# Committee-Based Sampling For Training Probabilistic Classifiers

**Ido Dagan** and **Sean P. Engelson**
Department of Mathematics and Computer Science
Bar-Ilan University
52900 Ramat Gan, Israel
{dagan,engelson}@bimacs.cs.biu.ac.il

## Abstract

In many real-world learning tasks, it is expensive to acquire a sufficient number of labeled examples for training. This paper proposes a general method for efficiently training probabilistic classifiers, by selecting for training only the more informative examples in a stream of unlabeled examples. The method, *committee-based sampling*, evaluates the informativeness of an example by measuring the degree of disagreement between several model variants. These variants (the committee) are drawn randomly from a probability distribution conditioned by the training set selected so far (Monte-Carlo sampling). The method is particularly attractive because it evaluates the expected information gain from a training example implicitly, making the model both easy to implement and generally applicable.

We further show how to apply committee-based sampling for training Hidden Markov Model classifiers, which are commonly used for complex classification tasks. The method was implemented and tested for the task of tagging words in natural language sentences with parts-of-speech. Experimental evaluation of committee-based sampling versus standard sequential sampling showed a substantial improvement in training efficiency.

## 1 INTRODUCTION

In supervised concept learning, a classifier is trained on a set of labeled examples. In many real-world concept learning tasks, however, acquiring labeled examples is expensive. Hence, we wish to develop automated methods that reduce training cost by *active learning*, in which the learner has some control over the choice of examples which are labeled and used for training.

There are two main types of active learning. The first uses *membership queries*, in which the learner constructs examples and asks a teacher to label them (Angluin, 1988; MacKay, 1992b; Plutowski and White, 1993). While this approach provides proven computational advantages (Angluin, 1987), it is not always applicable since it is not always possible to construct meaningful and informative unlabeled examples for training, without knowing the target concept in advance. This difficulty may be overcome when a large set of unlabeled training data is available. In this case the second type of active learning, *selective sampling*, can be applied: The learner examines many unlabeled examples, and selects only the most informative for learning (Seung, Opper, and Sompolinsky, 1992; Freund et al., 1993; Cohn, Atlas, and Ladner, 1994; Lewis and Catlett, 1994; Lewis and Gale, 1994).

In this paper, we address the problem of selective sampling for training a probabilistic classifier. Classification in this framework is performed by a probabilistic model which, given an input example, assigns a probability to each possible classification and selects the most probable one.

Our research follows recent theoretical work on selective sampling in the Query By Committee paradigm (Seung, Opper, and Sompolinsky, 1992; Freund et al., 1993). In our *committee-based sampling* scheme, the learner receives a stream of unlabeled examples as input and decides for each of them whether to ask for its label or not. To that end, the learner constructs a 'committee' of classifiers based on the current training set. Each committee member then classifies the candidate example, and the learner measures the degree of disagreement among the committee members. The example is selected for labeling probabilistically, such that the selection probability is proportional to the degree of disagreement among the committee members.

Committee members are generated randomly according to the posterior probability distribution of the possible classifiers, given the statistics acquired from the training data seen so far. Hence the degree of committee disagreement measures the amount of uncertainty in labeling the new example based on the current training data. This measure captures both the sensitivity of classification to parameter values (distance from class 'boundaries'), as well as the statistical significance of parameter estimates, which single model selection schemes, such as (Lewis and Catlett, 1994), do not. In addition, selecting examples from the input stream by a biased random decision implicitly models the distribution over the space of input examples, causing the model to be adapted to the distribution of examples observed.

Much previous work in selective sampling has either been theoretical in nature, or has been tested on toy problems. We, like Lewis and Catlett (1994), are motivated by complex, real-world problems in the area of statistical natural language and text processing. In this field, probabilistic classifiers are often used to select a preferred analysis of the linguistic structure of a text (for example, its syntactic structure (Black et al., 1993), word categories (Church, 1988), or word senses (Gale, Church, and Yarowsky, 1993)). The parameters of such a classification model are estimated from a training corpus (a collection of text).

