Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Speech and Language 20 (2006) 468–494
COMPUTER SPEECH AND LANGUAGE
www.elsevier.com/locate/csl

A study in machine learning from imbalanced data for sentence boundary detection in speech

Yang Liu [a,c,*], Nitesh V. Chawla [b], Mary P. Harper [c], Elizabeth Shriberg [a,d], Andreas Stolcke [a,d]

[a] *Speech Group, International Computer Science Institute, 1947 Center St., Ste 600, Berkeley, CA 94704, USA*
[b] *Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46530, USA*
[c] *Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA*
[d] *SRI International, Menlo Park, CA 94025, USA*

Received 4 August 2004; received in revised form 14 March 2005; accepted 16 June 2005
Available online 13 July 2005

Abstract

Enriching speech recognition output with sentence boundaries improves its human readability and enables further processing by downstream language processing modules. We have constructed a hidden Markov model (HMM) system to detect sentence boundaries that uses both prosodic and textual information. Since there are more nonsentence boundaries than sentence boundaries in the data, the prosody model, which is implemented as a decision tree classifier, must be constructed to effectively learn from the imbalanced data distribution. To address this problem, we investigate a variety of sampling approaches and a bagging scheme. A pilot study was carried out to select methods to apply to the full NIST sentence boundary evaluation task across two corpora (conversational telephone speech and broadcast news speech), using both human transcriptions and recognition output. In the pilot study, when classification error rate is the performance measure, using the original training set achieves the best performance among the sampling methods, and an ensemble of multiple classifiers from different downsampled training sets achieves slightly poorer performance, but has the potential to reduce computational effort. However, when performance is measured using receiver operating characteristics (ROC) or area under the curve (AUC), then the sampling approaches outperform the original training set. This observation is important if the

* Corresponding author. Tel.: +1 510 666 2993; fax: +510 666 2956.
E-mail addresses: yangl@icsi.berkeley.edu (Y. Liu), nchawla@cse.nd.edu (N.V. Chawla), harper@ecn.purdue.edu (M.P. Harper), ees@speech.sri.com (E. Shriberg), stolcke@speech.sri.com (A. Stolcke).

0885-2308/$ - see front matter
doi:10.1016/j.csl.2005.06.002

sentence boundary detection output is used by downstream language processing modules. Bagging was found to significantly improve system performance for each of the sampling methods. The gain from these methods may be diminished when the prosody model is combined with the language model, which is a strong knowledge source for the sentence detection task. The patterns found in the pilot study were replicated in the full NIST evaluation task. The conclusions may be dependent on the task, the classifiers, and the knowledge combination approach.

1. Introduction

Speech recognition technology has improved significantly during the past few decades. However, current automatic speech recognition systems simply output a stream of words. Sentence boundary information is not provided by these systems, and yet this type of information can make the output of the recognizer easier to process for both humans and downstream language processing modules. For example, a sentence-like unit is typically expected by machine translation systems or parsers. The following example shows word transcriptions with and without the annotation of sentence boundaries (marked by '/'). Apparently, without the sentence boundary information, transcriptions are more difficult for humans to read or for language processing modules to process.

```
no what is that I have not heard of that
no / what is that / I have not heard of that /
```

The sentence boundary detection problem can be represented as a classification task, that is, at each interword boundary, we determine whether or not it is a sentence boundary. To detect sentence boundaries, we have constructed a hidden Markov model (HMM) system that uses both prosodic and textual information. The prosodic information is modeled by a decision tree classifier. Because sentence boundaries are less frequent than nonsentence boundaries in the training data, the prosody model needs to be designed to deal with the imbalanced data set distribution. Our previous approach (Shriberg et al., 2000) was to randomly downsample the training set to obtain a balanced data set for decision tree training and adjust the posterior probability estimation from the prosody model on the test set. In this paper, we investigate several sampling approaches to cope with the imbalanced data distribution, as well as a bagging scheme, in an attempt to build more effective classifiers for the prosody model.

We first conduct a pilot study that uses a small training set in order to extensively evaluate all the methods. In this study, human transcriptions will be used to factor out the effect of the speech recognition errors. Then, based on the findings of the pilot study, we will choose the most successful methods to evaluate on the full NIST sentence boundary evaluation task, which involves two genres (conversational telephone speech and broadcast news). To our knowledge, this is the first study on the imbalanced data set problem for the sentence boundary detection task. This study will provide groundwork for future classification tasks related to spoken language processing, such as finding disfluencies in conversational speech (Liu et al., 2003) or hot spots in meetings (Wrede and Shriberg, 2003), where the class distribution is also imbalanced.

This paper is organized as follows. In Section 2, we introduce the sentence boundary detection problem, as well as the data and the evaluation metrics we will use. In Section 3, we describe the HMM approach used for sentence boundary detection and summarize related work. In Section 4, we describe the imbalanced data problem and the methods we will use to address it. Section 5 shows the experimental results from the pilot study. In Section 6, we apply the best techniques from the pilot study to the official NIST sentence boundary detection task across two corpora. Conclusions appear in Section 7.

2. Sentence boundary detection task

2.1. Task representation

The sentence boundary detection problem is represented as a classification task, with the word boundaries being identified as either a sentence boundary or not at each interword boundary. In the training data, sentence boundaries are marked by annotators using the information both in the transcriptions and in the recorded speech. For testing, given a word sequence (from a human transcriptions or speech recognition output) $w_1 w_2 \cdots w_n$ and the speech signal, we use several knowledge sources (in this case, prosodic and textual information) to determine whether or not there should be a sentence boundary at each interword boundary. Fig. 1 shows an example of a waveform, with the corresponding pitch and energy contour, the word alignment, and sentence boundary information.

