Trained Named Entity Recognition Using Distributional Clusters

Dayne Freitag
HNC Software, LLC
3661 Valley Centre Drive
San Diego, CA 92130
DayneFreitag@fairisaac.com

Abstract

This work applies boosted wrapper induction (BWI), a machine learning algorithm for information extraction from semi-structured documents, to the problem of named entity recognition. The default feature set of BWI is augmented with features based on distributional term clusters induced from a large unlabeled text corpus. Using no traditional linguistic resources, such as syntactic tags or special-purpose gazetteers, this approach yields results near the state of the art in the MUC 6 named entity domain. Supervised learning using features derived through unsupervised corpus analysis may be regarded as an alternative to bootstrapping methods.

1 Introduction

The problem of named entity recognition (NER) has recently received increasing attention. Identification of generic semantic categories in text—such as mentions of people, organizations, locations, and temporal and numeric expressions—is a necessary first step in many applications of information extraction, information retrieval, and question answering. To a large extent, knowledge-poor methods suffice to yield good recognition performance. In particular, supervised learning can be used to produce a system with performance at or near the state of the art (Bikel et al., 1997).

In the supervised learning framework, a corpus of (typically) a few hundred documents is annotated by hand to identify the entities of interest. Features of local context are then used to train a system to distinguish instances from non-instances in novel texts. Such features may include literal word tests, patterns of orthography, parts of speech, semantic categories, or membership in special-purpose gazetteers.

While supervised training greatly facilitates the development of a robust NER system, the requirement of a substantial training corpus remains an impediment to the rapid deployment of NER in new domains or new languages. A number of researchers have therefore sought to exploit the availability of unlabeled documents, typically by bootstrapping a classifier using automatic labellings (Collins and Singer, 1999; Cucerzan and Yarowsky, 1999; Thelen and Riloff, 2002).

bush peters reagan noriega …
john robert james david …
president chairman head owner …
japan california london chicago …

Table 1: Sample members of four clusters from the Wall Street Journal corpus.

Here, we investigate a different approach. Using a distributional clustering technique called co-clustering, we produce clusters which, intuitively, should be useful for NER. Table 1 shows example terms from several sample clusters induced using a collection of documents from the Wall Street Journal (WSJ). Several papers have shown that distributional clustering yields categories that have high agreement with part of speech (Schütze, 1995; Clark, 2000). As the table illustrates, these clusters also tend to have a useful semantic dimension. Clustering on the WSJ portion of the North American News corpus yields two clusters that clearly correspond to personal names, one for first names and one for last names. As an experiment, we scanned the MUC6 NER data set for token sequences consisting of zero or more members of the first name cluster (or an initial followed by a period), followed by one or more members of the last name cluster. This simple procedure identified 64% of personal names with 77% precision.

In this paper, we attempt to improve on this result by converting the clusters into features to be exploited by a general-purpose machine learning algorithm for information extraction. In Section 2, we provide a brief description of Boosted Wrapper Induction (BWI), a pattern learner that has yielded promising results on semi-structured information extraction problems (Freitag and Kushmer-

ick, 2000). In Section 3, we describe our clustering approach and its particular application. Section 4 presents the results of our experiments. Finally, in Section 5, we assess the significance of our contribution and attempt to identify promising future directions.

2 BWI

BWI decomposes the problem of recognizing field instances into two Boolean classification problems: recognizing field-initial and field-terminal tokens. Given a target field, a separate classifier is learned for each of these problems, and the distribution of field lengths is modeled as a frequency histogram. At application time, tokens that test positive for *initial* are paired with those testing positive for *terminal*. If the length of a candidate instance, as defined by such a pair, is determined to have non-zero likelihood using the length histogram, a prediction is returned.

Each of the three parts of a full prediction—initial boundary, terminal boundary, and length—is assigned a real-valued confidence. The confidence of a boundary detection is its strength as determined by AdaBoost, while that of the length assessment is the empirical length probability, which is determined using the length histogram. The confidence of the full prediction is the product of these three individual confidence scores. In the event that overlapping predictions are found in this way (a rare event, empirically), the predictions with lower confidence are discarded.

In this section, we sketch those aspects of BWI relevant to the current application. More details are available in the paper in which BWI was defined (Freitag and Kushmerick, 2000).

