HOLMES (Schoenmackers et al., 2008), an inference engine based on MLNs (Domingos and Lowd, 2009) (an SRL approach that combines first-order logic and Markov networks) to infer additional facts. However, MLNs include all possible type-consistent groundings of the rules in the corresponding Markov net, which, for larger datasets, can result in an intractably large graphical model. To overcome this problem, HOLMES uses a specialized model construction process to control the grounding process. Unlike MLNs, BLPs naturally employ a more "focused" approach to grounding by including only those literals that are directly relevant to the query.

Doppa et al. (2010) use FARMER (Nijssen and Kok, 2003), an existing ILP system, to learn first-order rules. They propose several approaches to score the rules, which are used to infer additional facts using purely logical deduction. Sorower et al. (2011) propose a probabilistic approach to modeling implicit information as missing facts and use MLNs to infer these missing facts. They learn first-order rules for the MLN by performing exhaustive search. As mentioned earlier, inference using both these approaches, logical deduction and MLNs, have certain limitations, which BLPs help overcome.

DIRT (Lin and Pantel, 2001) and RESOLVER (Yates and Etzioni, 2007) learn inference rules, also called entailment rules that capture synonymous relations and entities from text. Berant et al. (Berant et al., 2011) propose an approach that uses transitivity constraints for learning entailment rules for typed predicates. Unlike the systems described above, these systems do not learn complex first-order rules that capture common sense knowledge. Further, most of these systems do not use extractions from an IE system to learn entailment rules, thereby making them less related to our approach.

3 Bayesian Logic Programs

Bayesian logic programs (BLPs) (Kersting and De Raedt, 2007; Kersting and Raedt, 2008) can be considered 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 Horn-clause proofs of the query are constructed and used to build a Bayes net for answering the 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 as reviewed by Koller and Friedman (2009). The parameters in the BLP model can be learned using the methods described by Kersting and De Raedt (2008).

4 Learning BLPs to Infer Implicit Facts

4.1 Learning Rules from Extracted Data

The first step involves learning commonsense knowledge in the form of first-order Horn rules from text. We first extract facts that are explicitly stated in the text using SIRE (Florian et al., 2004), an IE system developed by IBM. We then learn first-order rules from these extracted facts using LIME (Mccreath and Sharma, 1998), an ILP system designed for noisy training data.

We first identify a set of target relations we want to infer. Typically, an ILP system takes a set of positive and negative instances for a target relation, along with a background knowledge base (in our case, other facts extracted from the same document) from which the positive instances are potentially inferable. In our task, we only have direct access to positive instances of target relations, i.e the relevant facts extracted from the text. So we artificially generate negative instances using the *closed world assumption*, which states that any instance of a relation that is not extracted can be considered a negative instance. While there are exceptions to this assumption, it typically generates a useful (if noisy) set of negative instances. For each relation, we generate all possible type-consistent instances using all constants in the domain. All instances that are not extracted facts (i.e. positive instances) are labeled as negative. The total number of such closed-world negatives can be intractably large, so we randomly sample a fixed-size subset. The ratio of 1:20 for positive to negative instances worked well in our approach.

Since LIME can learn rules using only positive instances, or both positive and negative instances, we learn rules using both settings. We include all unique rules learned from both settings in the final set, since the goal of this step is to learn a large set of potentially useful rules whose relative strengths will be determined in the next step of parameter learning. Other approaches could also be used to learn candidate rules. We initially tried using the popular ALEPH ILP system (Srinivasan, 2001), but it did not produce useful rules, probably due to the high level of noise in our training data.

4.2 Learning BLP Parameters

The parameters of a BLP include the CPT entries associated with the Bayesian clauses and the parameters of combining rules associated with the Bayesian predicates. For simplicity, we use a deterministic logical-and model to encode the CPT entries associated with Bayesian clauses, and use *noisy-or* to combine evidence coming from multiple ground rules that have the same head (Pearl, 1988). The noisy-or model requires just a single parameter for each rule, which can be learned from training data.

