Available online at www.sciencedirect.com

SCIENCE DIRECT®

Speech Communication 45 (2005) 171–186

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Combining active and semi-supervised learning for spoken language understanding

Gokhan Tur [a,*], Dilek Hakkani-Tür [a], Robert E. Schapire [b]

[a] *AT&T Labs—Research, Florham Park, NJ 07932, USA*

[b] *Department of Computer Science, Princeton University, Princeton, NJ 08544, USA*

Received 2 March 2004; received in revised form 27 July 2004; accepted 30 August 2004

Abstract

In this paper, we describe active and semi-supervised learning methods for reducing the labeling effort for spoken language understanding. In a goal-oriented call routing system, understanding the intent of the user can be framed as a classification problem. State of the art statistical classification systems are trained using a large number of human-labeled utterances, preparation of which is labor intensive and time consuming. Active learning aims to minimize the number of labeled utterances by automatically selecting the utterances that are likely to be most informative for labeling. The method for active learning we propose, inspired by certainty-based active learning, selects the examples that the classifier is the least confident about. The examples that are classified with higher confidence scores (hence not selected by active learning) are exploited using two semi-supervised learning methods. The first method augments the training data by using the machine-labeled classes for the unlabeled utterances. The second method instead augments the classification model trained using the human-labeled utterances with the machine-labeled ones in a weighted manner. We then combine active and semi-supervised learning using selectively sampled and automatically labeled data. This enables us to exploit all collected data and alleviates the data imbalance problem caused by employing only active or semi-supervised learning. We have evaluated these active and semi-supervised learning methods with a call classification system used for AT&T customer care. Our results indicate that it is possible to reduce human labeling effort significantly.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Active learning; Semi-supervised learning; Spoken language understanding; Call classification

[*] Corresponding author. Tel.: +1 973 360 5710.

E-mail addresses: gtur@research.att.com (G. Tur), dtur@research.att.com (D. Hakkani-Tür), schapire@cs.princeton.edu (R.E. Schapire).

0167-6393/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2004.08.002

1. Introduction

Spoken dialog systems aim to identify the intended meaning of human utterances, expressed in natural language, and to take actions accordingly to satisfy their request (Gorin et al., 2002). Typically, in a natural spoken dialog system, the speaker's utterance is first recognized using an automatic speech recognizer (ASR). Then, the intent of the speaker is identified from the recognized sequence using a spoken language understanding (SLU) component. This step can be framed as a classification problem for goal-oriented call routing systems (Tur et al., 2002; Natarajan et al., 2002; Kuo et al., 2002). In this study, we have used a Boosting-style classification algorithm for call classification (Schapire, 2001). As a call classification example, consider the utterance *I would like to know my account balance* in a customer care application. Assuming that the utterance is recognized correctly, the corresponding intent or call-type would be *Request(Account_Balance)*, and the action would be learning the account number and prompting the balance to the user, or routing this call to the Billing Department.

When statistical classifiers are used in such systems (e.g., Schapire and Singer, 2000; Haffner et al., 2003), they are trained using large amounts of task data which is usually transcribed and then labeled by humans, a very expensive and laborious process. By "labeling", we mean assigning one or more of the predefined call-types to each utterance. It is clear that the bottleneck in building an accurate statistical system is the time spent for high quality labeled data. Building better call classification systems in a shorter time frame motivates us to develop novel techniques. To this end, we employ active and semi-supervised learning algorithms.

Typically, the examples to be labeled are chosen randomly so that the training data matches the test set. Thus, the motto of any statistical learning algorithm is, "There's no data like more data." In the machine learning literature, learning from randomly selected examples is called *passive learning*. Recently, a new set of learning algorithms in which the learner actively selects the examples to be labeled has been proposed; this approach is called *active learning* (Cohn et al., 1994). Active learning aims at reducing the number of training examples to be labeled by selectively sampling a subset of the unlabeled data. This is done by inspecting the unlabeled examples and selecting the most *informative* ones, with respect to a given cost function, for a human to label (Cohn et al., 1994). In other words, the goal of the active learning algorithm is to select the examples which will have the largest improvement on performance, hence reducing the amount of human labeling effort.

If there are more candidate utterances than can be labeled manually with the available human resources, one can ask the labelers to label the selectively sampled utterances. This is indeed the case in a deployed natural dialog system where a constant stream of raw data is collected from the field to continuously improve the performance of the system. The aim of active learning then is to come up with a smaller subset of all utterances collected from the field for human labeling. In our work, the main criterion in selective sampling is the classifier confidence scores attached to utterances. The intuition is that there is a reverse correlation between the confidence score given by the classifier and the informativeness of that utterance. That is, the higher the classifier's confidence score for an utterance, the less informative that utterance.

We also focus on the complementary problem: how can we exploit the remaining set of utterances which are not labeled by a human? This is another crucial component of this study as part of our goal of training classification models with higher performance in a shorter time frame. Knowing that the remaining utterances are classified with confidence scores more than some threshold, we can exploit the unlabeled examples by adding them to the smaller set of labeled data in a weighted manner to improve the performance of the classifier.

In this paper, we present two semi-supervised learning methods for combining labeled and unlabeled utterances so as to speed up the building of accurate call-type classification systems. The first method simply adds the machine-labeled utterances to the training data. The second method is specific to the Boosting algorithms that we use, and augments the classification model trained using the human-labeled utterances with the machine-labeled ones in a weighted manner.

In the following section, we present briefly the active and semi-supervised learning literature, as well as review some of the related work in language processing. In Section 3, we briefly explain the Boosting algorithm. In Sections 4 and 5, we describe our active and semi-supervised learning algorithms, and Section 6 contains methods to combine these two complementary learning algorithms. Then in Section 7 we present our experiments, results, and a discussion of possible future work.

2. Related work

The search for effective training data sampling algorithms has been studied under the title of active learning. In order to have better systems with less annotated data, the learner is given some control over the inputs on which it trains.

Previous work in active learning has concentrated on two approaches: certainty-based methods and committee-based methods. In *certainty-based methods*, an initial system is trained using a small set of annotated examples. Then, the system examines and labels the unannotated examples and determines the "certainty" or "confidence" of each of its predictions. The examples with the lowest certainty levels are then presented to the labelers for annotation. Cohn et al. (1994) introduced certainty-based active learning for classification based on earlier work on artificial membership queries (Angluin, 1988). Lewis and Catlett (1994) used selective sampling with decision trees for text categorization and obtained a 10-fold reduction in the amount of labeled data needed. Active learning has also been applied to support-vector machines (Schohn and Cohn, 2000; Tong and Koller, 2001), as well as Boosting and Bagging (Abe and Mamitsuka, 1998). In the language processing framework, certainty-based methods have been used for natural language parsing and information extraction (Thompson et al., 1999; Tang et al., 2002) and word segmentation (Sassano, 2002). In our previous work, we presented certainty-based active learning approaches for reducing the amount of labeled data needed for spoken language understanding (Tur et al., 2003); these techniques were also applied to automatic speech recognition by Hakkani-Tür et al. (2002).

