Multistrategy Learning for Information Extraction

Dayne Freitag
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
dayne@cs.cmu.edu

Abstract

Information extraction (IE) is the problem of filling out pre-defined structured summaries from text documents. We are interested in performing IE in non-traditional domains, where much of the text is often ungrammatical, such as electronic bulletin board posts and Web pages. We suggest that the best approach is one that takes into account many different kinds of information, and argue for the suitability of a multistrategy approach. We describe learners for IE drawn from three separate machine learning paradigms: rote memorization, term-space text classification, and relational rule induction. By building regression models mapping from learner confidence to probability of correctness and combining probabilities appropriately, it is possible to improve extraction accuracy over that achieved by any individual learner. We describe three different multistrategy approaches. Experiments on two IE domains, a collection of electronic seminar announcements from a university computer science department and a set of newswire articles describing corporate acquisitions from the Reuters collection, demonstrate the effectiveness of all three approaches.

1 INTRODUCTION

Information extraction (IE) poses the following problem: Suppose each document in a collection describes some entity or event drawn from a semantically coherent domain. For example, the collection may consist of newswire articles describing terrorist attacks in Latin America, or of personal home pages from a university computer science departments. Given a document from the collection and a set of questions defined for the domain, find the answer to each question in the form of a fragment of text from the document. In the case of articles on terrorism, the object might be to find the title of the group responsible for the attack, the instrument of the attack, and the victim's name; from home pages, we might seek to extract the owner's name, home address, and university affiliation.

There are many possible uses for a successful IE system. As a front end, an IE system can enable database mining and knowledge discovery in textual domains, where such processing would otherwise be limited or impossible. In hypertext, it can support directed and efficient automatic navigation. It can serve as a source of high-quality features for document categorization. And the output of an IE system can be viewed as a kind of succinct and directed summarization.

Although traditional IE (Cowie & Lehnert, 1996) concentrates on domains consisting of grammatical prose, we are interested in extracting information from "messy" text, such as Web pages, email, and finger plan files. Our goal is the development of machine learning methods for such domains. To perform well, these methods must be prepared to exploit non-linguistic information, such as stock phrases, document formatting, meta-textual structure (e.g., in HTML), and term frequency statistics.

Several learning IE systems have been proposed which are also targeted at such domains (Soderland, 1997) (Califf & Mooney, 1997) (Kushmerick, 1997). These previous investigations all take a single approach or attack a particular kind of domain. However, given the wealth of information in a typical document and the difficulty of adequately representing this information for learning, we surmise that no individual learning approach is best for all IE problems. An individual learner embodies biases that make it more suitable for some kinds of information and aspects of a problem than for others. A statistical learner like Naive Bayes, for example, is useful for problems in which each feature contributes some evidence toward the determination of class membership, and in which violations of the independence assumption do not predominate. It is less suitable for problems involving elaborate feature sets, in which some features are abstractions or combinations of others (i.e., where the independence assumption is directly violated). Symbolic learners, on the other hand, work quite well for problems with elaborate feature sets, especially for those classes expressible in logical terms using a small subset of features. These considerations suggest at *multistrategy* approach.

Multistrategy learning is an attempt to devise systems which, by employing multiple constituent learners, which are typically drawn from diverse paradigms, achieve performance superior to any single learner (Michalski & Tecuci, 1994). The bulk of emphasis in past research in this area has been on systems which combine analytical and empirical techniques. Our work, however, is an example of what has been called "empirical multistrategy learning" (Domingos, 1996). All constituent learners are inductive, each designed to solve the IE problem individually. Elsewhere we have shown that heuristic combination of two learners from different paradigms can yield substantial performance improvements for the IE problem (Freitag, 1997). Here, we ask how we might profitably combine component learners by treating them as black boxes. This approach has been called "meta-learning" in the literature (Chan & Stolfo, 1993). Although we might expect a heuristic combination to achieve better performance, there are clear advantages to the meta-learning approach. It is modular and flexible, making no assumptions about the design of component learners or the number of learners available.

In this paper, we introduce three machine learning algorithms for IE, each drawn from a different paradigm and each suitable for particular kinds of IE problems. Next, we describe three ways of combining the basic learners, all variations of the meta-learning idea. Finally, we describe a set of experiments on two IE domains.