2.1 Boosting

BWI uses generalized AdaBoost to produce each boundary classifier (Schapire and Singer, 1998). Boosting is a procedure for improving the performance of a "weak learner" by repeatedly applying it to a training set, at each step modifying example weights to emphasize those examples on which the learner has done poorly in previous steps. The output is a weighted collection of weak learner hypotheses. Classification involves having the individual hypotheses "vote," with strengths proportional to their weights, and summing overlapping votes.

Although this is the first application of BWI to NER, boosting has previously been shown to work well on this problem. Differing from BWI in the details of the application, two recent papers nevertheless demonstrate the effectiveness of the boosting paradigm for NER in several languages (Carreras et al., 2002; Wu et al., 2002), one of them achieving the best overall performance in a comparison of several systems (Sang, 2002).

`Cap`	Initial capital
`AllCap`	All capitals
`Uncap`	Initial lower case
`Alpha`	Entirely alphabetic characters
`ANum`	Entirely alpha-numeric characters
`Punc`	Punctuation
`Num`	Entirely numeric characters
`Schar`	Single alphabetic character
`Any`	Anything

Table 2: Default wildcards used in these experiments.

2.2 Boundary Detectors

The output of a single invocation of the weak learner in BWI is always an individual pattern, called a *boundary detector*. A detector has two parts, one to match the text leading up to a boundary, the other for trailing text. Each part is a list of zero or more elements. In order for a boundary to match a detector, the tokens preceding the boundary (or following it) must match the corresponding elements in sequence. For example, the detector `[ms .][jones]` matches boundaries preceded by the (case-normalized) two-token sequence "`ms .`" and followed by the single token "`jones`".

Detectors are grown iteratively, beginning with an empty detector and repeatedly adding the element that best increases the ability of the current detector to discriminate true boundaries from false ones, using a cost function sensitive to the example weighting. A *look-ahead* parameter allows this decision to be based on several additional context tokens. The process terminates when no extensions yield a higher score than the current detector.

2.3 Wildcards

The elements of the detector `[ms .][jones]` are literal elements, which match tokens using case-normalized string comparison. More interesting elements can be introduced by defining token *wildcards*. Each wildcard defines some Boolean function over the space of tokens.

Table 2 lists the baseline wildcards. Using wildcards from this list, the example detector can be generalized to match a much broader range of boundaries (e.g., `[ms <Any>][<Cap>]`). By defining new wildcards, we can inject useful domain knowledge into the inference process, potentially

improving the performance of the resulting extractor. For example, we might define a wildcard called "`Honorific`" that matches any of "`ms`", "`mr`", "`mrs`", and "`dr`".

2.4 Boundary Wildcards

In the original formulation of BWI, boundaries are identified without reference to the location of the opposing boundary. However, we might expect that the end of a name, say, would be easier to identify if we know where it begins. We can build detectors that exploit this knowledge by introducing a special wildcard (called `Begin`) that matches the beginnings of names.

In these experiments, therefore, we modify boundary detection in the following way. Instead of two detector lists, we learn four—the two lists as in the original formulation (call them H_{Init} and H_{Term}), and two more lists ($H_{\text{Init}*}$ and $H_{\text{Term}*}$). In generating the latter two lists, we give the learner access to these special wildcards (e.g., the wildcard `End` in generating $H_{\text{Init}*}$).

At extraction time, H_{Init} and H_{Term} are first used to detect boundaries, as before. These detections are then used to determine which tokens match the "special" wildcards used by $H_{\text{Init}*}$ and $H_{\text{Term}*}$. Then, instead of pairing H_{Init} predictions with those of H_{Term}, they are paired with those made by $H_{\text{Term}*}$ (and $H_{\text{Init}*}$ with H_{Term}). In informal experiments, we found that this procedure tended to increase F1 performance by several points on a range of tasks. We adopt it uniformly in the experiments reported here.

3 Co-Clustering

As in Brown, et al (1992), we seek a partition of the vocabulary that maximizes the mutual information between term categories and their contexts. To achieve this, we use *information theoretic co-clustering* (Dhillon et al., 2003), in which a space of entities, on the one hand, and their contexts, on the other, are alternately clustered to maximize mutual information between the two spaces.