In *committee-based methods*, a distinct set of classifiers is created using the small set of annotated examples. The unannotated instances whose classifications differ most when presented to different classifiers are given to the labelers for annotation. Seung et al. (1992) introduced this approach calling it *query by committee*. Freund et al. (1997) provided an extensive analysis of this algorithm, especially for learning of perceptrons. Liere and Tadepalli (1997) employed committee-based active learning for text categorization and obtained 2–30-fold reductions in the amount of human-labeled data required, depending on the size of labeled dataset. Argamon-Engelson and Dagan (1999) formalized this algorithm for probabilistic classifiers introducing a metric called *vote entropy* to compute the disagreement of the committee members. They demonstrated its use for the task of part-of-speech tagging. One drawback with certainty-based methods is that it is hard to distinguish an informative example from an outlier. On the other hand, in order to train multiple classifiers required by committee-based active learning, one may need to divide the training data or feature set into multiple parts, and this may result in many weak classifiers instead of a single strong one.

Recently, semi-supervised learning algorithms that use both labeled and unlabeled data have been used for text classification in order to reduce the need for labeled training data. Blum and Mitchell (1998) proposed an approach called co-training. For using co-training, the features in the problem domain should naturally divide into two sets. Then the examples which are classified with high confidence scores with one view can be used as the training data of other views. For example, for web page classification, one view can be the text in them and another view can be the text in the hyperlinks pointing to those web pages. For the same task, Nigam et al. (2000) used an algorithm for learning from labeled and unlabeled documents based on the combination of Expectation Maximization (EM) and a Naive Bayes classifier. Nigam and Ghani (2000) then combined co-training and EM, coming up with the Co-EM algorithm which is the probabilistic version of co-training. Ghani (2002) later combined the Co-EM algorithm with error correcting output coding

(ECOC) to exploit the unlabeled data, in addition to the labeled data. For natural language call routing, Iyer et al. (2002) proposed using speech recognizer output instead of transcribing the utterances during training, without losing accuracy. However, hand-labeling of the utterances with the correct call-type is not mentioned. For spoken language understanding, Tur and Hakkani-Tür (2003) presented semi-supervised learning approaches for exploiting unlabeled data.

The idea of combining active and semi-supervised learning was first introduced by McCallum and Nigam (1998). They combined the committee-based active learning algorithm with EM-style semi-supervised learning to assign class labels to those examples that remain unlabeled. For the task of text categorization, using a Bayesian classifier, they employed EM for each of the committee members. Muslea et al. (2002) extended this idea to use Co-EM as the semi-supervised learning algorithm. They called this new algorithm Co-EMT. Exploiting multiple views for both active and semi-supervised learning has been shown to be very effective. Once the committee members are formed for active learning, it is straightforward to employ co-training or its variant Co-EM for semi-supervised learning. Riccardi and Hakkani-Tür (2003) used a combination of active and semi-supervised learning for automatic speech recognition and have shown improvements for statistical language modeling. In that work, they mainly exploited confidence scores for words and utterances computed from ASR word lattices.

Combining the data with prior task knowledge (e.g., rules) is also considered in the literature for building natural language dialog systems in a shorter time frame. Schapire et al. (2002) have extended Boosting so as to handle initial hand-written rules during classification. When there is little labeled data, this approach is shown to be very effective.

3. Boosting

We begin with a review of Boosting-style algorithms. Boosting aims to combine "weak" base classifiers to come up with a "strong" classifier (Freund and Schapire, 1997, 2001). Boosting is an iterative procedure; on each iteration, a weak classifier is trained on a weighted training set, and at the end, the weak classifiers are combined into a single, combined classifier.

The algorithm generalized for multi-class and multi-label classification is given in Fig. 1. This is a variant of Freund and Schapire (1997)'s AdaBoost algorithm called AdaBoost.MH, and is due to Schapire and Singer (2000). Let $\mathscr{X}$ denote the domain of possible training examples and let $\mathscr{Y}$ be a finite set of classes of size $|\mathscr{Y}| = k$. For $Y \subseteq \mathscr{Y}$, let $Y[l]$ for $l \in \mathscr{Y}$ be

$$Y[l] = \begin{cases} +1 & \text{if } l \in Y, \\ -1 & \text{otherwise.} \end{cases}$$

The algorithm begins by initializing a uniform distribution $D_1(i,l)$ over training examples i and labels l. After each round this distribution is updated so that the example-class combinations which are easier to classify get lower weights and vice versa. The intended effect is to force the weak learning algorithm to concentrate on the examples and labels that will be the most beneficial to the overall goal of finding a highly accurate classification rule.

Instead of just a raw real-valued classification score, it is possible to estimate the probability of a particular class using a logistic function (Friedman et al., 2000):

$$\Pr(Y[l] = +1 \mid x) = \frac{1}{1 + \mathrm{e}^{-2f(x,l)}}.$$

This algorithm can be seen as a procedure for finding a linear combination of base classifiers which attempts to minimize an exponential loss function (Schapire and Singer, 1999), which in this case is

$$\sum_i \sum_l \mathrm{e}^{-Y_i[l]f(x_i,l)}.$$

An alternative would be to minimize a logistic loss function as suggested by Friedman et al. (2000), namely

$$\sum_i \sum_l \ln(1 + \mathrm{e}^{-Y_i[l]f(x_i,l)}).$$

A modified version of AdaBoost for this loss function is given by Collins et al. (2002). In this case, the logistic function used to obtain probability estimates can be computed as:

```
• Given training data from the instance space
  $S = \{(x_1, Y_1), ..., (x_m, Y_m)\}$ where $x_i \in \mathcal{X}$ and $Y_i \subseteq \mathcal{Y}$.
• Initialize the distribution $D_1(i, l) = 1/mk$.
• For each iteration $t = 1, ..., T$ do
  · Train a base learner $h_t$ using distribution $D_t$.
  · Update

      $D_{t+1}(i, l) = \frac{D_t(i, l)e^{-\alpha_t Y_i[l] h_t(x_i, l)}}{Z_t}$

    where $Z_t$ is a normalization factor and $\alpha_t$ is the weight of
    the base learner.
• Output the final classifier defined as:

    $H(x, l) = \text{sign}(f(x, l))$

  where

    $f(x, l) = \sum_{t=1}^{T} \alpha_t h_t(x, l).$
```

Fig. 1. The algorithm *Adaboost.MH*.

$$\Pr(Y[l] = +1 \mid x) = \frac{1}{1 + \mathrm{e}^{-f(x,l)}}.$$

A more detailed explanation and analysis of this algorithm can be found in (Schapire, 2001). In our experiments, we used the BoosTexter tool, which is an implementation of the Boosting algorithm (Schapire and Singer, 2000). For text categorization, BoosTexter uses word *n*-grams as features, and each weak classifier (or "decision stump") checks the absence or presence of a feature.