2 LEARNING TO EXTRACT

In the simplest version of the information extraction problem, a single set of questions is applied to each document in a domain, and a single text fragment is sought as the answer to each question. We call a single question a *field*; the answer fragment from an individual document is a *field instance* or *instantiation*. For example, in a domain consisting of newswire articles describing terrorist attacks, one field might be the *perpetrator* of the attack, and the instantiation of this field in a given article might be "FMLN."

A field can be formalized as a function $\mathcal{F}(D) = (b_b, b_e)$ that maps a document to the boundaries of a text fragment (b_b and b_e are the indexes of the beginning and ending boundary terms, respectively). Given a set of documents in which this mapping is labeled, the goal of a ML system is to learn the function $\widehat{\mathcal{F}}$ that best approximates $\mathcal{F}$. This can be realized in the form of an auxiliary function $\mathcal{G}(D, b_b, b_e) = \mathbf{R} \cup \{\text{nil}\}$, which, given a candidate fragment, either returns a confidence that it is a field instance or declines to issue a confidence (*nil*). The form of $\mathcal{G}$ has a convenient affinity with any number of ML algorithms (the *nil* in its range constitutes a failure to match, for algorithms that include a notion of matching). The three approaches we will discuss, all based on standard ideas from ML, each implement $\mathcal{G}$.

Note that this learning task is only a part of the functionality of a typical participating system at the Message Understanding Conference (MUC) (Cardie, 1997). What we have called *fields* correspond to *slots* in the MUC setting. A slot is a component of a larger structure, called a *template*, which summarizes the relevant information contained in a document. In addition to the slot-filling task, which we address here, the more general MUC problem includes tasks such as document relevance determination, discourse analysis, and template merging. Thus, our results are best regarded as a piece of the larger IE puzzle.

2.1 ROTE LEARNING

Perhaps the simplest possible learning approach to the IE problem is to memorize field instances verbatim. Presented with a novel document, this memorizing learner simply matches text fragments against its "learned" dictionary, saying "field instance" to any matching fragments and rejecting all others.

As a slightly more sophisticated approach, we can estimate the probability that the matched fragment is indeed a field instance. The dictionary learner we experiment with here, which we call Rote, does exactly this. Training Rote involves scanning the training corpus and storing all distinct field instances verbatim in its dictionary. Dictionary construction is followed by a second pass through the training corpus. For each text fragment in its dictionary, Rote counts the number of times it appears as a field instance (pos) and the number of times it occurs over all (tot). During test, Rote's confidence in a prediction is the value $(pos+1)/(tot+2)$, i.e., a Laplace estimate that the matching fragment is genuine.

This approach, simple as it is, is nevertheless surprisingly applicable in a wide variety of domains. Its confidence, moreover, correlates well with actual probability of correctness. Because of this, even low-confidence predictions are potentially useful.

2.2 TERM-SPACE LEARNING

It is straightforward to adapt ideas from document classification to the IE setting. A simple mapping might transform every field instance into a miniature "document" and apply "bag-of-words" algorithms directly, such as Rocchio with TFIDF term weighting or Naive Bayes. Such an approach could be viewed as a generalization of Rote.

In contrast with document classification, however, positive examples in an IE setting always occur embedded within some larger context. This context is often critical in disambiguating field instances from other fragments. Although it is hard to exploit contextual regularities by memorizing, statistical approaches are well suited for this.

We base our bag-of-words learner, which we call Bayes, on the Naive Bayes algorithm, as used in document classification and elsewhere (originally in (Maron, 1961)). Each fragment of text in a document (of appropriate size) is regarded as a competing hypothesis. Given a document, we want to find the most likely hypothesis (the fragment most likely to be a field instance). Bayes Rule tells us how to maintain our belief in a set of disjoint hypotheses (H_i) in reaction to observed data (D):

$$\Pr(H_i|D) = \frac{\Pr(D|H_i)\Pr(H_i)}{\sum_{j=1}^{n}\Pr(D|H_j)\Pr(H_j)}$$

As in Naive Bayes as used elsewhere, the important terms to estimate are $\Pr(H_i)$ (the *prior* probability) and $\Pr(D|H_i)$ (the conditional *data* probability).

We assume a hypothesis takes the form, "the field instance starts at token s and is k tokens long" (let $H_{s,k}$ represent such a hypothesis). In other words, a single hypothesis consists of two parts, *position* and *length*. We can estimate the probability of a particular position or length from training data. In our implementation we treat these two estimates as independent, which is different from the typical Naive Bayes data independence assumption, but similar in spirit. Thus, our prior $\Pr(H_{s,k})$ is simply the product of $\Pr(\text{position} = s)$ and $\Pr(\text{length} = k)$.

