A Hybrid Approach for the Acquisition of Information Extraction Patterns

Mihai Surdeanu, Jordi Turmo, and Alicia Ageno
Technical University of Catalunya
Barcelona, Spain
{surdeanu,turmo,ageno}@lsi.upc.edu

Abstract

In this paper we present a hybrid approach for the acquisition of syntactico-semantic patterns from raw text. Our approach co-trains a decision list learner whose feature space covers the set of all syntactico-semantic patterns with an Expectation Maximization clustering algorithm that uses the text words as attributes. We show that the combination of the two methods always outperforms the decision list learner alone. Furthermore, using a modular architecture we investigate several algorithms for pattern ranking, the most important component of the decision list learner.

1 Introduction

Traditionally, Information Extraction (IE) identifies domain-specific events, entities, and relations among entities and/or events with the goals of: populating relational databases, providing event-level indexing in news stories, feeding link discovery applications, etcetera.

By and large the identification and selective extraction of relevant information is built around a set of domain-specific linguistic patterns. For example, for a "financial market change" domain one relevant pattern is <NOUN fall MONEY to MONEY>. When this pattern is matched on the text "London gold fell $4.70 to $308.35", a change of $4.70 is detected for the financial instrument "London gold".

Domain-specific patterns are either hand-crafted or acquired automatically (Riloff, 1996; Yangarber et al., 2000; Yangarber, 2003; Stevenson and Greenwood, 2005). To minimize annotation costs, some of the latter approaches use lightly supervised bootstrapping algorithms that require as input only a small set of documents annotated with their corresponding category label. The focus of this paper is to improve such lightly supervised pattern acquisition methods. Moreover, we focus on robust bootstrapping algorithms that can handle real-world document collections, which contain many domains.

Although a rich literature covers bootstrapping methods applied to natural language problems (Yarowsky, 1995; Riloff, 1996; Collins and Singer, 1999; Yangarber et al., 2000; Yangarber, 2003; Abney, 2004) several questions remain unanswered when these methods are applied to syntactic or semantic pattern acquisition. In this paper we answer two of these questions:

(1) Can pattern acquisition be improved with text categorization techniques?

Bootstrapping-based pattern acquisition algorithms can also be regarded as incremental text categorization (TC), since in each iteration documents containing certain patterns are assigned the corresponding category label. Although TC is obviously not the main goal of pattern acquisition methodologies, it is nevertheless an integral part of the learning algorithm: each iteration of the acquisition algorithm depends on the previous assignments of category labels to documents. Hence, if the quality of the TC solution proposed is bad, the quality of the acquired patterns will suffer.

Motivated by this observation, we introduce a co-training-based algorithm (Blum and Mitchell, 1998) that uses a text categorization algorithm as reinforcement for pattern acquisition. We show, using both a direct and an indirect evaluation, that the combination of the two methodologies always improves the quality of the acquired patterns.

(2) Which pattern selection strategy is best?

While most bootstrapping-based algorithms follow the same framework, they vary significantly in what they consider the most relevant patterns in each bootstrapping iteration. Several approaches have been proposed in the context of word sense disambiguation (Yarowsky, 1995), named entity (NE) classification (Collins and Singer, 1999), pattern acquisition for IE (Riloff, 1996; Yangarber, 2003), or dimensionality reduction for text categorization (TC) (Yang and Pedersen, 1997). However, it is not clear which selection approach is the best for the acquisition of syntactico-semantic patterns. To answer this question, we have implemented a modular pattern acquisition architecture where several of these ranking strategies are implemented and evaluated. The empirical study presented in this paper shows that a strategy previously proposed for feature ranking for NE recognition outperforms algorithms designed specifically for pattern acquisition.

The paper is organized as follows: Section 2 introduces the bootstrapping framework used throughout the paper. Section 3 introduces the data collections. Section 4 describes the direct and indirect evaluation procedures. Section 5 introduces a detailed empirical evaluation of the proposed system. Section 6 concludes the paper.

