Appears in *Proceedings of the Sixteenth International Machine Learning Conference*, pp.406-414, Bled, Slovenia, June 1999

Active Learning for Natural Language Parsing and Information Extraction

Cynthia A. Thompson
CSLI, Ventura Hall
Stanford University
Stanford, CA 94305
cthomp@csli.stanford.edu

Mary Elaine Califf
Dept. of Applied Computer Science
Illinois State University
Normal, IL 61790
mecalif@ilstu.edu

Raymond J. Mooney
Dept. of Computer Sciences
University of Texas
Austin, TX 78712
mooney@cs.utexas.edu

Abstract

In natural language acquisition, it is difficult to gather the annotated data needed for supervised learning; however, unannotated data is fairly plentiful. Active learning methods attempt to select for annotation and training only the most informative examples, and therefore are potentially very useful in natural language applications. However, existing results for active learning have only considered standard classification tasks. To reduce annotation effort while maintaining accuracy, we apply active learning to two non-classification tasks in natural language processing: semantic parsing and information extraction. We show that active learning can significantly reduce the number of annotated examples required to achieve a given level of performance for these complex tasks.

1 INTRODUCTION

Active learning is an emerging area in machine learning that explores methods that, rather than relying on a benevolent teacher or random sampling, actively participate in the collection of training examples. The primary goal of active learning is to reduce the number of supervised training examples needed to achieve a given level of performance. Active learning systems may construct their own examples, request certain types of examples, or determine which of a set of unsupervised examples are most usefully labeled. The last approach, *selective sampling* (Cohn, Atlas, & Ladner, 1994), is particularly attractive in natural-language learning, since there is an abundance of text, and we would like to annotate only the most informative sentences. For many language learning tasks, annotation is particularly time-consuming since it requires specifying a complex output rather than just a category label, so reducing the number of training examples required can greatly increase the utility of learning.

An increasing number of researchers are successfully applying machine learning to natural language processing (see Brill and Mooney (1997) for an overview). However, only a few have utilized active learning, and those have addressed two particular tasks: part of speech tagging (Dagan & Engelson, 1995) and text categorization (Lewis & Catlett, 1994; Liere & Tadepalli, 1997). Both of these are fundamentally classification tasks, while the tasks we address, semantic parsing and information extraction, are not. Many language learning tasks require annotating natural language text with a complex output, such as a parse tree, semantic representation, or filled template. However, the application of active learning to tasks requiring such complex outputs has not been well studied. Our research shows how active learning methods can be applied to such problems and demonstrates that it can significantly decrease annotation costs for important and realistic natural-language tasks.

The remainder of this paper is organized as follows. Section 2 presents background on active learning, and Section 3 introduces the two language-learning systems to which we apply active learning. Sections 4 and 5 describe the application of active learning to parser acquisition together with experimental results. Sections 6 and 7 describe the application of active learning to learning information extraction rules and present experimental results for this task. Section 8 suggests directions for future research. Finally, Section 9 describes some related research, and Section 10 presents our conclusions.

2 BACKGROUND ON ACTIVE LEARNING

Because of the relative ease of obtaining on-line text, we focus on selective sampling methods of active learning. In this case, learning begins with a small pool of annotated examples and a large pool of unannotated examples, and the learner attempts to choose the most informative additional examples for annotation. Existing work in the area has emphasized two approaches, *certainty-based* methods (Lewis & Catlett, 1994), and *committee-based* methods (Freund, Seung, Shamir, & Tishby, 1997; Liere & Tadepalli, 1997; Dagan & Engelson, 1995; Cohn et al., 1994).

In the certainty-based paradigm, a system is trained on a small number of annotated examples to learn an initial classifier. Next, the system examines unannotated examples, and attaches certainties to the predicted annotation of those examples. The k examples with the lowest certainties are then presented to the user for annotation and retraining. Many methods for attaching certainties have been used, but they typically attempt to estimate the probability that a classifier consistent with the prior training data will classify a new example correctly.

In the committee-based paradigm, a diverse committee of classifiers is created, again from a small number of annotated examples. Next, each committee member attempts to label additional examples. The examples whose annotation results in the most disagreement amongst the committee members are presented to the user for annotation and retraining. A diverse committee, consistent with the prior training data, will produce the highest disagreement on examples whose label is most uncertain with respect to the possible classifiers that could be obtained by training on that data.