Bayes's data likelihood estimate, $\Pr(D|H_{s,k})$, is based on the terms that occur in and around the text fragment to which $H_{s,k}$ corresponds. This estimate is formed in a way similar to Naive Bayes for document classification (a product of individual term estimates), but with a few modifications for the IE setting. In particular, a context window parameter w is set prior to training, and the w tokens on either side of a fragment are used to form the estimate, in addition to the in-field tokens. The algorithm is described in greater detail elsewhere (Freitag, 1997).

2.3 RELATIONAL LEARNING

Both Bayes and Rote are hobbled by their inability to take into account anything but simple term frequency statistics. It may be the case, however, that the information needed to perform information extraction comes in other forms. More abstract clues may be important, such as linguistic syntax, document layout, or simple orthography. In addition, statistical approaches like Bayes work by summing all available evidence, whereas in IE a more fruitful approach may involve identifying simple patterns that serve to distinguish sub-classes of a field.

Symbolic learning algorithms from the "covering" family form hypotheses that match such data spaces well. Previous research has shown the effectiveness of such methods for the IE problem (Soderland, 1996) (Califf & Mooney, 1997). Our relational learner, called SRV, is a variant of FOIL (Quinlan, 1990). Its example space consists of all text fragments from the training document collection as long (in number of tokens) as the smallest field instance in the training corpus but no longer than the largest. A negative example is any fragment that is not tagged as a field instance. Note that this includes fragments that contain, are contained by, and overlap with field instances.

Induction proceeds as with FOIL: Starting with a null rule that matches all examples not covered by previously learned rules, SRV greedily adds predicates using FOIL's information gain metric. In addition to the tagged document collection, SRV takes as input a set of features to use in conducting search. These features come in two varieties, *simple* features, which map from an individual token to an arbitrary value (e.g., capitalized? or noun?), and *relational* features, which map from a token to another token (e.g., next-token or subject-verb).

An individual predicate in SRV belongs to one of a few predefined types:

- length(Relop N): The number of tokens in a fragment is less than, greater than, or equal to some integer.

- some(Var Path Feat Value): This is a feature-value test for some token in the sequence (e.g., "the fragment contains some token that is capitalized"). One argument to this predicate is a variable. For a rule to match a text fragment, each distinct variable in a rule (used in this or either of the position predicates below) must bind to a distinct token in the fragment.

- every(Feat Value): Every token in a fragment passes some feature-value test (e.g., "every token in the fragment is non-numeric").

- position(Var From Relop N): This constrains the position of a token bound by a some-predicate in the current rule. The position is specified relative to the beginning or end of the sequence.

- relpos(Var1 Var2 Relop N): This constrains the ordering and distance between two tokens bound by distinct variables in the current rule.

Relational features are used only in the Path argument to the some predicate. This argument can be empty, in which case the some predicate is asserting a feature-value test for a token actually occurring within a field, or it can be a list of relational features. In the latter case, it is positing both a relationship about a field token with some other nearby token, as well as a feature-value for the other token. For example, the assertion:

```
some(?A [prev-token prev-token] capitalized true)
```

amounts to the English statement, "There is some token preceded by a capitalized token two tokens back."

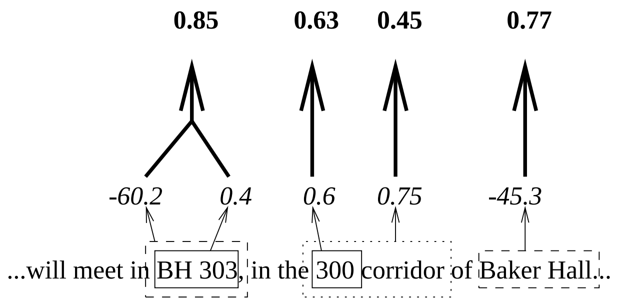


Figure 1: Hypothetical Extraction of a Seminar Location. Each box style is intended to represent a different learner. By combining evidence from multiple learners, we can correct for the mistakes of individual learners.

In order to enable SRV to return confidences with its predictions, training is followed by a validation step. Rather than train on the entire training collection, we set aside a fraction of the documents (one-third here) for validation. With each rule learned by SRV we store its performance on the hold-out set. From this performance we estimate a rule's actual accuracy. The confidence of a prediction made by SRV is formed from the estimated accuracy of matching rules. For additional details on SRV, please refer to (Freitag, 1998).