2 The Pattern Acquisition Framework

In this section we introduce a modular pattern acquisition framework that co-trains two different views of the document collection: the first view uses the collection words to train a text categorization algorithm, while the second view bootstraps a decision list learner that uses all syntactico-semantic patterns as features. The rules acquired by the latter algorithm, of the form $p \rightarrow y$, where p is a pattern and y is a domain label, are the output of the overall system. The system can be customized with several pattern selection strategies that dramatically influence the quality and order of the acquired rules.

2.1 Co-training Text Categorization and Pattern Acquisition

Given two views of a classification task, co-training (Blum and Mitchell, 1998) bootstraps a separate classifier for each view as follows: (1) it initializes both classifiers with the same small amount of labeled data (i.e. seed documents in our case); (2) it repeatedly trains both classifiers using the currently labeled data; and (3) after each learning iteration, the two classifiers share all or a subset of the newly labeled examples (documents in our particular case).

The intuition is that each classifier provides new, informative labeled data to the other classifier. If the two views are conditional independent and the two classifiers generally agree on unlabeled data they will have low generalization error. In this paper we focus on a “naive” co-training approach, which trains a different classifier in each iteration and feeds its newly labeled examples to the other classifier. This approach was shown to perform well on real-world natural language problems (Collins and Singer, 1999).

Figure 1 illustrates the co-training framework used in this paper. The feature space of the first view contains only lexical information, i.e. the collection words, and uses as classifier Expectation Maximization (EM) (Dempster et al., 1977). EM is actually a class of iterative algorithms that find maximum likelihood estimates of parameters using probabilistic models over incomplete data (e.g. both labeled and unlabeled documents) (Dempster et al., 1977). EM was theoretically proven to converge to a local maximum of the parameters’ log likelihood. Furthermore, empirical experiments showed that EM has excellent performance for lightly-supervised text classification (Nigam et al., 2000). The EM algorithm used in this paper estimates its model parameters using the Naive Bayes (NB) assumptions, similarly to (Nigam et al., 2000). From this point further, we refer to this instance of the EM algorithm as NB-EM.

The feature space of the second view contains the syntactico-semantic patterns, generated using the procedure detailed in Section 3.2. The second learner is the actual pattern acquisition algorithm implemented as a bootstrapped decision list classifier.

The co-training algorithm introduced in this paper interleaves one iteration of the NB-EM algorithm with one iteration of the pattern acquisition algorithm. If one classifier converges faster (e.g. NB-EM typically converges in under 20 iterations, whereas the acquisition algorithms learns new patterns for hundreds of iterations) we continue bootstrapping the other classifier alone.

2.2 The Text Categorization Algorithm

The parameters of the generative NB model, $\hat{\theta}$, include the probability of seeing a given category,

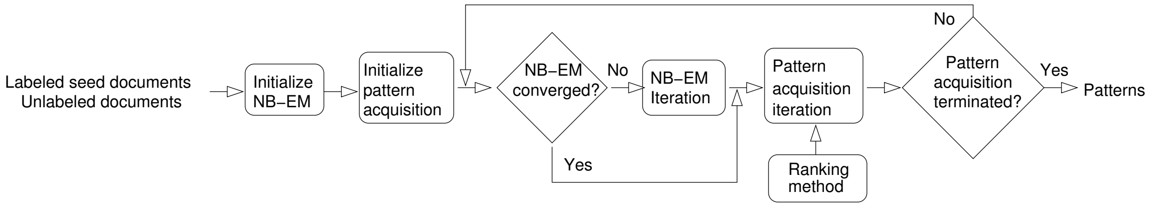


Figure 1: Co-training framework for pattern acquisition.

```
1. Initialization:
    • Initialize the set of labeled examples with $n$ labeled seed documents of the form ($d_i$, $y_i$). $y_i$ is the label of the $i$th document $d_i$. Each document $d_i$ contains a set of patterns $\{p_{i1}, p_{i2}, ..., p_{im}\}$.
    • Initialize the list of learned rules $R = \{\}$.
2. Loop:
    • For each label $y$, select a small set of pattern rules $r = p \rightarrow y$, $r \notin R$.
    • Append all selected rules $r$ to $R$.
    • For all non-seed documents $d$ that contain a pattern in $R$, set $label(d) = \arg\max_{p,y} strength(p, y)$.
3. Termination condition:
    • Stop if no rules selected or maximum number of iterations reached.
```