Figure 1 presents abstract pseudocode for both certainty-based and committee-based selective sampling. In an ideal situation, the batch size, k, would be set to one to make the most intelligent decisions in future choices, but for efficiency reasons in retraining batch learning algorithms, it is frequently set higher. Results on a number of classification tasks have demonstrated that this general approach is effective in reducing the need for labeled examples (see citations above). Our current work has explored certainty-based approaches, although committee-based approaches for our tasks of learning parsers and information extraction rules is a topic for future research.

```
Apply the learner to n bootstrap examples, creating
  one classifier or a committee of them.
Until there are no more examples or the annotator
  is unwilling to label more examples, do:
    Use most recently learned classifier/committee
      to annotate each unlabeled instance.
    Find the k instances with the lowest annotation
      certainty/most disagreement amongst
      committee members.
    Annotate these instances.
    Train the learner on the bootstrap examples
      and all examples annotated to this point.
```

Figure 1: Selective Sampling Algorithm

3 NATURAL LANGUAGE LEARNING SYSTEMS

3.1 PARSER ACQUISITION

CHILL is a system that, given a set of training sentences each paired with a meaning representation, learns a parser that maps sentences into this semantic form (Zelle & Mooney, 1996). It uses *inductive logic programming* (ILP) methods (Muggleton, 1992; Lavrač & Džeroski, 1994) to learn a deterministic shift-reduce parser written in Prolog. CHILL solves the parser acquisition problem by learning rules to control the step by step actions of an initial, overly-general parsing shell. While the initial training examples are sentence/representation pairs, the examples given to the ILP system are positive and negative examples of states of the parser in which a particular operator should or should not be applied. These examples are automatically constructed by determining what sequence of operator applications (e.g., shift and reduce) leads to the correct parse. However, the overall learning task for which user feedback is provided is not a classification task.

This paper will focus on one application in which CHILL has been tested, learning an interface to a geographical database. In this domain, CHILL learns parsers that map natural-language questions directly into Prolog queries that can be executed to produce an answer. Following are two sample queries for a database on U.S. geography paired with their corresponding Prolog query:

What is the capital of the state with the biggest population?

```
answer(C, (capital(S,C), largest(P,
    (state(S), population(S,P))))).
```

What state is Texarkana located in?

```
answer(S, (state(S),
   eq(C,cityid(texarkana,_)),
   loc(C,S))).
```

Given a sufficient corpus of such sentence/representation pairs, CHILL is able to learn a parser that correctly parses many novel sentences into logical queries.

3.2 INFORMATION EXTRACTION

We have also developed a system, RAPIER, that learns rules for information extraction (IE) (Califf, 1998). The goal of an IE system is to find specific pieces of information in a natural-language document. The specification of the information to be extracted generally takes the form of a template with a list of slots to be filled with substrings from the document (Lehnert & Sundheim, 1991). IE is particularly useful for obtaining a structured database from unstructured documents and is being used for a growing number of Web and Internet applications.

RAPIER is a bottom-up relational learner, and acquires rules in the form of a sequence of patterns that identify relevant phrases in the document. The patterns are similar to regular expressions that include constraints on the words, part-of-speech tags, and semantic classes of the extracted phrase and its surrounding context; however, in the results in this paper, we use the simplest version of the system which only makes use of words. We have found that part-of-speech tags may be useful in some domains, but that words alone provide most of the power.

Like semantic parsing, IE is not a classification task; although, like parsing in CHILL, it can be mapped to a series of classification subproblems (Freitag, 1998; Bennett, Aone, & Lovell, 1997). However, RAPIER does not approach the problem in this manner, and in any case, the example annotations provided by the user are in the form of filled templates, not class labels.

In our active learning research, we have focused on one of the three tasks on which RAPIER has been extensively tested, that of extracting information about computer-related jobs from netnews postings. Figure 2 shows an example with part of the corresponding filled template. The task is to extract information for 17 slots appropriate for the development of a jobs database. The slots vary in their applicability to different postings. Relatively few postings provide salary information, while most provide information about the job's location. A number of the slots may have more than one filler; for example, there are slots for the platform(s) and language(s) that the prospective employee will use.

4 ACTIVE LEARNING FOR SEMANTIC PARSING

Applying certainty-based sample selection to both of these systems requires determining the certainty of a complete annotation of a potential new training example, despite the fact that individual learned rules perform only part of the overall annotation task. Therefore, our general approach is to compute certainties for each individual decision made during the processing of an example, and combine these to obtain an overall certainty for an example. Since both systems learn rules with no explicit uncertainty parameters, simple metrics based on coverage of training examples are used to assign certainties to rule-based decisions.