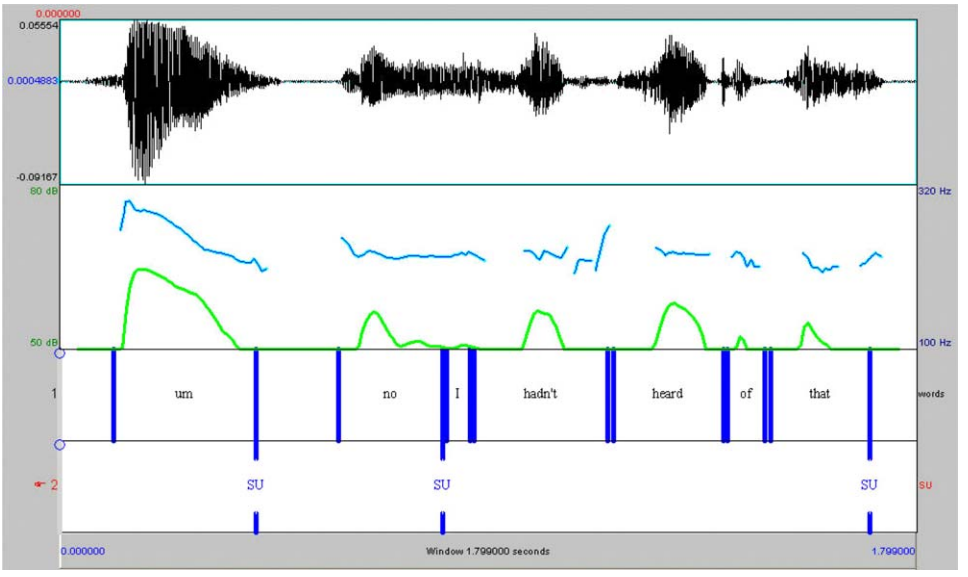

Fig. 1. The waveform, pitch and energy contour, word alignment, and sentence boundaries (denoted "SU") for the utterance "*um no I hadn't heard of that*".

2.2. Data

All data used in this investigation is taken from the official EARS RT-03 evaluation (training set, development set, and evaluation set) (National Institute of Standards and Technology, 2003). The data was annotated with sentence boundaries according to the V5.0 guideline (Strassel, 2003) developed by LDC as part of the DARPA EARS program (DARPA, 2003). Note that in spoken language, a sentence is not as well defined as in written text. In the DARPA EARS program, the sentence-like unit is called an 'SU'; hence, in this paper we will use the term 'SU boundary detection'. The following is an example of utterances annotated with SU information (an SU boundary is marked by a '/'):

```
no / not too often / but this was a just a good
opportunity to go see that movie / it was uh /
```

SUs typically express a speaker's complete thought or idea unless there is some disruption, such as can be seen in the last SU in the above example. An SU can be an entire well-formed sentence, a phrase, or a single word. Sometimes a unit is semantically complete but smaller than a sentence, for example, the second SU in the example above (*not too often*). Sometimes it is hard to judge whether there is an SU boundary at a location. The inter-annotator agreement on SU boundaries is about 93% (Strassel and Walker, 2003).

SU classification performance will be evaluated on two different corpora, conversational telephone speech (CTS) and broadcast news (BN) (National Institute of Standards and Technology, 2003). In conversational telephone speech, participants were paired by a computer-driven "robot operator" system that set up the phone call, selected a topic for discussion from a predefined set of topics, and recorded speech into separate channels until conversation was complete. BN contains news broadcasts, such as from ABC, CNN and CSPAN television networks, and NPR and PRI radio networks. In conversational speech, sentences are generally shorter than in broadcast news. In addition, there are many backchannels that are often one or two words long. The difference in speaking styles between these two corpora is reflected in the SU boundary class distribution in the data set: SU boundaries comprise about 8% of all the interword boundaries in BN, compared to 13% in CTS.

Evaluation is conducted based on two different types of transcriptions: human-generated transcriptions and speech recognition output. Using the reference transcriptions provides the best-case scenario for the algorithms, as they contain many fewer transcription errors.

2.3. Evaluation metrics

Several different performance measure metrics have been used for sentence boundary detection:

- *Classification error rate (CER)*. If the SU detection problem is treated as a classification problem, performance can be easily measured using CER. CER is defined as the number of incorrectly classified samples divided by the total number of samples. For this measure, the samples are all the interword boundaries.

- *F-measure*. The F-measure is defined as

$$\mathrm{F-measure} = \frac{(1+\beta^2) * \mathrm{recall} * \mathrm{precision}}{\beta^2 * \mathrm{recall} + \mathrm{precision}} \quad (1)$$

where precision $= \frac{\mathrm{TP}}{\mathrm{TP+FP}}$, recall $= \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, and TP and FP denote the number of true positives and false positives, respectively. FN represents the number of false negatives and β corresponds to the relative importance of precision versus recall. β is set to 1 if false alarms and misses are considered equally costly.[1] In this measure, the minority class (SU boundaries) is the positive class.

- *ROC and AUC*. Receiver operating characteristics (ROC) curves (Hand, 1997; Bradley, 1997) can be used to enable visual judgments of the trade-off between true positives and false positives for a classification or detection task. Depending on the application, an appropriate operating point from the ROC curve can be selected (Duda and Hart, 1973).[2] The area under the curve (AUC) can tell us whether a randomly chosen majority class example has a higher majority class membership than a randomly chosen minority class example; thus, it can provide insight on the ranking of the positive class (SU) examples.
- *SU error rate*. In the DARPA EARS program, SU detection is measured somewhat differently from the metrics above. The errors are the number of misclassified points (missed and falsely detected points) per reference SU. When recognition transcriptions are used for SU detection, scoring tools first align the reference and hypothesis words, map the hypothesized SU events to the reference SU events, and then calculate the errors. When recognition output words do not align perfectly with those in the reference transcriptions, an alignment that minimizes the word error rate (WER) is used. See http://www.nist.gov/speech/tests/rt/rt2003/fall/mdex.htm for more details on the NIST scoring metric.The SU error rate and the classification error rate tend to be highly correlated. When a reference transcription is used, the SU errors in NIST's metric correspond most directly to classification errors. The major difference lies in the denominator. The number of reference SUs is used in the NIST scoring tools; whereas, the total number of word boundaries is used in the CER measure. In the NIT scoring scheme, the SU is the unit used in the mapping step, yet in the boundary-based metric each interword boundary is the unit for scoring. It is not well defined how to map SU boundaries when recognition output is used, since there may be an imperfect alignment to the reference transcriptions (especially when there are inserted or deleted words). Hence, the boundary-based metric (i.e., classification error rate) is used only in the reference transcription condition, and not for the recognition output.The SU error rate in the NIST metric can be greater than 100%. The following example shows a system SU hypothesis aligned with the reference SUs:

```
Reference:   w1          w2   w3   /    w4
system:      w1   /      w2   w3        w4
                  ins                 del
```

where w_i is a word and '/' indicates an SU boundary. Two boundaries are misclassified: one insertion error and one deletion error (indicated by 'ins' and 'del') in the example

[1] More studies are needed to see whether missing an SU boundary or inserting an SU boundary has the same impact on human understanding or language processing modules.

[2] For the SU boundary detection task, a threshold is selected to minimize the overall classification error rate.

above. Since there is only one reference SU boundary, the NIST SU error rate for this system output is 200%; whereas, since there are four word boundaries, among which two are misclassified, the CER is 2/4 = 50%.