Figure 2: Pattern acquisition meta algorithm

$P(c|\hat{\theta})$, and the probability of seeing a word given a category, $P(w|c; \hat{\theta})$. We calculate both similarly to Nigam (2000). Using these parameters, the word independence assumption typical to the Naive Bayes model, and the Bayes rule, the probability that a document d has a given category c is calculated as:

$$P(c|d; \hat{\theta}) = \frac{P(c|\hat{\theta})P(d|c; \hat{\theta})}{P(d|\hat{\theta})} \quad (1)$$

$$= \frac{P(c|\hat{\theta})\Pi_{i=1}^{|d|}P(w_i|c; \hat{\theta})}{\sum_{j=1}^{q} P(c_j|\hat{\theta})\Pi_{i=1}^{|d|}P(w_i|c_j; \hat{\theta})} \quad (2)$$

2.3 The Pattern Acquisition Algorithm

The lightly-supervised pattern acquisition algorithm iteratively learns domain-specific IE patterns from a small set of labeled documents and a much larger set of unlabeled documents. During each learning iteration, the algorithm acquires a new set of patterns and labels more documents based on the new evidence. The algorithm output is a list R of rules $p \rightarrow y$, where p is a pattern in the set of patterns P, and y a category label in $Y = \{1...k\}$, k being the number of categories in the document collection. The list of acquired rules R is sorted in descending order of rule importance to guarantee that the most relevant rules are accessed first. This generic bootstrapping algorithm is formalized in Figure 2.

Previous studies called the class of algorithms illustrated in Figure 2 “cautious” or “sequential” because in each iteration they acquire 1 or a *small* set of rules (Abney, 2004; Collins and Singer, 1999). This strategy stops the algorithm from being over-confident, an important restriction for an algorithm that learns from large amounts of unlabeled data. This approach was empirically shown to perform better than a method that in each iteration acquires *all* rules that match a certain criterion (e.g. the corresponding rule has a strength over a certain threshold).

The key element where most instances of this algorithm vary is the **select** procedure, which decides which rules are acquired in each iteration. Although several selection strategies have been previously proposed for various NLP problems, to our knowledge no existing study performs an empirical analysis of such strategies in the context of acquisition of IE patterns. For this reason, we implement several selection methods in our system (described in Section 2.4) and evaluate their performance in Section 5.

The label of each collection document is given by the *strength* of its patterns. Similarly to (Collins and Singer, 1999; Yarowsky, 1995), we define the strength of a pattern p in a category y as the precision of p in the set of documents labeled with category y, estimated using Laplace smoothing:

$$strength(p, y) = \frac{count(p, y) + \epsilon}{count(p) + k\epsilon} \quad (3)$$

where $count(p, y)$ is the number of documents labeled y containing pattern p, $count(p)$ is the overall number of labeled documents containing p, and k is the number of domains. For all experiments presented here we used $\epsilon = 1$.

Another point where acquisition algorithms differ is the initialization procedure: some start with a small number of hand-labeled documents (Riloff, 1996), as illustrated in Figure 2, while others start with a set of seed rules (Yangarber et al., 2000; Yangarber, 2003). However, these approaches are conceptually similar: the seed rules are simply used to generate the seed documents.

This paper focuses on the framework introduced in Figure 2 for two reasons: (a) “cautious” algorithms were shown to perform best for several NLP problems (including acquisition of IE patterns), and (b) it has nice theoretical properties: Abney (2004) showed that, regardless of the selection procedure, "sequential" bootstrapping algorithms converge to a local minimum of K, where K is an upper bound of the negative log likelihood of the data. Obviously, the quality of the local minimum discovered is highly dependent of the selection procedure, which is why we believe an evaluation of several pattern selection strategies is important.

