Towards Learning Rules from Natural Texts*

Janardhan Rao Doppa, Mohammad NasrEsfahani, Mohammad S. Sorower
Thomas G. Dietterich, Xiaoli Fern, and Prasad Tadepalli
School of EECS, Oregon State University
Corvallis, OR 97330, USA
{doppa,nasresfm,sorower,tgd,xfern,tadepall}@cs.orst.edu

Abstract

In this paper, we consider the problem of inductively learning rules from specific facts extracted from texts. This problem is challenging due to two reasons. First, natural texts are *radically incomplete* since there are always too many facts to mention. Second, natural texts are *systematically biased* towards novelty and surprise, which presents an unrepresentative sample to the learner. Our solutions to these two problems are based on building a generative observation model of what is mentioned and what is extracted given what is true. We first present a *Multiple-predicate Bootstrapping* approach that consists of iteratively learning if-then rules based on an implicit observation model and then imputing new facts implied by the learned rules. Second, we present an iterative *ensemble co-learning* approach, where multiple decision-trees are learned from bootstrap samples of the incomplete training data, and facts are imputed based on weighted majority.

1 Introduction

One of the principal goals of learning by reading is to make the vast amount of natural language text which is on the web accessible to automatic processing. There are at least three different ways in which this can be done. First, factual knowledge on the web can be extracted as formal relations or tuples of a data base. A number of information extraction systems, starting from the WebKb project (Craven et al., 2000), to Whirl (Cohen, 2000) to the TextRunner (Etzioni et al., 2008) project are of this kind. They typically learn patterns or rules that can be applied to text to extract instances of relations. A second possibility is to learn general knowledge, rules, or general processes and procedures by reading natural language descriptions of them, for example, extracting formal descriptions of the rules of the United States Senate or a recipe to make a dessert. A third instance of machine reading is to generalize the facts extracted from the text to learn more general knowledge. For example, one might learn by generalizing from reading the obituaries that most people live less than 90 years, or people tend to live and die in the countries they were born in. In this paper, we consider the problem of learning such general rules by reading about specific facts.

At first blush, learning rules by reading specific facts appears to be a composition of information extraction followed by rule induction. In the above example of learning from obituaries, there is reason to believe that this reductionist approach would work well. However, there are two principal reasons why this approach of learning directly from natural texts is problematic. One is that, unlike databases, the natural texts are *radically incomplete*. By this we mean that many of the facts that are relevant to predicting the target relation might be missing in the text. This

* This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-09-C-0179. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA, or the Air Force Research Laboratory (AFRL). We thank the reviewers for their insightful comments and helpful suggestions.

Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading, pages 70–77,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

is so because in most cases the set of relevant facts is open ended.

The second problem, in some ways more worrisome, is that the natural language texts are *systematically biased* towards newsworthiness, which correlates with infrequency or novelty. This is sometimes called "the man bites a dog phenomenon."[1] Unfortunately the novelty bias violates the most common assumption of machine learning that the training data is representative of the underlying truth, or equivalently, that any missing information is missing at random. In particular, since natural langauge texts are written for people who already possess a vast amount of prior knowledge, communication efficiency demands that facts that can be easily inferred by most people are left out of the text.

To empirically validate our two hypotheses of radical incompleteness and systematic bias of natural texts, we have examined a collection of 248 documents related to the topics of people, organizations, and relationships collected by the Linguistic Data Consortium (LDC). We chose the target relationship of the birth place of a person. It turned out that the birth place of some person is only mentioned 23 times in the 248 documents, illustrating the radical incompleteness of texts mentioned earlier. Moreover, in 14 out of the 23 mentions of the birth place, the information violates some default inferences. For example, one of the sentences reads:

"Ahmed Said Khadr, an Egyptian-born Canadian, was killed last October in Pakistan."

Presumably the phrase "Egyptian-born" was considered important by the reporter because it violates our expectation that most Canadians are born in Canada. If Khadr was instead born in Canada, the reporter would mostly likely have left out "Canadian-born" because it is too obvious to mention given he is a Canadian. In all the 9 cases where the birth place does not violate the default assumptions, the story is biographical, e.g., an obituary.

In general, only a small part of the whole truth is ever mentioned in a given document. Thus, the reporter has to make some choices as to what to mention and what to leave out. The key insight of this paper is that considering how these choices are made is important in making correct statistical inferences. In the above example, wrong probabilities would be derived if one assumes that the birth place information is missing at random.