Note that the CER, F-measure, and the SU error rate all correspond to a single operating point; whereas, the ROC analysis covers a set of operating points. Also note that a baseline result can be obtained when the majority class (non-SU boundary) is hypothesized at each interword boundary. In this case, the CER is equal to the prior probability of the SU class, the NIST SU error rate is 100% (all due to deletion errors), the recall is 0, and the F-measure is not well defined.

Currently, there exist no standard tests for significance using the NIST scoring method. The problem is that the metric is not based on a consistent segment (Ostendorf and Hillard, 2004). For the CER metric, the sign test can be utilized to test significance at the word boundary level, which we will report when appropriate. Designing a significance test for the NIST metric is beyond the scope of this work; however, it is likely that the results found to be significant using the sign test on CER will also be significant using a significance test designed for the NIST metric.

3. SU boundary detection approach

3.1. Description of the HMM approach

To detect an SU boundary in speech, we will employ several knowledge sources, in particular textual and prosodic cues, to reduce the ambiguity inherent in each type of knowledge alone. By "textual information", we mean the information obtained from the word strings in the transcriptions generated either by a human or by an automatic speech recognition system. Textual information is no doubt very important. For example, words like "*I*" often start a new SU. However, since textual information may be ambiguous, prosody can provide additional, potentially disambiguating cues for determining where an SU boundary is.

In this paper, we focus on the HMM approach used for the SU detection task, which builds upon the prior work of Shriberg et al. (2000). There are three components: the prosody model, the hidden event language model, and the HMM method for combining these two models. Each component is described in the following. We have investigated other approaches for combining knowledge sources, such as using maximum entropy modeling (Liu et al., 2004) and conditional random fields (Liu et al., 2004); however, these efforts are orthogonal to the issues addressed in this paper, that is, the imbalanced data problem in the prosody model.

All our models (i.e., the prosody model and the language model) are trained using human transcriptions and the SU boundary annotations associated with them. When evaluating on the recognition output condition, there is a potential mismatch between the model training and testing conditions. However, to train the model using recognition output would require running recognition on all the training data (which is computationally expensive), and mapping the SU annotation available in human transcriptions to the errorful speech recognizer output (which is imperfect). In addition, our earlier experiments on similar tasks have shown that such a mismatch does not degrade performance.

3.1.1. The prosody model

The speech signal contains more information than can be represented in a traditional word transcription. Prosody the "rhythm" and "melody" of speech, is an important knowledge source for detecting structural information imposed on speech. Past research results suggest that speakers use prosody to convey structure in both spontaneous and read speech (Shriberg et al., 2000; Campbell, 1993; Lickley and Bard, 1996; De Pijper and Sanderman, 1994; Hirst, 1993; Price et al., 1991; Potisuk, 1995; Scott, 1982; Swerts, 1997; Nakatani and Hirschberg, 1994; Kompe, 1996). Examples of important prosodic indicators include pause duration, change in pitch range and amplitude, global pitch declination, vowel duration lengthening, and speaking rate variation. Because these features provide information complementary to the word sequence, they are an additional potentially valuable source of information for SU boundary detection. In addition, prosodic cues can be selected to be independent of word identity and thus may be more robust in the face of recognition errors.

At each word boundary, we extract a vector of prosodic features based on the word- and phone-level alignment information (Shriberg et al., 2000). These features reflect information concerning timing, intonation, and energy contours of the speech surrounding the boundary, as shown in Fig. 1. Features such as word duration, pause duration, and phone-level duration are normalized by the overall phone duration statistics and speaker-specific statistics. To obtain F0 features, pitch tracks are extracted from the speech signal and then post-processed to obtain stylized pitch contours (Sonmez et al., 1998) from which F0 features are extracted. Examples of F0 features are the distance from the average pitch in the word to the speaker's pitch floor and the difference of the average stylized pitch across a word boundary. Energy features are extracted from the stylized energy values that are determined in a way similar to F0 features. Some nonprosodic information is included, too, such as speaker gender and turn change. Note that the prosodic features could be very noisy, because, for example, the forced alignment may provide the wrong location for a word boundary, or the pitch extraction may be imperfect. Table 1 shows examples of the prosodic features we use. A full list of all the features in our prosodic feature set, 101 in total, can be found in Ferrer (2002).

The prosodic feature set includes both continuous features (e.g., pause duration after a word) and categorical features (e.g., whether there is a speaker change or whether the pitch contour is falling or rising before a boundary point). In addition, some feature values may be missing. For example, several features are based on F0 estimation, but F0 does not exist in unvoiced

Table 1
Examples of the prosodic features used for the SU detection problem

Prosodic feature	Description
PAU_DUR	Pause duration after a word
LAST_VOW_DUR_Z_bin	Binned normalized duration of the last vowel in the word
WORD_DUR	Word duration
PREV_PAU_DUR	Pause duration before the word
STR_RHYME_DUR_PH_bin	Binned normalized duration of the stressed rhyme in the word
TURN_F	Whether there is a turn change after the word
FOK_INWRD_DIFF	The log ratio of the first and the last stylized F0 value for the word

The "word" in the feature description is the word just before the boundary.

regions; hence, the F0-related features are missing for some samples. Another crucial aspect of the prosodic features is that they are highly correlated. These properties make the prosodic features difficult to model directly in the generative HMM approach, which will be described later. Hence, we have chosen to use a modular approach: the prosodic information is modeled separately by a prosody model classifier.

The goal of the prosody model in the SU detection task is to determine the class membership (SU and not-SU) for each word boundary using the prosodic features. We choose a decision tree classifier to implement the prosody model for several reasons. First, a decision tree classifier offers the distinct advantage of interpretability. This is crucial for obtaining a better understanding of how prosodic features are used to signal an SU boundary. Second, our preliminary studies have shown that the decision tree performs as well as other classifiers, such as neural networks, Bayes classifies, or mixture models. Third, the decision tree classifier can handle missing feature values, as well as both continuous and categorical features. Fourth, the decision tree can produce probability estimates that can be easily combined with a language model, which is very important since the textual knowledge source plays a key role for the SU detection task.

During training, the decision tree algorithm selects a single feature that has the highest predictive value at each node, that is, reduces entropy the most, for the classification task in question. Various pruning techniques are commonly employed to avoid overfitting the model to the training data. We use the CART algorithm for learning decision trees and the cost-complexity pruning approach, both of which are implemented in the IND package (Buntime and Caruana, 1992). During testing, the decision tree generates probabilistic estimates based on the class distribution at the leaves given the prosodic features. These probabilities are representative of each class at a leaf given the prosodic features. An example of a decision tree is shown in Fig. 2. The features used in this tree are described in Table 1.