2.4 Selection Criteria

The pattern selection component, i.e. the **select** procedure of the algorithm in Figure 2, consists of the following: (a) for each category y all patterns p are sorted in descending order of their scores in the current category, $score(p, y)$, and (b) for each category the top k patterns are selected. For all experiments in this paper we have used $k = 3$. We provide four different implementations for the pattern scoring function $score(p, y)$ according to four different selection criteria.

Criterion 1: Riloff

This selection criterion was developed specifically for the pattern acquisition task (Riloff, 1996) and has been used in several other pattern acquisition systems (Yangarber et al., 2000; Yangarber, 2003; Stevenson and Greenwood, 2005). The intuition behind it is that a qualitative pattern is yielded by a compromise between pattern precision (which is a good indicator of relevance) and pattern frequency (which is a good indicator of coverage). Furthermore, the criterion considers only patterns that are positively correlated with the corresponding category, i.e. their precision is higher than 50%. The Riloff score of a pattern p in a category y is formalized as:

$$score(p,y) = \begin{cases} prec(p,y)\log(count(p,y)), \\ \quad if\ prec(p,y) > 0.5; \\ 0, otherwise. \end{cases} \quad (4)$$

$$prec(p,y) = \frac{count(p,y)}{count(p)} \quad (5)$$

where $prec(p, y)$ is the raw precision of pattern p in the set of documents labeled with category y.

Criterion 2: Collins

This criterion was used in a lightly-supervised NE recognizer (Collins and Singer, 1999). Unlike the previous criterion, which combines relevance and frequency in the same scoring function, Collins considers only patterns whose raw precision is over a hard threshold T and ranks them by their global coverage:

$$score(p,y) = \begin{cases} count(p), & if\ prec(p,y) > T; \\ 0, & otherwise. \end{cases} \quad (6)$$

Similarly to (Collins and Singer, 1999) we used $T = 0.95$ for all experiments reported here.

Criterion 3: χ^2 (Chi)

The χ^2 score measures the lack of independence between a pattern p and a category y. It is computed using a two-way contingency table of p and y, where a is the number of times p and y co-occur, b is the number of times p occurs without y, c is the number of times y occurs without p, and d is the number of times neither p nor y occur. The number of documents in the collection is n. Similarly to the first criterion, we consider only patterns positively correlated with the corresponding category:

$$score(p,y) = \begin{cases} \chi^2(p,y), & if\ prec(p,y) > 0.5; \\ 0, & otherwise. \end{cases} \quad (7)$$

$$\chi^2(p,y) = \frac{n(ad-cb)^2}{(a+c)(b+d)(a+b)(c+d)} \quad (8)$$

The χ^2 statistic was previously reported to be the best feature selection strategy for text categorization (Yang and Pedersen, 1997).

Criterion 4: Mutual Information (MI)

Mutual information is a well known information theory criterion that measures the independence of two variables, in our case a pattern p and a category y (Yang and Pedersen, 1997). Using the same contingency table introduced above, the MI criterion is estimated as:

$$score(p,y) = \begin{cases} MI(p,y), & if\ prec(p,y) > 0.5; \\ 0, & otherwise. \end{cases} \quad (9)$$

$$MI(p,y) = \log\frac{P(p \wedge y)}{P(p) \times P(y)} \quad (10)$$

$$\approx \log\frac{na}{(a+c)(a+b)} \quad (11)$$

3 The Data

3.1 The Document Collections

For all experiments reported in this paper we used the following three document collections: (a) the *AP* collection is the Associated Press (year 1999) subset of the AQUAINT collection (LDC catalog number LDC2002T31); (b) the *LATIMES* collection is the Los Angeles Times subset of the TREC-5 collection[1]; and (c) the *REUTERS* collection is the by now classic Reuters-21578 text categorization collection[2].

[1] *http://trec.nist.gov/data/docs_eng.html*

[2] *http://trec.nist.gov/data/reuters/reuters.html*

Collection	# of docs	# of categories	# of words	# of patterns
AP	5000	7	24812	140852
LATIMES	5000	8	29659	69429
REUTERS	9035	10	12905	36608

Table 1: Document collections used in the evaluation

Similarly to previous work, for the REUTERS collection we used the ModApte split and selected the ten most frequent categories (Nigam et al., 2000). Due to memory limitations on our test machines, we reduced the size of the AP and LATIMES collections to their first 5,000 documents (the complete collections contain over 100,000 documents).