3.1 Background

The input to our algorithm is two finite sets of symbols, say $X = \{x_0, x_1, \cdots, x_{N_X}\}$ (e.g., terms) and $Y = \{y_0, y_1, \cdots, y_{N_Y}\}$ (e.g., term contexts), together with a set of co-occurrence count data consisting of a non-negative integer $n_{x_i y_j}$ for every pair of symbols (x_i, y_j) from X and Y. The output is two partitions: $X^* = \{x_0^*, ..., x_{N_{X^*}}^*\}$ and $Y^* = \{y_0^*, ..., y_{N_{Y^*}}^*\}$, where each x_i^* is a subset of X (a "cluster"), and each y_i^* a subset of Y. The co-clustering algorithm chooses the partitions X^* and Y^* to (locally) maximize the mutual information between them, under a constraint limiting the total number of clusters in each partition.

Recall that the *entropy* or *Shannon information* of a discrete distribution is:

$$\mathrm{I}_X = -\sum_x \mathrm{P}(x) \ln \mathrm{P}(x). \qquad (1)$$

This quantifies average improvement in one's knowledge upon learning the specific value of an event drawn from X. It is large or small depending on whether X has many or few probable values.

The *mutual information* between random variables X and Y can be written:

$$\mathrm{M}_{XY} = \sum_{xy} \mathrm{P}(x, y) \ln \frac{\mathrm{P}(x, y)}{\mathrm{P}(x)\mathrm{P}(y)} \qquad (2)$$

This quantifies the amount that one expects to learn indirectly about X upon learning the value of Y, or vice versa.

3.2 The Algorithm

Let X be a random variable over vocabulary terms as found in some text corpus. We define Y to range over immediately adjacent tokens, encoding co-occurrences in such a way as to distinguish left from right occurrences.

Given co-occurrence matrices tabulated in this way, we perform an approximate maximization of $M_{X^*Y^*}$ using a simulated annealing procedure in which each trial move takes a symbol x or y out of the cluster to which it is tentatively assigned and places it into another. Candidate moves are chosen by selecting a non-empty cluster uniformly at random, randomly selecting one of its members, then randomly selecting a destination cluster other than the source cluster. When temperature 0 is reached, all possible moves are repeatedly attempted until no further improvements are possible.

For efficiency and noise reduction, we first cluster only the 5000 most frequent terms and context terms. The remaining terms in the corpus vocabulary are then added by assigning each term to the cluster that maximizes the mutual information objective function.

4 Evaluation

We experimented with the MUC 6 named entity data set, which consists of a training set of 318 documents, a validation set of 30 documents, and a test set of 30 documents.

All documents are annotated to identify three types of name (PERSON, ORGANIZATION,

DATE	[][september]
	[in <Num>][<Punc>]
TIME	[][5 <ANum> . <ANum>]
	[midnight][]
MONEY	[][$]
	[$ <Any> billion][]
PCT.	[<Alph> <Punc>][<Any> %]
	[<Num> percentage <Alph>][]
PERSON	[mr <Any>][<Cap>]
	[<Cap>][, vice]
ORG.	[][nissan]
	[inc <Any>][]
LOC.	[in][<Cap> , <Alph> <Punc>]
	[germany][]

Table 3: Sample boundary detectors for the seven MUC 6 fields produced by BWI using the baseline feature set. An initial and terminal detector is shown for each field.

LOCATION), two types of temporal expression (DATE, TIME), and two types of numeric expression (MONEY, PERCENT). It is common to report performance in terms of precision, recall, and their harmonic mean (F1), a convention to which we adhere.

4.1 Baseline

Using the wildcards listed in Table 2, we trained BWI for 500 boosting iterations on each of the seven entity fields. The output out each of these training runs consists of $500 \times 4 = 2000$ boundary detectors. Look-ahead was set to 3.

Table 3 shows a few of the boundary detectors induced by this procedure. These detectors were selected manually to illustrate the kinds of patterns generated. Note how some of the detectors amount to field-specific gazetteer entries. Others have more interesting (and typically intuitive) structure. We defer quantitative evaluation to the next section, where a comparison with the cluster-enhanced extractors will be made.

4.2 Adding Cluster Features

The MUC 6 dataset was produced using articles from the Wall Street Journal. In order to produce maximally relevant clusters, we used documents from the WSJ portion of the North American News corpus as input to co-clustering—some 119,000 documents in total. Note that there is a temporal disparity between the MUC 6 corpus and this clustering corpus, which has an undetermined impact on performance.