Since the decision tree learning algorithm can be inductively biased toward the majority class (in this case to non-SU boundaries), the minority class may not be well modeled. Hence, in prior work (Shriberg et al., 2000), we have randomly downsampled our training data to address the

```
PAU_DUR < 8:  0.7123 0.2877 0
   LAST_VOW_DUR_Z_bin < 0.1:  0.8139 0.1861 0
      WORD_DUR < 36.5:  0.8491 0.1509 0
         PREV_PAU_DUR < 152:  0.8716 0.1284 0
            TURN_F = 0:  0.8766 0.1234 0
            TURN_F = T:  0.007246 0.9928 S
         PREV_PAU_DUR >= 152:  0.3054 0.6946 S
            STR_RHYME_DUR_PH_bin < 7.75:  0.5474 0.4526 0
            STR_RHYME_DUR_PH_bin >= 7.75:  0.1349 0.8651 S
      WORD_DUR >= 36.5:  0.544 0.456 0
         FOK_INWRD_DIFF < 0.49861:  0.5554 0.4446 0
         FOK_INWRD_DIFF >= 0.49861:  0.3454 0.6546 S
   LAST_VOW_DUR_Z_bin >= 0.1:  0.4927 0.5073 S
PAU_DUR >= 8:  0.1516 0.8484 S
```

Fig. 2. An example of a decision tree for SU detection. Each line represents a node in the tree, "with the associated question regarding one particular prosodic feature, the class distribution, and the most likely class among the examples going through this node (S for SU boundary, and 0 for non-SU boundary). The indentation represents the depth of the nodes in the decision tree. The features used in this tree are described in Table 1.

skewed class distribution, in order to allow the decision trees to learn the inherent properties for the SU boundaries.

Intuitively, this can make the model more sensitive to the minority class (i.e., SU boundaries in this task) so that it can learn about features predictive of this class. A potential problem with this approach is that many majority class samples (i.e., non-SU boundaries) are not used for model training, and thus downsampling may actually degrade performance. We will investigate several other methods to address the skewed class distribution problem in this paper.

3.1.2. The language model (LM)

For SU boundary detection, the goal of the LM is to model the structural information contained in a word sequence. Our approach uses a hidden event LM (Stolcke and Shriberg, 1996) that models the joint distribution of boundary types and words. Let W represent the string of spoken words, $w_1, w_2, \ldots,$ and E represent the sequence of interword events, $e_1, e_2, \ldots,$ The hidden event language model describes the joint distribution of words and events, $P(W, E) = P(w_1, e_1, w_2, e_2, \ldots w_n, e_n)$.

For training such a hidden event LM, the hand-labeled data (in this case the data annotated by LDC) is used such that an SU boundary event is represented by an additional nonword token ($\langle$SU$\rangle$) that is explicitly included in the training set, for example:

```
no ⟨SU⟩ what is that ⟨SU⟩ I have not heard of that ⟨SU⟩
```

The event "SU" is an additional token in the vocabulary that is inserted in the word sequence for LM training. This modeling approach is shown to be better than modeling the word sequence W and the event sequence E separately using two N-grain LMs (Liu, 2004). Note that we do not explicitly include the "non-SU" event hi the word sequence in order to make more effective use of the contextual information to avoid fragmenting the training data.

During testing (using the LM only), an HMM approach is employed, in which the word/event pairs correspond to states and the words to observations, with the transition probabilities given by the hidden event N-gram model. Given a word sequence W, a forward–backward dynamic programming algorithm (Rabiner and Juang, 1986) is used to compute the posterior probability $P(E_i|W)$ of an event E_i at position i. For our boundary detection task, we choose the event sequence $\widehat{E}$ that maximizes the posterior probability $P(E_i|W)$ at each individual boundary. This approach minimizes the expected per-boundary classification error rate.

3.1.3. Model combination using HMM

Because prosodic and textual cues provide complementary types of information, the combination of the models should yield superior performance over each model alone, as was found in Shriberg et al. (2000). Posterior probabilities at an interword boundary can be determined from both the prosody model and the hidden event LM. A simple combination of the models can be obtained by linearly interpolating the posterior probabilities from each model, with the interpolation weight set based on the held-out set.

Another approach is to more tightly integrate the two models within an HMM framework which has been found to perform better than linear interpolation (Shriberg et al., 2000). An integrated HMM approach models the joint distribution $P(W, F, E)$ of word sequence W, prosodic

features F, and the hidden event types E in a Markov model. The goal of this approach is to find the event sequence $\widehat{E}$ that maximizes the posterior probability $P(E|W,F)$:

$$\widehat{E} = \arg\max_E P(E|W,F) = \arg\max_E P(W,E,F). \tag{2}$$

Prosodic features are modeled as emissions from the hidden states with likelihood $P(F_i|E_i, W)$, where F_i corresponds to the prosodic features for an event boundary E_i at location i. Under the assumption that prosodic observations are conditionally independent of each other given the event type E_i and the words W, we can rewrite $P(W,E,F)$ as follows:

$$P(W,E,F) = P(W,E)\prod_i P(F_i|E_i,W). \tag{3}$$

Note that even though we use $P(F_i|E_i, W)$ in Eq. (3), prosodic observations depend only on the phonetic alignment (denoted by W_t), ignoring word identity. Word identity information is directly captured only in the hidden event LM term $P(W,E)$. This may make prosodic features more robust to recognition errors. Eq. (3) can therefore be rewritten using only the phonetic alignment information W_t for the second term:

$$P(W,E,F) = P(W,E)\prod_i P(F_i|E_i,W_t). \tag{4}$$

An estimation of $P(F_i|E_i, W_t)$ can be obtained from the decision tree class posterior probabilities $P_{\mathrm{DT}}(E_i|F_i, W_t)$:

$$P(F_i|E_i,W_t) = \frac{P(F_i|W_t)P_{\mathrm{DT}}(E_i|F_i,W_t)}{P(E_i|W_t)} \approx \frac{P(F_i|W_t)P_{\mathrm{DT}}(E_i|F_i,W_t)}{P(E_i)}. \tag{5}$$

We approximate $P(E_i|W_t)$ as $P(E_i)$ above, given the fact that W_t contains only the alignment information. The first term in the numerator, $P(F_i|W_t)$, is independent of E and can thus be ignored when substituting Eq. (5) into Eq. (4) and then Eq. (2). Hence, we obtain the following expression for the most likely event sequence, using the hidden event LM $P(W,E)$, the decision tree estimation $P_{\mathrm{DT}}(E_i|F_i, W_t)$, and the prior probabilities of the events $P(E_i)$:

$$\widehat{E} = \arg\max_E P(E|W,F) \approx \arg\max_E P(W,E)\prod_i \frac{P_{\mathrm{DT}}(E_i|F_i,W_t)}{P(E_i)}. \tag{6}$$