In CHILL, this approach is complicated slightly by the fact that the current learned parser may get stuck, and not even complete a parse for a potential new training example. This can happen because a control rule learned for an operator may be overly specific, preventing its correct application, or because an operator required for parsing the sentence may not have been needed for any of the training examples, so the parser does not even include it. If a sentence cannot be parsed, its annotation is obviously very uncertain and it is therefore a good candidate for selection. However, there are often more unparsable sentences than the batch size (k), so we must distinguish between them. This is done by counting the maximum number of sequential operators successfully applied while attempting to parse the sentence and dividing by the number of words in the sentence to give an estimate of how close the parser came to completing a parse. The sentences with a lower value for this metric are preferred for annotation.

If the number of unparsable examples is less than k, then the remaining examples selected for annotation are chosen from the parsable ones. A certainty for each parse, and thus each potential training example, is obtained by considering the sequence of operators applied to produce it. Recall that the control rules for each operator are induced from positive and negative examples of the contexts in which the operator should be applied. As a simple approximation, the number

Posting from Newsgroup

```
Telecommunications. SOLARIS Systems Administrator. 38-44K.
Immediate need

Leading telecommunications firm in need of an energetic
individual to fill the following position in the Atlanta office:

       SOLARIS SYSTEMS ADMINISTRATOR
       Salary: 38-44K with full benefits
       Location: Atlanta Georgia, no relocation assistance provided
```

Filled Template

```
computer_science_job
title: SOLARIS Systems Administrator
salary: 38-44K
state: Georgia
city: Atlanta
platform: SOLARIS
area: telecommunications
```

Figure 2: Sample Message and Filled Template

of examples used to induce the specific control rule used to select an operator is used as a measure of the certainty of that parsing decision. We believe this is a reasonable certainty measure in rule learning, since, as shown by Holte, Acker, and Porter (1989), *small disjuncts* (rules that correctly classify few examples) are more error prone than large ones. We then average this certainty over all operators used in the parse of the sentence to obtain the metric used to rank the example.

To increase the diversity of examples included in a given batch, we do not include sentences that vary only in known names for database constants (e.g., city names) from already chosen examples, nor sentences that contain a subset of the words present in an already chosen sentence.

5 EXPERIMENTAL RESULTS: SEMANTIC PARSING

For the experimental results in this paper, we use the following general methodology. For each trial, a random set of test examples is used and the system is trained on subsets of the remaining examples. First, n bootstrap examples are randomly selected from the training examples, then in each step of active learning, the best k examples of the remaining training examples are selected and added to the training set. The result of learning on this set is evaluated after each round. When comparing to random sampling, the k examples in each round are chosen randomly.

The initial corpus used for evaluating parser acquisition contains 250 questions about U.S. geography, paired with Prolog queries. This domain was chosen due to the availability of an existing hand-build natural language interface to a simple geography database containing about 800 facts. The original interface, *Geobase*, was supplied with Turbo Prolog 2.0 (Borland International, 1988). The questions were collected from uninformed undergraduates and mapped into logical form by an expert. Examples from the corpus were given in Section 3.1. The parser that is learned from the training data is used to process the test examples, the resulting queries submitted to the database, the answers compared to those generated by the correct representation, and the percentage of correct answers recorded.

In tests on this data, test examples were chosen independently for 10 trials with $n = 25$ bootstrap examples and a batch size of $k = 25$. The results are shown in Figure 3, where CHILL refers to random sampling, CHILL+Active refers to sample selection, and *Geobase* refers to the hand-built benchmark. Initially, the advantage of sample selection is small, since there