In this paper we introduce the notion of a "mention model," which models the generative process of what is mentioned in a document. We also extend this using an "extraction model," which represents the errors in the process of extracting facts from the text documents. The mention model and the extraction model together represent the probability that some facts are extracted given the true facts.

For learning, we could use an explicit mention model to score hypothesized rules by calculating the probability that a rule is satisfied by the observed evidence and then pick the rules that are most likely given the evidence. In this paper, we take the simpler approach of directly adapting the learning algorithms to an *implicit* mention model, by changing the way a rule is scored by the available evidence.

Since each text document involves multiple predicates with relationships between them, we learn rules to predict each predicate from the other predicates. Thus, the goal of the system is to learn a sufficiently large set of rules to infer all the missing information as accurately as possible. To effectively bootstrap the learning process, the learned rules are used on the incomplete training data to impute new facts, which are then used to induce more rules in subsequent iterations. This approach is most similar to the coupled semi-supervised learning of (Carlson et al., 2010) and general bootstrapping approaches in natural language processing (Yarowsky, 1995). Since this is in the context of multiple-predicate learning in inductive logic programming (ILP) (DeRaedt and Lavraøc, 1996), we call this approach "Multiple-predicate Bootstrapping."

One problem with Multiple-predicate Bootstrapping is potentially large variance. To mitigae this, we consider the bagging approach, where multiple rule sets are learned from bootstrap samples of the training data with an implicit mention model to score the rules. We then use these sets of rules as an ensemble to impute new facts, and repeat the process.

We evaluate both of these approaches on real world data processed through synthetic observation models. Our results indicate that when the assump-

[1] "When a dog bites a man, that is not news, because it happens so often. But if a man bites a dog, that is news," attributed to John Bogart of New York Sun among others.

tions of the learner suit the observation model, the learner's performance is quite good. Further, we show that the ensemble approach significantly improves the performance of Multiple-predicate Bootstrapping.

2 Probabilistic Observation Model

In this section, we will introduce a notional probabilistic observation model that captures what facts are extracted by the programs from the text given the true facts about the world and the common sense rules of the domain of discourse.

The observation model is composed of the *mention model* and the *extraction model*. The mention model $P(MentDB|TrueDB, Rules)$ models the probability distribution of mentioned facts, $MentDB$, given the set of true facts $TrueDB$ and the rules of the domain, $Rules$. For example, if a fact is always true, then the novelty bias dictates that it is *not* mentioned with a high probability. The same is true of any fact entailed by a generally valid rule that is common knowledge. For example, this model predicts that since it is common knowledge that Canadians are born in Canada, the birth place is not mentioned if a person is a Canadian and was born in Canada.

The extraction model $P(ExtrDB|MentDB)$ models the probability distribution of extracted facts, given the set of mentioned facts $MentDB$. For example, it might model that explicit facts are extracted with high probability and that the extracted facts are corrupted by coreference errors. Note that the extraction process operates only on the mentioned part of the database $MentDB$; it has no independent access to the $TrueDB$ or the $Rules$. In other words, the mentioned database $MentDB$ d-separates the extracted database $ExtrDB$ from the true database $TrueDB$ and the $Rules$, and the conditional probability decomposes.

We could also model multiple documents generated about the same set of facts $TrueDB$, and multiple databases independently extracted from the same document by different extraction systems. Given an explicit observation model, the learner can use it to consider different rule sets and evaluate their likelihood given some data. The posterior probability of a rule set given an extracted database can be obtained by marginalizing over possible true and mentioned databases. Thus, in principle, the maximum likelihood approach to rule learning could work by considering each set of rules and evaluating its posterior given the extracted database, and picking the best set. While conceptually straightforward, this approach is highly intractable due to the need to marginalize over all possible mentioned and true databases. Moreover, it seems unnecessary to force a choice between sets of rules, since different rule sets do not always conflict. In the next section, we describe a simpler approach of adapting the learning algorithms directly to score and learn rules using an *implicit* mention model.