The collection words were pre-processed as follows: (i) stop words and numbers were discarded; (ii) all words were converted to lower case; and (iii) terms that appear in a single document were removed. Table 1 lists the collection characteristics after pre-processing.

3.2 Pattern Generation

In order to extract the set of patterns available in a document, each collection document undergoes the following processing steps: (a) we recognize and classify named entities[3], and (b) we generate full parse trees of all document sentences using a probabilistic context-free parser.

Following the above processing steps, we extract Subject-Verb-Object (SVO) tuples using a series of heuristics, e.g.: (a) nouns preceding active verbs are subjects, (b) nouns directly attached to a verb phrase are objects, (c) nouns attached to the verb phrase through a prepositional attachment are indirect objects. Each tuple element is replaced with either its head word, if its head word is not included in a NE, or with the NE category otherwise. For indirect objects we additionally store the accompanying preposition. Lastly, each tuple containing more than two elements is generalized by maintaining only subsets of two and three of its elements and replacing the others with a wildcard.

Table 2 lists the patterns extracted from one sample sentence. As Table 2 hints, the system generates a large number of candidate patterns. It is the task of the pattern acquisition algorithm to extract only the relevant ones from this complex search space.

4 The Evaluation Procedures

4.1 The Indirect Evaluation Procedure

The goal of our evaluation procedure is to measure the quality of the acquired patterns. Intuitively, the learned patterns should have high coverage and low ambiguity. We indirectly measure the quality of the acquired patterns using a text categorization strategy: we feed the acquired rules to a decision-list classifier, which is then used to classify a new set of documents. The classifier assigns to each document the category label given by the first rule whose pattern matches. Since we expect higher-quality patterns to appear higher in the rule list, the decision-list classifier never changes the category of an already-labeled document.

Text	*The Minnesota Vikings beat the Arizona Cardinals in yesterday's game.*

Patterns	`s(ORG) v(beat)`
	`v(beat) o(ORG)`
	`s(ORG) o(ORG)`
	`v(beat) io(in game)`
	`s(ORG) io(in game)`
	`o(ORG) io(in game)`
	`s(ORG) v(beat) o(ORG)`
	`s(ORG) v(beat) io(in game)`
	`v(beat) o(ORG) io(in game)`

Table 2: Patterns extracted from one sample sentence. *s* stands for subject, *v* for verb, *o* for object, and *io* for indirect object.

The quality of the generated classification is measured using micro-averaged precision and recall:

$$P = \frac{\sum_{i=1}^{q} TruePositives_i}{\sum_{i=1}^{q}(TruePositives_i + FalsePositives_i)} \quad (12)$$

$$R = \frac{\sum_{i=1}^{q} TruePositives_i}{\sum_{i=1}^{q}(TruePositives_i + FalseNegatives_i)} \quad (13)$$

where q is the number of categories in the document collection.

For all experiments and all collections with the exception of REUTERS, which has a standard document split for training and testing, we used 5-fold cross validation: we randomly partitioned the collections into 5 sets of equal sizes, and reserved a different one for testing in each fold.

We have chosen this evaluation strategy because this indirect approach was shown to correlate well with a direct evaluation, where the learned patterns were used to customize an IE system (Yangarber et al., 2000). For this reason, much of the following work on pattern acquisition has used this approach as a *de facto* evaluation standard (Yangarber, 2003; Stevenson and Greenwood, 2005). Furthermore, given the high number of domains and patterns (we evaluate on 25 domains), an evaluation by human experts is extremely costly. Nevertheless, to show that the proposed indirect evaluation correlates well with a direct evaluation, two human experts have evaluated the patterns in several domains. The direct evaluation procedure is described next.

[3] We identify six categories: persons, locations, organizations, other names, temporal and numerical expressions.

