Active Learning for Information Extraction with Multiple View Feature Sets

Rosie Jones ROSIE.JONES@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Rayid Ghani RAYID.GHANI@ACCENTURE.COM
Accenture Technology Labs, 161 N. Clark St, Chicago, IL 60601 USA

Tom Mitchell TOM.MITCHELL@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA

Ellen Riloff RILOFF@CS.UTAH.EDU
School of Computing, University of Utah, Salt Lake City, UT 84112 USA

Abstract

A major problem with machine learning approaches to information extraction is the high cost of collecting labeled examples. Active learning seeks to make efficient use of a labeler's time by asking for labels based on the anticipated value of that label to the learner. We consider active learning approaches for information extraction problems where each example is described by two distinct sets of features, either of which is sufficient to approximate the function; that is, they fit the cotraining problem setting. We discuss a range of active learning algorithms and show that using feature set disagreement to select examples for active learning leads to improvements in extraction performance regardless of the choice of initially labeled examples. The result is an active learning approach to multiple view feature sets in general, and noun phrase extraction in particular, that significantly reduces training effort and compensates for errors in initially labeled data.

1. Introduction

One difficulty with machine learning techniques for information extraction is the high cost of collecting labeled examples. We can make more efficient use of the trainer's time by asking them to label only instances that are most useful for the learner. Research in active learning has shown that using a pool of unlabeled examples and prompting the user to only label examples that have high anticipated value reduces the number of examples required for tasks such as text classification , parsing, and information extraction (Thompson et al., 1999; Soderland, 1999). Bootstrapping algorithms have been proposed for similar learning problems (Blum & Mitchell, 1998; Nigam & Ghani, 2000; Collins & Singer, 1999; Muslea et al., 2000) that fall into the cotraining setting *ie.* they have the property that each example can be described by multiple feature sets, any of which are sufficient to approximate the function.

We present an active learning framework for problems that fall into the cotraining setting. Our approach differs from work by Muslea on co-testing (Muslea et al., 2000) in that semi-supervised learning , using both labeled and unlabeled data, is interleaved in the active learning framework. Instead of learning with only labeled data, we use a botstrapping algorithm to learn from both labeled and unlabeled data and then select examples to be labeled by the user at every iteration. We do this by adapting co-EM to information extraction tasks and develop active learning techniques that make use of multiple feature sets.

We focus on extracting noun phrases that belong to a predefined set of semantic classes. Using the words within the noun phrase and the words surrounding it as two distinct feature sets, we describe active learning algorithms that make effective use of this division. Instead of relying only on a fixed, prelabeled set of examples, the active learning system keeps the user in the loop and presents them with examples to label at each iteration. We find that by utilizing the redundancy inherent in the data because of multiple feature sets, active learning approaches can significantly reduce the effort required to train information extraction systems. We also show that active learning can compensate for a bad choice of initial labeled examples and that the labeling effort is better spent *during* the active learning process rather than at the beginning, as done in standard supervised and semi-supervised learning.

2. Task and Data Set Representation

We focus on extracting noun phrases that correspond to organizations, people, and locations from a set of web pages. These semantic classes are often identified using *named entity recognizers* (e.g. (Collins & Singer, 1999)), but those tasks are usually limited to proper names, such as "John Smith" or "California". Our task is to identify all relevant noun phrases, such as "a telecommunications company" or "software engineer". As our data set, we used 4392 web pages from corporate websites collected for the WebKB project (Craven et al., 1998). 4160 were used for training and 232 were set aside as a test set. We preprocessed the web pages by removing HTML tags, and adding periods to the end of sentences when necessary. To label our data set, we extract all noun phrases (NPs) and manually label them with one or more of the semantic classes of interest (organizations, people, or locations). If a noun-phrase does not belong to any of these classes, it is assigned *none*.