3 Multiple-predicate Bootstrapping with an Implicit Mention Model

Our first approach, called "Multiple-predicate Bootstrapping," is inspired by several pieces of work including co-training (Blum and Mitchell, 1998), multitask learning (Caruana, 1997), coupled semi-supervised learning (Carlson et al., 2010) and self-training (Yarowsky, 1995). It is based on learning a set of rules for all the predicates in the domain given the others and using them to infer (impute) the missing facts in the training data. This is repeated for several iterations until no more facts can be inferred. The support of a rule is measured by the number of records which satisfy the body of the rule, where each record roughly corresponds to a collection of related facts that can be independently generated, e.g., information about a single football game or a single news item. The higher the support, the more statistical evidence we have for judging its predictive accuracy. To use a rule to impute facts, it needs to be "promoted," which means it should pass a certain *threshold support* level. We measure the precision of a rule as the ratio of the number of records that non-trivially satisfy the rule to the number that satisfy its body, which is a proxy for the conditional probability of the head given the body. A rule is non-trivially satisfied by a record if the rule evaluates to true on that record for all possible instantiations of its variables, and there is at least one instantiation that satisfies its body. Given multiple promoted rules which apply to a given instance, we pick the rule with the highest precision to impute its value.

3.1 Implicit Mention Models

We adapt the multiple-predicate bootstrapping approach to the case of incomplete data by adjusting the scoring function of the learning algorithm to respect the assumed mention model. Unlike in the maximum likelihood approach discussed in the previous section, there is no explicit mention model used by the learner. Instead the scoring function is optimized for a presumed *implicit* mention model. We now discuss three specific mention models and the corresponding scoring functions.

Positive Mention Model: In the "positive mention model," it is assumed that any missing fact is false. This justifies counting evidence using the negation by failure assumption of Prolog. We call this scoring method "conservative." For example, the text "Khadr, a Canadian citizen, was killed in Pakistan" is counted as not supporting the rule `citizen(X,Y)` ⇒ `bornIn(X,Y)`, as we are not told that `bornIn(Khadr,Canada)`. Positive mention model is inapplicable for most instances of learning from natural texts, except for special cases such as directory web pages.

Novelty Mention Model: In the "novelty mention model," it is assumed that facts are missing only when they are entailed by other mentioned facts and rules that are common knowledge. This suggests an "aggressive" or optimistic scoring of candidate rules, which interprets a missing fact so that it supports the candidate rule. More precisely, a rule is counted as non-trivially satisfied by a record if there is some way of imputing the missing facts in the record without causing contradiction. For example, the text "Khadr, a Canadian citizen was killed in Pakistan" is counted as non-trivially supporting the rule `citizen(X,Y)` ⇒ `bornIn(X,Y)` because, adding `bornIn(Khadr, Canada)` supports the rule without contradicting the available evidence. On the other hand, the above text does not support the rule `killedIn(X,Y)` ⇒ `citizen(X,Y)` because the rule contradicts the evidence, assuming that `citizen` is a functional relationship.

Random Mention Model: In the "random mention model," it is assumed that facts are missing at random. Since the random facts can be true or false, this mention model suggests counting the evidence fractionally in proportion to its predicted prevalence. Following the previous work on learning from missing data, we call this scoring method "distributional" (Saar-Tsechansky and Provost, 2007). In distributional scoring, we typically learn a distribution over the values of a literal given its argument and use it to assign a fractional count to the evidence. This is the approach taken to account for missing data in Quinlan's decision tree algorithm (Quinlan, 1986). We will use this as part of our Ensemble Co-Learning approach of the next section.