4. Active learning: Selection of data to label

Inspired by the certainty-based active learning methods, we select the examples that we predict the classifier is the least confident about for labeling, leaving out the ones that have been classified with high confidence scores.

The certainty-based active learning algorithm is described in Fig. 2 and depicted in Fig. 3. We first train a classifier using a small set of labeled data, S_t. Using this classifier, we classify the utterances that are candidates for labeling, $S_p = \{s_1, \ldots, s_n\}$. We then use the confidence score for the top scoring call-type $CS(s_i)$ for each utterance $s_i \in S_p$ to predict which candidates are misclassified:

$$CS(s_i) = \max_{c_j} CS_{c_j}(s_i),$$

where $CS_{c_j}(s_i)$ is the confidence score assigned by the classifier to utterance s_i for the call-type c_j:

$$CS_{c_j}(s_i) = \Pr(Y[j] = +1 \mid S_i).$$

We then label the utterances which have the lowest confidence scores:

$$S_k = \{s_i : CS(s_i) < \text{th}\}.$$

This approach is independent of the classifier used. The threshold th is mainly determined by the capacity of the manual labeling effort or by the performance of the current classifier. The other parameter is the filtering criterion. One can come up with a different criterion for selecting the utterances for labeling other than the maximum confidence scores, such as the difference of the top

```
(1) Given some amount of training data $S_t$, and a larger amount
    of unlabeled data in the pool $S_p = \{s_1, ..., s_n\}$
(2) While (labelers/utterances are available) do
    2.1 Train a classifier using the current training data $S_t$
    2.2 Classify the utterances in $S_p$ using this classifier
        computing the corresponding confidence scores
        $CS(s_i), \quad i = 1, ..., n$
    2.3 Manually label the set $S_k = \{s_i : CS(s_i) < th\}$
    2.4 $S_t = S_t \bigcup S_k$
    2.5 $S_p = S_p - S_k$
```

Fig. 2. The certainty-based active learning algorithm.

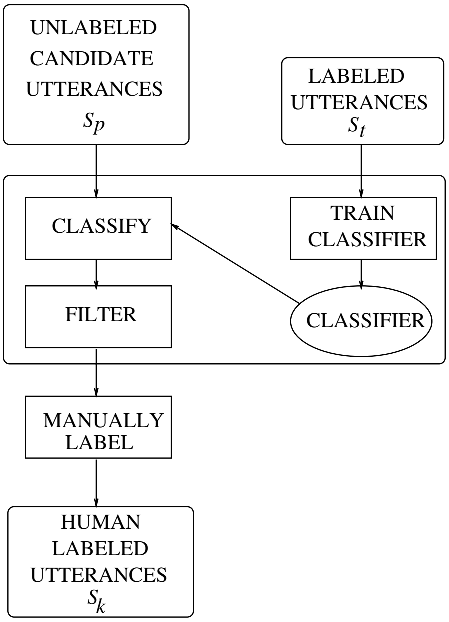

Fig. 3. The certainty-based active learning algorithm.

two call-type confidence scores, or by using other or all the call-type scores.

One other parameter of the algorithm is the characteristics of the pool used in the algorithm. Instead of assuming a fixed pool, one can use a dynamic pool. Assuming a constant stream of incoming data traffic better reflects an actual SLU system. In such a case, the aim of active learning is again to select the most informative utterances from a given pool, as in the above algorithm. The difference is that, at each iteration, a new pool is provided to the algorithm. In the above algorithm, Item [2.5] needs to be updated as follows:

```
2.5 Get new $S_p$
```

Note that the distribution of the call-types in the selectively sampled training data has skewed from its priors. That is, the distribution of call-types has become different in training and test data. The classes which have more examples in the training data or which can be easy to classify are underrepresented by selective sampling. In other words, the classifier trained on selectively sampled data can be biased to infrequent or hard to classify classes. Divergence from the priors is a problem that deteriorates the performance of the classifier, and has already been noted in the literature (Lewis and Catlett, 1994). The solution of Lewis and Catlett is to adjust the classifier parameters (in their case, decision trees) so that false positives are more costly than false rejections. Another solution is to adjust the priors by up-sampling or down-sampling the data. In this work, we propose a novel solution to this problem as described in Section 6.

5. Semi-supervised learning: exploiting the unlabeled data

The aim of semi-supervised learning is to exploit the unlabeled utterances in order to improve

```
(1) Given some amount of human-labeled training data $S_t$, and
    unlabeled data in the pool $S_p = \{s_1, ..., s_n\}$
(2) Train a classifier using the current training data $S_t$
(3) Classify the utterances in $S_p$ using this classifier
    computing the confidence scores, $CS(s_i), \quad i = 1, ..., n$
(4) Let $S_m = \{s_i : CS(s_i) \geq th\}$
(5) $S_t = S_t \bigcup S_m$
(6) Train a classifier using the augmented training data $S_t$
```

Fig. 4. The first semi-supervised learning algorithm: augmenting the data.

the performance of the classifier. To this end, we propose two methods. Both methods assume that there is some amount of training data available for training an initial classifier. The basic idea is to use this classifier to label the unlabeled data automatically, and to then improve the classifier performance using the machine-labeled call-types as the labels of those unlabeled utterances, thus reducing the amount of human-labeling effort necessary to come up with better statistical classifiers.

5.1. Augmenting the data

This simpler semi-supervised learning method is summarized in Fig. 4. First we train an initial model using the human-labeled data, and then classify the unlabeled ones. Then we add the unlabeled utterances directly to the training data using the machine-labeled call-types as seen in Fig. 5. In order to reduce the noise added because of classifier errors, we only add those utterances which are classified with the call-types with confidence scores higher than some threshold, th.

$$S_m = \{s_i : CS(s_i) \geqslant \text{th}\},$$

where

$$CS(s_i) = \max_{c_j} CS_{c_j}(s_i).$$

It is then straightforward to use the call-types exceeding that threshold for each utterance during re-training:

$$Y_i[l] = \begin{cases} +1 & \text{if } CS_{c_l}(s_i) \geqslant \text{th}, \\ -1 & \text{otherwise.} \end{cases}$$

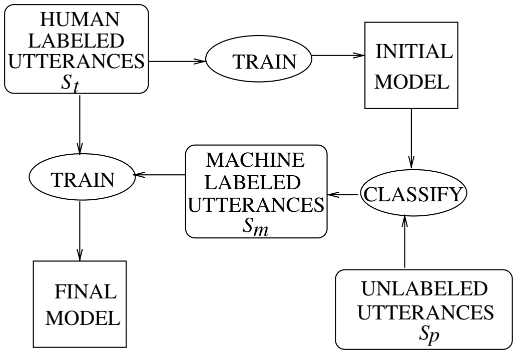
HUMAN LABELED UTTERANCES S_t
TRAIN
INITIAL MODEL
TRAIN
MACHINE LABELED UTTERANCES S_m
CLASSIFY
FINAL MODEL
UNLABELED UTTERANCES Sp

Fig. 5. The first semi-supervised learning method: augmenting the data.