We learn the noisy-or parameters using the EM algorithm adapted for BLPs by Kersting and De Raedt (2008). In our task, the supervised training data consists of facts that are extracted from the natural language text. However, we usually do not have evidence for inferred facts as well as noisy-or nodes. As a result, there are a number of variables in the ground networks which are always hidden, and hence EM is appropriate for learning the requisite parameters from the partially observed training data.

4.3 Inference of Additional Facts using BLPs

Inference in the BLP framework involves backward chaining (Russell and Norvig, 2003) from a specified query (SLD resolution) to obtain all possible deductive proofs for the query. In our context, each target relation becomes a query on which we backchain. We then construct a ground Bayesian network using the resulting deductive proofs for all target relations and learned parameters using the standard approach described in Section 3. Finally, we perform standard probabilistic inference to estimate the marginal probability of each inferred fact. Our system uses Sample Search (Gogate and Dechter, 2007), an approximate sampling algorithm developed for Bayesian networks with deterministic constraints (0 values in CPTs). We tried several exact and approximate inference algorithms on our data, and this was the method that was both tractable and produced the best results.

5 Experimental Evaluation

5.1 Data

For evaluation, we used DARPA's machine-reading intelligence-community (IC) data set, which consists of news articles on terrorist events around the world. There are $10,000$ documents each containing an average of 89.5 facts extracted by SIRE (Florian et al., 2004). SIRE assigns each extracted fact a confidence score and we used only those with a score of 0.5 or higher for learning and inference. An average of 86.8 extractions per document meet this threshold.

DARPA also provides an ontology describing the entities and relations in the IC domain. It 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 superclasses. For instance, a GeopoliticalEntity can be a HumanAgent as well as a Location. This can cause some problems for systems that rely on a strict typing system, such as MLNs which rely on types to limit the space of ground literals that are considered. Some sample relations are attendedSchool, approximateNumberOfMembers, mediatingAgent, employs, hasMember, hasMemberHumanAgent, and hasBirthPlace.

5.2 Methodology

We evaluated our approach using 10-fold cross validation. We learned first-order rules for the 13 target relations shown in Table 3 from the facts extracted from the training documents (Section 4.1). These relations were selected because the extractor's recall for them was low. Since LIME does not scale well to large data sets, we could train it on at most about $2,500$ documents. Consequently, we split the $9,000$ training documents into four disjoint subsets and learned first-order rules from each subset. The final knowledge base included all unique rules learned from any subset. LIME learned several rules that had only entity types in their bodies. Such rules make many incorrect inferences; hence we eliminated them. We also eliminated rules violating type constraints. We learned an average of 48 rules per fold. Table 1 shows some sample learned rules.

We then learned parameters as described in Section 4.2. We initially set all noisy-or parameters to 0.9 based on the intuition that if exactly one rule for a consequent was satisfied, it could be inferred with a probability of 0.9.

governmentOrganization(A) ∧ employs(A,B) → hasMember(A,B) *If a government organization A employs person B, then B is a member of A*
eventLocation(A,B) ∧ bombing(A) → thingPhysicallyDamaged(A,B) *If a bombing event A took place in location B, then B is physically damaged*
isLedBy(A,B) → hasMemberPerson(A,B) *If a group A is led by person B, then B is a member of A*
nationState(B) ∧ eventLocationGPE(A,B) → eventLocation(A,B) *If an event A occurs in a geopolitical entity B, then the event location for that event is B*
mediatingAgent(A,B) ∧ humanAgentKillingAPerson(A) → killingHumanAgent(A,B) *If A is an event in which a human agent is killing a person and the mediating agent of A is an agent B, then B is the human agent that is killing in event A*

Table 1: A sample set of rules learned using LIME

For each test document, we performed BLP inference as described in Section 4.3. We ranked all inferences by their marginal probability, and evaluated the results by either choosing the top n inferences or accepting inferences whose marginal probability was equal to or exceeded a specified threshold. We evaluated two BLPs with different parameter settings: *BLP-Learned-Weights* used noisy-or parameters learned using EM, *BLP-Manual-Weights* used fixed noisy-or weights of 0.9.