3 COMBINING LEARNERS

Certain features of the IE problem make it particularly amenable to a multistrategy approach. Among these are the following:

- **Examples have multiple representations.** Because documents and text fragments are "natural" objects which must be mapped to appropriate representations for learning, multiple mappings are possible. Although some information is necessarily lost in any one mapping, we can hope that taking multiple views of a document will permit better overall performance.

- **The problem is essentially Boolean.** As outlined above, performing extraction can be reduced to the task of accepting or rejecting candidate text fragments. Consequently, we can gauge a learner's performance on validation documents in an attempt to model the relationship between prediction confidence and probability of correctness.

- **Each document is a case study.** In contrast with a traditional classification problem, each per-

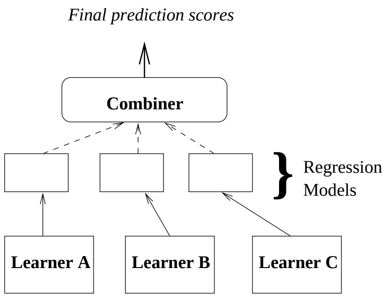


Figure 2: The Basic Combination Scheme. Regression models based on learner performance on hold-out sets are used to map raw confidence scores to probabilities. The combiner uses these probabilities to order all predictions.

formance unit, a document, is a collection of test problems. *Overgeneration*, the problem of saying *yes* to too many text fragments, can be regarded as an asset when multiple learners are available. It both affords more data for our attempt to model a learner's usefulness, and holds forward the hope that the poor predictions of a single learner can be corrected by checking them against those of other learners.

Figure 1 shows a hypothetical excerpt from a seminar announcement and how such correction might take place.

3.1 BASIC COMBINATION METHOD

Within the constraint that all learners assign a confidence to any predictions they make (any fragments they accept), a wide range of behaviors is possible. In particular, for a number of reasons, we cannot assume that the confidences bear any resemblance to true probability of correctness, or even that they are comparable across learners. Bayes's confidences are large negative log probabilities, for example.

We *do* assume, however, that probability of correctness increases with increasing confidence for all learners. The basic idea, therefore, is to attempt to compute a mapping for each learner from confidence to probability of correctness. Figure 2 shows this in outline. The specific steps involved are:

1. **Validate performance on a hold-out set.** Reserve a part of the training set for validation. After training each learner, store its predictions, with confidences, on the hold-out set.

2. **Use regression to map confidences to probabilities.** Based on the learner's performance on the hold-out set, attempt to model how its performance varies with confidence. What is modeled, and the kind of regression used, depends on the combination method.

3. **Use the regression models and calculated probabilities to make the best choice on the test set.**

We experimented with three basic methods of combination. The first two, which we will call Max and Prob, both attempt to work with regression models that map directly from confidence to probability of correctness. The third, which we will call CBayes, uses Bayes Rule to make combination decisions.

3.2 REGRESSION TO ESTIMATE CORRECTNESS

If a learner's confidence numbers are meaningful, then the probability that a prediction is correct will increase with increasing confidence. We use linear regression to model the rate at which this probability increases. For each prediction made we create a datapoint (x, y), where x is the prediction confidence, and y is 1, if the prediction was correct (the corresponding fragment *was* a field instance), else 0.

The result is a line equation which we use directly to map from learner confidence to probability of success. Both Max and Prob use the resulting estimates to arbitrate among multiple learners' predictions for a document. Estimates are computed for each learner's predictions, and the prediction with the highest estimate is chosen as the top combined prediction. The two methods differ only in how they handle the case in which multiple learners offer predictions for the same text fragment. In such an event, Max simply takes the larger estimate as the probability that the fragment is a field instance.

We believe, however, that the fact that two or more learners agree on a prediction provides more information than either prediction alone. Indeed, if we assume that two probability estimates of an event, P_a and P_b, are independent, then the combined probability is the probability that they are not both wrong, i.e., $1-(1-P_a)(1-P_b)$. Prob's estimate is based on this assumption. Given a set of probability estimates P_i, its estimate for the combined probability is $1-\prod_i(1-P_i)$.

3.3 BAYESIAN PREDICTION COMBINATION

Although Prob may exploit the availability of predictions from multiple learners better than Max, it still leaves something to be desired. In particular, it ignores some of the available information, such as the frequency with which a learner tends to predict at a given confidence level and any notion of prior probabilities.