Our goal is to recognize the semantic class of a word or phrase *in context*. Many words can belong to different semantic classes when they appear in different contexts. For example, the word "leader" can refer to a person, as in "a number of world leaders and other experienced figures" but can also occur in phrases which do not represent people, such as in "the company is a world leader". We have identified three common situations where the semantic class of a word can vary depending on the local context:

- **General Polysemy:** many words have multiple meanings. For example, "company" can refer to a commercial entity or to companionship.

- **General Terms:** many words have a broad meaning that can refer to entities of various types. For example, "customer" can refer to a person or a company.

- **Proper Name Ambiguity:** proper names can be associated with entities of different types. For example, "John Hancock" can refer to a person or a company, which reflects the common practice of naming companies after people.

Although the semantic category of a noun phrase may be ambiguous, the context in which the noun phrase occurs is often sufficient to resolve its category. Therefore, we cast our problem as one of classifying each instance of a noun phrase that appears within a document, based on both the noun phrase and its surrounding context. Each *noun phrase instance* (or example) consists of two items: (1) the noun phrase itself, and (2) a lexico-syntactic context surrounding the noun phrase. We used the AutoSlog (Riloff, 1996) system to generate patterns representing the lexico-syntactic contexts. For the remainder of the paper we will refer to these lexico-syntactic contexts simply as *contexts.*

By using both the noun phrases and the contexts surrounding them, we provide two different types of features to our classifier. In many cases, the noun phrase itself will be unambiguous and clearly associated with a semantic category (e.g., "the corporation" will nearly always be an organization). In these cases, the noun phrase alone would be sufficient for correct classification. In other cases, the context itself is highly predictive. For example, the context "subsidiary of <>" nearly always refers to an organization. In those cases, the context alone is sufficient. There will also be cases where either of these features by itself will be ambiguous with respect to the semantic class. We discuss these ambiguities in the next section and measure the extent to which these are present in our data set.

2.1. Ambiguity of Classes

Since our training corpus is unlabeled, we cannot measure the ambiguity directly. However, the examples in our test set were randomly drawn from the same distribution, and manually labeled, so ambiguity of noun-phrases and contexts in the test set is indicative of their ambiguity in the training set as well. During the labeling process, when an example was judged as belonging to multiple classes, multiple labels were assigned. An example is the sentence "We welcome feedback", where the "We" could refer to an organization, or the people of the organization. This kind of ambiguity also occurs when countries (locations) act as agents (organizations). Tables 1 and 2 summarize the ambiguity in noun-phrases and contexts in our test set, by showing how many noun-phrases were ambiguous with respect to which classes. Noun-phrases that did not fall into any of the categories `location`, `organization` or `person` were labeled as `none`.

Noun phrases are mostly unambiguous (only 2% of the

Ambiguity	**Class(es)**	**Number of NPs**
No Ambiguity	none	3574
	loc	114
	org	451
	person	189
Belonging to TWO classes	loc, none	6
	org, none	31
	person, none	25
	loc, org	6
	org, person	13
Belonging to THREE classes	loc, org, none	1
	org, person, none	3

Table 1. Noun-Phrase Ambiguity: The number of NPs that belong to each combination of classes. NPs are relatively unambiguous (4328 out of 4413 only belong to a single class).

Ambiguity	**Class(es)**	**Number of Contexts**
No Ambiguity	none	1068
	loc	25
	org	98
	person	59
Belonging to TWO classes	loc, none	51
	org, none	271
	person, none	206
	loc, org	5
	org, person	50
Belonging to THREE classes	loc, org, none	18
	org, person, none	83
Belonging to all FOUR classes	loc, org, person, none	6

Table 2. Context Ambiguity: The number of Contexts that belong to each combination of classes. Contexts are relatively ambiguous - 6 of the contexts were labeled as belonging to all 4 classes

4413 unique noun-phrases belong to 2 or more classes) but are relatively sparse in the training set; only 1887 of these noun-phrases had been seen in the training set. Thus for 57% of the noun-phrases in the test set, we have no training information at all. In contrast, 37% of the contexts are ambiguous but each of them occurs more often and can be modeled better. 91% of these 1940 contexts from the test set also appear in the training set. These measurements reinforce our previous assumption both the noun phrase and the context will play a role in determining the correct classification for each example.