5.3 Evaluation Metrics

The lack of ground truth annotation for inferred facts prevents an automated evaluation, so we resorted to a manual evaluation. We randomly sampled 40 documents (4 from each test fold), judged the accuracy of the inferences for those documents, and computed *precision*, the fraction of inferences that were deemed correct. For probabilistic methods like BLPs and MLNs that provide certainties for their inferences, we also computed *precision at top n*, which measures the precision of the n inferences with the highest marginal probability across the 40 test documents. Measuring recall for making inferences is very difficult since it would require labeling a reasonable-sized corpus of documents with *all* of the correct inferences for a given set of target relations, which would be extremely time consuming. Our evaluation is similar to that used in previous related work (Carlson et al., 2010; Schoenmackers et al., 2010).

SIRE frequently makes incorrect extractions, and therefore inferences made from these extractions are also inaccurate. To account for the mistakes made by the extractor, we report two different precision scores. The "unadjusted" (UA) score, does not correct for errors made by the extractor. The "adjusted" (AD) score does not count mistakes due to extraction errors. That is, if an inference is incorrect because it was based on incorrect extracted facts, we remove it from the set of inferences and calculate precision for the remaining inferences.

5.4 Baselines

Since none of the existing approaches have been evaluated on the IC data, we cannot directly compare our performance to theirs. Therefore, we compared to the following methods:

- *Logical Deduction*: This method forward chains on the extracted facts using the first-order rules learned by LIME to infer additional facts. This approach is unable to provide any confidence or probability for its conclusions.

- *Markov Logic Networks (MLNs)*: We use the rules learned by LIME to define the structure of an MLN. In the first setting, which we call *MLN-Learned-Weights*, we learn the MLN's parameters using the generative weight learning algorithm (Domingos and Lowd, 2009), which we modified to process training examples in an online manner. In online generative learning, gradients are calculated and weights are estimated after processing each example and the learned weights are used as the starting weights for the next example. The pseudo-likelihood of one round is obtained by multiplying the pseudo-likelihood of all examples.

	UA	AD
Precision	29.73 (443/1490)	35.24 (443/1257)

Table 2: Precision for logical deduction. "UA" and "AD" refer to the unadjusted and adjusted scores respectively

In our approach, the initial weights of clauses are set to 10. The average number of iterations needed to acquire the optimal weights is 131. In the second setting, which we call *MLN-Manual-Weights*, we assign a weight of 10 to all rules and maximum likelihood prior to all predicates. MLN-Manual-Weights is similar to BLP-Manual-Weights in that all rules are given the same weight. We then use the learned rules and parameters to probabilistically infer additional facts using the MC-SAT algorithm implemented in Alchemy,[1] an open-source MLN package.

6 Results and Discussion

6.1 Comparison to Baselines

Table 2 gives the unadjusted (UA) and adjusted (AD) precision for logical deduction. Out of $1,490$ inferences for the 40 evaluation documents, 443 were judged correct, giving an unadjusted precision of 29.73%. Out of these $1,490$ inferences, 233 were determined to be incorrect due to extraction errors, improving the adjusted precision to a modest 35.24%.

MLNs made about $127,000$ inferences for the 40 evaluation documents. Since it is not feasible to manually evaluate *all* the inferences made by the MLN, we calculated precision using only the top 1000 inferences. Figure 1 shows both unadjusted and adjusted precision at top-n for various values of n for different BLP and MLN models. For both BLPs and MLNs, simple manual weights result in superior performance than the learned weights. Despite the fairly large size of the overall training sets (9,000 documents), the amount of data for each target relation is apparently still not sufficient to learn particularly accurate weights for both BLPs and MLNs. However, for BLPs, learned weights do show a substantial improvement initially (i.e. top 25–50 inferences), with an average of 1 inference per document at 91% adjusted precision as opposed to an average of 5 inferences per document at 85% adjusted precision for BLP-Manual-Weights. For MLNs, learned weights show a small improvement initially only with respect to adjusted precision. Between BLPs and MLNs, BLPs perform substantially better than MLNs at most points in the curve. However, MLN-Manual-Weights improve marginally over BLP-Learned-Weights at later points (top 600 and above) on the curve, where the precision is generally very low. Here, the superior performance of BLPs over MLNs could be possibly due to the focused grounding used in the BLP framework.

