Multi-View Semi-Supervised Learning for Dialog Act Segmentation of Speech

Umit Guz, *Member, IEEE*, Sébastien Cuendet, Dilek Hakkani-Tür, *Senior Member, IEEE*, and Gokhan Tur, *Senior Member, IEEE*

***Abstract*—Sentence segmentation of speech aims at determining sentence boundaries in a stream of words as output by the speech recognizer. Typically, statistical methods are used for sentence segmentation. However, they require significant amounts of labeled data, preparation of which is time-consuming, labor-intensive, and expensive. This work investigates the application of multi-view semi-supervised learning algorithms on the sentence boundary classification problem by using lexical and prosodic information. The aim is to find an effective semi-supervised machine learning strategy when only small sets of sentence boundary-labeled data are available. We especially focus on two semi-supervised learning approaches, namely, self-training and co-training. We also compare different example selection strategies for co-training, namely, agreement and disagreement. Furthermore, we propose another method, called self-combined, which is a combination of self-training and co-training. The experimental results obtained on the ICSI Meeting (MRDA) Corpus show that both multi-view methods outperform self-training, and the best results are obtained using co-training alone. This study shows that sentence segmentation is very appropriate for multi-view learning since the data sets can be represented by two disjoint and redundantly sufficient feature sets, namely, using lexical and prosodic information. Performance of the lexical and prosodic models is improved by 26% and 11% relative, respectively, when only a small set of manually labeled examples is used. When both information sources are combined, the semi-supervised learning methods improve the baseline F-Measure of 69.8% to 74.2%.**

***Index Terms*—Boosting, co-training, prosody, self-training, semi-supervised learning, sentence segmentation.**

Manuscript received October 24, 2008; revised July 01, 2009. First published July 24, 2009; current version published November 20, 2009. This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) CALO (FA8750-07-D-0185, Delivery Order 0004), in part by the Scientific and Technological Research Council of Turkey (TUBITAK) funding at SRI, in part by a J. William Fulbright Post-Doctoral Research Fellowship, Isik University Research Fund (Projects: 05B304, 09A301), DARPA GALE (HR0011-06-C-0023) and in part by the Swiss National Science Foundation through the research network, IM2 fundings at ICSI. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ruhi Sarikaya.

U. Guz is with the International Computer Science Institute (ICSI), Speech Group, Berkeley, CA 94704 USA, and also with Engineering Faculty, Department of Electronics Engineering, Isik University, 34980 Sile, Istanbul, Turkey (e-mail: guz@isikun.edu.tr; guz@icsi.berkeley.edu).

S. Cuendet is with Optaros, Zurich CH-8037, Switzerland. He has been with the International Computer Science Institute (ICSI), Speech Group, Berkeley, CA 94704 USA, and on leave from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne CH-1015, Switzerland (e-mail: cuendet@icsi.berkeley.edu).

D. Hakkani-Tür is with the International Computer Science Institute (ICSI), Speech Group, Berkeley, CA 94704 USA (e-mail: dilek@icsi.berkeley.edu).

G. Tur is with the Speech Technology and Research (STAR) Laboratory, SRI International, Menlo Park, CA 94025 USA (e-mail: gokhan@speech.sri.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2009.2028371

I. Introduction

DIALOG act or sentence segmentation for speech[1] aims at finding sentence units in a stream of words, output by the speech recognizer. Sentence segmentation is a preliminary step for many speech processing applications, such as parsing, machine translation, and information extraction. Speech recognizer output typically lacks certain entities, such as headers, paragraphs, sentence punctuation, and capitalization. However, speech provides extra nonlexical cues, related to features like pitch, energy, pause and word durations, named as prosodic features. It has been shown that for segmentation of speech into sentences, prosodic and lexical cues provide complementary information [1].

For sentence segmentation, statistical methods are employed. However, such data-driven methods require significant amounts of labeled data, which is expensive, time-consuming, and laborious to prepare. In this paper the goal is building a better sentence segmentation system using less data. In our earlier work, we proposed supervised model adaptation methods for sentence segmentation using a small amount of labeled in-domain data and a large amount of labeled out-of-domain data [2]. This paper, on the other hand, focuses on multi-view semi-supervised training of sentence segmentation models without exploiting any out-of-domain data. We compare the well-known semi-supervised learning approaches in the machine learning and speech and language processing literature, including co-training and self-training. Furthermore, previously, we investigated the effects of co-training methods on the sentence segmentation problem [3]. That work employed a single pass co-training algorithm with only 1000 labeled examples and showed an F-Measure improvement from 69.8% to 71.2%.

In this paper, we study the effect of iterating the co-training algorithm and show that F-Measure further increases to 74.2%. The optimum number of iterations is determined using a held-out set. Furthermore, we investigate the effect of the size of the initial seed of manually labeled data on this task using these semi-supervised learning methods. More specifically, we show results using 1000, 3000, and 6000 annotated examples. The performance improvement is consistently seen in all cases using semi-supervised learning, however by a lesser amount as the data sizes increase, as expected.

This study also proposes a novel multi-view semi-supervised learning approach, called, *self-combined* learning combining self-training and co-training algorithms. We show that this

[1]Sentence segmentation from here on.

method outperforms co-training and self-training for the first iteration, however after multiple iterations, co-training results in better performance.

After giving the experimental results, we present an analysis of cases which demonstrate the use of co-training for sentence segmentation. Basically we compare examples that are classified correctly with high confidence by the lexical model but incorrectly by the prosodic model and vice versa.

In Section II, we present related work on sentence segmentation and semi-supervised learning methods, and then describe our sentence segmentation and semi-supervised learning approaches in Section III. We provide experimental results using self-training and different semi-supervised learning strategies with the ICSI Meeting Recorder Dialog Act (MRDA) corpus [4] in Section IV and our conclusions in Section V.

II. Related Work

Before presenting the details of the semi-supervised learning methods employed in this study, we first present the related work for sentence segmentation. Then we present the literature on the semi-supervised learning methods.

A. Sentence Segmentation

Segmenting an utterance into sentential units has different interpretations depending on the domain. In prewritten speech, such as broadcast news, a sentential unit is equivalent to a regular grammatical sentence. However, work on multiparty meetings has been more recent [5], [6], among others. The problem with the meetings domain is that, unlike prewritten speech, it is not always clear where the sentence boundaries are. The interannotator agreement has been low during the segmentation of the meetings [5]. For the example utterance *okay no problem*, it is unclear whether or not this is a single sentence. To alleviate this problem, the segmentation task is redefined for the meetings domain as the task of dialog act segmentation, and dialog acts are defined for conversational speech using various standards such as DAMSL [7] or MRDA [8]. According to these, the example *okay no problem* has two dialog act tags (or sentential units): *okay* and *no problem*.