The threshold th can be set using a separate held-out set. Obviously, there is a trade-off in selecting the threshold. If it is set to a lower value, that means a larger amount of noisy data, and if it is set to a higher value, that means less amount of useful or informative data. Finally, all the data, including both human- and machine-labeled utterances, are used for training the classifier.

5.2. Augmenting the classification model

For the second method for semi-supervised learning, we again train a classifier using some amount of labeled data. This method is the same as the previous one until the last step where we instead augment the initial model by machine-labeled examples in a weighted manner as described in Fig. 6. Fig. 7 depicts the process proposed for this method.

```
(1) Given some amount of human-labeled training data $S_t$, and
    unlabeled data in the pool $S_p = \{s_1, ..., s_n\}$
(2) Train an initial classifier using the current training data
    $S_t$
(3) Classify the utterances in $S_p$ using this classifier
    computing the confidence scores $CS(s_i), \quad i = 1, ..., n$
(4) Let $S_m = \{s_i : CS(s_i) \geq th\}$
(5) Augment the classifier by changing the loss function so that
    it fits both the initial model and the new machine-labeled
    data $S_m$
```

Fig. 6. The second semi-supervised learning algorithm: augmenting the model.

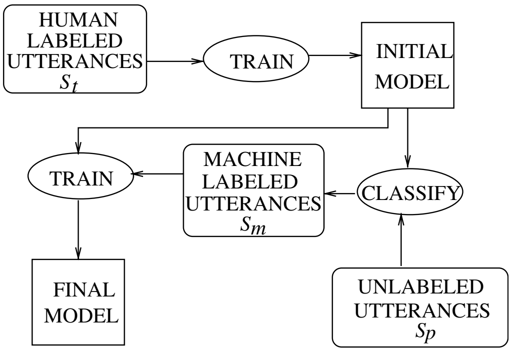

Fig. 7. The second semi-supervised learning method: augmenting the model.

This method is similar to incorporating prior knowledge into Boosting (Schapire et al., 2002). In that work, a model which fits both the training data and the task knowledge is trained. In our case, the aim is to train a model that fits both the human-labeled and machine-labeled data. For this purpose, we first train an initial model using the human-labeled data. Then the Boosting algorithm tries to fit both the machine-labeled data and the prior model using the following loss function:

$$\sum_i \sum_l (\ln(1 + \mathrm{e}^{-Y_i[l]f(x_i,l)}) + \eta KL(\Pr(Y_i[l] = +1 \mid x_i) \| \rho(f(x_i, l)))),$$

where

$$KL(p\|q) = p \ln\left(\frac{p}{q}\right) + (1-p)\ln\left(\frac{1-p}{1-q}\right)$$

is the Kullback–Leibler divergence (or binary relative entropy) between two probability distributions p and q. In our case, they correspond to the distribution from the prior model $\Pr(Y_i[l] = +1|x_i)$ and to the distribution from the constructed model $\rho(f(x_i, l))$, where $\rho(x)$ is the logistic function $1/(1 + \mathrm{e}^{-x})$. This term is basically the distance from the initial model built by human-labeled data to the new model built with machine-labeled data. In the marginal case, if these two distributions are always the same, then the KL term will be zero and the loss function will be exactly the same as the first term, which is nothing but the logistic loss. Here, η is used to control the relative importance of these two terms. This weight may be determined empirically on a held-out set. In addition, similar to the first method, in order to reduce the noise added because of classifier errors, we only exploit those utterances that are classified with a confidence score higher than some threshold.

Note that most classifiers support a way of combining models or augmenting the existing model, so although this implementation is classifier (Boosting) dependent, the idea is more general. For example, in a Naive Bayes classifier, this can be implemented as linear model interpolation.

The challenge with semi-supervised learning is that only the utterances which are classified with a confidence score larger than some threshold may be exploited in order to reduce the noise introduced by the classifier errors. Intuitively, the noise introduced would be less with better initial models, but in such a case, additional data will be less useful. So one may expect such semi-supervised techniques to be less useful with very little or very large amounts of data. We have also observed this behavior in our experiments presented in Section 7. Instead of using a threshold to select machine-labeled data, an alternative approach would be to modify the classifier so that at each iteration the confidence scores of the call-types contribute to the data distribution.

6. Combining active and semi-supervised learning

The examples which are not selected by active learning can be exploited using semi-supervised learning methods.

We propose the algorithm described in Fig. 8 to combine active learning and semi-supervised learning. A simplified flowchart is depicted in Fig. 9. This algorithm is a combination of the active learning algorithm presented in Section 4 and the second semi-supervised learning algorithm presented in Section 5. Instead of leaving out the utterances classified with high confidence scores, this algorithm exploits them.

Note that in this algorithm, instead of assuming a fixed pool, we have assumed that there is a constant stream of incoming data, which, we believe, better reflects a real-life scenario. It is also straightforward to modify this algorithm to handle cases with a fixed pool. Another variant of this approach would be to ignore Step [2.1] and use the classifier built by augmenting the machine-labeled data after the first iteration.

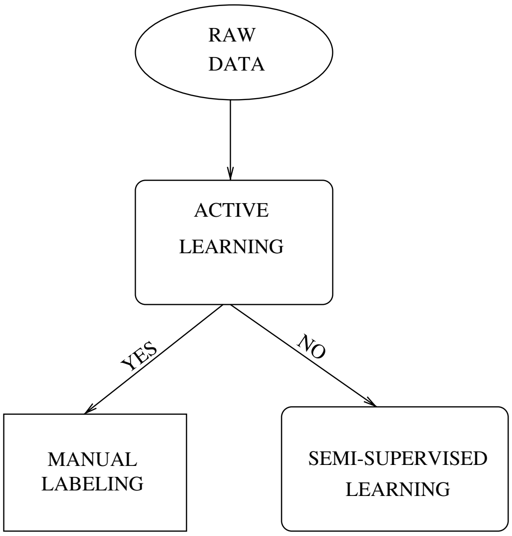

Fig. 9. Combining active and semi-supervised learning.