In the common case of supervised training, the learner uses a corpus in which each sentence is manually annotated with the correct analysis. Manual annotation is typically very expensive. As a consequence, few large annotated corpora exist, mainly for the English language, covering only a few genres of text. Selective sampling is an appropriate way to reduce annotation cost, as it is easy to obtain large volumes of raw text from which small subsets will be selected for annotation.

We have applied committee-based sampling to learning Hidden Markov Models (HMMs) for part-of-speech tagging of English sentences. Part-of-speech tagging is the task of labeling each word in the sentence with its appropriate part of speech (for example, labeling an occurrence of the word 'hand' as a noun or a verb). This task is non-trivial since determining a word's part of speech depends on its linguistic context. HMMs have been used extensively for this task (eg, (Church, 1988; Merialdo, 1991)), in most cases trained from corpora which have been manually annotated with the correct part of speech for each word. Our experiments on part-of-speech tagging, described in Section 6, show that using committee-based selection results in substantially faster learning rates, enabling the learner to achieve a given level of accuracy using far fewer training examples than by sequential training using all of the text.

# 2 BACKGROUND

## 2.1 QUERY BY COMMITTEE

Query By Committee (QBC) (Seung, Opper, and Sompolinsky, 1992; Freund et al., 1993) is a method for selecting informative training examples out of a stream of unlabeled examples. When an example is selected the learner queries the teacher for its correct label and adds it to the training set. As examples are selected for training, they restrict the set of *consistent concepts*, i.e, the set of concepts that label all the training examples correctly (the version space). This method was proposed for learning binary concepts in cases where there exists a prior probability distribution measure over the concept class (a Bayesian learning framework).

A simple version of QBC, which was analyzed in (Freund et al., 1993) (summarized in (Freund, 1994)), uses the following selection algorithm:

1. Draw an unlabeled input example at random from the probability distribution of the example space.
2. Select at random two hypotheses according to the probability distribution of the concept class, restricted to the set of consistent concepts.
3. Use the example for training if the two hypotheses disagree on its classification.

Freund et al. prove that, under some assumptions, this algorithm achieves an exponential reduction in the number of labeled examples required to achieve a desired classification accuracy, compared with random selection of training examples. This speedup is achieved because the algorithm tends to select examples that split the version space into two parts of similar size. One of these parts is eliminated from the version space after the example and its correct label are added to the training set.

## 2.2 SELECTIVE SAMPLING FOR PROBABILISTIC CLASSIFIERS

Probabilistic classifiers do not fall within the framework addressed in the theoretical QBC work. Training a probabilistic classifier involves estimating the values of model parameters which determine a probability estimate for each possible classification of an example. While we expect that a good classifier would, in most cases, assign the highest probability to the correct class, this is not guaranteed to occur for all training examples. Accordingly, the notion of a consistent hypothesis is generally not applicable to probabilistic classifiers. Thus, the posterior distribution over classifiers given the training data cannot be defined as the restriction of the prior to the set of consistent hypotheses. Rather, within a Bayesian framework, the posterior distribution is defined by the statistics of the training set, assigning higher probability to those classifiers which are more likely given the statistics.

We now consider the desired properties of examples that are selected for training. Generally speaking, a training example contributes data to several statistics, which in turn determine the estimates of several parameter values. An informative example is therefore one whose contribution to the statistics leads to a useful improvement of parameter estimates. We identify three properties of parameters for which acquiring additional statistics is most beneficial:

1. The current estimate of the parameter is uncertain due to insufficient statistics in the training set. An uncertain estimate is likely to be far from the true value of the parameter and can cause incorrect classification. Additional statistics would bring the estimate closer to the true value.
2. Classification is sensitive to changes in the current estimate of the parameter. Otherwise, acquiring additional statistics is unlikely to affect classification and is therefore not beneficial.
3. The parameter takes part in calculating class probabilities for a large proportion of examples. Parameters that are only relevant for classifying few examples, as determined by the probability distribution of the input examples, have low utility for future estimation.