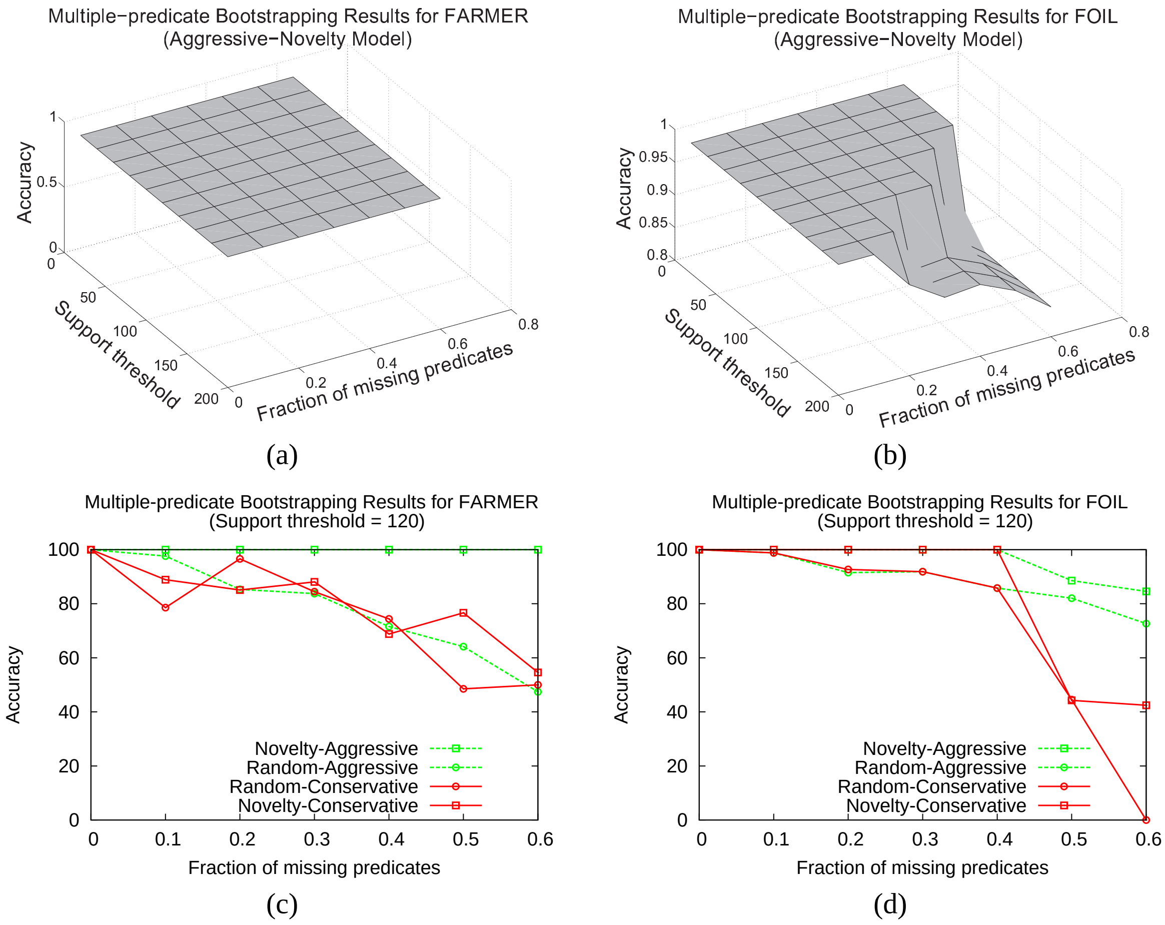
3.2 Experimental Results

We evaluated Multiple-predicate Bootstrapping with implicit mention models on the schema-based NFL database retrieved from `www.databasefootball.com`. We developed two different synthetic observation models. The observation models are based on the Novelty mention model and the Random mention model and assume perfect extraction in each case. The following predicates are manually provided:

- `gameWinner (Game, Team)`,
- `gameLoser(Game, Team)`,
- `homeTeam(Game, Team)`,
- `awayTeam(Game, Team)`, and
- `teamInGame(Team, Game)`,

with the natural interpretations. To simplify arithmetic reasoning we replaced the numeric team scores in the real database with two defined predicates `teamSmallerScore(Team, Game)` and `teamGreaterScore(Team, Game)` to indicate the teams with the smaller and the greater scores.

We generate two sets of synthetic data as follows. In the Random mention model, each predicate except the `teamInGame` predicate is omitted independently with probability p. The Novelty mention model, on the other hand, relies on the fact that `gameWinner`, `gameLoser`, and `teamFinalScore` are mutually correlated, as are `homeTeam` and `awayTeam`. Thus, it picks one predicate



(a)

(b)

(c)

(d)

Figure 1: Multiple-predicate bootstrapping Results for (a) FARMER using aggressive-novelty model (b) FOIL using aggressive-novelty model (c) FARMER with support threshold 120 (d) FOIL with support threshold 120

from the first group to mention its values, and omitseach of the other predicates independently with some probability q. Similarly it gives a value to one of the two predicates in the second group and omits the other predicate with probability q. One consequence of this model is that it always has one of the predicates in the first group and one of the predicates in the second group, which is sufficient to infer everything if one knew the correct domain rules. We evaluate two scoring methods: the aggressive scoring and the conservative scoring.