What remains is to explain how $P_{\mathrm{DT}}(E_i|F_i, W_t)$ is calculated during testing. We know that the decision tree prosody model can generate the posterior probabilities of the class membership for the test samples. However, if there is a mismatch between the class distributions in the training and test sets, the posterior probabilities may need to be adjusted accordingly (Bishop, 1995). For a classification problem, the posterior probability of the class membership C_K for a sample X can be expressed using Bayes's theorem:

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)}. \tag{7}$$

If training and test sets differ significantly in class distribution, then it is appropriate to use Bayes's theorem to make necessary corrections in the posterior probabilities for the test set. This can be done by dividing the output posterior probabilities from the classifier by the prior probabilities of

the classes corresponding to the training set, multiplying them by the new prior probabilities of the classes in the test set,[3] and then normalizing the results. For example, if we sample the training set to obtain a balanced training set for training the prosodic classifies, the various event classes all have the same prior probability in the new training set. When using this decision tree on the original test set (i.e., without sampling), the posterior probability estimation from the decision tree needs to be adjusted (by multiplying by the prior probabilities of the classes in the original training set and then normalizing). Equivalently, the class priors can be canceled out with the denominator term $P(E_i)$ in Eq. (6) in this case. Obtaining good posterior probability estimates is an important factor that we need to consider in order to combine the prosody model with the LM properly.

Notice that the formulas above are derived to obtain the most likely event sequence $\widehat{E}$. As mentioned earlier, we use a forward–backward algorithm to find the most likely event for each interword location, rather than using the Viterbi algorithm to determine the most likely SU event sequence.

3.2. Related work on sentence boundary detection

Some research has been done on sentence boundary detection in text (Schmid, 2000; Palmer and Hearst, 1994; Reynar and Ratnaparkhi, 1997). However, that task is to identify sentence boundaries in text where punctuation is available; hence, the problem is effectively reduced to deciding which symbols that potentially denote sentence boundaries (periods, question marks, and exclamation marks) actually do. For example, in "*I watch C. N. N.*" only the final period denotes the end of a sentence. When dealing with spoken language, there is no punctuation information, the words are not capitalized, and the transcriptions from the recognition output are errorful. This lack of punctuation in speech is partly compensated for by prosodic information (timing, pitch, and energy patterns).

In the prior work on detecting SU boundaries (or punctuation) in speech, some approaches do not use prosodic information and focus only on textual information (Beeferman et al., 1998; Stevenson and Gaizauskas, 2000). There are also systems that use only the prosodic features (Wang and Narayanan, 2004). Other approaches combine prosodic information and textual information to find SUs and their subtypes (Shriberg et al., 2000; Chen, 1999; Gotoh and Renals, 2000; Kim and Woodland, 2001; Christensen et al., 2001; Huang and Zweig, 2002). Chen (1999) treated punctuation marks as words in the dictionary, with acoustic baseforms of silence, breath, and other nonspeech sounds, and he also modified the language model to include punctuation. Gotoh and Renals (2000) combine a pause duration model and a language model. Shriberg et al. (2000) and Kim and Woodland (2001) use a much richer prosodic feature set to train a prosodic decision tree classifier and combine it with a language model for SU and punctuation detection. Christensen et al. (2001) also investigated using a multilayer perceptron to model the prosodic features. Huang and Zweig (2002) developed a maximum entropy based method to add punctuation into transcriptions, combining textual information and pause information. Most of the prior studies evaluate only on the reference transcriptions, and not on the speech recognition output. It is hard

[3] Although in reality the distribution for the test set is not available, it can be estimated based on the distribution in the original nonresampled training set.

to compare the results of the above approaches to each other as they were obtained under different conditions (different corpora and different transcriptions) and used different performance metrics.

In the DARPA EARS program, all the SU detection systems participating in the DARPA RT-03F Metadata Extraction evaluation (National Institute of Standards and Technology, 2003) were based on an HMM framework, in which word/tag sequences are modeled by N-gram language models. Additional prosodic features are modeled as observation likelihoods attached to the N-gram states of the HMM, similar to the approach we described in Section 3.1.3. Because these approaches all used the same corpora and metrics, these efforts can be compared to each other and to the work in this paper.

Prior work has found that textual cues are a valuable knowledge source for determining SU boundary or punctuation in speech, and that prosody provides additional important information for spoken language. Most of these efforts have focused on combining multiple knowledge sources (either features themselves or the combination approach), and used either the downsampled training set or the original training set in the prosody model; however, none of the previous studies have systematically investigated the imbalanced data problem, which is the focus of this paper.

4. Addressing the imbalanced data set problem

4.1. Imbalanced data set problem

In a classification problem, the training set is *imbalanced* when one class is more heavily represented than the other. Clearly, this problem arises in our SU boundary detection task. As we have mentioned earlier, only about 13% of the interword boundaries correspond to SU boundaries in conversational speech, and 8% in broadcast news speech.

The imbalanced data set problem has received much attention from statisticians and the machine learning community (Chawla et al., 2003, 2002; Kubat and Matwin, 1997; Laurikkaka, 2001; Kubat et al., 1997; Japkowicz and Stephen, 2002; Provost and Fawcett, 2001; Lee, 2000; Chan and Stolfo, 1998; Ling and Li, 1998). Various approaches try to balance the class distribution in the training set by either oversampling the minority class or downsampling the majority class. Some variations on these approaches use sophisticated ways to choose representative majority class samples (instead of randomly choosing samples to match the size of the majority sample to that of the minority class), synthetically generate additional samples for the minority class (rather than replicating the existing samples), or combine classifiers trained from both the downsampled and the oversampled data sets. It is important to note that most of the techniques are focused on improving the minority class prediction (due to its relatively higher importance in many applications).

Weiss and Provost (2003) observed that the naturally occurring distribution is not always the optimal distribution; when using an ROC as a performance criterion, a balanced distribution is usually a preferred choice. While sampling methodologies generally improve the prediction of the minority class, they tend to penalize the majority-class cases. However, for the SU boundary detection task defined in the DARPA EARS program, both false positives and false negatives are equally costly. Therefore, changing the distribution to have relatively more minority class samples may not produce a classifier with the best performance. Our goal is thus to evaluate techniques to address the imbalance in the data sets for the SU detection task.

Which sampling method is the best greatly depends on the chosen classifier[4] and the properties of the application, such as how the samples are distributed in the multidimensional space or the extent to which the different classes are mixed. Therefore, a systematic investigation of different sampling approaches for our SU boundary detection task is important for building better decision tree classifiers. In addition to sampling methods, we investigate the use of bagging. Bagging samples the same training set multiple times, and has been shown to outperform a single classifier trained from the training set (Breiman, 1996).