4.2 The Direct Evaluation Procedure

The task of manually deciding whether an acquired pattern is relevant or not for a given domain is not trivial, mainly due to the ambiguity of the patterns. Thus, this process should be carried out by more than one expert, so that the relevance of the ambiguous patterns can be agreed upon. For example, the patterns *s(ORG) v(score) o(goal)* and *s(PER) v(lead) io(with point)* are clearly relevant only for the sports domain, whereas the patterns *v(sign) io(as agent)* and *o(title) io(in DATE)* might be regarded as relevant for other domains as well.

The specific procedure to manually evaluate the patterns is the following: (1) two experts separately evaluate the acquired patterns for the considered domains and collections; and (2) the results of both evaluations are compared. For any disagreement, we have opted for a strict evaluation: all the occurrences of the corresponding pattern are looked up in the collection and, whenever at least one pattern occurrence belongs to a document assigned to a different domain than the domain in question, the pattern will be considered as not relevant.

Both the ambiguity and the high number of the extracted patterns have prevented us from performing an exhaustive direct evaluation. For this reason, only the top (most relevant) 100 patterns have been evaluated for one domain per collection. The results are detailed in Section 5.2.

5 Experimental Evaluation

5.1 Indirect Evaluation

For a better understanding of the proposed approach we perform an incremental evaluation: first, we evaluate only the various pattern selection criteria described in Section 2.4 by disabling the NB-EM component. Second, using the best selection criteria, we evaluate the complete co-training system.

In both experiments we initialize the system with high-precision manually-selected seed rules which yield seed documents with a coverage of 10% of the training partitions. The remaining 90% of the training documents are maintained unlabeled. For all experiments we used a maximum of 400 bootstrapping iterations. The acquired rules are fed to the decision list classifier which assigns category labels to the documents in the test partitions.

Evaluation of the pattern selection criteria

Figure 3 illustrates the precision/recall charts of the four algorithms as the number of patterns made available to the decision list classifier increases. All charts show precision/recall points starting after 100 learning iterations with 100-iteration increments. It is immediately obvious that the Collins selection criterion performs significantly better than the other three criteria. For the same recall point, Collins yields a classification model with much higher precision, with differences ranging from 5% in the REUTERS collection to 20% in the AP collection.

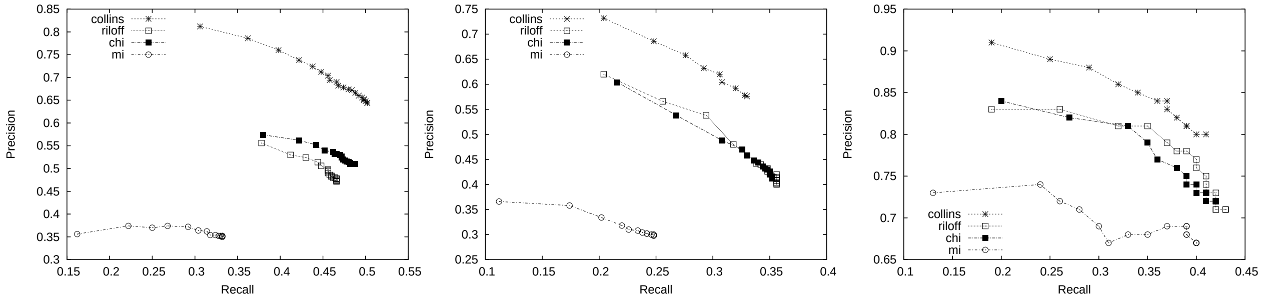
Theorem 5 in (Abney, 2002) provides a theoretical explanation for these results: if certain independence conditions between the classifier rules are satisfied and the precision of each rule is larger than a threshold T, then the precision of the final classifier is larger than T. Although the rule independence conditions are certainly not satisfied in our real-world evaluation, the above theorem indicates that there is a strong relation between the precision of the classifier rules on labeled data and the precision of the final classifier. Our results provide the empirical proof that controling the precision of the acquired rules (i.e. the Collins criterion) is important.