The committee-based sampling scheme, as we describe further below, tends to select examples that affect parameters with the above three properties. Property 1 is addressed by randomly picking parameter values for committee members from the posterior distribution of parameter estimates (given the current statistics). When the statistics for a parameter are insufficient the variance of the posterior distribution of the estimates is large, and hence there will be large differences in the values of the parameter picked for different committee members. Note that property 1 is not addressed when uncertainty in classification is only judged relative to a *single* model (as in, eg, (Lewis and Gale, 1994)). Such an approach captures uncertainty with respect to given parameter values, in the sense of property 2, but it does not model uncertainty about the choice of these values in the first place (the use of a single model is also criticized in (Cohn, Atlas, and Ladner, 1994)).

Property 2 is addressed by selecting examples for which committee members highly disagree in classification. Thus, the algorithm tends to acquire statistics where uncertainty in parameter estimates entails uncertainty in actual classification (this is analogous to splitting the version space in QBC). Finally, property 3 is addressed by independently examining input examples which are drawn from the input distribution. In this way, we implicitly model the expected utility of the statistics in classifying future examples. Such modeling is absent in 'batch' selection schemes, where examples with maximal classification uncertainty are selected from a large batch of examples (see also (Freund et al., 1993) for further discussion).

## 3 COMMITTEE-BASED SAMPLING

Let $X$ denote a set of instances that we want to classify, and let $C$ denote a set of possible classes. For each $x \in X$ we wish to determine its true class $c \in C$. For example, $C$ may be a set of people we know and $X$ may be the (infinite) set of all pictures of the faces of the people in $X$. In the probabilistic classification scheme, classification by a classifier $M$ is performed on the basis of a conditional probability function $P_M(c|x)$, which assigns a probability to every class $c$ given an instance $x$. The classifier then assigns to $x$ the class $c$ which maximizes $P_M(c|x)$.

The function $P_M$ (hence the classifier $M$) is defined by a set of parameters, $\{\alpha_i\}$. During training, the parameters are estimated from a set of statistics, $S$, extracted from a training set of labeled examples. For committee-based sampling, we assume that it is possible to estimate the posterior probability distributions of the parameter values given the training data, $P(\alpha_i = a_i|S)$. This can usually be done through the application of Bayes' theorem, assuming some prior distribution on the values of the parameters.

The selection algorithm operates as follows. First, the set of statistics $S$ is initialized from a small set of labeled training examples. Thereafter, the program examines a stream of unlabeled examples, and chooses which examples it wants labeled for training. If an example gets labeled, its statistics are incorporated into $S$, and the next example is examined.

The core of the algorithm is the procedure for deciding when an example should be labeled for training. This decision is made as follows:

1. Draw $k$ models randomly to serve as committee members, by drawing their parameters from the posterior distributions $P(\alpha_i = a_i|S)$. $S$ are the statistics acquired from examples that were labeled so far.
2. Classify the input example by each committee member and measure the degree of disagreement in their classifications. In this work, we quantify the degree of disagreement as the entropy of the committee classifications (as described below). Other measures for disagreement might also be used.
3. Make a biased random decision whether or not to select the example for labeling, based on the degree of committee disagreement on the example.

The bias is such that higher disagreement entails a higher probability for selecting the example.

This algorithm tends to select examples whose classification is uncertain according to the current statistics. We assume that model parameters are independent given training data. This is true for the HMMs which we use in this work, and is a reasonable assumption in many other cases.

## 4 HMMS AND PART-OF-SPEECH TAGGING

A first-order Hidden Markov Model (HMM) is a probabilistic finite-state string generator (Rabiner, 1989), defined as a set of states $Q = \{q_i\}$, a set of output symbols $\Sigma$, a set of *transition probabilities* $P(q_i \rightarrow q_j)$ of each transition between states $q_i$ and $q_j$, a set of *output probabilities* $P(o|q)$ for each state $q$ to output each symbol $o \in \Sigma$, and a distinguished *start state* $q_0$. The probability of a string $s = a_1 a_2 \cdots a_n$ being generated by an HMM is given by

$$\sum_{q_1 \cdots q_n \in Q^n} \left( \prod_{i=1}^{n} P(q_{i-1} \rightarrow q_i) P(a_i|q_i) \right) \quad ,$$

the sum, for each path through the HMM, of the probability that the path was traversed and that it output the string. In contrast with ordinary Markov Models, in an HMM it is not known which sequence of states generated a given string (hence the term 'hidden').