3. Active Learning Problem Setting

Active learning is the problem of determining which unlabeled instances to label next as learning proceeds, in order to learn most accurately from the least labeling effort. The detailed problem setting varies with the form of the target function, the pool of unlabeled instances available, and the type of training information sought from the trainer. This section defines the active learning problem setting we consider, by placing it along each of these dimensions.

Form of target function to be learned. We consider active learning of a target function $f : X \to Y$ that maps a set X of instances to a set Y of possible values. We only consider target functions where instances are described by two distinct sets of features X_1 and X_2 (i.e., $X = X_1 \times X_2$), such that the target function can be approximated either in terms of X_1 or in terms of X_2. In our information extraction task, X_1 describes the noun phrase itself, and X_2 describes the context in which it appears.

For example, consider a problem where each instance $x \in X$ is a noun phrase along with its surrounding linguistic context (e.g., "drove to ⟨New York ⟩"), where the target function f specifies whether or not the noun phrase refers to a location, where the first set of features X_1 consists of the noun phrase itself (e.g., ⟨New York ⟩), and where the second set of features X_2 consists of the linguistic context (e.g., "drove to ⟨⟩"). Ideally, we assume that the target function f can be expressed in terms of X_1 alone, and also in terms of X_2 alone (e.g., that it is possible to determine whether the instance refers to a location, based solely on the context "I drove to ⟨⟩", and also based solely on the noun phrase "⟨New York ⟩." Put more formally, we assume $X = X_1 \times X_2$, where there exist functions $g_1 : X_1 \to Y$ and $g_2 : X_2 \to Y$ such that $f(x) = g_1(x_1) = g_2(x_2)$ for all $x = x_1|x_2$. In the real-world domains considered here, this ideal assumption is not fully satisfied, as described in Section 2 and Tables 1 and 2.

Pool of unlabeled data available. A second dimension for defining the active learning problem involves assumptions about how and when unlabeled instances are made available to the active learner. We begin with the usual PAC-learning assumption (Ehrenfeucht et al., 1989), that the instances X are generated according to some fixed but unknown probability distribution $P(X)$, and that the goal of the learner is to minimize the probability that it will misclassify future instances drawn randomly according to this same distribution. We can make several assumptions about how unlabeled instances are obtained by the active learner. We could assume that a fixed pool containing n instances is collected in advance according to the distribution $P(X)$, and that this fixed pool is all that is available to the active learner. This setting is considered in (McCallum & Nigam, 1998). We call this the *fixed random pool* setting. An alternative is to assume that the active learner can draw new instances at random from $P(X)$ during learning, so that it is not limited to the fixed pool. This setting is considered in (Cohn et al., 1994), and we will refer to this as the *ongoing random sampling* setting. A further alternative is to assume the learner can synthesize any syntactically legal instance in X, regardless of $P(X)$, and ask the trainer for information about this instance. While this setting is interesting, it can lead to synthetic examples that are not intelligible to the trainer (Baum & Lang, 1992). This setting is considered in (Angluin, 1988) . We will call this the *synthesized instances* setting, though it has sometimes been referred to as "membership querying." In this paper, we consider only the fixed random pool setting for active learning.

Information provided by the trainer. A third dimension concerns what information is to be provided to and by the trainer. In the *standard labeling* setting the trainer is provided an unlabeled instance, and in return provides the label. A different possibility is that the trainer is provided part of the description (e.g., provided only "drove to ⟨⟩"), and required to label it. We will call this the *single feature set labeling* setting. Another possibility is that the trainer is allowed to demur in some cases, providing a label only when certain (e.g., the trainer may decline to label "occurred in ⟨⟩" because of its ambiguity, but agree to label "drove to ⟨⟩" as a reliable location context).