For BLPs, as n increases towards including all of the logically sanctioned inferences, as expected, the precision converges to the results for logical deduction. However, as n decreases, both adjusted and unadjusted precision increase fairly steadily. This demonstrates that probabilistic BLP inference provides a clear improvement over logical deduction, allowing the system to accurately select the best inferences that are most likely to be correct. Unlike the two BLP models, MLN-Manual-Weights has more or less the same performance at most points on the curve, and it is slightly better than that of purely-logical deduction. MLN-Learned-Weights is worse than purely-logical deduction at most points on the curve.

6.2 Results for Individual Target Relations

Table 3 shows the *adjusted* precision for each relation for instances inferred using logical deduction, BLP-Manual-Weights and BLP-Learned-Weights with a confidence threshold of 0.95. The probabilities estimated for inferences by MLNs are not directly comparable to those estimated by BLPs. As a result, we do not include results for MLNs here. For this evaluation, using a confidence threshold based cutoff is more appropriate than using top-n inferences made by the BLP models since the estimated probabilities can be directly compared across target relations.

For logical deduction, precision is high for a few relations like employs, hasMember, and hasMemberHumanAgent, indicating that the rules learned for these relations are more accurate than the ones

[1] `http://alchemy.cs.washington.edu/`

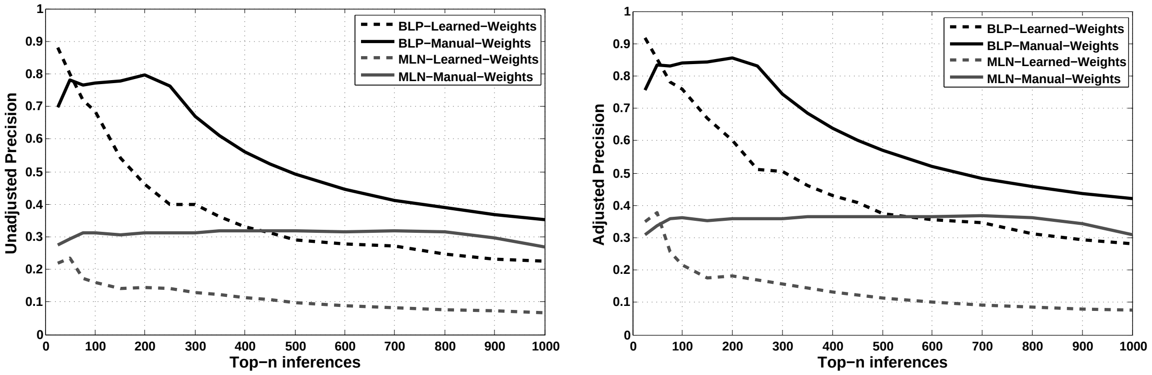

Figure 1: Unadjusted and adjusted precision at top-n for different BLP and MLN models for various values of n

learned for the remaining relations. Unlike relations like hasMember that are easily inferred from relations like employs and isLedBy, certain relations like hasBirthPlace are not easily inferable using the information in the ontology. As a result, it might not be possible to learn accurate rules for such target relations. Other reasons include the lack of a sufficiently large number of target-relation instances during training and lack of strictly defined types in the IC ontology.

Both BLP-Manual-Weights and BLP-Learned-Weights also have high precision for several relations (eventLocation, hasMemberHumanAgent, thingPhysicallyDamaged). However, the actual number of inferences can be fairly low. For instance, 103 instances of hasMemberHumanAgent are inferred by logical deduction (i.e. 0 confidence threshold), but only 2 of them are inferred by BLP-Learned-Weights at 0.95 confidence threshold, indicating that the parameters learned for the corresponding rules are not very high. For several relations like hasMember, hasMemberPerson, and employs, no instances were inferred by BLP-Learned-Weights at 0.95 confidence threshold. Lack of sufficient training instances (extracted facts) is possibly the reason for learning low weights for such rules. On the other hand, BLP-Manual-Weights has inferred 26 instances of hasMemberHumanAgent, out which all are correct. These results therefore demonstrate the need for sufficient training examples to learn accurate parameters.