HMMs have been used widely in speech and language processing. In particular, an HMM can be used to provide a classification model for characters in strings: If we need to classify each character in a string, we encode each possible class by a state in an HMM. Training the HMM amounts to estimating the values of the transition and output probabilities. Then, given a string for classification, we assume that it was generated by the HMM and compute the most likely state sequence for the string, using the Viterbi algorithm[1] (Viterbi, 1967).

An HMM can be used for part-of-speech tagging by encoding each possible part-of-speech tag, $t$ (noun, verb, adjective, etc.), as an HMM state. The output probabilities, $P(w|t)$, give the probability of producing each word $w$ in the language conditioned on the current tag $t$. The transition probabilities, $P(t_1 \rightarrow t_2)$, give the probability of generating the tag $t_2$ given that the previous tag is $t_1$, constituting a weak syntactic model of the language. This model is often termed the *tag-bigram model.*

Given an input sentence $W = w_1 \cdots w_n$, the most likely tag sequence $T = t_1 \cdots t_n$ is the one which maximizes

$$P(T, W) = \prod_{i=1}^{n} P(t_{i-1} \rightarrow t_i) P(w_i|t_i)$$

For technical convenience, we use Bayes' theorem to replace each $P(w_i|t_i)$ term by the term $\frac{P(t_i|w_i)P(w_i)}{P(t_i)}$, noting that $P(w_i)$ does not effect the maximization over tag sequences and can therefore be omitted (following Church (1988)). The parameters of a part-of-speech model, then, are: *tag probabilities* $P(t_i)$, *transition probabilities* $P(t_{i-1} \rightarrow t_i)$, and *lexical probabilities* $P(t|w)$.

Supervised training of the tagger is performed using a tagged corpus (text collection), which was manually labeled with the correct part-of-speech for each word. Maximum likelihood estimates (MLEs) for the parameters are easily computed from word and tag counts from the corpus. For example, the MLE of $P(t)$ is the fraction of tag occurrences in the corpus that were the tag $t$, whereas $P(t|w)$ is the ratio between the count for the word $w$ being labeled with the tag $t$ and the total count for $w$. In our committee-based sampling scheme, these counts are used also to compute the posterior distributions for parameter estimates, as discussed below in Section 5.1.

It should be noted that practical implementations of part-of-speech tagging typically employ a tag-trigram model, in which the probability of a tag depends on the last two tags rather than just the last one. The committee-based sampling method, which we apply here to the bigram model, is equally applicable to the trigram case.

## 5 COMMITTEE-BASED SAMPLING FOR HMMS

In this section we describe the application of our committee-based sampling scheme to the HMM classification framework. First we will discuss how to compute the posterior distributions over the HMM parameters $P(t_i \rightarrow t_j)$ and $P(t|w)$, given training statistics.[2] We then discuss the question of how to define an example for training—an HMM deals with (in principle) infinite strings; on what substrings do we make decisions about labeling? Finally, we describe how to measure the amount of disagreement between committee members.

[1] An alternative classification scheme is to compute the most likely state for each individual character (instead of the most likely state sequence) by the Forward-Backward algorithm (Rabiner, 1989). We do not address here this alternative, which is computationally more expensive and is typically not used for part-of-speech tagging. It is easy, however, to apply the committee-based sampling method for this type of classification.

[2] We do not sample the model space over the tag probability parameters, since the amount of data for tag frequencies is large enough to make their MLEs quite definite.

### 5.1 POSTERIOR DISTRIBUTIONS FOR MULTINOMIAL PARAMETERS

In this section, we consider how to approximate the posterior parameter distributions $P(\alpha_i = a_i|S)$ for an HMM, assuming a non-informative (uniform) prior. First note that the parameters of an HMM define a set of multinomial probability distributions. Each multinomial corresponds to a conditioning event and its values are given by the corresponding set of conditioned events. For example, a transition probability parameter $P(t_i \rightarrow t_j)$ has conditioning event $t_i$ and conditioned event $t_j$.