PERS.	[<C95>][<C73>]
	[<C144> <Any> <C106>][<Uncap>]
ORG.	[][<C178> express]
	[bank <ANum> <C146>][]
LOC.	[][<C72> korea]
	[<C160>][<Punc>]

Table 4: Sample boundary detectors for the seven MUC 6 fields produced by BWI using the expanded feature set.

72	general south north poor ...
73	john robert james david ...
95	says adds asks recalls ...
106	clinton dole johnson gingrich ...
144	mr ms dr sen ...
146	japan american china congress ...
160	washington texas california ...
178	american foreign local ...

Table 5: Most frequent members of clusters referenced by detectors in Table 4.

We used this data to produce 200 clusters, as described in Section 3. Treating each of these clusters as an unlabeled gazetteer, we then defined corresponding wildcards. For example, the value of wildcard <C35> only matches a term belonging to Cluster 35. In order to reduce the training time of a given boundary learning problem, we tabulated the frequency of wildcard occurrence within three tokens of any occurrences of the target boundary and omitted from training wildcards testing true fewer than ten times.[1]

Table 4, which lists sample detectors from these runs, includes some that are clearly impossible to express using the baseline feature set. An example is the first row, which matches a third-person present-tense verb used in quote attribution, followed by a first name (see Table 5). At the same time, some of the new wildcards are employed trivially, such as the use of <C178> in the field-initial detector for the ORGANIZATION field.

Table 6 shows performance of the two variants on the individual MUC 6 fields, tested over the "dryrun" and "formal" test sets combined. In this table, we scored each field individually using our own evaluation software. An entity instance was judged to be correctly extracted if a prediction precisely identified its boundaries (ignoring "ALT" at-

[1] For the TIME field, which occurs a total of six times in the training set, this cut-off was a single occurrence.

Field		*F1*	*Prec*	*Rec*
DATE	*Base*	0.766	0.765	0.768
	Clust	0.782	0.776	0.789
TIME	*Base*	0.667	1.000	0.500
	Clust	0.667	1.000	0.500
MONEY	*Base*	0.938	0.926	0.949
	Clust	0.943	0.938	0.949
PERCENT	*Base*	0.922	0.855	1.000
	Clust	0.930	0.869	1.000
PERSON	*Base*	0.827	0.810	0.844
	Clust	0.892	**0.859**	**0.927**
ORG.	*Base*	0.587	0.811	0.460
	Clust	0.733	0.796	**0.680**
LOCATION	*Base*	0.726	0.675	0.785
	Clust	0.724	0.648	0.821

Table 6: Performance on the seven MUC 6 fields, without (Base) and with (Clust) cluster-based features. Significantly better precision or recall scores, at the 95% confidence level, are in boldface.

tributes). Non-matching predictions and missed entities were counted as false positives and false negatives, respectively. We assessed the statistical significance of precision and recall scores by computing beta confidence intervals at the 95% level. In the table, the higher precision or recall is in boldface if its separation from the lower score is significant.

Except for TIME and LOCATION, all fields benefit from inclusion of the cluster features. TIME, which is scarce in the training and test sets, is insensitive to their inclusion. The effect on LOCATION is more interesting. It shares in the general tendency of cluster features to increase recall, but loses precision as a result.[2] Although the increase in recall is approximately the same as the loss in precision, the F1 score, which is more heavily influenced by the lower of precision and recall, drops slightly.

While the effect of the cluster features on precision is inconsistent, they typically benefit recall. This effect is most dramatic in the case of ORGANIZATION, where, at the expense of a small drop in precision, recall increases by more than 20 points. The somewhat counter-intuitive improvements in precision on some fields (particularly the significant improvement on PERSON) is attributable to our learning framework. Boosting for a sufficient number of iterations forces a learner to account for all boundary tokens through one or more detectors. To the exent that the baseline's features are unable to account for as many of the boundary tokens, it is forced to learn a larger number of over-specialized detectors that rely on questionable patterns in the data. Depending on the task, these detectors can lead to a larger proportion of false positives.

The relatively weak result for DATE comes as a surprise. Inspection of the data leads us to attribute this to two factors. On the one hand, there is considerable temporal drift between the training and test sets. Many of the dates are specific to contemporaneous events; patterns based on specific years, therefore, generalize in only a limited way. At the same time, the notion of date, as understood in the MUC 6 corpus, is reasonably subtle. Meaning roughly "non-TIME temporal expression," it includes everything from shorthand date expressions to more interesting phrases, such as, "the first six months of fiscal 1994."