For our final combination method, we attempt to apply Bayes Rule, which tells us how to maintain our probability estimates in response to incoming data. Using Bayes Rule offers two advantages over Prob: It allows us to incorporate priors into our estimates, and it tells us how to maintain our hypothesis space so that the resulting estimates are closer to true probabilities—an advantage in terms of the accuracy-coverage trade-off.

Here, a hypothesis H_i takes the form, "*the fragment at this place in the document is a field instance.*" Let $P_{ai} = C$ be the event, *Learner A predicted fragment i is a field instance with confidence C.* For each fragment i chosen by any of the learners, we maintain two hypotheses explicitly, H_i and $\neg H_i$. Individual learner predictions $P_{ai} = C$ are treated as events which cause us to update hypotheses. We want, therefore, to model $\Pr(P_{ai} = C|H_i)$ and $\Pr(P_{ai} = C|\neg H_i)$. It is more convenient, however, to model the event $P_{ai} >= C$, i.e., the probability of a prediction with confidence *at least* C. Modeling the cumulative probability yields better statistics and allows us to avoid the arbitrary decisions inherent in binning.

We use exponential regression to model these two probabilities, i.e., we perform linear regression on pairs of the form $(x, \log(y))$, where x is a confidence level, and y is the cumulative probability of seeing a prediction for a fragment given that it either is or is not a field instance. As an example, consider the problem of creating the "positive" model $\Pr(P_{ai} >= C|H_i)$ for some learner A. Let F be the total number of field instances in the validation set, and let $G_a(C)$ be the number of field instances identified by Learner A with predictions having confidence equal to or greater than C. For every prediction made by Learner A, we add a regression datapoint $(x, \log(y))$, where x is the confidence of the prediction and $y = G_a(x)/F$. The "negative" model $\Pr(P_{ai} >= c|\neg H_i)$ is constructed in the same way, except over non-field-instance fragments—any fragment in the validation set identified by any of the learners. We settled on exponential regression empirically, but it is easy to see why it works better than linear regression. Low-confidence predictions tend to be more frequent than high-confidence ones, obeying something like Zipf's Law.

Table 1: Accuracy-Coverage Results for the Seminar Announcement Domain.

	speaker		location	
	Acc	*Cov*	*Acc*	*Cov*
Rote	57.4 ± 8.8	11.8	89.5 ± 2.2	64.9
Bayes	36.1 ± 3.5	70.8	59.6 ± 2.8	98.7
SRV	60.4 ± 3.0	96.6	75.9 ± 2.6	92.3
Max	59.8 ± 3.0	98.8	75.6 ± 2.5	99.7
Prob	60.8 ± 3.0	98.8	76.0 ± 2.5	99.7
CBayes	62.5 ± 3.0	98.8	75.6 ± 2.5	99.7
	stime		etime	
Rote	73.7 ± 2.5	99.6	75.1 ± 3.7	95.4
Bayes	98.2 ± 0.7	100.0	96.1 ± 1.6	99.6
SRV	98.6 ± 0.7	99.8	94.1 ± 2.0	98.4
Max	96.6 ± 1.0	100.0	93.6 ± 2.0	100.0
Prob	99.3 ± 0.5	100.0	95.4 ± 1.7	100.0
CBayes	99.3 ± 0.5	100.0	96.3 ± 1.6	100.0

With each prediction, we use the two models associated with a learner to adjust the posterior probabilities of the two mutually exclusive hypotheses regarding the affected fragment, always normalizing so they sum to 1.

4 EXPERIMENTS

We experimented with data from two IE domains. One consists of 485 postings to electronic bulletin boards, which describe upcoming seminars in a university environment. The earliest of these announcements dates to October, 1982; the most recent was posted in August, 1995. We manually tagged these announcements for four fields: `speaker`, `location`, `stime` (start time), and `etime` (end time). The other domain is a collection of 600 newswire articles on corporate acquisitions from the Reuters data set (Lewis, 1992). We defined nine fields for this domain and manually annotated the collection to identify all instances of them. We selected five of the fields for these experiments: `acquired` (the official name of the company or resource that is being purchased), `purchaser`, `acqabr` (the short name for `acquired` used in the body of the article), `purchabr`, and `dlramt` (the price paid).

The performance numbers we report here are the result of five-fold experiments in each domain. In each iteration the datasets were randomly divided into two partitions of equal size. One partition was used for training, the other for testing.