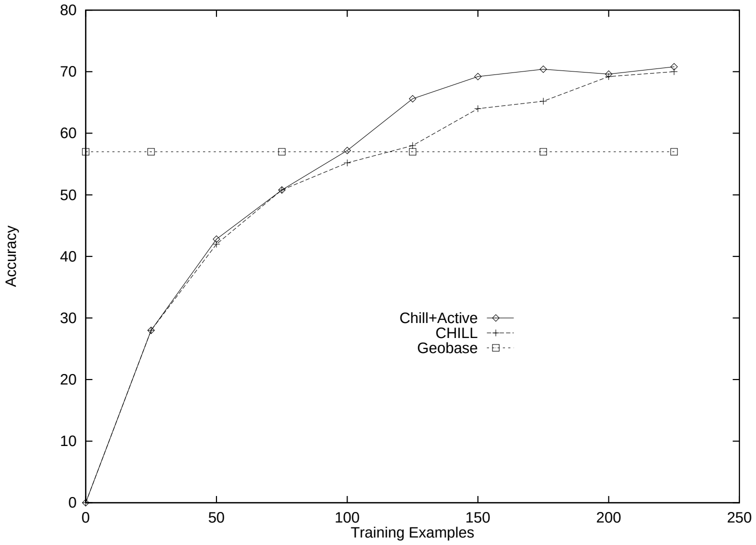

Figure 3: Parser Acquisition Results for Geography Corpus

is insufficient information to make an intelligent choice of examples; but after 100 examples, the advantage becomes clear. Eventually, the training set becomes exhausted, the active learner has no choice in picking the remaining examples, and both approaches use the full training set and converge to the same performance. However, the number of examples required to reach this level is significantly reduced when using active learning. To get within 5% of the final accuracy requires 125 selected examples but 175 random examples, a savings of 29%. Also, to surpass the performance of *Geobase* requires under 100 selected examples versus 125 random examples, a savings of 20%. According to a t-test, the differences between active and random choice at 125 and 175 training examples are statistically significant at the .05 level or better.

We also ran experiments on a larger, more diverse corpus of geography queries, where additional examples were collected from undergraduate students in an introductory AI course. The set of questions in the previous experiments was collected from students in introductory German, with no instructions on the complexity of queries desired. The AI students tended to ask more complex and diverse queries: their task was to give 5 interesting questions and the associated logical form for a homework assignment. There were 221 new sentences, for a total of 471. This data was split into 425 training sentences and 46 test sentences, for 10 random splits. For this corpus, we used $n = 50$ and $k = 25$. The results are shown in Figure 4. Here, the savings with active learning is about 150 examples to reach an accuracy close to the maximum, or about a 35% annotation savings. The curve for selective sampling does not reach 425 examples because of our elimination of sentences that vary only in database names and those that contain a subset of the words present in an already chosen sentence. Obviously this is a more difficult corpus, but active learning is still able to choose examples that allow significant savings in annotation cost.

6 ACTIVE LEARNING FOR INFORMATION EXTRACTION

A similar approach to certainty-based sample selection was used with Rapier. A simple notion of the certainty of an individual extraction rule is based on its coverage of the training data: $pos - 5 \cdot neg$, where pos is the number of correct fillers generated by the rule and neg is the number of incorrect ones. Again, "small disjuncts" that account for few examples are deemed less certain. Also, since Rapier, unlike Chill, prunes rules to prevent overfitting, they may generate spurious fillers for the training data; therefore, a significant penalty is included for such errors.

Given this notion of rule certainty, Rapier determines the certainty of a filled slot for an example being evaluated for annotation certainty. In the case where a single rule finds a filler for a slot, the certainty for the slot is the certainty of the rule that filled it. However, when more than one slot-filler is found, the certainty of the slot is defined as the minimum of the certainties

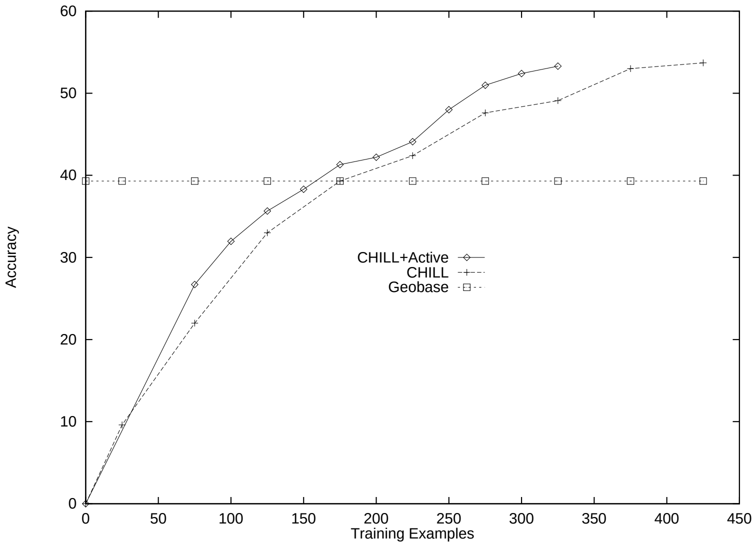

Figure 4: Parser Acquisition Results for a Larger Geography Corpus

of the rules that produced these fillers. The minimum is chosen since we want to focus attention on the least certain rules and find examples that either confirm or deny them.