In the literature, typically sentence or dialog act segmentation is treated as a boundary classification problem where the goal is finding the most likely boundary tag sequence T given the features F

$$\text{argmax}_T P(T|F)$$

To this end, mostly generative, discriminative, or hybrid models are used. The most popular generative model is the hidden event language model, as introduced by Stolcke and Shriberg [9]. In this approach, sentence boundaries are treated as the hidden events, and the above optimization is simply done by the Viterbi decoding using only lexical features, i.e., language model, to model $P(F|T)$. $P(T)$ is simply considered to be a constant, optimized according to the tradeoff between false alarms and misses:

$$\text{argmax}_T P(T|F) = \text{argmax}_T P(F|T) \times P(T)$$

Decision trees were also used to build hybrid models to improve this approach by using additional prosodic features [1]. The posterior probabilities obtained from the decision trees were simply converted to state observation likelihoods by dividing to their priors following the well-known Bayes rule

$$\text{argmax}_T \frac{P(T|F)}{P(T)} = \text{argmax}_T P(F|T).$$

With the advances in discriminative classification algorithms, researchers tried using conditional random fields [10], boosting [2], and hybrid approaches using boosting and maximum entropy classification algorithms [11]. In this paper, we employed a discriminative classifier namely, boosting which is described below.

Recent research has focused on model adaptation methods for improving dialog act segmentation for meetings using spontaneous telephone conversations, and speaker-specific prosodic [12] and lexical modeling [2].

B. Semi-Supervised Learning

The goal of semi-supervised learning is to reduce the amount of labeled data needed to train statistical models. In most real-life applications it is easier to collect, than to label, data. For example, one can easily record audio from broadcast news, but transcribing or annotating the audio with some labels is a harder task.

Semi-supervised learning methods assume that there is a small amount of in-domain labeled data and a relatively larger amount of unlabeled data. Then the goal is exploiting the unlabeled data to improve performance of the system. In all semi-supervised learning methods, first an initial classifier is trained using the available data. Then the basic idea is to use this classifier to label the unlabeled data automatically, and improve classifier performance using the machine-labeled examples. Making this process iterative and exploiting a small number of automatically labeled examples in each iteration enables incremental use of unlabeled data. This is in contrast to using *all* the unlabeled data (even in a weighted manner), and the behavior of the models significantly changes accordingly [13].

1) Self-Training: Self-training is the most popular semi-supervised learning method. For self-training, the given model estimates the classes for the unlabeled portion of the data. Then the examples that are classified automatically are added to the training set, the model is retrained, and the whole process is iterated [13]. To eliminate the noise introduced by falsely classified samples, only those classified with a high confidence may be exploited. Similarly, those classified with very high confidence may also be ignored to avoid the bias toward easy-to-classify classes.

Self-training is very closely related to unsupervised model adaptation typically employed for speech and speaker processing systems. A very popular adaptation approach is maximum *a posteriori* (MAP) adaptation [14]. With some simplification, the MAP adaptation can be reduced to a weighted linear interpolation of the prior model and the model trained with automatically classified samples. That is nothing but self-training. This approach has been applied to unsupervised

acoustic and language model (LM) adaptation and speaker adaptation. Bacchiani and Roark have applied iterative unsupervised LM adaptation to voicemail transcription [15]. Hakkani-Tür *et al.* have employed unsupervised LM adaptation for new call center spoken dialog applications [16]. Gretter and Riccardi have exploited word confidences obtained from word confusion networks during unsupervised LM adaptation [17].

Self-training has been applied to a number of language processing tasks, such as part of speech tagging [18], word sense disambiguation [19], and syntactic parsing [20].

2) Co-Training: Contrary to the traditional single view machine learning concept, the set of features of the multi-view approach consists of two or more different feature subsets (views), each of which is distinguishable and sufficient for learning from itself individually. The main difference between these two approaches is that in the multi-view concept the views are bootstrapped from each other, while in the single view the algorithm trains itself.

Co-training is a very effective machine-learning technique that has been used successfully in several classification tasks, such as web page classification, word sense disambiguation, and named-entity recognition. Co-training is a semi-supervised learning method that aims to improve performance of a supervised learning algorithm by incorporating large amounts of unlabeled data into the training data set. Co-training algorithms work by generating two or more classifiers trained on different views of the input labeled data that are then used to label the unlabeled data separately. The most confidently labeled examples of the automatically labeled data can then be added to the set of manually labeled data. The process may continue for several iterations. In this paper, we describe the application of the co-training method for sentence segmentation where we use prosodic and lexical information as two views of the data.

The co-training approach was first introduced and performed by Blum and Mitchell [21], [22]. The main goal is using multiple views together with unlabeled data to augment a much smaller set of labeled examples. More specifically, the presence of multiple distinct views of each example can be used to train separate models for the same task, and then each classifier's predictions on the unlabeled examples are used to augment the training set of the other classifier. The task Blum and Mitchell used was identifying the web pages of academic courses from a large collection of web pages collected from several computer science departments. Their co-training implementation had two natural feature sets: the words present in the course web page and the words used in the links pointing to that web page. For this task, both views of examples are considered as sufficient for learning. Blum and Mitchell showed that co-training is probably approximately correct (PAC) learnable when the two views are individually sufficient for classification and conditionally independent given the class. Their results showed that the error rate of the combined classifier was reduced from 11% to 5%.

There has been much effort on investigating the effectiveness of the co-training algorithm in different domains and applications. In recent work [23], it is shown that the independence assumption can be relaxed, and co-training is still effective under a weaker independence assumption. In that work, a greedy algorithm to maximize the agreement on unlabeled data is proposed. This resulted in improved results in a co-training experiment for named entity classification. It is shown that the rate of disagreement between two classifiers with weak independence is an upper bound on the co-training error rate.

Kiritchenko and Matwin applied co-training to the e-mail classification task [24]. In this work, it was found that performance of the co-training was sensitive to the learning algorithm used. In particular, co-training with Naïve Bayes did not result in better performance. However, this was not the case with support vector machines. The authors explained this situation with the inability of the Naïve Bayes to deal with large sparse datasets. This explanation was also confirmed by significantly better results after feature selection.

Nigam and Ghani demonstrated the relationship between the expectation-maximization (EM) algorithm and the semi-supervised learning methods, such as self-training and co-training [13]. They also proposed a hybrid approach, called *Co-EM*, an iterative semi-supervised learning method in which all the unlabeled data is exploited in each iteration. They performed experiments to investigate the sensitivity of the co-training to the assumptions of conditional independence and redundant sufficiency. In the first experiment, co-training was applied to the web page database from Blum and Mitchell [21]. The results showed that the use of co-training was not better than expectation maximization even when there is a natural split of features. Both expectation maximization and co-training improved performance of the initial classifier by approximately 10%. The second experiment was performed on a dataset that had been created in a semi-artificial manner so that the two feature sets are truly conditionally independent. In addition, the condition of redundantly sufficient features was met, since the classifier trained on each of the data sets separately was able to obtain a small error rate. It was found that co-training well outperformed expectation–maximization, and even outperformed the classifier trained with all examples labeled. Their third experiment involved performing co-training on a dataset whereby a natural split of feature sets is not used. The two feature sets were chosen by randomly assigning all the features of the dataset into two different groups. This was tried for two datasets, one with a clear redundancy of features, and one with an unknown level of redundancy and nonevident natural split in features. The results indicated that the presence of redundancy in the feature sets gave the co-training algorithm a bigger advantage over expectation maximization. However, performance of the co-EM method is similar to that of co-training. The results of these experiments verified that co-training has a considerable dependence on the assumptions of conditional independence and redundant sufficiency. However, even when either or both of the assumptions are violated, the performance of co-training can still be quite useful in improving a classifier's performance. We believe that the sentence segmentation task demonstrates a sufficient amount of redundancy since ends of sentences are typically marked with lexical and prosodic cues.