In passing we note a few potentially relevant idiosyncrasies in these experiments. Most significant is a representational choice we made in tokenizing the cluster corpus. In tallying frequencies we treated all numeric expressions as occurrences of a special term, "`*num*`". Consequently, the tokens "`1989`" and "`10,000`" are treated as instances of the same term, and clustering has no opportunity to distinguish, say, years from monetary amounts.

The (perhaps) disappointing performance on the relatively simple fields, TIME and PERCENT, somewhat under-reports the strength of the learner. As noted above, TIME occurs only very infrequently. Consequently, little training data is available for this field and mistakes (BWI missed one of the three instances in the test set) have a large effect on the TIME-specific scores. In the case of PERCENT, we ignored MUC instructions not to attempt to recognize instances in tabular regions. One of the documents contains a significant number of unlabeled percentages in such a table. BWI duly recognized these—to the detriment of the reported precision.

4.3 MUC Evaluation

For comparison with numbers reported in the literature, we used the learned extractors to produce mark-up and evaluated the result using the MUC 6 scorer. The MUC 6 evaluation framework differs from ours in two key ways. Most importantly, all entity types are to be processed simultaneously. We benefit from this framework, since spurious predictions for one entity type may be superseded by correct predictions for a related type. The opportunity is greatest for the three name types; in inspecting the false positives, we observed a number of confu-

[2] Note, however, that none of the differences observed for LOCATION are significant at the 95% level.

Field		*F1*	*Prec*	*Rec*
DATE	*Base*	0.91	0.91	0.91
	Clust	0.92	0.90	0.94
TIME	*Base*	0	0	0
	Clust	0	0	0
MONEY	*Base*	0.95	0.94	0.96
	Clust	0.95	0.95	0.96
PERCENT	*Base*	0.97	0.94	1.0
	Clust	1.0	1.0	1.0
PERSON	*Base*	0.88	0.91	0.86
	Clust	0.94	0.94	0.95
ORG.	*Base*	0.62	0.78	0.52
	Clust	0.79	0.84	0.74
LOCATION	*Base*	0.86	0.86	0.87
	Clust	0.86	0.80	0.92
ALL	*Base*	0.79	0.85	0.73
	Clust	0.87	0.88	0.86

Table 7: Performance on the markup task, as scored by the MUC 6 scorer.

sions among these fields.[3] The MUC scorer is also more lenient than ours, awarding points for extraction of alternative strings and forgiving the inclusion of certain functional tokens in the extracted text.

In moving to the multi-entity extraction setting, the obvious approach is to collect predictions from all extractors simultaneously. However, this requires a strategy for dealing with overlapping predictions (e.g., a single text fragment labeled as both a person and organization). We resolve such conflicts by preferring in each case the extraction with the highest confidence. In order to render confidence scores more comparable, we normalized the weights of detectors making up each boundary classifier so they sum to one.

A comparison of Table 7 with Table 6 suggests the extent to which BWI benefits from the multi-field mark-up setting. Note that, here, we used only the "formal" test set for evaluation, in contrast with the numbers in Table 6, which combine the two test sets. The lift we observe from cluster features is also in evidence here, and is most evident as an increase in recall, particularly of PERSON and ORGANIZATION. There is now also an increase in global precision, attributable in large part to the benefit of extracting multiple fields simultaneously.

The F1 score produced by BWI is comparable to the best machine-learning-based results reported elsewhere. Bikel, at al (1997), reports summary F1 of 0.93 on the same test set, but using a model trained on 450,000 words. We count approximately 130,000 words in the experiments reported here. The numbers reported by Bennett, et al (1997), for PERSON, ORGANIZATION, and LOCATION (F1 of 0.947, 0.815, and 0.925, respectively), are slightly better than the numbers BWI reaches on the same fields. Note, however, that the features provided to their learner include syntactic labels and carefully engineered semantic categories, whereas we eschew knowledge- and labor-intensive resources. This has important implications for the portability of the approaches to new domains and languages.