If the manual labeling resources are very scarce, one can set a second threshold to eliminate noise from unlabeled data. That means using one threshold for active learning and another for semi-supervised learning. Otherwise combining active and semi-supervised learning eliminates the need for the second threshold. Since the utterances which

```
(1) Given some amount of human-labeled training data $S_t$, and a
    larger amount of unlabeled data in the pool $S_p = \{s_1, ..., s_n\}$
(2) $S_u = \emptyset$
(3) While (labelers/utterances are available) do
   2.1 Train a classifier using the current training data $S_t$
   2.2 Classify the utterances in $S_p$ using this classifier
       computing the confidence scores, $CS(s_i), \quad i = 1, ..., n$
   2.3 Manually label the set $S_k = \{s_i : CS(s_i) < th\}$
   2.4 $S_t = S_t \bigcup S_k$
   2.5 $S_u = S_u \bigcup (S_p \backslash S_k)$
   2.6 Augment the classifier by changing the loss function
       so that it fits both the initial model and new
       machine-labeled data, $S_u$
   2.7 Get new $S_p$
```

Fig. 8. Algorithm for combining active and semi-supervised learning.

are already classified with low confidence scores are selected by active learning and sent to a human for labeling, the noise is already reduced for semi-supervised learning. So it may not be necessary to find an optimal threshold to work with.

One other problem with the existing semi-supervised approaches is that, one particular call-type may be poorly trained using the initial human-labeled data, as there is very little or no data for that call-type. The proposed approaches are not supposed to improve the classification accuracy for such call-types. Combining active learning with semi-supervised learning may be a solution to this problem, too.

Another advantage is that since semi-supervised learning only chooses the utterances which are classified with confidence scores higher than some threshold, we expect the well represented or easy-to-classify call-types to dominate the automatically labeled data. This results in just the opposite effect of active learning which mostly trims such call-types, so the combination of these two learning methods may alleviate the data imbalance problem due to each separately.

One problem with semi-supervised learning is that it may introduce noise to the training data, even though a thresholding mechanism is employed. An utterance might be classified erroneously with very high confidence. This is a problem especially for combining certainty-based active and semi-supervised learning methods, since such examples will not be selected for manual labeling by active learning. In order to alleviate this problem, before using semi-supervised learning at Step [2.6] we have found it useful to re-train the classifier only with manually labeled examples and re-classify the unselected utterances S_u with the new model.

```
(1) Re-train the classifier using $S_t$
(2) Classify the unselected utter-
    ances $S_u$ using this classifier to
    get machine-labels.
```

This is expected to decrease the noise in the machine-labeled data, but is still a partial solution. Although this is a plausible problem, we did not encounter such a case in our experiments. It is possible (but unlikely) that a random initial training set will be highly non-representative of whatever it is that we are trying to learn. More empirical experimentation is needed to determine how serious a problem this may be.

7. Experiments and results

We have evaluated these active learning methods using the utterances from the database of the *How May I Help You?*^{SM} (*HMIHY*^{SM}) system for AT&T customer care (Tur et al., 2002; Gorin et al., 2002). In this natural dialog system, users are responding to the open-ended prompt "How may I help you?" by asking questions about their phone bills, calling plans, etc., and the system aims to classify them into one or more of a total of 49 call-types, such as *Request*(*Account_Balance*), or *Explain*(*Bill*). If the intent is not understood (i.e. the confidence score is less than some rejection threshold) or the intent is vague (e.g. "I have a problem."), the user would be re-prompted. If the system is not confident enough, a confirmation prompt would be played.

There are about 65,000 utterances in this corpus, including the responses of users to all of the prompts. We performed our tests using the BoosTexter tool (Schapire and Singer, 2000). For all experiments, we used word *n*-grams in transcriptions as features and iterated BoosTexter 500 times. In order to evaluate the classifier performance we used "classification error rate" (or "top class error rate") which is the fraction of utterances in which the top scoring call-type is not one of the true call-types assigned by a human-labeler. In this study we assumed that all candidate utterances are first recognized by the same automatic speech recognizer (ASR), so we deal with only text input of the same quality.

7.1. Confidence score evaluations

We began the experiments by checking the informativeness of using the classifier score with this data. We trained a classifier with BoosTexter using all the data. We used word *n*-grams as features for classification. For each classifier score

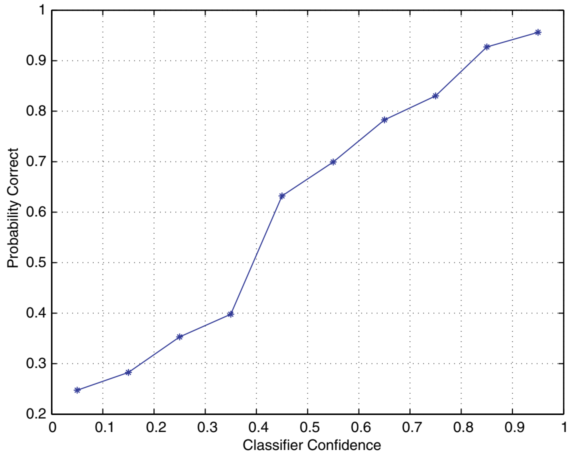

Fig. 10. Accuracy with respect to the classifier score.

bin, we computed the accuracy of its decision. As seen in Fig. 10, we got an almost diagonal curve, verifying that BoosTexter scores are indeed useful for distinguishing misclassified utterances.

7.2. Active learning experiments

In order to see the actual improvement obtained with active learning, we performed controlled experiments comparing our methods with random sampling. As a first experiment, we used 51,755 utterances from the HMIHY data set, excluding some garbage or empty utterances. We used 40,000 randomly selected examples for training (i.e., as potential candidates for labeling), and the remaining 11,755 utterances for testing. We incrementally trained the classifier every 2000 utterances sampled among all the unlabeled utterances and generated learning curves for classification error rate, one for random sampling and one for selective sampling. Fig. 11 shows the results. Also, the entire experiment was repeated ten times with different training and test sets and the results were averaged. It is evident that selective sampling significantly reduces the need for labeled data. For instance, achieving a test error of 26% requires 40,000 examples if they are randomly chosen, but only around 11,000 selected examples, a savings of 72.5%.