Some studies also consider using different classification algorithms instead of different views for co-training. For example, Wang *et al.* employ maximum entropy and hidden Markov models (HMMs) for part-of-speech tagging and parsing [18]. They also compare co-training with self-training. Similarly,

Mihalcea applied co-training to word sense disambiguation, comparing the effectiveness with self-training [19].

III. Approach

We first briefly present our sentence segmentation approach using lexical and prosodic features. Then, we present how we employ the semi-supervised learning algorithms for this task using various example selection mechanisms. We also provide a description of the self-training method commonly used for semi-supervised learning.

A. Sentence Segmentation

In this paper, the sentence segmentation task was considered as a binary sequence classification problem while s and n represent the sentence boundary and nonsentence boundary as the classes, respectively. In sentence segmentation, the aim is to estimate the classes for sentence boundaries $\{s_1, \ldots, s_N\}$ for a given word sequence $\{w_1, \ldots, w_N\}$, where $s_i, i = 1, \ldots, N-1$ is the boundary between the word w_i and the word w_{i+1}, and s_N is the last boundary following w_N. This is done by training a binary statistical classifier [25] to estimate the posterior probability $P(s_i = k|o_i)$, where $k \in \{+1, -1\}$ for sentence and nonsentence boundaries and o_i are the feature observations for the word boundary s_i. For each word boundary, a probability is emitted by the statistical classifier. Ideally, the decision of the classifier is the class with maximum probability $P(s_i = k|o_i)$. Nevertheless, in the sentence segmentation task, the estimated sentence boundary probabilities $P(s_i = +1|o_i)$ are compared with a threshold value. If the probability is higher than the threshold, a decision of sentence boundary is made; otherwise, the boundary is marked as nonsentence boundary. In this paper, we used a discriminative classifier, namely, Boosting, to estimate $P(s_i = +1|o_i)$.

1) Feature Design: Prosodic and lexical features are used to represent word boundaries to the classifier. The six lexical features are n-grams composed of the word following the boundary of interest and the two previous words. The 34 prosodic features are the pause duration between the two words at the word boundary of interest and at the preceding boundary, and various measures of the pitch and the energy of the voice of the speaker. The features include comparison of the value of the pitch or energy before and after the word boundary of interest, and their speaker normalized versions, following [1]. The range in which the value is measured is either the word or 200-ms time window before/after the word boundary, and the measure considers the maximum, minimum, and average values in this range.

2) Boosting: In this paper, we employed a discriminative classifier, namely, boosting. Boosting is an iterative learning algorithm that aims to combine weak base classifiers to come up with a strong classifier. At each iteration, a weak classifier is learned so as to minimize the training error, and a different distribution or weighting over the training examples is used to give more emphasis to examples that are often misclassified by preceding weak classifiers. In boosting, weighted sampling is used instead of random sampling to focus learning on most difficult examples. Furthermore, weak classifiers are combined using weighted voting instead of equal voting.

Initialization:

Given training data from the instance space $S = \{(x_1, y_1), ..., (x_m, y_m)\}$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y} = \{-1, +1\}$.

Initialize the distribution $D_1(i) = \frac{1}{m}$.

Algorithm:

for $t = 1, ..., T$: **do**

Train a weak learner $h_t : \mathcal{X} \rightarrow \mathrm{R}$ using distribution D_t.

Determine weight α_t of h_t.

Update the distribution over the training set:

$$D_{t+1}(i) = \frac{D_t(i)e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$$

where Z_t is a normalization factor chosen so that D_{t+1} will be a distribution.

end for

Final score:

$f(x) = \sum_{t=0}^{T} \alpha_t h_t(x)$ and $H(x) = sign(f(x))$

Fig. 1. AdaBoost algorithm for the binary classification task.

The basic boosting algorithm is shown in Fig. 1. Boosting assumes that the data set D consists of the training instances or examples x_i. Each example x_i in D is represented by a set of features (prosodic, $x_{i,\mathrm{pro}}$ and lexical, $x_{i,\mathrm{lex}}$). Each example has also the classes (labels) $y_i \in Y$, which are assigned by human labelers. As stated above, the sentence segmentation task can be considered as a binary classification problem, in which every word boundary must be labeled as a sentence boundary or as a nonsentence boundary. In this manner, the set of possible classes $Y = \{+1, -1\}$ is also referred to true or reference classes, where $+1$ and -1 represent the sentence and the nonsentence boundaries, respectively. Note that since sentence segmentation is treated as a binary classification task, $f(x_i = +1) = -f(x_i = -1)$ and hence $|f(x_i)|$ denote the score of the positive scoring class (i.e., highest scoring class), given for the sample, x_i and $H(x_i)$ denotes the boundary type.

B. Semi-Supervised Learning

In our experiments, we applied the semi-supervised methods described below on the sentence segmentation task.

1) Self-Training: The first semi-supervised learning approach we employ is iterative self-training. We believe that self-training will provide us a baseline to compare the performance of other semi-supervised methods. In self-training the feature set is considered to be single view, and there is only a single model automatically labeling the examples for itself. The algorithm for self-training for sentence segmentation is shown in Fig. 3. The scheme of the self-training algorithm that we used is illustrated in Fig. 2. Note that, to reduce the classification noise, we employ a threshold, θ, to select the most confident examples at each iteration. In our experiments this threshold is optimized using a held-out data set.

2) Co-Training: Co-training is one of the most effective multi-view semi-supervised machine learning approach based on learning a hypothesis in each view, and adding to the training set the most confident predictions made on the unlabeled examples repeatedly. In our approach, we use prosodic and lexical

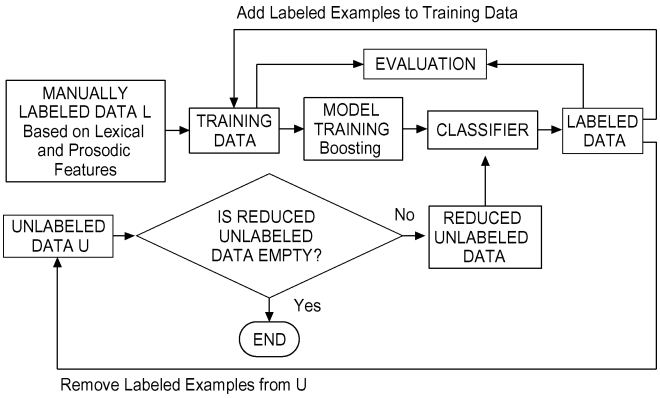


Fig. 2. Self-training scheme with lexical and prosodic features.

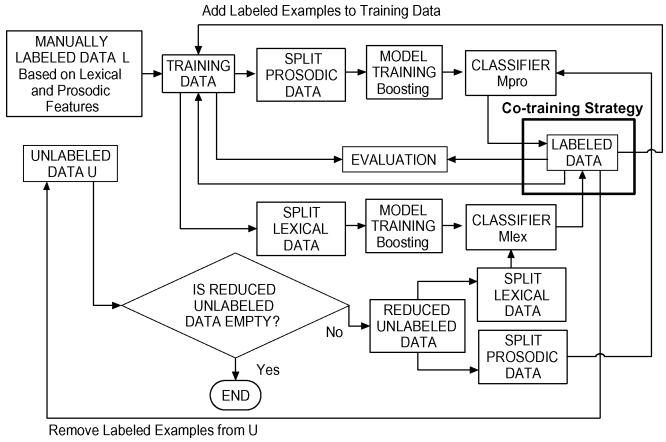


Fig. 4. Co-training scheme with lexical and prosodic features.