To summarize, we consider an active learning problem in which the target function follows the *cotraining assumption*, the data available to the active learner is a *fixed random pool*, and we compare *standard labeling* to *single feature set labeling.*

4. Algorithmic Overview

Our approach consists of the following steps: a small set of words (*seedwords* or *seeds*) and a set of documents are provided. Instances in the document collection are initially labeled using the seeds (*initial examples*) and the annotated documents (including the unlabeled instances) are given to the bootstrapping algorithm. After every iteration of the bootstrapping algorithm, a human labeler is asked to label a set of examples selected by the active learning method. The design of our information extraction system requires answering the following questions:

1. How to label the initial examples for the bootstrapping algorithm?

2. What bootstrapping method will be used to learn from a combination of labeled and unlabeled data?

Class	SeedWords
locations	australia, canada, china, england, france, germany, japan, mexico, switzerland, united states
organizations	inc., praxair, company, companies, arco dataram, halter marine group, xerox, rayonier timberlands, puretec
people	customers, subscriber, people, users, shareholders, individuals, clients, leader, director, customer

Table 3. Seedwords used for initialization of bootstrapping.

3. What is the best active learning algorithm for requesting additional labels from the trainer?

4. What is the best method to assign labels to *test* instances?

We discuss and answer these questions below.

4.1. Method for Initial Labeling

The set of seedwords we use to generate initially labeled examples for the three information extraction tasks are shown in Table 3. The `locations` seedwords are the same as those used in (Riloff & Jones, 1999). The seeds for `organizations` and `people` were chosen by sorting noun-phrases in the training set by frequency, and selecting the first ten matching the target class. Note that this method does not necessarily lead to the best choice of seedwords, but is a simple method not requiring skill and experience. A domain expert might be able to pick better seedwords but we wanted to experiment with words that a non-expert could easily generate. Seedwords may be of poor quality if they are either (1) infrequent in the documents, or (2) ambiguous. We run experiments to examine whether active learning can compensate for "poor" seedwords (in terms of both 1 and 2) and report results in Section 5.

There was ambiguity across all sets of seedwords. In particular, "leader" refers to an organization more often than to a person in our data set, but it was used as a person keyword during Fixed Initialization. We will show that our algorithms are robust enough to recover from this kind of ambiguity. We examine two methods for creating initial labeled examples to jumpstart the bootstrapping process, one of which allows us to correct these ambiguities at the beginning:

Fixed Initialization: All noun phrases whose head noun (right-most word) matches a seed word are considered to be positive training instances, regardless of the context in which they appeared. This approach was also used in (Riloff & Jones, 1999). This is frequently correct, for example labeling the city "Columbia" as a location in the example "... headquartered in Columbia". However, in the example "Columbia published ..." it refers to a publishing company, not a location.

Active Initialization: To address errors introduced by ambiguity in the automatic labeling phase, we tried a novel method that incorporates active learning. In *active initialization*, examples matching the seed words are interactively labeled by the trainer before beginning the bootstrapping process. We hypothesize that by actively labeling examples at the outset and correcting the errors introduced by ambiguous seedwords, we can provide the bootstrapping algorithms with better initial examples and thus improve extraction performance. For reasonably frequent seed words, this requires significant numbers of examples to be labeled at the outset; 669 examples for `locations`, 3406 for `organizations`, and 2521 for `people`, for the seed words in Table 3 and our training collection.