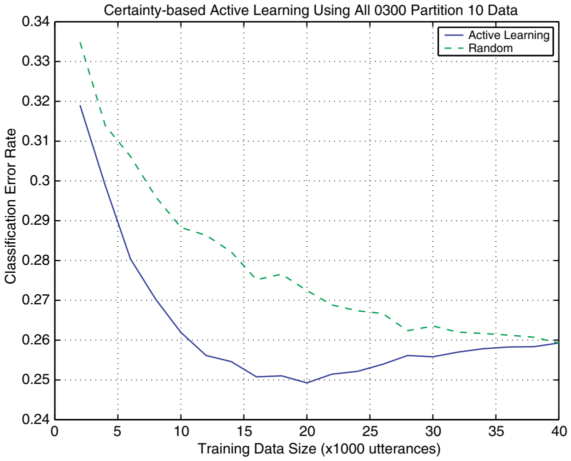

Fig. 11. The error rates achieved by BoosTexter using active and random selection of examples for labeling. At each iteration, utterances are selected from a fixed pool containing all the unlabeled utterances.

Note that because these are controlled experiments and we select utterances from a fixed pool, the performance using all available training data has to be the same for both random sampling and selective sampling.

The active learning curve also exhibits an interesting phenomenon previously noted by Schohn and Cohn (2000): better performance can actually be achieved using *fewer* training examples. Apparently, examples added toward the end are not only uninformative, but actually disinformative. With selective sampling, even though we add more examples to the training data, the classifier performance degrades and we achieve the best performance using around 20,000 utterances as training data, only half of all the data.

As a second set of experiments, we simulated a more likely case where we assume that each day we get 2000 utterances and select 500 of them for labeling, forgetting about the remainder. So, instead of using a fixed pool containing all unlabeled utterances, we used a dynamic and smaller-sized pool. We again used the HMIHY dataset, but in this case, we used all the data since our aim is to simulate actual deployment behavior. We randomly selected 7013 utterances for testing, and 58,000 utterances for training. We chronologically sorted the training data to get more realistic results.

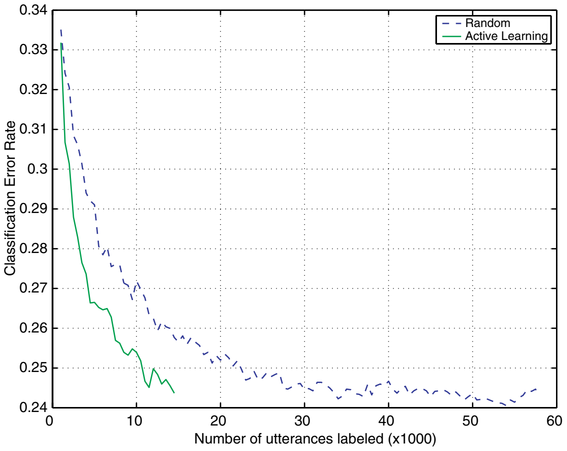

Fig. 12. The error rates achieved by BoosTexter using active and random selection of examples for labeling. At each iteration, utterances are selected from a dynamic smaller size pool.

Similar to the previous experiments, we re-trained the classifier after each pseudo-day.

The results are shown in Fig. 12. The axes are the same as before. But in this experiment, since we ignore 75% of the data, the active learning curve goes only until 14,500 utterances. Similar to the previous experiments, we observe a significant reduction in the amount of labeled data needed to achieve a given performance level. For instance, achieving a test error of 25% requires 23,500 examples if randomly chosen, but only 11,000 actively selected examples, a savings of 53.2%.

7.3. Semi-supervised learning experiments

We evaluated the semi-supervised learning methods using the same corpus. We used the same 7013 test utterances used for the second set of active learning experiments. We divided it into two and put 3500 utterances in the held-out set which we used to optimize the parameters of the algorithm (i.e., th and η), and 3513 utterances in the real test set. As in the first set of experiments, we did not consider active learning and evaluated semi-supervised learning methods on top of randomly selected base training data.

First, we selected the optimal threshold th of confidence scores, using the held-out set. Fig. 13 shows the trade-off for selecting optimal threshold th for the held-out set. We trained initial models using 2000, 4000, and 8000 human-labeled utterances and then augmented these as described in the first method with the remaining training data (only using machine-labeled call-types). The x-axis gives the different thresholds used to select from the unlabeled data, and the y-axis gives the classification error rate if that data is also exploited. A threshold of 0 means using all the machine-labeled data and 1 means using none. As can be seen from the figure, there is consistently a 1–1.5% absolute difference in classification error rates using various thresholds for each data set size. When we use all the data (i.e., th = 0), we do not achieve much improvement due to the noise introduced.

Fig. 14 depicts the performance using the two methods proposed by plotting the learning curves for various initial models trained on human-labeled data. In the figure, the x-axis is the number of human-labeled training utterances, and the y-axis is the classification error rate of the corresponding model on the full test set of 7013 utterances. The baseline is the top curve with the highest error rate where no machine-labeled data is used. The two curves below the baseline are obtained by the two proposed methods. In both methods, we selected 0.5 as the threshold for selecting machine-labeled data, and 0.9 as the weight η for all data sizes in

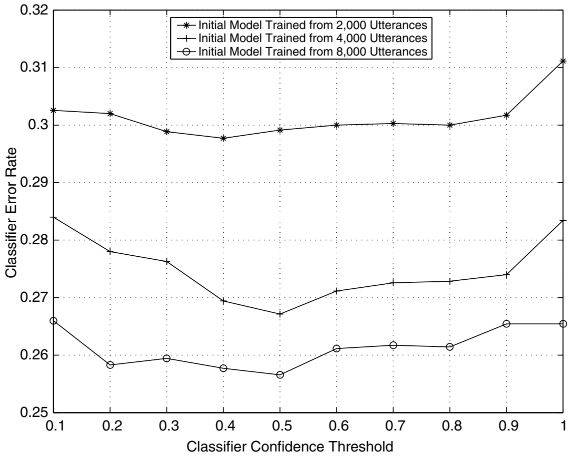

Fig. 13. Trade-off for choosing the threshold to select among the machine-labeled data on the held-out set.

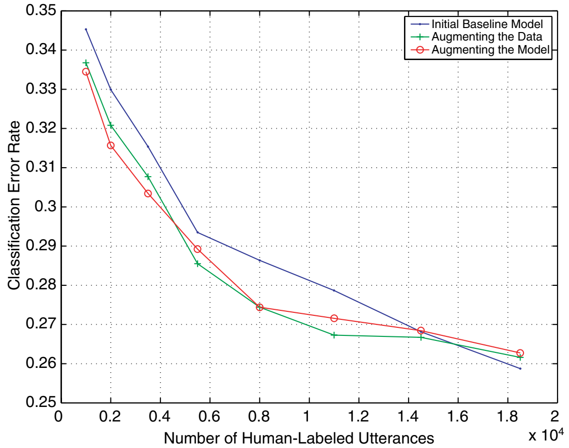

Fig. 14. Results using semi-supervised learning. Topmost learning curve is obtained using just human-labeled data as a baseline. Below that lie the learning curves using the first and second methods.

the second method. As in the case of the held-out set, we consistently obtained 1–1.5% classifier error rate reductions on the test set using both approaches when the labeled training data size is less than 15,000 utterances. The reduction in the need for human-labeled data to achieve the same classification performance is around 30%. For example we got the same performance when we used 5000 human-labeled utterances instead of 8000 if we augment the data with unlabeled utterances. This improvement disappears when the amount of manually labeled data exceeds 15,000 utterances, verifying our intuition that semi-supervised learning is expected to be less useful with better initial models.