Initialization:
1. Given a small set, L, of manually labeled examples
$L = \{(x_1, y_1), ..., (x_{|L|}, y_{|L|})\}$ where $x_i = (x_{i,pro}, x_{i,lex})$.
2. Given a large set, U, of unlabeled examples $U = \{(x_1), ..., (x_{|U|})\}$.
Algorithm:
while $U \neq \emptyset$ and
held-out data set error rate does not converge/increase **do**
  Learn all-features classifier M from L.
  for each $x_i \in U$ **do**
    if $|f_M(x_i)| > \theta$ **then**
      $U = U - \{x_i\}$.
      $L = L \cup \{(x_i, H_M(x_i))\}$.
    end if
  end for
end while

Fig. 3. Self-training algorithm.

Initialization:
1. Given a small set, L, of manually labeled examples
$L = \{(x_1, y_1), ..., (x_{|L|}, y_{|L|})\}$ where $x_i = (x_{i,pro}, x_{i,lex})$.
2. Given a large set, U, of unlabeled examples $U = \{(x_1), ..., (x_{|U|})\}$.
Algorithm:
while $U \neq \emptyset$ and
held-out data set error rate does not converge/increase **do**
  Obtain two sets,
  $L_{pro} = \{(x_{1,pro}, y_1), ..., (x_{|L|,pro}, y_{|L|})\}$ and
  $L_{lex} = \{(x_{1,lex}, y_1), ..., (x_{|L|,lex}, y_{|L|})\}$ from L.
  Learn classifier M_{pro} using L_{pro}.
  Learn classifier M_{lex} using L_{lex}.
  for each $x_i \in U$ **do**
    if $H_{M_{pro}}(x_{i,pro}) = H_{M_{lex}}(x_{i,lex})$ and
    $|f_{M_{pro}}(x_{i,pro})| + |f_{M_{lex}}(x_{i,lex})| > \theta$ **then**
      $U = U - \{x_i\}$.
      $L = L \cup \{(x_i, H_{M_{pro}}(x_{i,pro}))\}$.
    end if
  end for
  Learn all-features classifier M using L.
end while

Fig. 5. Co-training agreement algorithm.

information as two separate views for the sentence segmentation task. In this paper, we use an extended version of the basic co-training algorithm as shown in Fig. 5. The scheme of our co-training approach is illustrated in Fig. 4. Our co-training approach consists of multiple stages. In the first stage, we train two separate models (M_{pro} and M_{lex}) using only prosodic and only lexical features based on small amounts of manually labeled data L. Then we estimate the sentence boundaries for the unlabeled portion of the data U using these models. The examples x_i are sorted according to their confidence scores for both cases. At this point, we tried different example selection mechanisms to come up with the set of examples from both sides.

- *Agreement*: In this strategy, we consider only the examples that get high confidence scores according to both prosodic and lexical models M_{pro} and M_{lex}. In other words, both classifiers have the same decision, $H_{M_{\text{pro}}}(x_{i,\text{pro}}) = H_{M_{\text{lex}}}(x_{i,\text{lex}})$ and highly confident about the class of the $x_i \in U$. We add these examples to the training set of individual models, L_{pro} and L_{lex} with their agreed labels, and iterate. In this strategy, any unlabeled example is determined with its estimated class by using

$$H_{M_{\text{pro}}}(x_{i,\text{pro}}) = H_{M_{\text{lex}}}(x_{i,\text{lex}})$$

and

$$\left|f_{M_{\text{pro}}}(x_{i,\text{pro}})\right| + \left|f_{M_{\text{lex}}}(x_{i,\text{lex}})\right| > \theta$$

The algorithm of the co-training with the agreement strategy is shown in Fig. 5.

- *Disagreement*: In this strategy, we consider only the examples that are labeled with high confidence scores using one model and low confidence scores using the other model. We add these examples to the training set of the other model. The motivation here is to incorporate new examples that are hard to classify for the other model to its training data. Note that this is equivalent to the *max-t-min-s* method proposed by [18]. In this strategy, the examples whose classes estimated by the prosodic model $H_{M_{\text{pro}}}(x_{i,\text{pro}})$

Initialization:
1. Given a small set, L, of manually labeled examples
$L = \{(x_1, y_1), ..., (x_{|L|}, y_{|L|})\}$ where $x_i = (x_{i,pro}, x_{i,lex})$.
2. Given a large set, U, of unlabeled examples $U = \{(x_1), ..., (x_{|U|})\}$.
Algorithm:
Obtain two sets,
$L_{pro} = \{(x_{1,pro}, y_1), ..., (x_{|L|,pro}, y_{|L|})\}$ and
$L_{lex} = \{(x_{1,lex}, y_1), ..., (x_{|L|,lex}, y_{|L|})\}$ from L.
Obtain two sets, $U_{pro} = U_{lex} = U$.
while $U_{pro} \neq \emptyset$ and $U_{lex} \neq \emptyset$ and held-out data set error rate does not converge/increase **do**
  Learn classifier M_{pro} using L_{pro}.
  Learn classifier M_{lex} using L_{lex}.
  for each $x_i \in U_{lex}$ **do**
    if $|f_{M_{pro}}(x_{i,pro})| - |f_{M_{lex}}(x_{i,lex})| > \theta_1$ **then**
      $U_{lex} = U_{lex} - \{x_i\}$.
      $L_{lex} = L_{lex} \cup \{(x_{i,lex}, H_{M_{pro}}(x_{i,pro}))\}$.
      $L = L \cup \{(x_i, H_{M_{pro}}(x_{i,pro}))\}$.
    end if
  end for
  for each $x_i \in U_{pro}$ **do**
    if $|f_{M_{lex}}(x_{i,lex})| - |f_{M_{pro}}(x_{i,pro})| > \theta_2$ **then**
      $U_{pro} = U_{pro} - \{x_i\}$.
      $L_{pro} = L_{pro} \cup \{(x_{i,pro}, H_{M_{lex}}(x_{i,lex}))\}$.
      $L = L \cup \{(x_i, H_{M_{lex}}(x_{i,lex}))\}$.
    end if
  end for
  Learn all-features classifier M using L.
end while

Fig. 6. Co-training disagreement algorithm.

with a high probability are assigned to the training set of the lexical model L_{lex} when

$$\left|f_{M_{\text{pro}}}(x_{i,\text{pro}})\right| - |f_{M_{\text{lex}}}(x_{i,\text{lex}})| > \theta$$

and vice versa. Note that $f(x=+1) - f(x=-1)$ is closer to 0 when the classifier is less confident about its decision.

This process may be iterated until the unlabeled data set is exploited completely or the models do not improve any more as determined by a held-out set. After that, one can train a single model using both the lexical and prosodic features of the automatically and manually labeled examples to combine the models. The algorithm of the co-training with the disagreement strategy is shown in Fig. 6.