By taking a few post-processing steps, it is possible to realize further improvements. For example, the learner occasionally identifies terms and phrases which some simple rules can reliably reject. By suppressing any prediction that consists entirely of a stopword, we increase the precision of both ORGANIZATION and LOCATION to 0.86 (from 0.84 and 0.80) and overall F1 to 0.88.

We can also exploit what Cucerzan and Yarowsky (1999) call the *one sense per discourse* phenomenon, the tendency of terms to have a fixed meaning within a single document. By marking up unmarked strings that match extracted entity instances in the same document, we can improve the recall of some fields. We added this post-processing step for the PERSON and ORGANIZATION fields. This increased recall of PERSON from 0.95 to 0.98 and of ORGANIZATION from 0.74 to 0.79 with minimal changes to precision and a slight improvement in summary F1.

4.4 Analysis and Related Work

The promise of this general method—supervised learning on small training set using features derived from a larger unlabeled set—lies in the support it provides for rapid deployment in novel domains and languages. Without relying on any linguistic resources more advanced than a tokenizer and some orthographic features, we can produce a NER module using only a few annotated documents.

How few depends ultimately on the difficulty of the domain. We might also expect the benefit of distributional features to decrease with increasing training set size. Figure 1 displays the F1 learning-curve performance of BWI, both with and without cluster features, on the two fields that benefit the greatest from these features, PERSON and ORGANIZATION. As expected, the difference appears to be greatest on the low end of the horizontal axis (al-

[3] For example, companies are occasionally named after people (e.g., *Liz Claiborne*).

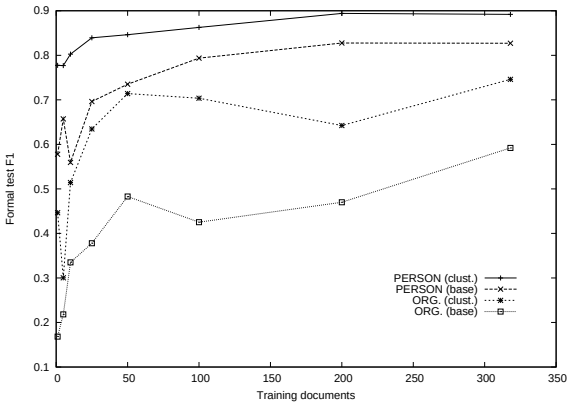

Figure 1: F1 as a function of training set size in number of documents.

though overfitting complicates the comparison). At the same time, the improvement is fairly consistent at all training set sizes. Either the baseline feature set is ultimately too impoverished for this task, or, more likely, the complete MUC 6 training set (318 documents) is small for this class of learner.

Techniques to lessen the need for annotation for NER have received a fair amount of attention recently. The prevailing approach to this problem is a bootstrapping technique, in which, starting with a few hand-labeled examples, the system iteratively adds automatic labels to a corpus, training itself, as it were. Examples of this are Cucerzan and Yarowsky (1999), Thelen and Riloff (2002), and Collins and Singer (1999).

These techniques address the same problem as this paper, but are otherwise quite different from the work described here. The labeling method (*seeding*) is an indirect form of corpus annotation. The promise of all such approaches is that, by starting with a small number of seeds, reasonable results can be achieved at low expense. However, it is difficult to tell how much labeling corresponds to a given number of seeds, since this depends on the coverage of the seeds. Note, too, that any bootstrapping approach must confront the problem of instability; poor initial decisions by a bootstrapping algorithm can lead to large eventual performance degradations. We might expect a lightly supervised learner with access to features based on a full-corpus analysis to yield more consistently strong results.

Of the three approaches mentioned above, only Cucerzan and Yarowsky do not presuppose a syntactic analysis of the corpus, so their work is perhaps most comparable to this one. Of course, comparisons must be strongly qualified, given the different labeling methods and data sets. Nevertheless, performance of cluster-enhanced BWI at the low end of the horizontal axis compares favorably with the English F1 performance of 0.543 they report using 190 seed words. And, arguably, annotating 10-20 documents is no more labor intensive than assembling a list of 190 seed words.

Strong corroboration for the approach advocated in this paper is provided by Miller, et al (2004), in which cluster-based features are combined with a sequential maximum entropy model proposed in Collins (2002) to advance the state of the art. In addition, using active learning, the authors are able to reduce human labeling effort by an order of magnitude.