The present study has properties that are characteristic of machine learning tasks for speech and natural language processing in general: it involves rather large amounts of data, it involves inherent ambiguity (SU boundaries are sometimes a matter of judgment), the data is noisy because of both measurement errors (from imperfect forced alignments) and labeling errors (human labelers make errors), and the class distribution is skewed, the latter being the main issue addressed in this paper. In addition, the property that the majority and the minority classes are of equal interest is another attribute that makes this problem interesting. We believe that this study is therefore beneficial to both the machine learning and the speech and language processing communities.

4.2. Sampling approaches

In our experiments, we investigate the use of four different sampling approaches, as well as the original training set. We investigate these approaches because of the convenience in maintaining a particular class distribution (balanced data set or the original training set) to support the combination of the prosody model and the LM.

- *Random downsampling.* This approach randomly downsamples the majority class to equate the number of minority and majority class samples. Since this method uses only a subset of majority class samples, it may result in poorer performance for the majority class (Chawla et al., 2002; Kubat et al., 1997; Japkowicz and Stephen, 2002).
- *Oversampling using replication.* This approach replicates the minority class samples to equate the number of majority and minority class samples. All of the majority class samples are preserved; however, the minority class samples are replicated multiple times. If some of these are bad samples of the minority class, their addition can lead to poorer performance for the minority class (Chawla et al., 2002; Japkowicz and Stephen, 2002; Ling and Li, 1998).
- *Ensemble downsampling.* In random downsampling, many majority samples are ignored. Ensemble downsampling is a simple modification of this approach. We split the majority class into N subsets, each with roughly the same number of samples as the minority class (Chan and Stolfo, 1998). Then we use each of these subsets together with the minority class to train a classifier that is, the minority class is coupled with a disjoint subset of majority class data. In the end, we have N decision trees, each of which is trained from a balanced training set. On the test set, the posterior probabilities from these N decision trees are averaged to obtain the final decision. The samples used for this approach are the same as in the oversampling approach, that is, all the majority class is used plus the minority class samples replicated N times. The two

[4] The reasons for choosing a decision tree classifier for the prosody model are discussed in Section 3.1.1.

approaches differ only in how the decision trees are trained. The ensemble downsampling approach is more scalable since classifier training can be easily implemented in a distributed fashion.

- *Oversampling using synthetic samples – SMOTE.*[5] In the oversampling approach, the minority class samples are replicated multiple times. By contrast, the SMOTE (Chawla et al., 2002) approach generates synthetic minority class samples rather than replicating existing samples. Synthetic samples are generated in the neighborhood of the existing minority class examples. For the continuous feature values, SMOTE produces new values by multiplying a random number between 0 and 1 with the difference between the corresponding feature values of a minority class example and one of its nearest neighbors in the minority class. For nominal cases, SMOTE takes a majority vote among a minority class example and its k-nearest neighbors. The synthetic samples can potentially cause the classifier to create larger and less specific decision regions, which can potentially generalize better on the testing set than simple oversampling with replication.
- *Original data set*. There is no sampling in this method. We utilize the original training set as is.

4.3. Bagging

Bagging (Breiman, 1996) combines classifiers trained from different samples (with replacement) given a training set. The bagging algorithm is shown in Fig. 3. To maintain a fixed class distribution used for all the bagging trees (in order to easily combine the prosody model with the LM), we sample for each class separately. T sets of samples are generated, each of which is used to train a classifier, and the final classifier is built from the T classifiers, equally weighted. Since each classifier generates the posterior probability for the test sample, the outputs from these classifiers can be averaged to obtain the final probability for a test sample, which is then combined with the LM.

Bagging has several advantages. First, because different classifiers make different errors, combining multiple classifiers generally leads to superior performance when compared to a single classifier (Dietterich, 2000), and thus it is more noise tolerant. Second, bagging can be computationally efficient in training because it can be implemented in a parallel or distributed fashion (Chawla et al., 2003). Finally, bagging is able to maintain the class distribution of the training set on which bagging is applied. This is important since the prosody model is combined with the LM to achieve the best possible performance (Shriberg et al., 2000; Liu, 2004). One disadvantage of bagging is that it produces multiple decision trees that can mask the advantage of the easy interpretation of the important features, which is available for a single decision tree.

Boosting (Freund, 1996; Freund and Schapire, 1996) is another algorithm that combines multiple classifiers and that has been shown to generally improve classification performance over a single classifier. However, our preliminary experiments have found no gain from boosting compared to bagging for the SU detection task. In addition, the classifier training runs sequentially and therefore is not as computationally efficient as bagging. Finally, boosting does not preserve the

[5] SMOTE stands for 'Synthetic Minority Over-sampling Technique'.

Input: training set S, number of bagging T

Bagging (T, S)
for $i = 1$ to T {
  S'_1 = sample from class 1 in S (with replacement)
  S'_2 = sample from class 2 in S (with replacement)
  $S' = S'_1 + S'_2$
  train a decision tree C_i from S'
}

Output: T classifiers

Fig. 3. The bagging algorithm. T is 50 in our experiments. In each bag, the class distribution is the same as in the original data S.

class distributions in different sampled training sets, thus making the combination of the prosody model and the LM much less straightforward. Given these considerations, we choose not to use boosting for this investigation.

5. Pilot study

5.1. Experimental setup

In this pilot study, we use a small subset of the CTS data from the RT-03 training data in order to evaluate each of the methods described above. Table 2 describes the data set that we used in our experiments, containing 128 conversations annotated according to the annotation guidelines (Strassel, 2003). These conversations are from the first section of the corpus released by LDC for the RT-03 FALL DAKPA EARS evaluation. We split these conversations into training and testing sets.[6] The training data set contains about 128 K word boundaries, of winch 16.8 K are in the SU class, with the remaining being non-SUs. The test set consists of about 16 K word boundaries.

Because it is time consuming to train a decision tree with a large number of features and also to synthetically generate minority samples using the SMOTE approach for this pilot study, we first trained a decision tree from a downsampled training set using all the prosodic features described in Section 3. We then used the features selected by this decision tree (25 features in total) to evaluate the various sampling approaches in order to minimize the computational effort for the pilot work.

For these initial investigations, we evaluate only on human transcriptions to factor out the impact of recognition errors on our investigation of the prosody model. Also, results are reported using the CER, F-measure, and ROC and AUC measurement, to better focus on the machine learning aspects of the problem.

[6] The list of training and test conversations can be found at http://www..berkeley.edu/~yangl/CSL-ML.