4.2. Bootstrapping Method: coEM

Unlike previous work in active learning where the classifiers are only learned on labeled data, we use a bootstrapping method to learn from both labeled and unlabeled data. The bootstrapping algorithm we use for the information extraction task is coEM. *coEM* is a hybrid algorithm, proposed by (Nigam & Ghani, 2000), combining features from both co-training and Expectation-Maximization (EM). coEM is iterative, like EM, but uses the feature split present in the data, like co-training. The separation into feature sets we used is that of noun-phrases and contexts. The algorithm proceeds by initializing the noun-phrase classifier $\hat{g_1}(x_1)$ using the labeled data only. Then $\hat{g_1}(x_1)$ is used to probabilistically label all the unlabeled data. The context classifier $\hat{g_2}(x_2)$ is then trained using the original labeled data plus the unlabeled data with the labels provided by $\hat{g_1}$. Similarly, $\hat{g_2}$ then relabels the data for use by $\hat{g_1}$, and this process iterates until the classifiers converge. For final predictions over the test set, $\hat{g_1}$ and $\hat{g_2}$ predictions are combined by assuming independence, and assigning the test example probability proportional to $\hat{g_1}(x_1)\hat{g_2}(x_2)$.

In earlier work (Ghani & Jones, 2002), we compared coEM with metabootstrapping (Riloff & Jones, 1999) and found coEM to be better.

4.3. Active Learning Methods in the Cotraining Setting

The cotraining problem structure lends itself to a variety of active learning algorithms. In *co-testing* (Muslea et al., 2000) the two classifiers are trained only on available labeled data, then run over the unlabeled data. A *contention set* of examples is then created, consisting of all unlabeled examples on which the classifiers disagree. Examples from this contention set are selected at random, a label is requested from the trainer, both classifiers are retrained, and the process repeats.

While this naïve co-testing algorithm was shown to be quite effective, it represents just one possible approach to active learning in the co-training setting. It is based on training the two classifiers $\hat{g_1}$ and $\hat{g_2}$ using labeled examples only, whereas work by (Collins & Singer, 1999; Blum & Mitchell, 1998; Riloff & Jones, 1999) has shown that unlabeled data can bootstrap much more accurate classifiers. In this paper, we use both labeled and unlabeled data to create our classifiers. Also, instead of selecting new examples uniformly at random from the contention set, one might rank the examples in the contention set according to some criterion reflecting the value of obtaining their label. In this paper, we propose and experiment with active learning algorithms that use unlabeled data for training $\hat{g_1}$ and $\hat{g_2}$, in addition to using $\hat{g_1}$ and $\hat{g_2}$ to determine which unlabeled example to present to the trainer. We also consider a variety of strategies for selecting the best example from the contention set:

Uniform Random Selection: This baseline method selects examples according to a uniform distribution. Each noun-phrase, context pair that occurs at least once in the training set is selected with equal probability. Example frequency is ignored.

Density Selection: The most frequent unlabeled noun-phrase context pair is selected for labeling at each step. This method is based on the assumption that labeling frequently occurring examples would be beneficial for the learner.

Feature Set Disagreement: Since we learn two distinct classifiers that apply to the same instance, one way to select instances where a human trainer can provide useful information is to identify instances where these two classifiers disagree. This approach can be viewed as a form of *query-by-committee* (QBC), (Freund et al., 1997; Liere & Tadepalli, 1997) or *uncertainty sampling* (Thompson et al., 1999), where the committee consists of models that use different feature sets and is similar to that used by (McCallum & Nigam, 1998). Our selection criterion is based on Kullback-Leibler (KL) divergence. Our final ranking gave each example a density-weighted KL score, by multiplying $KL(\hat{P}_{g_1}(+|x), \hat{P}_{g_2}(+|x))$ by the frequency of the example. Examples were selected deterministically, with the next unlabeled example taken each time. For these experiments we used only a single committee member per feature set, though an obvious extension is to have multiple committee members per feature set.