The Collins criterion controls the recall of the learned model by favoring rules with high frequency in the collection. However, since the other two criteria do not use a high precision threshold, they will acquire more rules, which translates in better recall. For two out of the three collections, Riloff and Chi obtain a slightly better recall, about 2% higher than Collins', albeit with a much lower precision. We do not consider this an important advantage: in the next section we show that co-training with the NB-EM component further boosts the precision and recall of the Collins-based acquisition algorithm.

The MI criterion performs the worst of the four evaluated criteria. A clue for this behavior lies in the following equivalent form for MI: $MI(p, y) = \log P(p|y) - \log P(p)$. This formula indicates that, for patterns with equal conditional probabilities $P(p|y)$, MI assigns higher scores to patterns with lower frequency. This is not the desired behavior in a TC-oriented system.

Evaluation of the co-training system

Figure 4 compares the performance of the stand-alone pattern acquisition algorithm ("bootstrapping") with the performance of the acquisition algorithm trained in the co-training environ-



(a) (b) (c)

Figure 3: Performance of the pattern acquisition algorithm for various pattern selection strategies and multiple collections: (a) AP, (b) LATIMES, and (c) REUTERS

ment ("co-training"). For both setups we used the best pattern selection criterion for pattern acquisition, i.e. the Collins criterion. To put things in perspective, we also depict the performance obtained with a baseline system, i.e. the system configured to use the Riloff pattern selection criterion and without the NB-EM algorithm ("baseline"). To our knowledge, this system, or a variation of it, is the current state-of-the-art in pattern acquisition (Riloff, 1996; Yangarber et al., 2000; Yangarber, 2003; Stevenson and Greenwood, 2005). All algorithms were initialized with the same seed rules and had access to all documents.

Figure 4 shows that the quality of the learned patterns *always* improves if the pattern acquisition algorithm is "reinforced" with EM. For the same recall point, the patterns acquired in the co-training environment yield classification models with precision (generally) much larger than the models generated by the pattern acquisition algorithm alone. When using the same pattern acquisition criterion, e.g. Collins, the differences between the co-training approach and the stand-alone pattern acquisition method ("bootstrapping") range from 2-3% in the REUTERS collection to 20% in the LATIMES collection. These results support our intuition that the sparse pattern space is insufficient to generate good classification models, which directly influences the quality of all acquired patterns.

Furthermore, due to the increased coverage of the lexicalized collection views, the patterns acquired in the co-training setup generally have better recall, up to 11% higher in the LATIMES collection.

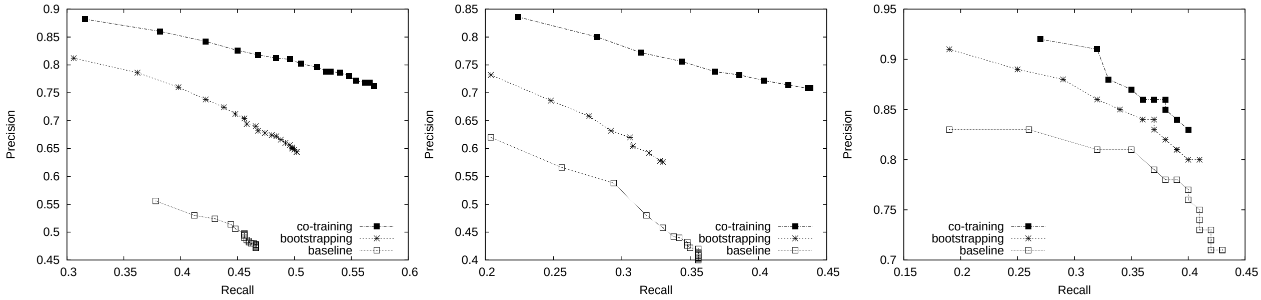
Lastly, the comparison of our best system ("co-training") against the current state-of-the-art (our "baseline") draws an even more dramatic picture: for the same recall point, the co-training system obtains a precision up to 35% higher for AP and LATIMES, and up to 10% higher for REUTERS.

Collection	Domain	Relevant patterns baseline	Relevant patterns co-training	Initial inter-expert agreement
AP	Sports	22%	68%	84%
LATIMES	Financial	67%	76%	70%
REUTERS	Corporate Acquisitions	38%	46%	66%

Table 3: Percentage of relevant patterns for one domain per collection by the baseline system (Riloff) and the co-training system.