A final consideration is determining the certainty of an empty slot. In some tasks, some slots are empty a large percentage of the time. For example, in the jobs domain, the salary is present less than half the time. On the other hand, some slots are always (or almost always) filled, and the absence of fillers for such slots should decrease confidence in an example's labeling. Consequently, we record the number of times a slot appears in the training data with no fillers and use that count as the confidence of the slot when no filler for it is found. Once the confidence of each slot has been determined, the confidence of an example is found by summing the confidence of all slots.

In order to allow for the more desirable option of actively selecting a single example at a time ($k = 1$), an incremental version of Rapier was created. This version still requires remembering all of the training examples but reuses and updates existing rules as new examples are added. The resulting system can incrementally incorporate new training examples reasonably efficiently, allowing each chosen example to immediately effect the result and therefore the choice of the next example.

7 EXPERIMENTAL RESULTS: INFORMATION EXTRACTION

The computer-related job-posting corpus used to test active learning in Rapier consists of 300 postings to the local newsgroup `austin.jobs`, as illustrated in Figure 2. Training and test sets were generated using 10-fold cross-validation, and learning curves generated by training on randomly or actively selected subsets of the training data for each trial. For active learning, there were $n = 10$ bootstrap examples and subsequent examples were selected one at a time from the remaining 260 examples.

In information extraction, the standard measurements of performance are precision (the percentage of items that the system extracted which should have been extracted) and recall (the percentage of items that the system should have extracted which it did extract). In order to combine these measurements to simplify comparisons, it is common to use F-measure: $F = (2 \cdot \mathit{precision} \cdot \mathit{recall})/(\mathit{precision} + \mathit{recall})$. It is possible to weight the F-measure to prefer recall or precision, but we weight them equally. For the active learning results, we measured performance at 10-example intervals. The results for random sampling are measured less frequently.

Figure 5 shows the results, where Rapier uses random sampling and Rapier+Active uses selective sampling. From 30 examples on, Rapier+Active consistently outperforms Rapier. The difference between the curves is not large, but does represent a large re-

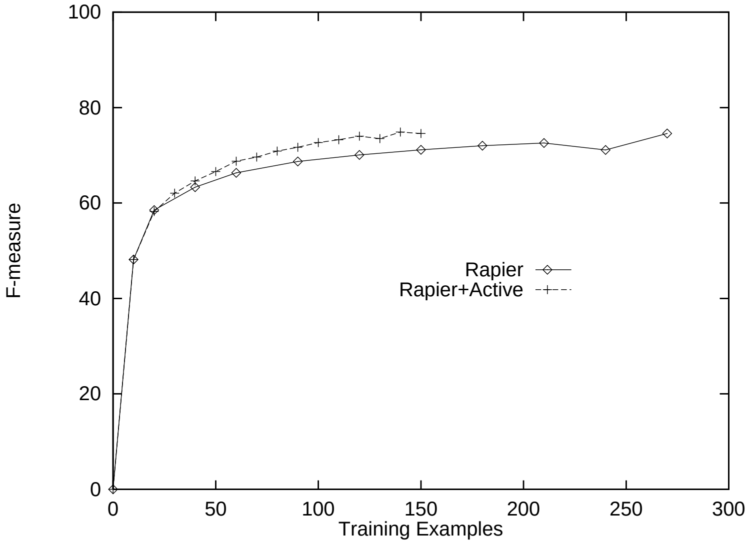

Figure 5: Information Extraction Results for Job Postings

duction in the number of examples required to achieve a given level of performance. At 150 examples, the average F-measure is 74.56, exactly the same as the average F-measure with 270 random examples. This represents a savings of 120 examples, or 44%. The differences in performance at 120 and 150 examples are significant at the 0.01 level according to a two-tailed paired t-test. The curve with selective sampling does not go all the way to 270 examples, because once the performance of 270 randomly chosen examples is reached, the information available in the data set has been exploited, and the curve will just level off as the less useful examples are added.

8 FUTURE WORK

Experiments on additional semantic parsing and information extraction corpora are needed to test the ability of this approach to reduce annotation costs in a variety of domains. It would also be interesting to explore active learning for other natural language processing problems such as syntactic parsing, word-sense disambiguation, and machine translation.