3) Self-Combined: In this approach, we consider the examples that have the highest confidence scores obtained from the self-training of the prosodic and lexical models individually in each iteration. First, as an initialization process, the manually labeled data L that contain both the prosodic and lexical features are separated into two parts, which consist of the prosodic and lexical features with their classes as L_{pro} and L_{lex}. Then we train the prosodic M_{pro} and lexical M_{lex} models by using this small amount of manually labeled data separately, and we obtain the estimated confidence scores $f_{M_{\text{pro}}}(x_{i,\text{pro}})$ and $f_{M_{\text{lex}}}(x_{i,\text{lex}})$ of the larger set of unlabeled prosodic U_{pro} and lexical U_{lex} data by using these two trained models. The examples having the highest confidence scores from both sides are determined by sorting the confidence scores in descending order and employing different thresholds. For example, if the confidence scores of the examples x_i estimated by the prosodic model are bigger than a threshold and the estimated classes agree

$$\left|f_{M_{\text{pro}}}(x_{i,\text{pro}})\right| > \theta_1 \text{ and } H_{M_{\text{pro}}}(x_{i,\text{pro}}) = H_{M_{\text{lex}}}(x_{i,\text{lex}})$$

the estimated classes $H_{M_{\text{pro}}}(x_{i,\text{pro}})$ are assigned to these examples and they are added to labeled examples with their prosodic and lexical features and estimated classes. In the other case, if the confidence scores of the examples x_i estimated by the lexical model are bigger than a threshold and the estimated classes agree

$$|f_{M_{\text{lex}}}(x_{i,\text{lex}})| > \theta_2 \text{ and } H_{M_{\text{lex}}}(x_{i,\text{lex}}) = H_{M_{\text{pro}}}(x_{i,\text{pro}})$$

the estimated classes $H_{M_{\text{lex}}}(x_{i,\text{lex}})$ are assigned to these examples and added to labeled examples with their prosodic and lexical features and estimated classes. It should be noted that, in both cases, the examples labeled with different tags (conflict examples) by the prosodic and lexical classifiers are excluded in each iteration. The whole process is iterated until the unlabeled sets are completely labeled or the models do not improve any more as determined by a held-out set. In this method, in each iteration the prosodic and lexical classifiers are trained individually (self-training based on one view). The performance evaluation is realized on the held-out and test data sets in each iteration after obtaining newly labeled data which contain both the prosodic and lexical features. In this stage, we train a model in each iteration using whole labeled data and experiment on the held-out set and test set. The algorithm of our new semi-supervised learning approach is given in Fig. 7.

IV. Experiments and Results

All experiments are performed using manual transcriptions to avoid the noise introduced by the speech recognition system. The prosodic features are computed using the forced alignments of the manual transcriptions. We perform experiments using different sizes of initial manually labeled data and compare different semi-supervised learning methods. In our experiments, we used the BoosTexter tool (described in [26]) as a classifier. For all experiments we iterated Boosting 500 times.

A. Data Sets

The ICSI meeting corpus [4] contains approximately 72 h of multichannel conversational speech data. Generally, for sentence segmentation experiments 73 out of the total 75 available meetings (two meetings are excluded because of their very different character from the rest of the data) are used. The 73 meetings are split into a training set (51 meetings, approximately 539K words), a development set (11 meetings, approximately 110K words), and a test set (11 meetings, approximately 102K words). In our experiments, we use the lexical and prosodic features of the ICSI Meeting Recorder Dialog Act (MRDA) Corpus. We use 51 meetings, which have in total 538 956 examples with prosodic and lexical features, as training data. We use three different random orderings of the training set to get different feature distributions and remove the biasing effect in the evaluation stage. Then we report the average performance. In addition to this, both the development and test sets consist of

Initialization:

1. Given a small set, L, of manually labeled examples $L = \{(x_1, y_1), ..., (x_{|L|}, y_{|L|})\}$ where $x_i = (x_{i,pro}, x_{i,lex})$.
2. Given a large set, U, of unlabeled examples $U = \{(x_1), ..., (x_{|U|})\}$.

Algorithm:

Obtain two sets, $U_{pro} = U_{lex} = U$.
while $U_{pro} \neq \emptyset$ and $U_{lex} \neq \emptyset$ and held-out data set error rate does not converge/increase **do**
  Obtain two sets, $L_{pro} = \{(x_{1,pro}, y_1), ..., (x_{|L|,pro}, y_{|L|})\}$ and $L_{lex} = \{(x_{1,lex}, y_1), ..., (x_{|L|,lex}, y_{|L|})\}$ from L.
  Learn classifier M_{pro} using L_{pro}.
  Learn classifier M_{lex} using L_{lex}.
  for each $x_i \in U_{pro}$ **do**
    if $(|f_{M_{pro}}(x_{i,pro})| > \theta_1$ and $H_{M_{pro}}(x_{i,pro}) = H_{M_{lex}}(x_{i,lex}))$ **then**
      $U_{pro} = U_{pro} - \{x_i\}$.
      $U_{lex} = U_{lex} - \{x_i\}$.
      $L_{pro} = L_{pro} \cup \{(x_{i,pro}, H_{M_{pro}}(x_{i,pro}))\}$.
      $L = L \cup \{(x_i, H_{M_{pro}}(x_{i,pro}))\}$.
    end if
  end for
  for each $x_{i,lex} \in U_{lex}$ **do**
    if $(|f_{M_{lex}}(x_{i,lex})| > \theta_2$ and $H_{M_{lex}}(x_{i,lex}) = H_{M_{pro}}(x_{i,pro}))$ **then**
      $U_{lex} = U_{lex} - \{x_i\}$.
      $U_{pro} = U_{pro} - \{x_i\}$.
      $L_{lex} = L_{lex} \cup \{(x_i, H_{M_{lex}}(x_{i,lex}))\}$.
      $L = L \cup \{(x_i, H_{M_{lex}}(x_{i,lex}))\}$.
    end if
  end for
  Learn all-features classifier M from L.
end while

Fig. 7. Self-combined algorithm.

11 meetings and they have 110 851 and 101 510 examples, respectively. The test and development sets are kept the same for all the experiments.

B. Evaluation Metrics

For the sentence segmentation, performance of the baseline and the semi-supervised methods is evaluated by the F-measure and the NIST error rate metrics. The F-measure, which is often used in information retrieval and natural language processing, is the weighted harmonic mean of the precision and recall measures for the classes hypothesized by the classifier to those assigned by human labelers. The NIST error rate is the ratio of the number of insertion and deletion errors for sentence boundaries made by the classifier to the number of reference sentence boundary classes. Therefore, if no boundaries are marked by sentence segmentation, it is 100%, but it can exceed 100%; the maximum error rate is the ratio of number of words to number of correct boundaries.

C. Experimental Results

In the experiments we use three different sizes of initial manually labeled data, which have 1000, 3000, and 6000 examples, respectively. For each amount of manually labeled data points, we repeat all the experiments by adding different amounts of automatically labeled data such as 100, 250, 500, 1000, 2000, 3000, and 5000 examples in the following iterations. The total number of iterations used in the experiments is 25. In each iteration, we pick the number of automatically labeled examples that result in maximum performance on the development set.