Table 2: Accuracy-Coverage Results for the Acquisition Domain.

	acquired		purchaser	
	Acc	*Cov*	*Acc*	*Cov*
Rote	56.1 ± 5.6	20.5	47.5 ± 5.6	22.3
Bayes	22.4 ± 2.2	96.4	41.4 ± 2.6	99.7
SRV	41.1 ± 2.6	96.0	49.7 ± 2.7	97.8
Max	43.4 ± 2.5	99.8	51.4 ± 2.7	99.9
Prob	45.0 ± 2.5	99.8	53.2 ± 2.7	99.9
CBayes	45.8 ± 2.5	99.8	54.7 ± 2.6	99.9
	acqabr		purchabr	
Rote	31.7 ± 4.2	43.8	24.7 ± 4.1	38.5
Bayes	33.1 ± 2.8	99.7	52.0 ± 2.9	99.9
SRV	45.0 ± 3.0	99.8	54.0 ± 2.9	99.6
Max	42.7 ± 2.9	100.0	57.4 ± 2.9	100.0
Prob	47.6 ± 3.0	100.0	61.0 ± 2.9	100.0
CBayes	47.2 ± 3.0	100.0	60.0 ± 2.9	100.0
	dlramt			
Rote	77.2 ± 4.7	48.1		
Bayes	62.2 ± 4.3	76.9		
SRV	74.4 ± 3.5	90.1		
Max	72.0 ± 3.5	95.5		
Prob	73.1 ± 3.5	95.5		
CBayes	70.2 ± 3.5	95.5		

Table 3: *F1* Scores. Two scores are shown for each result: *Full*, the F1 score for the accuracy-coverage results reported in Tables 1 and 2, and *Peak*, the highest F1 score along the full accuracy-coverage curve.

	speaker		location		stime	
	Full	*Peak*	*Full*	*Peak*	*Full*	*Peak*
Rote	19.6	19.6	75.3	75.3	84.7	84.7
Bayes	47.8	48.0	74.3	75.1	99.1	99.1
SRV	74.3	74.3	83.3	83.3	99.2	99.2
Max	74.5	74.5	86.0	86.4	98.3	98.3
Prob	75.3	75.3	86.3	86.6	99.6	99.6
CBayes	76.6	76.6	86.0	86.0	99.6	99.6
	etime		acquired		purchaser	
Rote	84.0	84.0	30.0	30.0	30.3	30.3
Bayes	97.8	97.8	36.4	38.3	58.5	59.3
SRV	96.2	96.2	57.5	58.3	65.9	66.4
Max	96.7	96.7	60.5	61.2	67.9	68.0
Prob	97.6	97.6	62.0	62.7	69.5	69.7
CBayes	98.1	98.1	62.8	63.2	70.7	71.1
	acqabr		purchabr		dlramt	
Rote	36.8	37.2	30.1	30.8	59.3	59.3
Bayes	49.7	52.8	68.4	68.6	68.8	68.8
SRV	62.0	62.0	70.0	70.2	81.5	81.5
Max	59.8	59.8	72.9	72.9	82.1	82.1
Prob	64.5	64.5	75.8	75.8	82.8	82.8
CBayes	64.1	64.1	75.0	75.0	80.9	80.9

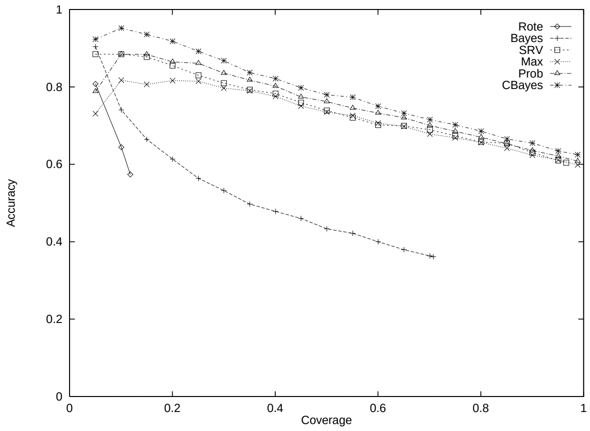
A third of the training set, randomly selected, was set aside for validation. Each learner was trained on the remaining two-thirds, and tested on the validation set. Following this validation step, each learner was again trained on the entire training set and tested on the test set. The goal of the combining methods was to use performance results on the validation set to arbitrate among predictions on the test set.