5.2 Direct Evaluation

As stated in Section 4.2, two experts have manually evaluated the top 100 acquired patterns for one different domain in each of the three collections. The three corresponding domains have been selected intending to deal with different degrees of ambiguity, which are reflected in the initial inter-expert agreement. Any disagreement between experts is solved using the algorithm introduced in Section 4.2. Table 3 shows the results of this direct evaluation. The co-training approach outperforms the baseline for all three collections. Concretely, improvements of 9% and 8% are achieved for the Financial and the Corporate Acquisitions domains, and 46%, by far the largest difference, is found for the Sports domain in AP. Table 4 lists the top 20 patterns extracted by both approaches in the latter domain. It can be observed that for the baseline, only the top 4 patterns are relevant, the rest being extremely general patterns. On the other hand, the quality of the patterns acquired by our approach is much higher: all the patterns are relevant to the domain, although 7 out of the 20 might be considered ambiguous and according to the criterion defined in Section 4.2 have been evaluated as not relevant.



(a) (b) (c)

Figure 4: Comparison of the bootstrapping pattern acquisition algorithm with the co-training approach: (a) AP, (b) LATIMES, and (c) REUTERS

Baseline	Co-training
s(he) o(game)	**v(win) o(title)**
v(miss) o(game)	s(I) v(play)
v(play) o(game)	**s(he) v(game)**
v(play) io(in LOC)	s(we) v(play)
v(go) o(be)	**v(miss) o(game)**
s(he) v(be)	**s(he) v(coach)**
s(that) v(be)	**v(lose) o(game)**
s(I) v(be)	s(I) o(play)
s(it) v(go) o(be)	v(make) o(play)
s(it) v(be)	**v(play) io(in game)**
s(I) v(think)	v(want) o(play)
s(I) v(know)	**v(win) o(MISC)**
s(I) v(want)	**s(he) o(player)**
s(there) v(be)	**v(start) o(game)**
s(we) v(do)	s(PER) o(contract)
v(do) o(it)	s(we) o(play)
s(it) o(be)	**s(team) v(win)**
s(we) v(are)	**v(rush) io(for yard)**
s(we) v(go)	**s(we) o(team)**
s(PER) o(DATE)	**v(win) o(Bowl)**

Table 4: Top 20 patterns acquired from the Sports domain by the baseline system (Riloff) and the co-training system for the AP collection. The correct patterns are in bold.

6 Conclusions

This paper introduces a hybrid, lightly-supervised method for the acquisition of syntactico-semantic patterns for Information Extraction. Our approach co-trains a decision list learner whose feature space covers the set of all syntactico-semantic patterns with an Expectation Maximization clustering algorithm that uses the text words as attributes. Furthermore, we customize the decision list learner with up to four criteria for pattern selection, which is the most important component of the acquisition algorithm.

For the evaluation of the proposed approach we have used both an indirect evaluation based on Text Categorization and a direct evaluation where human experts evaluated the quality of the generated patterns. Our results indicate that co-training the Expectation Maximization algorithm with the decision list learner tailored to acquire only high precision patterns is by far the best solution. For the same recall point, the proposed method increases the precision of the generated models up to 35% from the previous state of the art. Furthermore, the combination of the two feature spaces (words and patterns) also increases the coverage of the acquired patterns. The direct evaluation of the acquired patterns by the human experts validates these results.

References

S. Abney. 2002. Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

S. Abney. 2004. Understanding the Yarowsky algorithm. *Computational Linguistics*, 30(3).

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*.

M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of EMNLP*.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1).

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3).

E. Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*.

M. Stevenson and M. Greenwood. 2005. A semantic approach to ie pattern induction. In *Proceedings of the 43rd Meeting of the Association for Computational Linguistics*.

Y. Yang and J. O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*.

R. Yangarber, R. Grishman, P. Tapanainen, and S. Hutunen. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th International Conference of Computational Linguistics (COLING 2000)*.

R. Yangarber. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*.