6.3 Discussion

We now discuss the potential reasons for BLP's superior performance compared to other approaches. Probabilistic reasoning used in BLPs allows for a principled way of determining the most confident inferences, thereby allowing for improved precision over purely logical deduction. The primary difference between BLPs and MLNs lies in the approaches used to construct the ground network. In BLPs, only propositions that can be logically deduced from the extracted evidence are included in the ground network. On the other hand, MLNs include all possible type-consistent groundings of all rules in the network, introducing many ground literals which cannot be logically deduced from the evidence. This generally results in several incorrect inferences, thereby yielding poor performance.

Even though learned weights in BLPs do not result in a superior performance, learned weights in MLNs are substantially worse. Lack of sufficient training data is one of the reasons for learning less accurate weights by the MLN weight learner. However, a more important issue is due to the use of the closed world assumption during learning, which we believe is adversely impacting the weights learned. As mentioned earlier, for the task considered in the paper, if a fact is not explicitly stated in text, and hence not extracted by the extractor, it does not necessarily imply that it is not true. Since existing weight learning approaches for MLNs do not deal with missing data and open world assumption, developing such approaches is a topic for future work.

Apart from developing novel approaches for

Relation	Logical Deduction	BLP-Manual-Weights-.95	BLP-Learned-Weights-.95	No. training instances
employs	**69.44** (25/36)	**92.85** (13/14)	nil (0/0)	**18440**
eventLocation	18.75 (18/96)	**100.00** (1/1)	**100** (1/1)	6902
hasMember	**95.95** (95/99)	**97.26** (71/73)	nil (0/0)	1462
hasMemberPerson	43.75 (42/96)	**100.00** (14/14)	nil (0/0)	705
isLedBy	12.30 (8/65)	nil (0/0)	nil (0/0)	8402
mediatingAgent	19.73 (15/76)	nil (0/0)	nil (0/0)	**92998**
thingPhysicallyDamaged	25.72 (62/241)	**90.32** (28/31)	**90.32** (28/31)	**24662**
hasMemberHumanAgent	**95.14** (98/103)	**100.00** (26/26)	**100.00** (2/2)	3619
killingHumanAgent	15.35 (43/280)	33.33 (2/6)	**66.67** (2/3)	3341
hasBirthPlace	0 (0/88)	nil (0/0)	nil (0/0)	89
thingPhysicallyDestroyed	nil (0/0)	nil (0/0)	nil (0/0)	800
hasCitizenship	48.05 (37/77)	58.33 (35/60)	nil (0/0)	222
attendedSchool	nil (0/0)	nil (0/0)	nil (0/0)	2

Table 3: Adjusted precision for individual relations (highest values are in bold)

weight learning, additional engineering could potentially improve the performance of MLNs on the IC data set. Due to MLN's grounding process, several spurious facts like employs(a,a) were inferred. These inferences can be prevented by including additional clauses in the MLN that impose integrity constraints that prevent such nonsensical propositions. Further, techniques proposed by Sorower et al. (2011) can be incorporated to explicitly handle missing information in text. Lack of strict typing on the arguments of relations in the IC ontology has also resulted in inferior performance of the MLNs. To overcome this, relations that do not have strictly defined types could be specialized. Finally, we could use the deductive proofs constructed by BLPs to constrain the ground Markov network, similar to the model-construction approach adopted by Singla and Mooney (2011).

However, in contrast to MLNs, BLPs that use first-order rules that are learned by an off-the-shelf ILP system and given simple intuitive hand-coded weights, are able to provide fairly high-precision inferences that augment the output of an IE system and allow it to effectively "read between the lines."