Context Disagreement: All *active* selection algorithms described so far use the *standard labeling* paradigm, with the user labeling a pair consisting of a noun-phrase and its context. However, we can also label noun phrases independent of context, and since each noun phrase may occur in many contexts, this may lead to greater value in labeling. For example, "Italy" occurs with "centers in ⟨⟩", "operations in ⟨⟩", "introduced in ⟨⟩", "partners in ⟨⟩", and "offices in ⟨⟩", so labeling "Italy" provides us with information about all of these contexts. In addition, we can use the different contexts as votes by committee members about the label for the noun-phrase. Selecting the noun-phrase with the most *context disagreement* may provide the bootstrapping algorithm with the most informative labeling. This is can be thought of as query-by-committee (QBC) with the committee consisting of different cooccurrences of elements of one feature set with elements of the other feature set. We quantified context disagreement using density weighted KL divergence to the mean, as in feature set disagreement, and all the contexts of the noun-phrase were used as input to the KL divergence measure. We used the frequency of the noun-phrase to density-weight the KL divergence. The user then only labeled noun-phrases, in *single-feature set labeling.*

4.4. Extraction Method

The combination of bootstrapping and active learning results in a learned model consisting of noun-phrases and contexts, with corresponding learned probabilities for each. We use this model to assign scores to the unseen test instances by taking the product of the scores of the noun-phrase and context (both from the training set). Nouns and contexts that occur in the test set but have not been seen in the training set are assigned a prior to reflect the frequency of the classes in our dataset (0.027 for `locations`, 0.11 for `people` and 0.20 for `organizations`). Note that our examples include pronouns and other anaphoric references.

5. Results

We use coEM to label five examples per iteration, until 500 examples have been labeled. Then, we continue running coEM till convergence (usually around 400 iterations total) and use the learned model to score the test instances. We sort the test instances according to the score assigned by the extraction method and calculate precision-recall values.

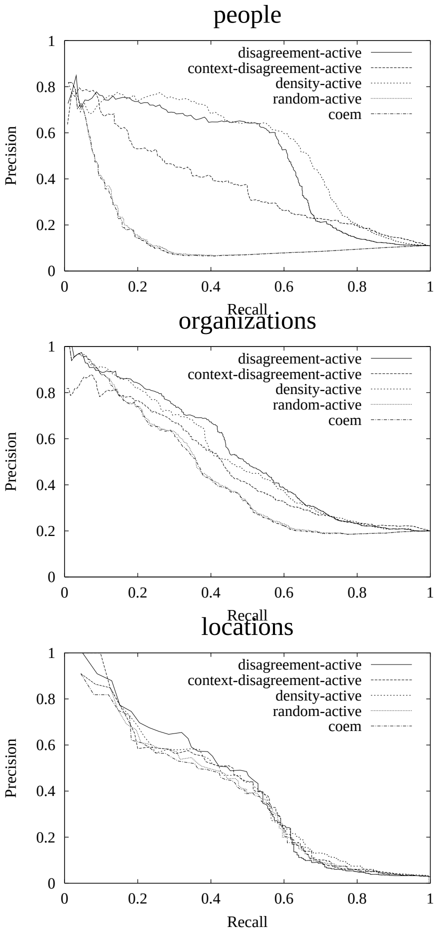

Figure 1. Comparison of active learning methods, for *locations*, *people* and *organizations.* for people, the sparsest class. Choosing examples to label based on disagreement between the two feature sets was most effective.

Disagreement and Density Active Learning Most Effective: Figure 1 shows how the different active learning methods perform after the user has labeled 500 examples. Feature set disagreement ("Disagreement") outperforms *all* other methods, except in the "people class" where density based active learning performs best. The people class contains many pronouns, which are frequently selected by density-based selection. Uniform random labeling of 500 examples does not improve over the baseline coEM using only the initial seeds. We believe this result to be significant as it shows that *randomly* selecting examples to label is no better than not labeling at all and letting coEM learn from the more promising initial labeled and unlabeled examples. This makes sense in our setting, since our positive classes are sparse, and random labeling does not provide much information about the positive class, compared with the dense information provided with the seeds. However, if we started with no seeds at all, some information would be gained by random labeling.