The performance of all methods is summarized in Table 1, for the seminar announcement fields, and Table 2, for the acquisition fields. The unit of measurement here, as elsewhere in this paper, is a document. When assessing a learner's performance for a single document, we can distinguish among four basic outcomes: no prediction from the learner, prediction on a document lacking a field instance (*spurious*), top prediction is incorrect (*wrong*), and top prediction is correct (*correct*). The coverage column (*Cov*) shows for what fraction of those documents containing a field instance a learner actually made a prediction. The number in the accuracy column (*Acc*) shows the fraction of correct predictions over documents for which the learner made a prediction *and* which contained a field instance, i.e., it ignores spurious predictions. Note that if any single learner makes a spurious prediction, all combining methods also make one, since they are limited to ordering the predictions made by actual learners. Thus, counting spurious predictions as errors, while generally appropriate, tends to obscure the differences between the learners and the combining methods.

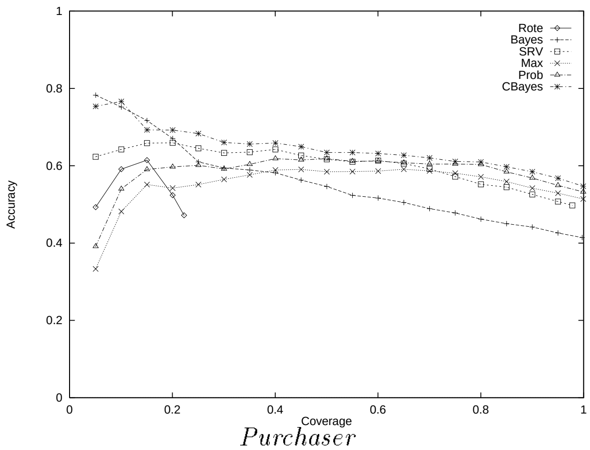
Both the accuracy and coverage values should be considered together. There are cases, for example, where the accuracy number makes Rote look like the strongest extraction method. Its accuracy, however, is usually measured over a much smaller number of documents. While it can typically recognize a fraction of field instances with reasonable accuracy (especially locations), it does not stand up well to overall comparison with the other learners. For convenience in comparing systems, it is common in information retrieval and information extraction to combine precision and recall into a single, summary number, called the *F-measure*:

$$F = \frac{(\beta^2 + 1.0)PR}{(\beta^2 P) + R}$$

The parameter β determines how much to favor recall over precision. Researchers in information extraction frequently report the F1 score of a system ($\beta = 1$), which weights precision and recall equally. We can do the same with our accuracy-coverage results. Table 3



Speaker



Purchaser

Figure 3: Plots of accuracy vs. coverage for all methods on two fields, `speaker` and `purchaser`.

shows the F1 scores for all learners and fields.

For the `purchabr` field there is clear statistical separation between the best individual learner (SRV) and the top two combining methods (Prob and CBayes). Note, as Table 3 makes clear, that even in the cases where the difference is less apparent, the combining methods tend to outperform the best individual method at *higher* coverage levels. Among the three combining methods there is not one case of statistical separation, but across all fields a clear picture emerges in which Prob and CBayes are better than Max. Note that even in cases where a combining method performs only as well as the best individual learner, it has served a valuable purpose—that of relieving us of the requirement of choosing a single learner. If a combining method can do this in most cases, while providing added value in a few, we account it a clear success.

Perhaps more interesting than summary statistics are *accuracy-coverage* (similar to *precision-recall*) graphs. Each point x along the horizontal axis represents the $x\%$ most confident predictions. The vertical value at this point is the accuracy of these predictions. If the accuracy-coverage curve declines monotonically, it suggests that the learner's confidence correlates well with actual accuracy.

Table 4: Overlap in Learner Behavior for the `Speaker` Field. Numbers are the probability that column learner predicted correctly, given that the row learner predicted correctly.

	Rote	Bayes	SRV	Max	Prob	CBayes
Rote	1	0.81	0.81	0.90	0.96	0.97
Bayes	0.22	1	0.68	0.86	0.89	0.86
SRV	0.09	0.30	1	0.93	0.93	0.97
Max	0.10	0.37	0.92	1	0.99	0.98
Prob	0.11	0.38	0.90	0.98	1	0.98
CBayes	0.11	0.35	0.92	0.93	0.95	1

Figure 3 shows the accuracy-coverage curves for all methods on two of the fields. The `speaker` and `purchaser` fields are the ones for which CBayes does best. These graphs make clear what the summary statistics cannot: That combining learners allows us to make better accuracy-coverage judgments than we can with a single learner. The anomalous high-confidence behavior of Prob and Max in the `purchaser` curve may be due to an over-reliance on Rote, which has similar behavior. Note that the high-confidence (low-coverage) end of the curve is the part with the least statistical certainty. Also, although CBayes appears better than any *individual* learner, an examination of the graphs for all fields does not support a preference of it over Prob, or vice versa. There are cases where CBayes has high-confidence difficulties similar to those shown here for Prob and Max. We believe that better regression models will mitigate some of these phenomena.