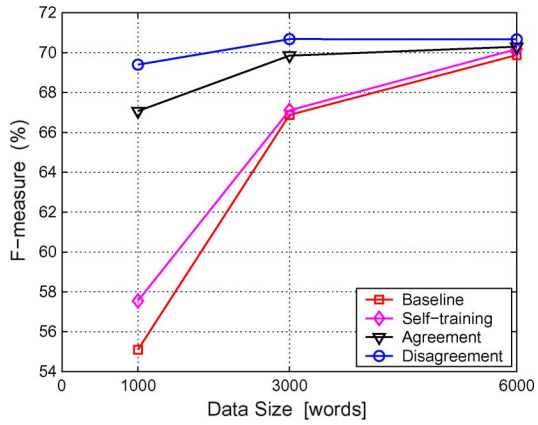


Fig. 8. F-measure results of the different strategies for the lexical features only.

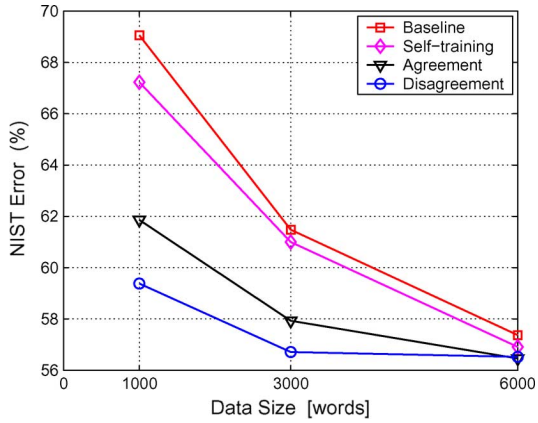


Fig. 9. NIST results of the different strategies for the lexical features only.

Performance of each method is evaluated in two phases. First, all the results of the different semi-supervised strategies are obtained by using only the lexical and only the prosodic models and compared with a baseline. The baseline result in each of these experiments is computed using only the set of manually labeled examples with supervised learning.

Figs. 8–11 present the results using self-training and the other co-training strategies (agreement and disagreement) against the baseline. The curves show performance improvement on the individual lexical and prosodic models when different sizes of initial manually labeled data are used. These figures show that co-training methods, especially the disagreement strategy, improve the results of the baseline significantly, especially when a lesser amount of labeled data is available. With only 1000 manually labeled examples, performance of the lexical model increases from 55.10% to 69.40%, an improvement of 25.95% relative by using the disagreement strategy (Fig. 8). Performance of the prosodic model increases from 58.42% to 64.61%, an improvement of 10.59% relative by using the disagreement strategy (Fig. 10).

The other performance evaluation was done by using the lexical and prosodic models together. In this case, the examples with all the lexical and prosodic features combined are used in the implementation of the methods. Figs. 12 and 13 illustrate complete results of different strategies using all the features. The highest performance improvement was obtained with

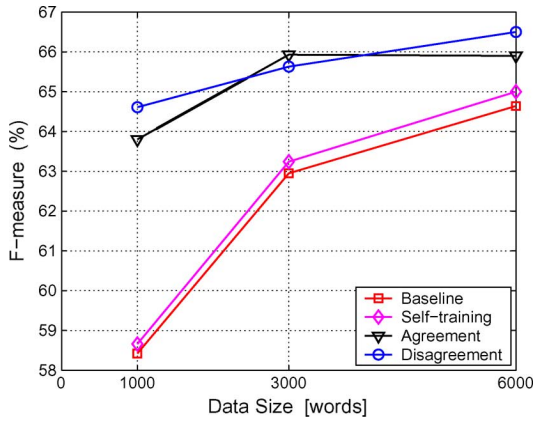


Fig. 10. F-measure results of the different strategies for the prosodic features only.

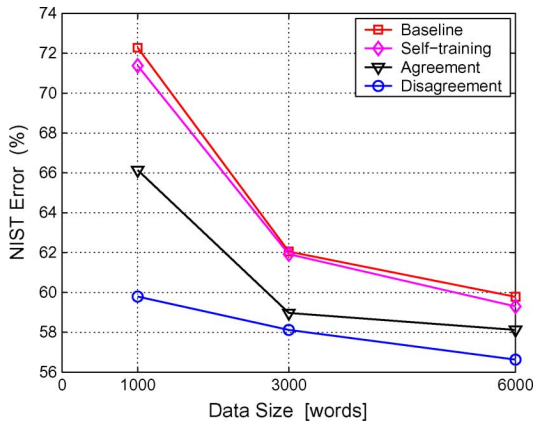


Fig. 11. NIST results of the different strategies for the prosodic features only.

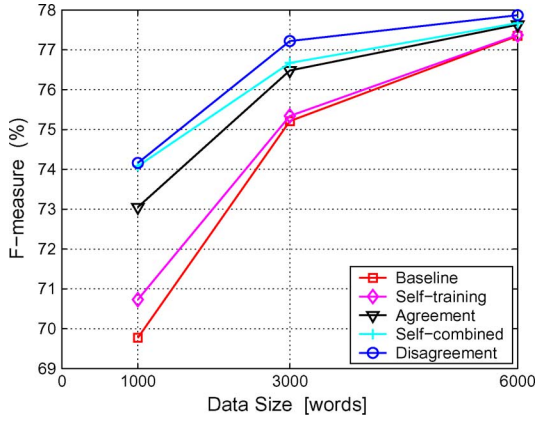


Fig. 12. F-measure results of the different strategies using all the features.

the disagreement strategy. F-measure of the baseline increases from 69.77% to 74.16%, an improvement of 6.29% relative by using the disagreement strategy, and the NIST error drops from 52.00% to 48.34%, a relative reduction of 7%. As seen, the disagreement strategy is slightly better than the agreement strategy. This behavior can be explained by reasoning that if both models are confident about an example it is relatively less informative. It is impressive that the co-training strategies significantly outperform self-training, which provides slight improvement over the baseline.

To expose the effect of the iterative process we use the combination of both lexical and prosodic features on the agreement, disagreement, self-training, and self-combined methods. We also compute the F-measure and NIST error results of the methods on the test set starting at the first iteration to the optimum number of iterations that maximize the development set. The effect of the iterative process is shown in Table I when only 1000, 3000, and 6000 manually labeled data examples are available. The table shows the maximum performance difference between the first iteration and the optimum number of iterations observed in the disagreement strategy when a small amount of manually labeled data is available. The improvement of the effect of the iterative process decreases relatively when the amount of manually labeled data is increased. The proposed self-combined method typically results in better performance than self-training and co-training with the agreement strategy, but is still worse than co-training with the disagreement strategy.

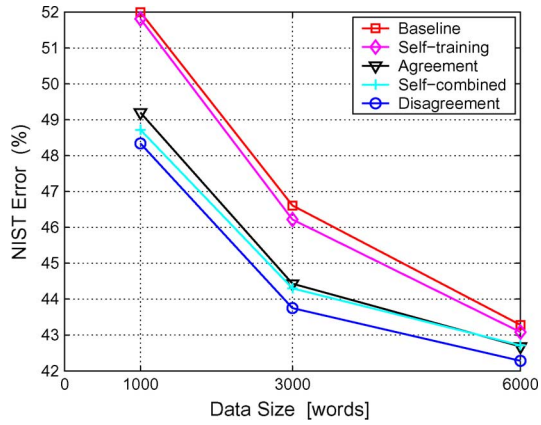


Fig. 13. NIST results of the different strategies using all the features.