7 Future Work

A primary goal for future research is developing an on-line structure learner for BLPs that can directly learn probabilistic first-order rules from uncertain training data. This will address important limitations of LIME, which cannot accept uncertainty in the extractions used for training, is not specifically optimized for learning rules for BLPs, and does not scale well to large datasets. Given the relatively poor performance of BLP parameters learned using EM, tests on larger training corpora of extracted facts and the development of improved parameter-learning algorithms are clearly indicated. We also plan to perform a larger-scale evaluation by employing crowdsourcing to evaluate inferred facts for a bigger corpus of test documents. As described above, a number of methods could be used to improve the performance of MLNs on this task. Finally, it would be useful to evaluate our methods on several other diverse domains.

8 Conclusions

We have introduced a novel approach using Bayesian Logic Programs to learn to infer implicit information from facts extracted from natural language text. We have demonstrated that it can learn effective rules from a large database of noisy extractions. Our experimental evaluation on the IC data set demonstrates the advantage of BLPs over logical deduction and an approach based on MLNs.

Acknowledgements

We thank the SIRE team from IBM for providing SIRE extractions on the IC data set. This research was funded by MURI ARO grant W911NF-08-1-0242 and Air Force Contract FA8750-09-C-0172 under the DARPA Machine Reading Program. Experiments were run on the Mastodon Cluster, provided by NSF grant EIA-0303609.

References

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACl-HLT 2011)*, pages 610–619.

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr., and T.M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 1306–1313. AAAI Press.

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

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

Janardhan Rao Doppa, Mohammad NasrEsfahani, Mohammad S. Sorower, Thomas G. Dietterich, Xiaoli Fern, and Prasad Tadepalli. 2010. Towards learning rules from natural texts. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading (FAM-LbR 2010)*, pages 70–77, Stroudsburg, PA, USA. Association for Computational Linguistics.

Radu Florian, Hany Hassan, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla, Xiaoqiang Luo, Nicolas Nicolov, and Salim Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2004)*, pages 1–8.

L. Getoor and B. Taskar, editors. 2007. *Introduction to Statistical Relational Learning*. MIT Press, Cambridge, MA.

Vibhav Gogate and Rina Dechter. 2007. Samplesearch: A scheme that searches for consistent samples. In *Proceedings of Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07)*.

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

Kristian Kersting and Luc De Raedt. 2008. *Basic principles of learning Bayesian Logic Programs*. Springer-Verlag, Berlin, Heidelberg.

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

Nada Lavrač and Saso Džeroski. 1994. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood.

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

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

Un Yong Nahm and Raymond J. Mooney. 2000. A mutually beneficial integration of data mining and information extraction. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI 2000)*, pages 627–632, Austin, TX, July.

Siegfried Nijssen and Joost N. Kok. 2003. Efficient frequent query discovery in FARMER. In *Proceedings of the Seventh Conference in Principles and Practices of Knowledge Discovery in Database (PKDD 2003)*, pages 350–362. Springer.

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

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

J. R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo,CA.

Stuart Russell and Peter Norvig. 2003. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 2 edition.

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

Stefan Schoenmackers, Oren Etzioni, and Daniel S. Weld. 2008. Scaling textual inference to the web. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, pages 79–88, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stefan Schoenmackers, Oren Etzioni, Daniel S. Weld, and Jesse Davis. 2010. Learning first-order Horn clauses from web text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 1088–1098, Stroudsburg, PA, USA. Association for Computational Linguistics.

Parag Singla and Raymond Mooney. 2011. Abductive Markov Logic for plan recognition. In *Twenty-fifth National Conference on Artificial Intelligence*.

Mohammad S. Sorower, Thomas G. Dietterich, Janardhan Rao Doppa, Orr Walker, Prasad Tadepalli, and Xiaoli Fern. 2011. Inverting Grice's maxims to learn rules from natural language extractions. In *Proceedings of Advances in Neural Information Processing Systems 24*.

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

Alexander Yates and Oren Etzioni. 2007. Unsupervised resolution of objects and relations on the web. In *Pro-*

ceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2007).