Let $\{u_i\}$ denote the set of possible values of a given multinomial variable, and let $S = \{n_i\}$ denote a set of statistics extracted from the training set, where $n_i$ is the number of times that the value $u_i$ appears in the training set. We denote the total number of appearances of the multinomial variable as $N = \sum_i n_i$. The parameters whose distributions we wish to estimate are $\alpha_i = P(u_i)$.

The maximum likelihood estimate for each of the multinomial's distribution parameters, $\alpha_i$, is $\hat{\alpha}_i = \frac{n_i}{N}$. In practice, this estimator is usually smoothed in some way to compensate for data sparseness. Such smoothing typically reduces the estimates for values with positive counts and gives small positive estimates for values with a zero count. For simplicity, we describe here the approximation of $P(\alpha_i = a_i|S)$ for the unsmoothed estimator[3].

We approximate $P(\alpha_i = a_i|S)$ by first assuming that the multinomial is a collection of independent binomials, each of which corresponds to a single value $u_i$ of the multinomial; we then separately apply the constraint that the parameters of all these binomials should sum to 1. For each such binomial, we approximate $P(\alpha_i = a_i|S)$ as a truncated normal distribution (restricted to [0,1]), with estimated mean $\mu = \frac{n_i}{N}$ and variance $\sigma^2 = \frac{\mu(1-\mu)}{N}$.[4] We found in practice, however, very small differences between parameter values drawn from this distribution, and consequently too few disagreements between committee members to be useful for sampling. We therefore also incorporate a 'temperature' parameter, $t$, which is used as a multiplier for the variance estimate $\sigma^2$. In other words, we actually approximate $P(\alpha_i = a_i|S)$ as a truncated normal distribution with mean $\mu$ and variance $\sigma^2 t$.

To generate a particular multinomial distribution, we randomly choose values for its parameters $\alpha_i$ from their binomial distributions, and renormalize them so that they sum to 1.

To generate a random HMM given statistics $S$, we note that all of its parameters $P(t_i \rightarrow t_j)$ and $P(t|w)$ are independent of each other. We thus independently choose values for the HMM's parameters from their multinomial distributions.

### 5.2 EXAMPLES IN HMM TRAINING

Typically, concept learning problems are formulated such that there is a set of training examples that are independent of each other. When training an HMM, however, each state/output pair is dependent on the previous state, so we are presented (in principle) with a single infinite input string for training. In order to perform selective sampling, we must divide this infinite string into (short) finite strings.

For part-of-speech tagging, this problem may be solved by considering each sentence as an individual example. More generally, we can break the text at any point where tagging is unambiguous. In particular, it is common to have a *lexicon* which specifies which parts-of-speech are possible for each word (i.e, which of the parameters $P(t|w)$ are positive). In bigram tagging, we can use unambiguous words (those with only one possible part of speech) as example boundaries. Similar natural breakpoints occur in other HMM applications; for example, in speech recognition we can consider different utterances separately. In other cases of HMM learning, where such natural breakpoints do not occur, some heuristic will have to be applied, preferring to break at 'almost unambiguous' points in the input.

### 5.3 QUANTIFYING DISAGREEMENT

Recall that our algorithm decides whether or not to select an example based on how much the committee members disagree on its labeling. For tagging, an example is a word sequence, where each word is labeled with a tag by each committee member.

We first quantify committee disagreement for an individual word by using the entropy of tag distribution of tags assigned by committee members to that word. Let $V(t,w)$ be the number of committee members (out of $k$ members) 'voting' for tag $t$ for the word $w$. Then $w$'s *vote entropy* is

$$VE(w) = -\sum_t \frac{V(t,w)}{k} \log \frac{V(t,w)}{k}$$

We found empirically that the average vote entropy for words which the tagger (after some training) classified correctly was 0.25, whereas the average entropy for incorrectly classified words was 0.66. This demonstrates

[3] In the implementation we smooth the MLE by interpolation with a uniform probability distribution, following Merialdo (1991). Adaptation of $P(\alpha_i = a_i|S)$ to the smoothed version of the estimator is simple.