TABLE I
EFFECT OF THE ITERATIVE PROCESS ON THE SEMI-SUPERVISED LEARNING METHODS WHEN ONLY 1000, 3000, AND 6000 MANUALLY LABELED EXAMPLES ARE AVAILABLE

Man. labeled data=1,000	*At first iteration*		*At opt. num. of iterations*	
	F-measure(%)	*NIST(%)*	*F-measure(%)*	*NIST(%)*
Baseline	69.77	52.00	69.77	52.00
Self-training	70.27	52.55	70.73	51.81
Agreement	70.45	51.49	73.05	49.20
Disagreement	70.53	51.43	**74.16**	**48.34**
Self-combined	**71.19**	**51.29**	74.08	48.72
Man. labeled data=3,000	*At first iteration*		*At opt. num. of iterations*	
	F-measure(%)	*NIST(%)*	*F-measure(%)*	*NIST(%)*
Baseline	75.21	46.61	75.21	46.61
Self-training	74.79	47.03	75.34	46.22
Agreement	75.09	46.36	76.48	44.43
Disagreement	75.32	**45.59**	**77.22**	**43.75**
Self-combined	**75.55**	45.91	76.67	44.30
Man. labeled data=6,000	*At first iteration*		*At opt. num. of iterations*	
	F-measure(%)	*NIST(%)*	*F-measure(%)*	*NIST(%)*
Baseline	77.35	43.28	77.35	43.28
Self-training	77.32	43.26	77.37	43.08
Agreement	77.07	43.65	77.63	42.68
Disagreement	**77.19**	**43.18**	**77.87**	**42.28**
Self-combined	77.05	43.33	77.67	42.71

D. Analysis and Discussion

An interesting question is whether there is a pattern for the optimal number of iterations for each co-training strategy. Table II

TABLE II
AVERAGE NUMBER OF ITERATIONS AND AMOUNT OF AUTOMATICALLY ANNOTATED DATA EXPLOITED WITH THE SEMI-SUPERVISED LEARNING METHODS WHEN ONLY 1000, 3000, AND 6000 MANUALLY LABELED EXAMPLES ARE AVAILABLE

Man. labeled data=1,000	*At first iteration*	*At opt. num. of iterations*	
	Avg. # Added	*Avg. # Iters*	*Avg. # Added*
Baseline	0	0	0
Self-training	3,783	3.33	8,366
Agreement	5,500	5.33	16,166
Disagreement	2,233	5.66	12,166
Self-combined	3,516	7.00	14,866
Man. labeled data=3,000	*At first iteration*	*At opt. num. of iterations*	
	Avg. # Added	*Avg. # Iters*	*Avg. # Added*
Baseline	0	0	0
Self-training	1,833	9.50	39,466
Agreement	4,000	11.66	23,000
Disagreement	5,116	2.66	16,666
Self-combined	7,166	4.33	26,166
Man. labeled data=6,000	*At first iteration*	*At opt. num. of iterations*	
	Avg. # Added	*Avg. # Iters*	*Avg. # Added*
Baseline	0	0	0
Self-training	400	6.50	39,883
Agreement	2,333	7.50	17,500
Disagreement	4,166	8.00	17,833
Self-combined	5,750	8.50	25,000

presents the average of the optimum iteration number and the amount of unlabeled data exploited for each co-training strategy corresponding to Table I. As seen, the average number of iterations are typically in single digits, and it is hard to find any correlation between the initial data sizes and semi-supervised learning techniques employed. We observe a relatively steep slope during the iterative process and later the performances have phased out as more number of less informative samples are introduced.

When we examine the errors of the lexical model when the prosodic model correctly classifies the example, we see that the prosodic model is very strong especially in the cases of disruptions and disfluencies. When the speaker rephrases the sentences or the speaker is interrupted by another speaker the prosodic model marks these examples as a sentence boundary while the lexical model may not.

Consider the following example excerpt from two speakers, where the first speaker is interrupted by the second one. The lexical model does not expect a new dialog act unit start after the word *you*, whereas the prosodic model may capture it. "(SB)" indicates a sentential unit boundary.

Speaker 1: if you (SB).
Speaker 2: great (SB).
Speaker 1: okay so that will get us through the next couple days (SB).

Another frequent case of conflict between two models occur around disfluencies such as self repetitions or repairs. In such cases the lexical model interprets these examples as a part of continuous speech and cannot detect the boundary. The correct decision is made by the prosodic model. Below are some examples of such cases.

Speaker: that's probably the (SB) this is probably channel error stuff (SB) huh (SB).
Speaker: I mean you're (SB) I think you (SB) I think you'd have to modify the standard deviation or something so that you make it wider or narrower (SB).
Speaker: are you using (SB) are you using the (SB) oh you oh you are (SB) we already talked about that (SB).

In our experiments, if we examine the other way around, we observe that the prosodic model sometimes misses the boundaries before conjunctive words and phrases such as "but," "so," and "because." These results are also verified by the audio data. In some of these examples the boundaries are not prosodically marked, whereas the lexical model usually marks them correctly as sentence boundaries.

Speaker: okay (SB) but . . .

V. CONCLUSION

We have investigated the application of the multi-view semi-supervised learning algorithms on the sentence boundary classification problem by using lexical and prosodic information. The experimental results on the ICSI MRDA corpus show the effectiveness of these algorithms for the task of sentence segmentation. Performance of the lexical and prosodic models is improved by 25.95% and 10.59% relative, respectively, when only a small set of manually labeled examples is used. When both information sources are combined, the semi-supervised learning methods improve the baseline F-Measure of 69.8% to 74.2%.

Our future work includes employing cross-adaptation methods instead of simply concatenating the data to improve performance. The classifiers trained with lexical and prosodic features can be treated as a committee of classifiers, and can be used for committee-based active learning. We also plan to investigate the application of committee-based active learning for this task and combine with co-training. Furthermore, we plan to experiment with speech recognition output. A similar approach could also be useful for other tasks that use prosodic and lexical features, such as emotion detection, topic segmentation, and dialog act tagging.

ACKNOWLEDGMENT

The authors would like to thank E. Shriberg, A. Stolcke, B. Favre, M. Zimmerman, and M. Magimai Doss for many helpful discussions.

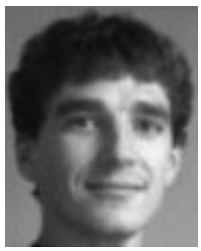
REFERENCES

[1] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tur, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech Commun.*, vol. 32, no. 1–2, pp. 127–154, 2000.
[2] S. Cuendet, D. Hakkani-Tür, and G. Tur, "Model adaptation for sentence segmentation from speech," in *Proc. IEEE/ACL Spoken Lang. Technol. (SLT) Workshop*, Aruba, 2006.
[3] U. Guz, D. Hakkani-Tür, S. Cuendet, and G. Tur, "Co-training using prosodic and lexical information for sentence segmentation," in *Proc. Interspeech*, Antwerp, Belgium, Aug. 2007.
[4] E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey, "The ICSI Meeting Recorder Dialog Act (MRDA) corpus," in *Proc. SIGdial Workshop Discourse and Dialogue*, 2004, pp. 97–100.
[5] J. Kolar, E. Shriberg, and Y. Liu, "Using prosody for automatic sentence segmentation of multi-party meetings," in *Proc. Int. Conf. Text, Speech, Dialogue (TSD)*, Czech Republic, 2006.
[6] J. Ang, Y. Liu, and E. Shriberg, "Automatic dialog act segmentation and classification in multiparty meetings," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Philadelphia, PA, Mar. 2005, pp. 1061–1064.