We employed two learning systems: Quinlan's FOIL, which learns relational rules using a greedy covering algorithm (Quinlan, 1990; Cameron-Jones and Quinlan, 1994), and Nijssen and Kok's FARMER, which is a relational data mining algorithm that searches for conjunctions of literals of large support using a bottom-up depth first search (Nijssen and Kok, 2003). Both systems were applied to learn rules for all target predicates. One important difference to note here is that while FARMER seeks all rules that exceed the necessary support threshold, FOIL only learns rules that are sufficient to classify all training instances into those that satisfy the target predicate and those that do not. Secondly, FOIL tries to learn maximally deterministic rules, while FARMER is parameterized by the minimum precision of a rule. We have not modified the way they interpret missing features during learning. However, after the learning is complete, the rules learned by both approaches are scored by interpreting the missing data either aggressively or conservatively as described in the previous section.

We ran both systems on synthetic data generated using different parameters that control the fraction of missing data and the minimum support threshold needed for promotion. In Figures 1(a) and 1(b), the X and Y-axes show the fraction of missing predicates and the support threshold for the novelty mention model and aggressive scoring of rules for FOIL and FARMER. On the Z-axis is the accuracy of predictions on the missing data, which is the fraction of the total number of initially missing entries that are correctly imputed. We can see that aggressive scoring of rules with the novelty mention model performs very well even for large numbers of missing values for both FARMER and FOIL. FARMER's performance is more robust than FOIL's because FARMER learns all correct rules and uses whichever rule fires. For example, in the NFL domain, it could infer `gameWinner` from `gameLoser` or `teamSmallerScore` or `teamGreaterScore`. In contrast, FOIL's covering strategy prevents it from learning more than one rule if it finds one perfect rule. The results show that FOIL's performance degrades at larger fractions of missing data and large support thresholds.

Figures 1(c) and 1(d) show the accuracy of prediction vs. percentage of missing predicates for each of the mention models and the scoring methods for FARMER and FOIL for a support threshold of 120. They show that agressive scoring clearly outperforms conservative scoring for data generated using the novelty mention model. In FOIL, aggressive scoring also seems to outperform conservative scoring on the dataset generated by the random mention model at high levels of missing data. In FARMER, the two methods perform similarly. However, these results should be interpreted cautiously as they are derived from a single dataset which enjoys deterministic rules. We are working towards a more robust evaluation in multiple domains as well as data extracted from natural texts.

4 Ensemble Co-learning with an Implicit Mention Model

One weakness of Multiple-predicate Bootstrapping is its high variance especially when significant amounts of training data are missing. Aggressive evaluation of rules in this case would amplify the contradictory conclusions of different rules. Thus, picking only one rule among the many possible rules could lead to dangerously large variance.