Although our active learning algorithms improve over the baseline in all cases, the improvement is most marked for the `people` class – this was the class with

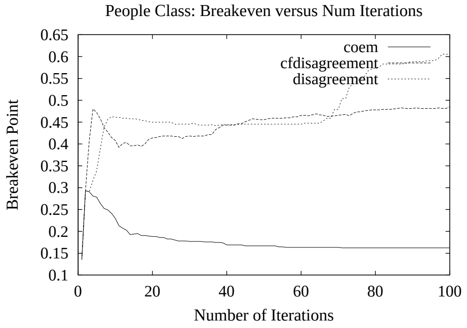

Figure 2. Breakeven point between precision and recall shown for each iteration for learning the people class. 5 examples are labeled per iteration for active learning, up to a total of 500 examples labeled. coEM improves slightly with the first few iterations, then degrades. With active learning, the most substantial improvements are made in the first few iterations, but labeling more data continues to improve results.

the most ambiguous initial seeds. This provides evidence that active learning can compensate for poor seed choices.

Labeling instances based on their frequency ("Density") was also very effective for `people` and `organizations`, which were frequent in our dataset. Context disagreement, which uses the single feature set labeling setting, did not perform as well as Disagreement or Density labeling, which use standard labeling. Single feature set labeling is useful (better than no active learning), but using both feature sets to select instances is a more effective technique.

Substantial Improvements with Small Amounts of Labeling: As can be seen in Figure 2, the most substantial improvements with active learning are made in the first few iterations, but labeling more data continues to improve results. This suggests that we can perform favorable trade-offs between labeling time and desired levels of accuracy.

Active Learning More Useful than Active Initialization: We found that active initialization (manually labeling and correcting errors in the initial labeled examples due to ambiguous seedwords) did not perform significantly better than fixed initialization. When our active learning method is provided a set of initial instances that are "clean" and unambiguous, the extraction performance does not improve. This suggests that the active learning methods are robust to ambiguous/noisy training data and can recover from poor initial seeds. This is shown in Figure 3. We also find that the active learning method (with 500 examples labeled for `locations`) performed better than using bootstrapping with coEM with active initialization

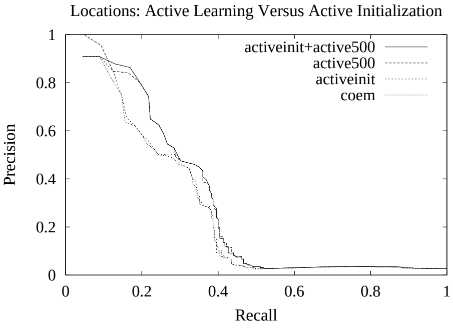

Figure 3. Labeling 500 noun-phrase-context pairs with active learning (and no initial labeling) does better than using active initialization, which requires labeling 693 examples at the outset. Labeling the initial examples and then labeling another 500 with active learning performs best.

(with 693 examples labeled). This is an important result since if we have a fixed amount of time to label instances, active learning can be a more effective use of this time than labeling the instances at the outset.

Active Learning Compensates for Infrequent Seeds: Figure 4 shows that selecting 20 country names uniformly from a set of 253 leads to variable results, with effectiveness related to frequency of initial seeds. The graph on the left contains three sets of 20 seeds each, matching 133, 129 and 34 examples in our training set, respectively. In the left-hand graph, the seed set matching only 34 examples performs poorly with the bootstrapping algorithm, but when combined with active learning, is able to produce results comparable with the other seed sets. The graph on the right shows the results for a frequent seed set (occurring 673 times in the training set). It is interesting that active learning improves results in all cases and compensates for seeds that are infrequent in the document set. Thus with active learning we can obtain results superior to bootstrapping on the best seed set, regardless of the seed set we choose.