[4] As noted by one of the anonymous reviewers, the normal approximation can be avoided. The posterior probability $P(\alpha_i = a_i|S)$ for the multinomial is given exactly by the Dirichlet distribution (Johnson, 1972) (which reduces to the Beta distribution in the binomial case).

that vote entropy is a useful measure of classification uncertainty (likelihood of error) based on the training data.

To measure disagreement over an entire word sequence $W$, we use the average, $\overline{VE}(W)$, of the voting entropy over all ambiguous words in the sequence. The probability of accepting the word sequence $W$ for labeling is then defined by

$$P_{\text{label}}(W) = \frac{e}{\log k}\overline{VE}(W)$$

where $e$ is an *entropy gain* system parameter, which controls the overall frequency with which examples are selected, and $1/\log k$ normalizes the entropy to between 0 and 1 (since the maximum entropy possible is $\log k$).

### 5.4 COMMITTEE-BASED SAMPLING AND MONTE-CARLO ESTIMATION

We can also view committee-based selection as a Monte-Carlo method for estimating the probability distribution of classes assigned to an example, over all possible models, given previous training data. The proportion of votes among committee members for a class $c$ on an example $x$ approximates the probability, for a model chosen randomly from the posterior model distribution, of assigning $c$ to $x$. That is, the committee-based method is a Monte-Carlo estimate of

$$P^*(c|x) = \int_{\mathcal{M}} T_M(c|x)P(M|S)dM$$

where $M$ ranges over possible models (vectors of parameter values) in the model space $\mathcal{M}$, $P(M|S)$ is the posterior probability density of model $M$ given statistics $S$, and $T_M(c|x) = 1$ if $c$ is the highest probability class for $x$ in $M$ (i.e, $c = \arg\max_{c_i} P_M(c_i|x)$), and 0 otherwise. Vote entropy as discussed above, then, is an approximation of the entropy of $P^*$, which is a direct measure of uncertainty in example classification over the possible models.

Note that we measure entropy over the *final classes* assigned to an example by possible models (i.e, $T_M$), not over the *class probabilities* given by possible models (i.e, $P_M$). Measuring entropy over $P_M$ (say, by looking at the expected probability over all models) would not properly address properties 1 and 2 discussed in Section 2.2 above.

## 6 EXPERIMENTAL RESULTS

In this section, we describe the results of applying the committee-based sampling method to bigram part-of-speech tagging, as compared with standard sequential sampling. Evaluation was performed using the University of Pennsylvania tagged corpus from the ACL/DCI CD-ROM I. For ease of implementation, we used a complete (closed) lexicon which contains all the words in the corpus[5]. Approximately 63% of the tokens (word occurences) in the corpus were ambiguous in the lexicon.