[7] M. Core and J. Allen, "Coding dialogs with the DAMSL annotation scheme," in *Proc. Working Notes Conf. Amer. Assoc. Artif. Intell. (AAAI) Fall Symp. Commun. Action in Humans Machines*, Cambridge, MA, Nov. 1997.
[8] E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey, "The ICSI Meeting Recorder Dialog Act (MRDA) corpus," in *Proc. SigDial Workshop*, Boston, MA, May 2004.
[9] A. Stolcke and E. Shriberg, "Statistical language modeling for speech disfluencies," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Atlanta, GA, May 1996, pp. 405–408.
[10] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Using conditional random fields for sentence boundary detection in speech," in *Proc. Annu. Meeting Assoc. Comput. Linguist. (ACL)*, Ann Arbor, MI, 2005.
[11] M. Zimmerman, D. Hakkani-Tür, J. Fung, N. Mirghafori, L. Gottlieb, E. Shriberg, and Y. Liu, "The ICSI+ multilingual sentence segmentation system," in *Proc. Int. Conf. Spoken Lang. Process. (ICSLP)*, Pittsburg, PA, 2006.
[12] J. Kolar, Y. Liu, and E. Shriberg, "Speaker adaptation of language models for automatic dialog act segmentation of meetings," in *Proc. Interspeech*, Antwerp, Belgium, 2007.
[13] K. Nigam and R. Ghani, "Understanding the behaviour of co-training," in *Proc. Workshop Text Mining 6th ACM SIGKDD at the KDD*, 2000.
[14] J.-L. Gauvain and C.-H. Lee, "Maximum *a posteriori* estimation for multivariate Gaussian mixture observation of Markov chains," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 291–298, Apr. 1994.
[15] M. Bacchiani and B. Roark, "Unsupervised language model adaptation," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Hong Kong, Apr. 2003, pp. 173–176.
[16] D. Hakkani-Tür, G. Tur, M. Rahim, and G. Riccardi, "Unsupervised and active learning in automatic speech recognition for call classification," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Montreal, QC, Canada, May 2004, pp. 429–432.
[17] R. Gretter and G. Riccardi, "On-line learning of language models with word error probability distributions," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Salt Lake City, UT, May 2001, pp. 557–560.
[18] W. Wang, Z. Huang, and M. Harper, "Semi-supervised leaning for part-of-speech tagging of mandarin transcribed speech," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Honolulu, HI, 2007, pp. 137–140.
[19] R. Mihalcea, "Co-training and self-training for word sense disambiguation," in *Proc. Conf. Comput. Natural Lang. Learn. (CoNLL)*, Boston, MA, May 2004.
[20] D. McClosky, E. Charniak, and M. Johnson, "Effective self-training for parsing," in *Proc. Conf. North Amer. Chapt. Assoc. Comput. Linguist. (NAACL)*, New York, Jul. 2006.
[21] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. Workshop Computational Learning Theory (COLT)*, Madison, WI, Jul. 1998.
[22] T. M. Mitchell, "The role of unlabeled data in supervised learning," in *Proc. 6th Int. Colloquium Cognitive Sci.*, San Sebastian, Spain, 1999.
[23] S. Abney, "Bootstrapping," in *Proc. Annu. Meeting Assoc. Comput. Linguist. (ACL)*, 2002.
[24] S. Kiritchenko and S. Matwin, "Email classification with co-training," in *Centre for Advanced Studies on Collaborative Research (CASCON)*, 2001.
[25] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Proc. MSRI Workshop Nonlinear Estimation and Classification*, Berkeley, CA, Mar. 2001.
[26] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Mach. Learn.*, vol. 39, no. 2/3, pp. 135–168, 2000.

Umit Guz (M'03) received the M.S. and Ph.D. degrees in electronics engineering from the Institute of Science, Istanbul University, in 1997 and 2002, respectively.

He is an Assistant Professor in the Department of Electronics Engineering, Engineering Faculty, Isik University, Istanbul, Turkey. He was awarded a Postdoctoral Research Fellowship by The Scientific and Technological Research Council of Turkey (TUBITAK) in 2006. He was accepted as an International Fellow by the SRI International Speech Technology and Research (STAR) Laboratory in 2006. He was awarded a J. William Fulbright Postdoctoral Research Fellowship for 2007. He was accepted as an International Fellow by the International Computer Science Institute (ICSI) Speech Group at the University of California at Berkeley in 2007 and 2008. His research interest covers speech processing, speech modeling, speech coding, speech compression, automatic speech recognition, natural language processing, and biosignal processing.



Sébastien Cuendet received the B.Sc. and M.Sc. degrees from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2004 and 2006, respectively.

He was a Research Scholar at the International Computer Science Institute (ICSI) when this work was realized. Since August 2007, he has been with Optaros, Zurich, Switzerland, a consulting company based on open source solutions for web technologies.

Dilek Hakkani Tür (M'01–SM'05) received the Ph.D. degree from the Department of Computer Engineering, Bilkent University, Ankara, Turkey, in 2000.

She is a Senior Researcher at the International Computer Science Institute (ICSI), Berkeley, CA. Prior to joining ICSI, she was a Senior Technical Staff Member in the Voice Enabled Services Research Department, AT&T Labs-Research, Florham Park, NJ. She worked on machine translation during her visit at the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, in 1997, and her visit to the Computer Science Department, The Johns Hopkins University, Baltimore, MD, in 1998. In 1998 and 1999, she visited the Speech Technology and Research Laboratory, SRI International Menlo Park, CA, and worked on using lexical and prosodic information for information extraction from speech. She has coauthored more than 100 papers in natural language and speech processing.

Dr. Hakkani Tür is a member of ISCA, the Association for Computational Linguistics, and was an Associate Editor of IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING.

Gokhan Tur (M'01–SM'05) received the B.S., M.S., and Ph.D. degrees from the Department of Computer Science, Bilkent University, Ankara, Turkey, in 1994, 1996, and 2000, respectively.

From 1997 to 1999, he visited the Center for Machine Translation, Carnegie Mellon University, Pittsburgh, PA, then the Department of Computer Science of The Johns Hopkins University, Baltimore, MD, and then the Speech Technology and Research (STAR) Laboratory, SRI International, Menlo Park, CA. He worked at AT&T Labs-Research from 2001 to 2006. He is currently with the Speech Technology and Research Laboratory, SRI International. His research interests include spoken language understanding (SLU), speech and language processing, machine learning, and information retrieval and extraction. He coauthored more than 75 papers published in refereed journals, and presented at international conferences.

Dr. Tur is a senior member the ACL and ISCA, and was a member of the IEEE Signal Processing Society (SPS), Speech and Language Technical Committee (SLTC) for 2006–2008.