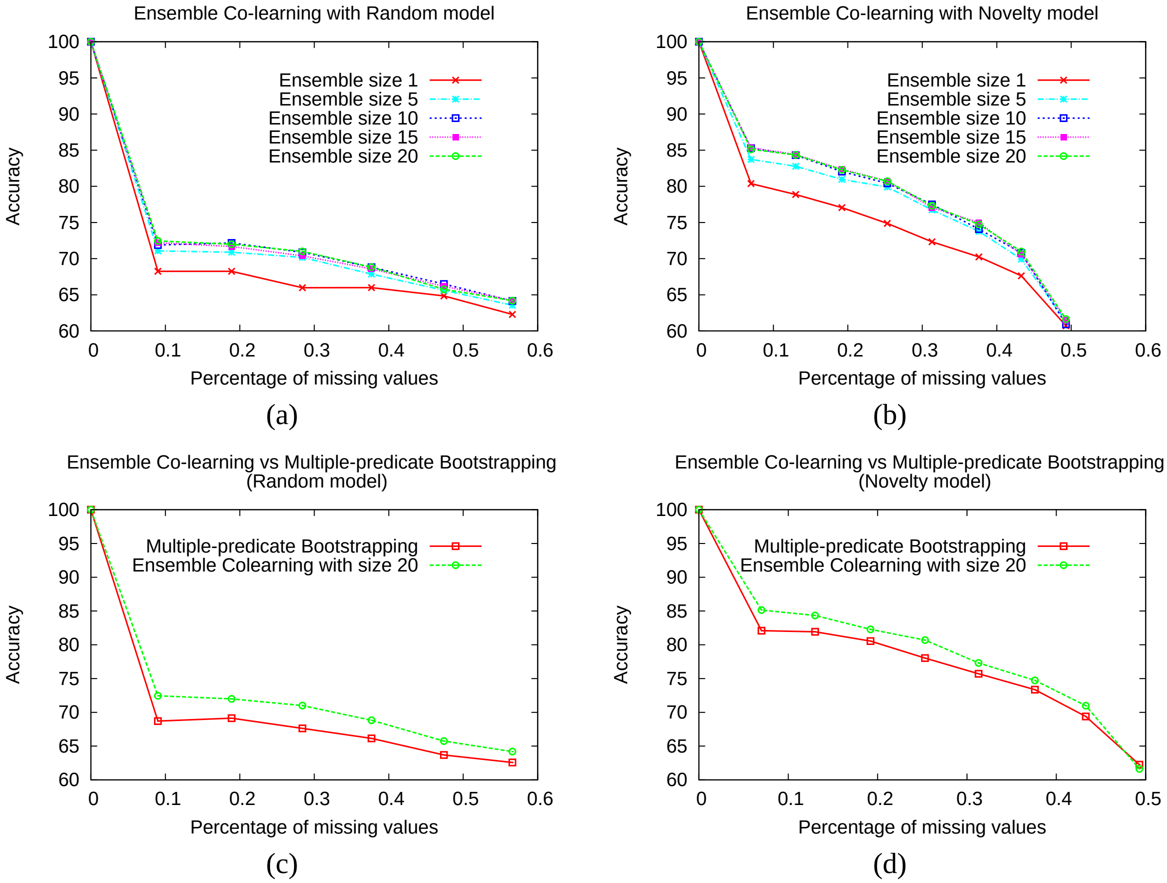
One way to guard against the variance problem is to use an ensemble approach. In this section we test the hypothesis that an ensemble approach would be more robust and exhibit less variance in the context of learning from incomplete examples with an implicit mention model. For the experiments in this section, we employ a decision tree learner that uses a distributional scoring scheme to handle missing data as described in (Quinlan, 1986).

While classifying an instance, when a missing value is encountered, the instance is split into multiple pseudo-instances each with a different value for the missing feature and a weight corresponding to the estimated probability for the particular missing value (based on the frequency of values at this split in the training data). Each pseudo-instance is passed down the tree according to its assigned value. After reaching a leaf node, the frequency of the class in the training instances associated with this leaf is returned as the class-membership probability of the pseudo-instance. The overall estimated probability of class membership is calculated as the weighted average of class membership probabilities over all pseudo-instances. If there is more than one missing value, the process recurses with the weights combining multiplicatively. The process is similar at the training time, except that the information gain at the internal nodes and the class probabilities at the leaves are calculated based on the weights of the relevant pseudo-instances.

We use the *confidence level* for pruning a decision tree as a proxy for support of a rule in this case. By setting this parameter to different values, we can obtain different degrees of pruning.

Experimental Results: We use the Congressional Voting Records[2] database for our experiments. The (non-text) database includes the party affiliation and votes on 16 measures for each member of the U.S House Representatives. Although this database (just like the NFL database) is complete, we generate two different synthetic versions of it to simulate the extracted facts from typical news stories on this topic. We use all the instances including those with unknown values for training, but do not count the errors on these unknown values. We ex-

[2] http://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records



(a)

(b)

(c)

(d)

Figure 2: Results for (a) Ensemble co-learning with Random mention model (b) Ensemble co-learning with Novelty mention model (c) Ensemble co-learning vs Multiple-predicate Bootstrapping with Random mention model (d) Ensemble co-learning vs Multiple-predicate Bootstrapping with Novelty mention model