Table 2
Description of the data set used in the pilot study

Training size	127,937 words
Test size	16,066 words
Class distribution	87% are SUs, 13% are non-SUs
Features	25 features (2 discrete)

Note that the number of words shown in this table may not be exactly the same as that in the released data due to our processing of the data (e.g., text normalization, forced alignment).

We evaluated all of the sampling and bagging approaches on the test set under two conditions:

- *Prosody model alone*. For the decision trees trained from a balanced data set, we must combine the priors and the posterior probabilities generated by the decision trees to obtain the adjusted posterior probabilities for the imbalanced test set, as described in Section 3.1.3. For the decision trees trained from the original data set, the posterior probabilities do not need to be adjusted under the assumption that the original training set and the test set have similar class distributions.
- *Combination with the LM*. We evaluate the HMM combination of the prosody model and the event N-gram LM on the test set. If the decision tree is trained from the balanced data set (no matter which downsampling or oversampling approach is used), the posterior probability from the decision tree needs to be adjusted, essentially canceling out the denominator in Eq. (6). For the decision trees trained from the original data set, the posterior probability generated by the decision tree is the posterior probability $P_{DT}(E_i|F_i,W_t)$ in the numerator of Eq. (6).

We do not modify the distribution of the testing set because our goal is to see if the training process can be improved to achieve better performance, on the naturally occurring data distribution for the SU detection task. This allows us to establish the generalization of the underlying classifier(s). Moreover, we correct the posterior probabilities with the modified priors to compensate for the oversampling- or downsampling-introduced balanced distributions.

5.2. Sampling results

Experimental results for the different sampling approaches are shown in Table 3. We use a β of 1 in the F-measure computation. In addition we set a threshold of 0.5 on the probability for

Table 3
Experimental results (CER in % and F-measure) for different sampling approaches in the SU detection task in the pilot study, using the prosody model alone and its combination with the LM

Approaches	Prosody alone		Prosody + LM	
	CER	F-measure	CER	F-measure
Chance	13	0	–	–
Downsampling	8.48	0.612	4.20	0.837
Oversampling	10.67	0.607	4.49	0.826
Ensemble	7.61	0.644	4.18	0.837
SMOTE	8.05	0.635	4.39	0.821
Original	7.32	0.654	4.08	0.836

The CER of the LM alone on the test set is 5.02%.

classification. Looking at the results using the prosody model alone, we notice that the downsampling methods out-perform the oversampling method that employs replication (the differences are significant at $p < 0.01$ using the sign test). Oversampling by replication can lead to overfitting. The CER from the oversampling (by replication) is much higher than any other technique in the table, thus indicating that the decision tree does not generalise well on the testing set. There is a slight improvement when using an ensemble of multiple decision trees over using a single randomly downsampled data set to train the prosody model (the difference is significant at $p < 0.01$ using the sign test).

SMOTE improves upon the results from both downsampling and oversampling approaches (significant at $p < 0.01$). SMOTE introduces new examples in the neighborhood of the minority class cases, thereby improving the coverage on the minority class cases. Since SMOTE enables the entire majority class set to be used in a single decision tree, it can improve the performance on the majority class (i.e., the nan-SU decision). This result is also supported by the discussion in the subsequent section. However, SMOTE can lead to a computational bottleneck for very large data sets, since the training set becomes even larger as a result of adding the synthetic samples.

Using the original training set achieves the best overall performance in terms of the F-measure and CER when using only the prosody model (the differences between this and the other sampling methods are significant at $p < 0.01$, except for the ensemble sampling). This is, potentially, due to the equal costs assigned to both the classes and using 0.5 as the threshold. Also, it is possible that there are sufficient examples belonging to the minority class in the training set to learn about the SU boundaries. Thus, the classification error is lower than for the other sampling approaches. However training a decision tree from the original training set takes much longer than from a downsampled training set. If the training set were large or heavily skewed, the advantage of the original training set may be diminished.

Fig. 4 compares the various techniques using the ROC curves and the AUC obtained by each of the approaches when using the prosody model alone. The ROC curves span the entire continuous region of classification thresholds, and hence provide a visualization of the trade-off between the true positives and false positives. For the CER measurement, using the original training set achieves the best performance (as shown in Table 3); however, the advantage of the sampling techniques is more pronounced when we look at the ROC curves (and the corresponding AUC value). The AUC of the sampling and ensemble techniques is significantly larger than the AUC obtained by training a decision tree on the original distribution. Ensemble sampling yields the best ROC and AUC among all the approaches. We observe that downsampling improves upon oversampling with replication, while SMOTE improves upon both oversampling and downsampling. This has also been observed by other researchers in the machine learning literature (Drummond and Holte, 2003; Chawla, 2003). As shown in Fig. 4, at lower false positive (FP) rates, the original distribution is competitive with the sampling techniques, while at higher FP rates, the sampling schemes significantly improve upon the original distribution. Thus, if relative costs were to be established between the false positives and true positives, an optional operating point could be selected. If the minority class were of greater importance, then one would tolerate more false positives and achieve a higher recognition on the minority class. If obtaining a high recall for the SU detection task is important, then based on the ROC analysis, the sampling techniques are definitely useful.

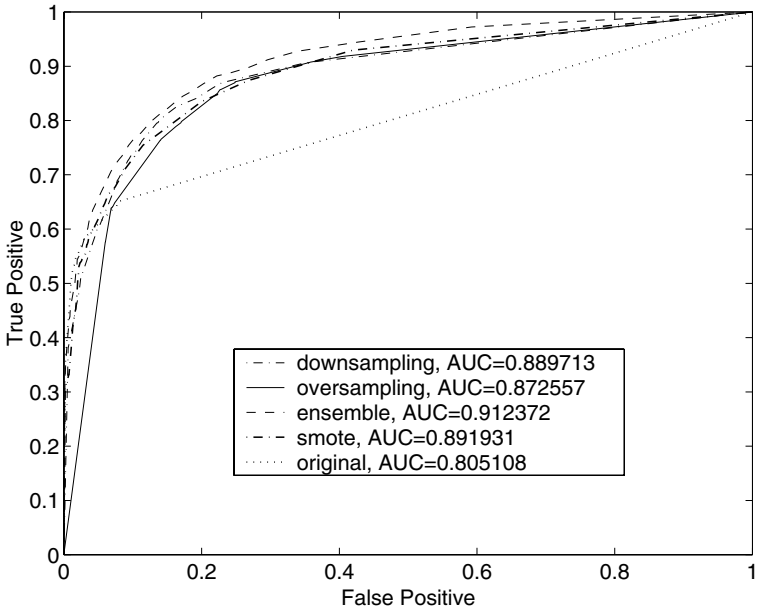

Fig. 4. ROC curves for the decision trees trained from different sampling approaches and the original training set.