The strength of a meta-learning approach depends on the mutual independence of the constituent learners. Table 4 shows where some of the power of combining learners comes from on the `speaker` field, a relatively challenging task. In this table we ask the question, given that Learner A has predicted correctly on some document, what is the probability that Learner B will also predict correctly? The number in entry (i, j) is the fraction of all documents correctly handled by method i which method j also correctly handled. Based on this table, it is evident that Rote and Bayes are more closely related to each other than either to SRV.

The column for a combining method allows us to infer which learners it depends on most for its performance. It appears from this that all three methods rely more on Rote than on Bayes. We would hope to see this, based on Figure 3, since the few Rote predictions that are available for this field tend to have higher accuracy than most Bayes predictions. It is also gratifying that all methods appear to rely heavily on SRV, since it is the best individual learner in this case.

5 CONCLUSION

The experimental results presented here show that multistrategy learning can be useful for the problem of information extraction. We present one form of multistrategy learning, in which the component learners are treated as black boxes and only their reliability, as a function of confidence, is modeled. Nothing in the basic framework requires the information extraction setting or makes any assumptions about the number or structure of component learners. It is only necessary that learners be instrumented to associate a confidence with any prediction they make, something which is already part of the design of many learners, and which can be readily added to others.

We do not claim that the multistrategy results reported here are the best that can be achieved. Many details remain to be filled in, such as how best to conduct validation and which statistical assumptions are appropriate. We have experimented with two kinds of regression to model learner reliability, but would not be surprised if other methods which we have not tried, such as logistic regression or a simple neural network, might afford increased accuracy. We regard this as future work.

It also remains to be seen how these results might be fit into a more traditional information extraction setting, in which slot filling is performed as part of a larger system and as one of several interacting tasks. Still, the approaches described here are immediately applicable to a number of unconventional information extraction problems. And we can begin to see how information extraction from ungrammatical text, and other "natural" problems admitting multiple abstract representations, can be addressed with machine learning methods.

Acknowledgements

This research was supported in part by the DARPA HPKB program under contract F30602-97-1-0215.

References

Califf, M. E., and Mooney, R. J. 1997. Relational learning of pattern-match rules for information extraction. In *Working Papers of ACL-97 Workshop on Natural Language Learning.*

Cardie, C. 1997. Empirical methods in information extraction. *AI Magazine* 18(4):65–79.

Chan, P., and Stolfo, S. 1993. Experiments on multistrategy learning by meta-learning. In *Proceedings of the Second International Conference on Information and Knowledge Management (CIKM 93)*, 314–323.

Cowie, J., and Lehnert, W. 1996. Information extraction. *Communications of the ACM* 39(1).

Domingos, P. 1996. Unifying instance-based and rule-based induction. *Machine Learning* 24(2):141–168.

Freitag, D. 1997. Using grammatical inference to improve precision in information extraction. In *Notes of the ICML-97 Workshop on Automata Induction, Grammatical Inference, and Language Acquisition.*

Freitag, D. 1998. Information extraction from html: Application of a general machine learning approach. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98).*

Kushmerick, N. 1997. *Wrapper Induction for Information Extraction.* Ph.D. Dissertation, University of Washington. Tech Report UW-CSE-97-11-04.

Lewis, D. 1992. *Representation and Learning in Information Retrieval.* Ph.D. Dissertation, Univ. of Massachusetts. CS Tech. Report 91–93.

Maron, M. 1961. Automatic indexing: An experimental inquiry. *Journal of the Association for Computing Machinery* 8:404–417.

Michalski, R., and Tecuci, G., eds. 1994. *Machine Learning: A Multistrategy Approach.* San Mateo, CA: Morgan Kaufmann.

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

Soderland, S. 1996. *Learning Text Analysis Rules for Domain-specific Natural Language Processing.* Ph.D. Dissertation, University of Massachusetts. CS Tech. Report 96-087.

Soderland, S. 1997. Learning to extract text-based information from the world wide web. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining.*