7.4. Active and semi-supervised learning experiments

In order to evaluate the performance of active and semi-supervised learning, we only considered the dynamic pool case, which is more realistic. Similar to the second set of experiments done for active learning, we simulated the more likely case where we assume each day we get 2000 utterances and select 500 of them for labeling, exploiting the remaining using semi-supervised learning. We again used the whole HMIHY dataset, with the same 7013 utterances for testing, and 58,000 utterances for training. Similar to the previous experiments, we retrained the classifier after each pseudo-day. In this experiment, we only used the second semi-supervised learning method using a variable threshold (since 500 utterances are selected by active learning on each day) and 0.9 as the weight η.

Fig. 15 shows the results using only active learning and also combining active learning with semi-supervised learning. In this experiment, we only used the second semi-supervised learning method. One impressive result is that, all along the learning curve, using semi-supervised learning improves the performance of the classifier by around 0.5% absolute. This is in contrast to using just semi-supervised learning where the improvement disappears after exceeding some amount of human-labeled data.

The following is a brief analysis of the effect of combining active and semi-supervised learning on call-type distribution: Fig. 16 shows the distribution of call-types after passive learning (i.e. random sampling), after active learning, and after combining active and semi-supervised learning. It is clear that active learning has trimmed the most frequent call-types giving more weight to infrequent ones. When we augment the automatically labeled data with the selectively sampled data, we

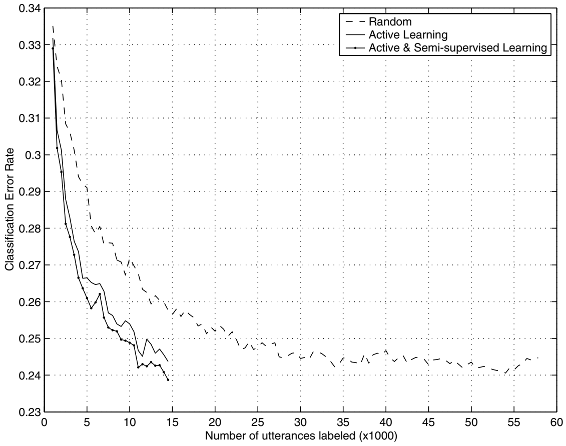

Fig. 15. Results using active and semi-supervised learning. Topmost learning curve is obtained using just randomly sampled human-labeled data as a baseline. Below that lie the learning curves using the active and semi-supervised learning methods.

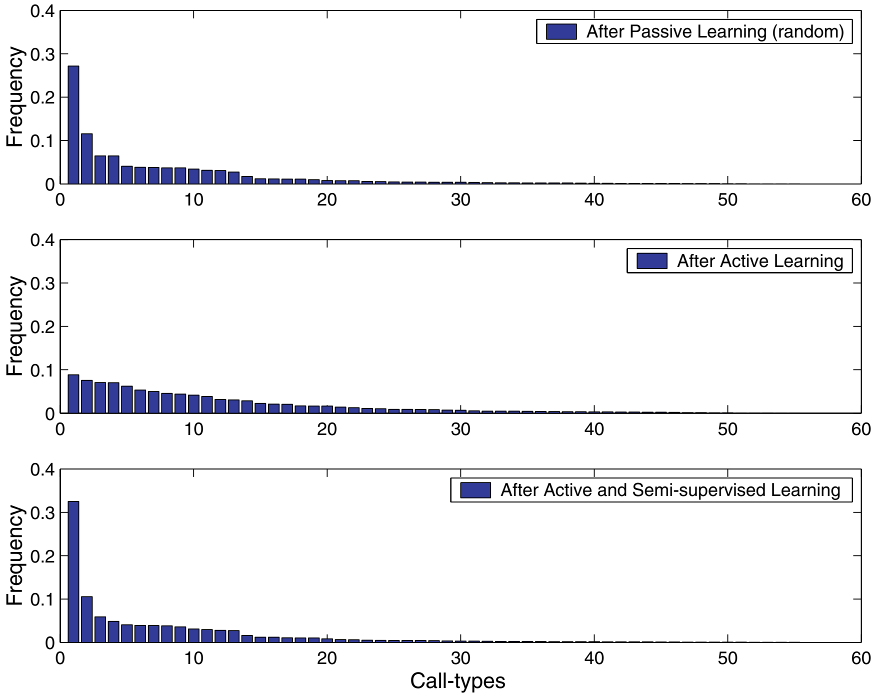

Fig. 16. Frequency of call-types using random sampling, active learning, and active and semi-supervised learning.

have approximated the original frequencies of the call-types.

The effect of data imbalance caused by active or semi-supervised learning is hard to measure. Instead we did the following experiment: We chose 5000 utterances randomly from a subset of training data containing 10,000 utterances. We trained a classifier, and evaluated its performance on our unseen test set. As seen in Table 1 the classification error rate happened to be 29.12%. Then we selected another 5000 utterances, in this case ignoring the utterances whose call-types have appeared more than some threshold. In other words, we trimmed the distribution in a brute-force way. This time, the error rate increased significantly to 30.81%. This experiment is an empirical proof of the problem caused by unbalanced data, hence by any selective sampling method that does not consider prior distributions, and in a way counterintuitive to the performance of a classifier trained with the selectively sampled data. Our opinion is that the performance we got with active learning may improve if we can find a solution to the data imbalance problem; this may explain in part why automatically labeled data helps.

Table 1
Effect of the data imbalance in training data to classification

	Classification error rate (%)
Random	29.12
Biased	30.81

8. Conclusions and discussion

We have presented active and semi-supervised learning algorithms for a spoken language understanding system, in this case a call classifier. The aim is to reduce the number of labeled training examples by selectively sampling a subset of the unlabeled data and exploiting the unselected ones. In other words, we have studied the questions: *which data to label?* and *what to do with the remaining unlabeled data?*

First, inspired by certainty-based active learning approaches, we have selectively sampled the candidate utterances with respect to the confidence score of the call classifier. Then, using semi-supervised learning methods, we have exploited the utterances, which are confidently classified by the classifier, hence ignored by active learning.

We have shown that, for the task of call classification, by combining active and semi-supervised learning, it is possible to speed up the learning rate of the classifier with respect to the number of labeled utterances. Our results indicate that we have achieved the same call classification accuracy using less than half of the labeled data, and have resolved partly the performance deterioration caused by the data imbalance problem introduced by using just active or semi-supervised learning.