Miller, et al, use a proprietary data set for training and testing, so it is difficult to make a close comparison of outcomes. At roughly comparable training set sizes, they appear to achieve a score of about 0.89 (F1) with a "conventional" HMM, versus 0.93 using the discriminative learner trained with cluster features (compared with 0.86 reached by BWI). Both the HMM and Collins model are constrained to account for an entire sentence in tagging it, making determinations for all fields simultaneously, in contrast to the individual, local boundary detections made by BWI. This characteristic probably accounts for the accuracy advantage they appear to enjoy.

An interesting distinguishing feature of Miller, et al, is their use of hierarchical clustering. While much is made of the ability of their approach to accomodate different levels of granularity automatically, no evidence is provided that the hierarchy provides real benefit. At the same time, our work shows that significant gains can be realized with a single, sufficiently granular partition of terms. It is known, moreover, that greedy agglomerative clustering leads to partitions that are sub-optimal in terms of a mutual information objective function (see, for example, Brown, et al (1992)). Ultimately, it is left to future research to determine how sensitive, if at all, the NER gains are to the details of the clustering.

5 Conclusion

There are several ways in which this work might be extended and improved, both in its particular form and in general:

- BWI models initial and terminal boundaries, but ignores characteristics of the extracted phrase other than its length. We are exploring mechanisms for modeling relevant phrasal *structure*.
- While global statistical approaches, such as se-

quential averaged perceptrons or CRFs (McCallum and Li, 2003), appear better suited to the NER problem than local symbolic learners, the two approaches search different hypothesis spaces. Based on the surmise that, by combining them, we can realize improvements over either in isolation, we are exploring mechanisms for integration.

- The distributional clusters we find are independent of the problem to which we want to apply them and may sometimes be inappropriate or have the wrong granularity. We are exploring ways to produce groupings that are sensitive to the task at hand.

Our results clearly establish that an unsupervised distributional analysis of a text corpus can produce features that lead to enhanced precision and, especially, recall in information extraction. We have successfully used these features in lieu of domain-specific, labor-intensive resources, such as syntactic analysis and special-purpose gazetteers. Distributional analysis, combined with light supervision, is an effective, stable alternative to bootstrapping methods.

Acknowledgments

This material is based on work funded in whole or in part by the U.S. Government. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors, and do not necessarily reflect the views of the U.S. Government.

References

S.W. Bennett and C. Aone. 1997. Learning to tag multilingual texts through observation. In *Proc. 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP-2)*, August.

D.M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Proc. 5th Conference on Applied Natural Language Processing (ANLP-97)*, April.

P.F. Brown, V.J. Della Pietra, P.V. deSouza, J.C. Lai, and R.L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

X. Carreras, L. Màrquez, and L. Padró. 2002. Named entity extraction using AdaBoost. In *Proceedings of CoNLL-2002*, Taipei, Taiwan.

A. Clark. 2000. Inducing syntactic categories by context distribution clustering. In *CoNLL 2000*, September.

M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proc. 1999 Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*.

M. Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP-2002*.

S. Cucerzan and D. Yarowsky. 1999. Language independent named entity recognition combining morphological and contextual evidence. In *Proc. 1999 Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*, pages 90–99.

I. S. Dhillon, S. Mallela, and D. S. Modha. 2003. Information-theoretic co-clustering. Technical Report TR-03-12, Dept. of Computer Science, U. Texas at Austin.

D. Freitag and N. Kushmerick. 2000. Boosted wrapper induction. In *Proc. 17th National Conference on Artificial Intelligence (AAAI-2000)*, August.

A. McCallum and W. Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proc. 7th Conference on Natural Language Learning (CoNLL-03)*.

S. Miller, J. Guinness, and A. Zamanian. 2004. Name tagging with word clusters and discriminative training. In *Proceedings of HLT/NAACL 04*.

E. F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*.

R.E. Schapire and Y. Singer. 1998. Improved boosting algorithms using confidence-rated predictions. In *Proc. 11th Annual Conference on Computational Learning Theory (COLT-98)*, pages 80–91, July.

H. Schütze. 1995. Distributional part-of-speech tagging. In *Proc. 7th EACL Conference (EACL-95)*, March.

M. Thelen and E. Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proc. 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*.

D. Wu, G. Ngai, M. Carpuat, J. Larsen, and Y. Yang. 2002. Boosting for named entity recognition. In *Proceedings of CoNLL-2002*, Taipei, Taiwan.