periment with two different implicit mention models: Random and Novelty. These are similar to those we defined in the previous section. In the Random mention model, each feature in the dataset is omitted independently with a probability p. Since we don't know the truely predictive rules here unlike in the football domain, we learn the novelty model from the complete dataset. Using the complete dataset which has n features, we learn a decision tree to predict each feature from all the remaining features. We use these n decision trees to define our novelty mention model in the following way. For each instance in the complete dataset, we randomly pick a feature and see if it can be predicted from all the remaining features using the predictive model. If it can be predicted, then we will omit it with probability p and mention it otherwise. We use different bootstrap samples to learn the ensemble of trees and impute the values using a majority vote. Note that, the decision tree cannot always classify an instance successfully. Therefore, we will impute the values only if the count of majority vote is greater than some minimum threshold (margin). In our experiments, we use a margin value equal to half of the ensemble size and a fixed support of 0.3 (i.e., the confidence level for pruning) while learning the decision trees. We employ J48, the WEKA version of Quinlan's C4.5 algorithm to learn our decision trees. We compute the accuracy of predictions on the missing data, which is the fraction of the total number of initially missing entries that are imputed correctly. We report the average results of 20 independent runs.

We test the hypothesis that the Ensemble Co-learning is more robust and exhibit less variance in the context of learning from incomplete examples when compared to Multiple-predicate Bootstrapping. In Figures 2(a)-(d), the X and Y-axes show the percentage of missing values and the prediction accuracy. Figures 2(a) and (b) shows the behavior with different ensemble sizes (1, 5, 10, 15 and 20) for both Random and Novelty mention model. We can see that the performance improves as the ensemble size grows for both random and novelty models. Figures 2(c) and (d) compares Multiple-predicate Bootstrapping with the best results over the different ensemble sizes. We can see that Ensemble Co-learning outperforms Multiple-predicate Bootstrapping.

5 Discussion

Learning general rules by reading natural language texts faces the challenges of radical incompleteness and systematic bias. Statistically, our notion of incompleteness corresponds to the Missing Not At Random (MNAR) case, where the probability of an entry being missed may depend on its value or the values of other observed variables (Rubin, 1976).

One of the key insights of statisticians is to build an explicit probabilistic model of missingness, which is captured by our mention model and extraction model. This missingness model might then be used in an Expectation Maximization (EM) approach (Schafer and Graham, 2002), where alternately, the missing values are imputed by their expected values according to the missingness model and the model parameters are estimated using the maximum likelihood approach. Our "Multiple-predicate Bootstrapping" is loosely analogous to this approach, except that the imputation of missing values is done implicitly while scoring the rules, and the maximum likelihood parameter learning is replaced with the learning of relational if-then rules.

In the Multiple Imputation (MI) framework of (Rubin, 1987; Schafer and Graham, 2002), the goal is to reduce the variance due to single imputation by combining the results of multiple imputations. This is analogous to Ensemble Co-learning, where we learn multiple hypotheses from different bootstrap samples of the training data and impute values using the weighted majority algorithm over the ensemble. We have shown that the ensemble approach improves performance.

References

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of COLT*, pages 92–100. Morgan Kaufmann Publishers.

R. M. Cameron-Jones and J. R. Quinlan. 1994. Efficient top-down induction of logic programs. In *ACM SIGART Bulletin*.

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010)*.

R. Caruana. 1997. Multitask learning: A knowledge-based source of inductive bias. *Machine Learning*, 28:41–75.

William W. Cohen. 2000. WHIRL: a word-based information representation language. *Artif. Intell.*, 118(1-2):163–196.

Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom M. Mitchell, Kamal Nigam, and Seán Slattery. 2000. Learning to construct knowledge bases from the world wide web. *Artif. Intell.*, 118(1-2):69–113.

Luc DeRaedt and Nada Lavraøc. 1996. Multiple predicate learning in two inductive logic programming settings. *Logic Journal of the IGPL*, 4(2):227–254.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Commun. ACM*, 51(12):68–74.

Siegfried Nijssen and Joost N. Kok. 2003. Efficient frequent query discovery in FARMER. In *PKDD*, pages 350–362.

Ross Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1(1):81–106.

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

D. B. Rubin. 1976. Inference and missing data. *Biometrika*, 63(3):581.

D. B. Rubin. 1987. *Multiple Imputation for non-response in surveys*. Wiley New York.

Maytal Saar-Tsechansky and Foster Provost. 2007. Handling missing values when applying classification models. *Journal of Machine Learning Research*, 8:1625–1657.

J. L. Schafer and J. W. Graham. 2002. Missing data: Our view of the state of the art. *Psychological methods*, 7(2):147–177.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL*.