Summary of Results We compared different metrics for selecting examples to label and found that using the disagreement between classifiers built with the two feature sets worked well. Manually correcting initial examples that were mislabeled due to ambiguous seeds is not as effective as providing the active learning algorithm with an arbitrary set of seeds and labeling examples during the learning process. Context-disagreement used the single feature set labeling setting, and did not perform as well as methods using standard labeling. Using only a single feature set for labeling may allow inaccuracies to creep into the labeled set, if any of the examples are ambiguous with respect to that feature set. In addition, disagreement

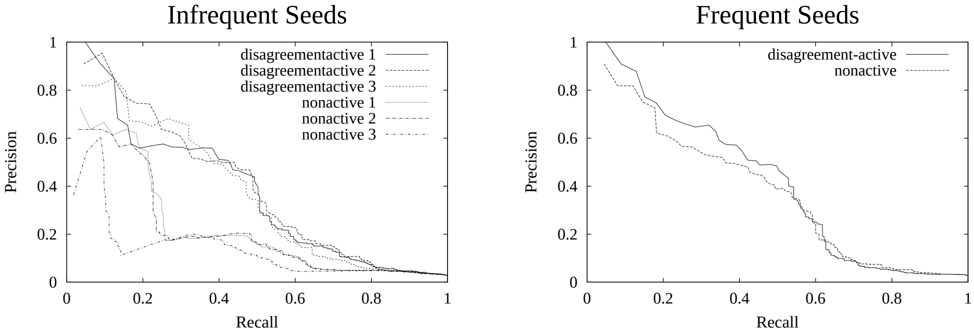

Figure 4. Active learning provides gains even with a good choice of frequent seeds (right hand graph). With a very poor set of initial seeds, active learning permits comparable results (left hand graph).

between members of a single feature set may reflect inherent ambiguity in the example, and not uncertainty in the learner.

6. Conclusions

We presented a framework for incorporating active learning in the semi-supervised learning paradigm by interleaving a bootstraping algorithm that learns from both labeled and unlabeled data with a variety of sample selection techniques that present the user with examples to label at each iteration. We show that employing the redundancy in feature sets and designing algorithms that exploit this redundancy enables both bootstrapping and active learning to be effective for training information extractors. The techniques presented in this paper are shown to be robust and are able to compensate for bad choice of initial seedwords. Although the results shown here are specific to the information extraction setting, our approach and framework are likely to be useful in designing active learning algorithms for settings where a natural, redundant division of features exists.

References

Angluin, D. (1988). Queries and concept learning. *Machine Learning*, *2*, 319–342.

Baum, E., & Lang, K. (1992). Query learning can work poorly when a human oracle is used. *International Joint Conference on Neural Networks.*

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *COLT-98.*

Cohn, D. A., Atlas, L., & Ladner, R. E. (1994). Improving generalization with active learning. *Machine Learning*, *15*, 201–221.

Collins, M., & Singer, Y. (1999). Unsupervised Models for Named Entity Classification. *EMNLP/VLC.*

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., & Slattery, S. (1998). Learning to Extract Symbolic Knowledge from the World Wide Web. *AAAI-98.*

Ehrenfeucht, A., Haussler, D., Kearns, M., , & Valiant, L. (1989). A general lower bound on the number of examples needed for learning. *Information and Computation*, *82*, 247–261.

Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, *28*, 133–168.

Ghani, R., & Jones, R. (2002). A comparison of efficacy of bootstrapping algorithms for information extraction. *LREC 2002 Workshop on Linguistic Knowledge Acquisition.*

Liere, R., & Tadepalli, P. (1997). Active learning with committees for text categorization. *AAAI-97.*

McCallum, A. K., & Nigam, K. (1998). Employing EM and pool-based active learning for text classification. *ICML.*

Muslea, I., Minton, S., & Knoblock, C. A. (2000). Selective sampling with redundant views. *AAAI/IAAI.*

Nigam, K., & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. *CIKM-2000.*

Riloff, E. (1996). An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. *85*, 101–134.

Riloff, E., & Jones, R. (1999). Learning Dictionaries for Information Extraction by Multi-level Boot-strapping. *AAAI-99.*

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, *34*, 233–272.

Thompson, C. A., Califf, M. E., & Mooney, R. J. (1999). Active learning for natural language parsing and information extraction. *ICML.*