Our current results have involved a certainty-based approach; however, proponents of committee-based approaches have convincing arguments for their theoretical advantages. Our initial attempts at adapting committee-based approaches to our systems were not very successful; however, additional research on this topic is indicated. One critical problem is obtaining diverse committees that properly sample the version space (Cohn et al., 1994).

Although they seem to work quite well, the certainty metrics used in both Chill and Rapier are quite simple and somewhat *ad hoc*. A more principled approach based on learning probabilistic models of parsing and information extraction could perhaps result in better estimates of certainty and therefore improved sample selection.

Finally, a more intelligent method for choosing batch sizes is needed. From initial informal experiments with Chill, we have observed that the optimal batch size seems to vary with the total amount of training data. At first, small batches are most beneficial, but later in learning, larger batches seem better. However, converting Chill to an incremental version as done with Rapier might sidestep this issue and allow efficient learning at one step increments.

9 RELATED WORK

Cohn et al. (1994) were among the first to discuss certainty-based active learning methods in detail. They focus on a neural network approach to actively searching a version-space of concepts. Liere and Tadepalli (1997) apply active learning with committees to the problem of text categorization. They show improvements with active learning similar to those that we obtain, but use a committee of Winnow-based learners on a traditional classification task. Dagan and Engelson (1995) also apply committee-based learning to part-of-speech tagging. In their work, a committee of hidden Markov models is used to select examples for annotation. Lewis and Catlett (1994) use *heterogeneous* certainty-based methods, in which a simple classifier is used to select examples that are then annotated and presented to a more powerful classifier. Again, their methods are applied to text classification.

One other researcher has recently applied active learning to information extraction. Soderland's (1999) Whisk system uses an unusual form of selective sampling. Rather than using certainties or committees, Whisk divides the pool of unannotated instances into three classes: 1) those covered by an existing rule, 2) those that are near misses of a rule, and 3) those not covered by any rule. The system then randomly selects a set of new examples from each of the three classes and adds them to the training set. Soderland shows that this method significantly improves performance in a management succession domain; however, it is unclear how more traditional sample selection methods would perform by comparison.

10 CONCLUSIONS

Active learning is a new area of machine learning that has been almost exclusively applied to classification tasks. We have demonstrated its successful application to two more complex natural language processing tasks, semantic parsing and information extraction. The wealth of unannotated natural language data, along with the difficulty of annotating such data, make selective sampling a potentially invaluable technique for natural language learning. Our results on realistic corpora for semantic parsing and information extraction indicate that example savings as high as 44% can be achieved by employing active sample selection using only simple certainty measures for predictions on unannotated data. Improved sample selection methods and applications to other important language problems hold the promise of continued progress in using machine learning to construct effective natural language processing systems.

Acknowledgements

This research was supported by the National Science Foundation under grant IRI-9704943.

References

Bennett, S., Aone, C., & Lovell, C. (1997). Learning to tag multilingual texts through observation. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pp. 109–116.

Borland International (1988). *Turbo Prolog 2.0 Reference Guide*. Borland International, Scotts Valley, CA.

Brill, E., & Mooney, R. (1997). An overview of empirical natural language processing. *AI Magazine*, *18*(4), 13–24.

Califf, M. E. (1998). *Relational Learning Techniques for Natural Language Information Extraction*. Ph.D. thesis, Department of Computer Sciences, University of Texas, Austin, TX. Also appears as Artificial Intelligence Laboratory Technical Report AI 98-276 (see http://www.cs.utexas.edu/users/ai-lab).

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, *15*(2), 201–221.

Dagan, I., & Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 150–157 San Francisco, CA. Morgan Kaufman.

Freitag, D. (1998). Multi-strategy learning for information extraction. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 161–169.

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

Holte, R. C., Acker, L., & Porter, B. (1989). Concept learning and the problem of small disjuncts. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 813–818 Detroit, MI.

Lavrač, N., & Džeroski, S. (1994). *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood.

Lehnert, W., & Sundheim, B. (1991). A performance evaluation of text-analysis technologies. *AI Magazine*, *12*(3), 81–94.

Lewis, D. D., & Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In

Proceedings of the Eleventh International Conference on Machine Learning, pp. 148–156 San Francisco, CA. Morgan Kaufman.

Liere, R., & Tadepalli, P. (1997). Active learning with committees for text categorization. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pp. 591–596 Providence, RI.

Muggleton, S. H. (Ed.). (1992). *Inductive Logic Programming*. Academic Press, New York, NY.

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

Zelle, J. M., & Mooney, R. J. (1996). Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence* Portland, OR.