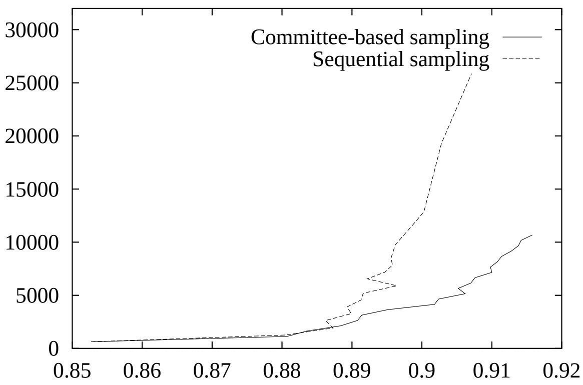


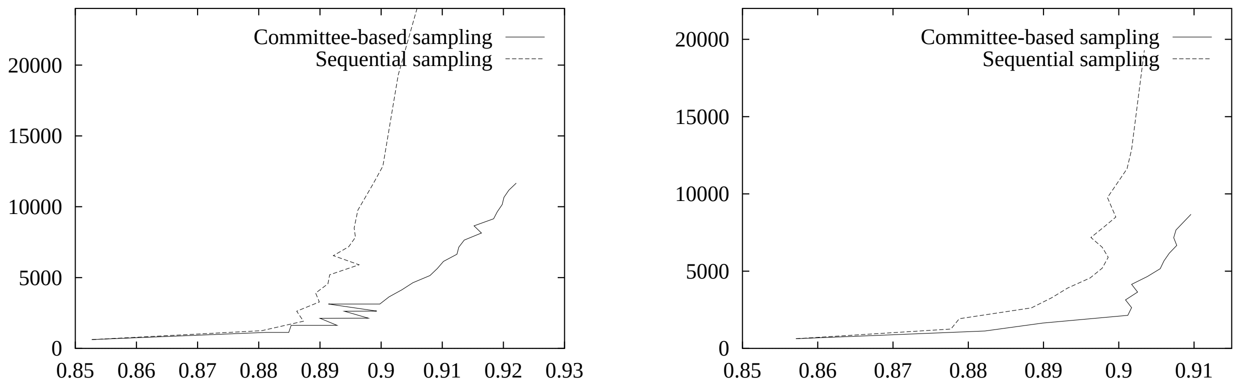
Figure 1: Amount of training (number of ambiguous words in the training sentences) plotted ($y$-axis) versus classification accuracy ($x$-axis). The committee-based sampling used $k = 10$ committee members, entropy gain $e = 0.3$, and temperature $t = 50$.

For evaluation, we compared the learning efficiency of our committee-based selection algorithm to that of sequential selection. In sequential selection, training examples are selected sequentially from the corpus. We make the common assumption that the corpus is a random stream of example sentences drawn from the distribution of the language.

The committee-based sampling algorithm was initialized using the first 1,000 words from the corpus (624 of which were unambiguous), and then sequentially examined the following examples in the corpus for possible labeling. Testing was performed on a separate portion of the corpus consisting of 20,000 words. We compare the amount of training required by the different methods to achieve a given tagging accuracy on the test set, where both the amount of training and tagging accuracy are measured only over ambiguous words.

In Figure 1, we present a plot of training effort versus accuracy achieved, for both sequential sampling and committee-based sampling ($k = 10$, $e = 0.3$, and $t = 50$). The curves start together, but the efficiency of committee-based selection begins to be evident when we seek 89% accuracy. Committee-based selection requires less than one-fourth the amount of training that sequential selection does to reach 90.5% accuracy. Fur-

[5] We used the lexicon provided with Brill's part-of-speech tagger (Brill, 1992). While in an actual application the lexicon would not be complete, our results using a complete lexicon are still valid, since the evaluation is comparative.



(a) (b)

Figure 2: Further results comparing committee-based and sequential selection. Amount of training (number of ambiguous words in the training sentences) plotted against desired accuracy. (a) Training vs. accuracy for $k = 5$, $e = 0.3$, and $t = 50$. (b) Training vs. accuracy for $k = 10$, $e = 0.5$ and $t = 50$, for a second test set of 20,000 words.

thermore, the efficiency improvement resulting from using committee-based sampling greatly increases with the desired accuracy. This is in line with the results of Freund et al. (1993) on Query By Committee sampling, in which they prove exponential speedup under certain theoretical assumptions.

Figure 2 demonstrates that our results are qualitatively the same for different values of the system parameters. Figure 2(a) shows a plot comparable to Figure 1, for sampling using 5 committee members. Results are substantially the same, though the speedup is slightly less and there is more oscillation in accuracy at low training levels, due to the greater coarseness of the evaluation of information content. In Figure 2(b) we show a similar plot on a different test set of 20,000 words, for sampling using 10 committee members and an entropy gain of 0.5. Again, we see a similar efficiency gain for committee-based sampling as compared with sequential sampling.

# 7 CONCLUSIONS

We have presented a general method for selective sampling to improve efficiency of training probabilistic classifiers. The method is applicable whenever it is possible to estimate a posterior distribution over the model space given the training data. We have also shown how to apply the method to training Hidden Markov Models. The method was implemented and tested for the complex task of part-of-speech tagging of natural language sentences, and produced a substantial improvement in training efficiency.