It is clear that the active and semi-supervised learning methods presented in this paper can be easily applied to other statistical SLU or classification tasks such as named entity extraction, part-of-speech tagging or text categorization.

Our future research includes selective sampling for semi-supervised learning which aims to reduce the noisy or uninformative examples in the machine-labeled data. Instead of using all the utterances which are confidently classified, we would like to explore ways to extract from the machine-labeled examples only the ones which will help the classification task.

Considering our domain of spoken dialog systems, it is also possible to use information other than classification confidence scores for active and semi-supervised learning. This may include dialog level features such as the previous prompt played or the previous call-type, or customer related features such as location or account features.

Acknowledgment

We would like to thank the anonymous reviewers, Giuseppe Riccardi, Mazin Rahim, and Jay Wilpon for their valuable comments and suggestions.

References

Abe, N., Mamitsuka, H., July 1998. Query learning strategies using Boosting and Bagging. In: Proc. Internat. Conf. on Machine Learning (ICML), Madison, WI.

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

Argamon-Engelson, S., Dagan, I., 1999. Committee-based sample selection for probabilistic classifiers. J. Artif. Intell. Res. 11, 335–360.

Blum, A., Mitchell, T., July 1998. Combining labeled and unlabeled data with co-training. In: Proc. Workshop on Computational Learning Theory (COLT), Madison, WI.

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

Collins, M., Schapire, R.E., Singer, Y., 2002. Logistic regression, AdaBoost and Bregman distances. Machine Learning 48 (1/2/3).

Freund, Y., Schapire, R., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. System Sci. 55 (1), 119–139.

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

Friedman, J., Hastie, T., Tibshirani, R., 2000. Additive logistic regression: A statistical view of boosting. Ann. Statist. 38 (2), 337–374.

Ghani, R., July 2002. Combining labeled and unlabeled data for multiclass text categorization. In: Proc. Internat. Conf. on Machine Learning (ICML), Sydney, Australia.

Gorin, A.L., Riccardi, G., Wright, J.H., 2002. Automated natural spoken dialog. IEEE Comput. Mag. 35 (4), 51–56.

Haffner, P., Tur, G., Wright, J., April 2003. Optimizing SVMs for complex call classification. In: Proc. Internat. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Hong Kong.

Hakkani-Tür, D., Riccardi, G., Gorin, A., May 2002. Active learning for automatic speech recognition. In: Proc. Internat. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Orlando, FL.

Iyer, R., Gish, H., McCarthy, D., May 2002. Unsupervised training techniques for natural language call routing. In: Proc. Internat. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Orlando, FL.

Kuo, J., Lee, C., Zitouni, I., Fosler-Lusser, E., Ammicht, E., September 2002. Discriminative training for call classification and routing. In: Proc. Internat. Conf. on Spoken Language Processing (ICSLP), Denver, CO.

Lewis, D.D., Catlett, J., July 1994. Heterogeneous uncertainty sampling for supervised learning. In: Proc. Internat. Conf. on Machine Learning (ICML), New Brunswick, NJ.

Liere, R., Tadepalli, P., July 1997. Active learning with committees for text categorization. In: Proc. Conf. of the American Association for Artificial Intelligence (AAAI), Providence, RI.

McCallum, A.K., Nigam, K., July 1998. Employing EM and pool-based active learning for text classification. In: Proc. Internat. Conf. on Machine Learning (ICML), Madison, WI.

Muslea, I., Minton, S., Knoblock, C.A., July 2002. Active + semi-supervised learning = robust multi-view learning. In: Proc. Internat. Conf. on Machine Learning (ICML), Sydney, Australia.

Natarajan, P., Prasad, R., Suhm, B., McCarthy, D., September 2002. Speech enabled natural language call routing: BBN call director. In: Proc. Internat. Conf. on Spoken Language Processing (ICSLP), Denver, CO.

Nigam, K., Ghani, R., November 2000. Analyzing the effectiveness and applicability of co-training. In: Proc. Internat. Conf. on Information and Knowledge Management (CIKM), McLean, VA.

Nigam, K., McCallum, A., Thrun, S., Mitchell, T., 2000. Text classification from labeled and unlabeled documents using EM. Machine Learning 39 (2/3), 103–134.

Riccardi, G., Hakkani-Tür, D., September 2003. Active and unsupervised learning for automatic speech recognition. In: Proc. European Conf. on Speech Communication and Technology (EUROSPEECH), Geneva, Switzerland.

Sassano, M., July 2002. An empirical study of active learning with support vector machines for Japanese word segmentation. In: Proc. Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA.

Schapire, R.E., Rochery, M., Rahim, M., Gupta, N., July 2002. Incorporating prior knowledge into boosting. In: Proc. Internat. Conf. on Machine Learning (ICML), Sydney, Australia.

Schapire, R.E., March 2001. The boosting approach to machine learning: an overview. In: Proc. MSRI Workshop on Nonlinear Estimation and Classification, Berkeley, CA.

Schapire, R.E., Singer, Y., 2000. Boostexter: a boosting-based system for text categorization. Machine Learning 39 (2/3), 135–168.

Schapire, R.E., Singer, Y., 1999. Improved boosting algorithms using confidence-rated predictions. Machine Learning 37 (3), 297–336.

Schohn, G., Cohn, D., June 2000. Less is more: Active learning with support vector machines. In: Proc. Internat. Conf. on Machine Learning (ICML), Palo Alto, CA.

Seung, H.S., Opper, M., Sompolinsky, H., July 1992. Query by committee. In: Proc. Workshop on Computational Learning Theory (COLT), Pittsburgh, PA.

Tang, M., Luo, X., Roukos, S., July 2002. Active learning for statistical natural language parsing. In: Proc. Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA.

Thompson, C., Califf, M.E., Mooney, R.J., June 1999. Active learning for natural language parsing and information extraction. In: Proc. Internat. Conf. on Machine Learning (ICML), Bled, Slovenia.

Tong, S., Koller, D., 2001. Support vector machine active learning with applications to text classification. J. Machine Learning Res. 2, 45–66.

Tur, G., Hakkani-Tür, D., September 2003. Unsupervised learning for spoken language understanding. In: Proc. European Conf. on Speech Communication and Technology (EUROSPEECH), Geneva, Switzerland.

Tur, G., Wright, J., Gorin, A., Riccardi, G., Hakkani-Tür, D., September 2002. Improving spoken language understanding using word confusion networks. In: Proc. Internat. Conf. on Spoken Language Processing (ICSLP). Denver, CO.

Tur, G., Schapire, R.E., Hakkani-Tür, D., May 2003. Active learning for spoken language understanding. In: Proc. Internat. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Hong Kong.