The committee-based sampling method addresses the three factors which relate the informativeness of a training example to the model parameters that it affects. These factors are: (1) the statistical significance of the parameter's estimate, (2) the parameter's effect on classification, and (3) the probability that the parameter will be used for classification in the future. The use of a committee models the uncertainty in classification relative to the entire model space, while randomized selection implicitly models the distribution of the examples. This is in contrast to previous empirical work in selective sampling for probabilistic classifiers (Lewis and Catlett, 1994), which uses a single model and a batch selection scheme.

We find the committee-based sampling method particularly attractive because it measures the expected information gain from a training example in an implicit manner. Explicit calculation of the expected information gain, on the other hand, is complex and is specific to the particular type of classification model used (MacKay, 1992b; MacKay, 1992a; Cohn, Ghahramani, and Jordan, 1995). It is unclear to us whether such explicit estimation can even be performed for complex classification models such as HMMs. Furthermore, the generality obtained from implicitly modeling information gain suggests using committee-based sampling also in non-probabilistic contexts, where explicit modeling of information gain may be impossible. In such contexts, committee members might be generated by randomly varying some of the decisions made in the learning algorithm. In the context of learning HMMs, this approach could be profitably applied to learning HMM structure (Stolcke and Omohundro, 1992), in addition to estimating HMM parameters.

### Acknowledgements

Discussions with Yoav Freund and Yishai Mansour greatly enhanced this work. The second author gratefully acknowledges the support of the Fulbright Foundation. We also thank the anonymous reviewers for their helpful comments.

## References

Angluin, Dana. 1987. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, November.

Angluin, Dana. 1988. Queries and concept learning. *Machine Learning*, 2:319–342.

Black, Ezra, Fred Jelinek, John Lafferty, David Magerman, Robert Mercer, and Salim Roukos. 1993. Towards history-based grammars: using richer models for probabilistic parsing. In *Proc. of the Annual Meeting of the ACL.*

Brill, Eric. 1992. A simple rule-based part of speech tagger. In *Proc. of ACL Conference on Applied Natural Language Processing.*

Church, Kenneth W. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proc. of ACL Conference on Applied Natural Language Processing.*

Cohn, David, Les Atlas, and Richard Ladner. 1994. Improving generalization with active learning. *Machine Learning*, 15.

Cohn, David A., Zoubin Ghahramani, and Michael I. Jordan. 1995. Active learning with statistical models. In G. Tesauro, D. Touretzky, and J. Alspector, editors, *Advances in Neural Information Processing*, volume 7. Morgan Kaufmann.

Freund, Y., H. S. Seung, E. Shamir, and N. Tishby. 1993. Information, prediction, and query by committee. In S. Hanson et al., editor, *Advances in Neural Information Processing*, volume 5. Morgan Kaufmann.

Freund, Yoav. 1994. Sifting informative examples from a random source. In *Working Notes of the Workshop on Relevance, AAAI Fall Symposium Series*, pages 85–89.

Gale, William, Kenneth Church, and David Yarowsky. 1993. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439.

Johnson, Norman L. 1972. *Continuous Multivariate Distributions.* John Wiley & Sons, New York.

Lewis, D. and J. Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings of the 11th International Conference.*

Lewis, D. and W. Gale. 1994. Training text classifiers by uncertainty sampling. In *Proceedings of ACM-SIGIR Conference on Information Retrieval.*

MacKay, David J. C. 1992a. The evidence framework applied to classification networks. *Neural Computation*, 4.

MacKay, David J. C. 1992b. Information-based objective functions for active data selection. *Neural Computation*, 4.

Merialdo, Bernard. 1991. Tagging text with a probabilistic model. In *Proc. Int'l Conf. on Acoustics, Speech, and Signal Processing.*

Plutowski, Mark and Halbert White. 1993. Selecting concise training sets from clean data. *IEEE Trans. on Neural Networks*, 4(2).

Rabiner, Lawrence R. 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2).

Seung, H. S., M. Opper, and H. Sompolinsky. 1992. Query by committee. In *Proc. ACM Workshop on Computational Learning Theory.*

Stolcke, A. and S. Omohundro. 1992. Hidden Markov Model induction by Bayesian model merging. In *Advances in Neural Information Processing*, volume 5. Morgan Kaufmann.

Viterbi, A. J. 1967. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Informat. Theory*, T-13.