The nonsmooth ROC using the original training set is largely due to its imperfect probability estimation. For example, the minimum posterior probability among all the test samples is 0.16 according to the decision tree; therefore, when the decision threshold is greater than 0.16, all the test samples are hypothesized as being in the positive class, resulting in a sharp change in the curve. In the sampling approaches, on the other hand, the posterior probabilities span over the entire region between 0 and 1. These observations based on the ROC analysis are especially useful if the output from an SU detection system is used by other downstream language processing modules, which generally need posterior probabilities of the SU class membership.

Looking at the results in Table 3 where the prosody model is combined with the LM, we find that the gain from the prosody model alone does not always hold up in combination with the LM. For example, the gain from the SMOTE method using the prosody model alone is lost when it is combined with the LM. This may occur because the synthesized samples to some extent are incompatible with what normally happens in language, or the samples with which SMOTE helps make correct decisions for the prosody model alone are the ones that are already well modeled by the LM. Similarly, the gain of the prosody model from the ensemble downsampling over the random downsampling approach is diminished when combined with the LM. This suggests that even though one classifier (prosody model) alone achieves good performance, other knowledge sources (language models) may mask this gain, especially when that knowledge source is a very strong one for the task, as is the case for the LM.

To focus on the error patterns for each sampling method, in Table 4 we show the precision and recall using the prosody model alone, as well as in combination with the LM. Using the prosody model alone, oversampling yields the best recall result, at the cost of lower precision. We had expected that using a balanced training set would be beneficial to the recall rate; however, contrary

Table 4
Recall and precision results (%) for the sampling methods in the pilot study

Approaches	Prosody alone		Prosody + LM	
	Recall	Precision	Recall	Precision
Downsampling	51.4	75.6	83.0	84.5
Oversampling	63.5	58.2	82.2	83.1
Ensemble	53.2	81.4	82.6	84.9
SMOTE	53.8	77.4	77.4	87.4
Original	53.5	84.7	80.0	87.5

Using the LM alone yields a recall of 74.6% and precision of 84.9%.

to our expectations, the recall performance from the downsampling and ensemble sampling approaches is not better than using the original training set. This is possibly because the downsampled data set is small and insufficient for training a robust decision tree. Note again that we use 0.5 as the threshold to make decisions using the posterior probabilities, and the false positive and false negative errors are equally costly. Thus, there are many fewer false positives if the original distribution is used in learning the decision tree. This leads to a much higher value of *precision* compared to any of the sampling techniques.

After the prosody model is combined with the LM, we notice that the recall rate is substantially improved for the downsampling and ensemble sampling approaches, resulting in a better recall rate than when the original training set is used. However SMOTE does not combine well with the LM: the recall rate is the worst after combining with the LM even though SMOTE yields a better recall rate than downsampling or ensemble sampling when the prosody model is used alone. Since SMOTE introduces new synthetic examples to the prosody model but not to the LM model, there could be a systematic disagreement between the prosody and the LM on the synthetic data that might reduce the overall performance of the system. The relative gain in the recall rate from the oversampling approach when the prosody model is used alone is also diminished when combined with the LM.

5.3. Bagging results

We have selected several sampling techniques on which to apply bagging. Since downsampling is an approach that is computationally efficient and does not significantly reduce classification accuracy, we first bagged the downsampled training set to construct multiple classifiers. We also tested bagging together with the ensemble approach. As described above, for the ensemble approach, we partitioned the majority class samples into N sets, each of which is combined with the minority class samples to obtain a balanced training set for decision tree training. The final classifier is the combination of the N base classifiers. We applied bagging (with trial number T) to each of N balanced sets, and thus obtained $T*N$ classifiers for combination. Finally, we applied bagging on the original training set. For all the bagging experiments, we used 50 bags. We do not combine bagging with any oversampling approaches because of their poorer performance compared to downsampling or using the original training set.

The bagging results are reported in Table 5. These results show that bagging always reduces the classification error rate over the corresponding method without bagging (significant at $p < 0.01$ using the sign test). Bagging the downsampled training set uses only a subset of the training

Table 5
Experimental results (CER in % and F-measure) with bagging applied to a randomly downsampled data set, ensemble of downsampled training sets (DS), and the original training set

Approaches	Prosody alone		Prosody + LM	
	CER	F-measure	CER	F-measure
Downsampling	8.48	0.612	4.20	0.837
Ensemble of downsampled sets	7.61	0.644	4.18	0.837
Original training set	7.32	0.654	4.08	0.836
Bagging on downsampled set	7.10	0.665	3.98	0.845
Bagging on ensemble DS	6.93	0.673	3.93	0.847
Bagging on original set	6.82	0.676	3.87	0.849

The results for the training conditions without bagging are also shown for comparison. The CER of the LM alone on the test set is 5.02%.

samples, yet achieves better performance than using the original training set without bagging. Bagging is able to construct an ensemble of diverse classifiers, and improves the generalization of decision tree classifiers; it mitigates the overfitting of a single decision tree classifier. The gain is more substantial when bagging is applied to the downsampled training set than to the original training set or the ensemble sampling sets.

Similarly to the study on sampling techniques, we plot the ROC curves for the three bagging schemes in Fig. 5, with a zoomed version of the curve shown at the bottom. The AUC is substantially better when bagging is employed compared to the results shown in Fig. 4, while the three bagging curves are very similar. Notice that the AUC is improved substantially when bagging is applied to the original training set. This can be attributed to the better posterior probability estimation, which is obtained from the average of multiple classifiers. Consistent with the results without bagging, applying bagging on the original training set yields a slightly poorer AUC than in the downsampled and ensemble bagging cases.

6. Evaluation on the NIST SU task

6.1. Experimental setup

We next evaluate some of the best techniques identified by the pilot study on the full NIST SU detection task (National Institute of Standards and Technology, 2003) in the DARPA EARS program. Evaluation is performed on two corpora that differ in speaking style: conversational telephone speech (CTS) and broadcast news (BN). Training and test data are those used in the DARPA RT-03 Fall evaluation.[7] The CTS data set contains roughly 40 hours for training and 6 hours (72 conversations) for testing. The BN data contains about 20 hours for training and 3 hours (6 shows) for testing. Training and test data are annotated with SUs by LDC, using guidelines detailed in Strassel (2003). Table 6 shows the data size of the training and the test set, the

[7] We combined both the development set and the evaluation set as the test set in this paper, in order to increase the test set size to make the results more robust.

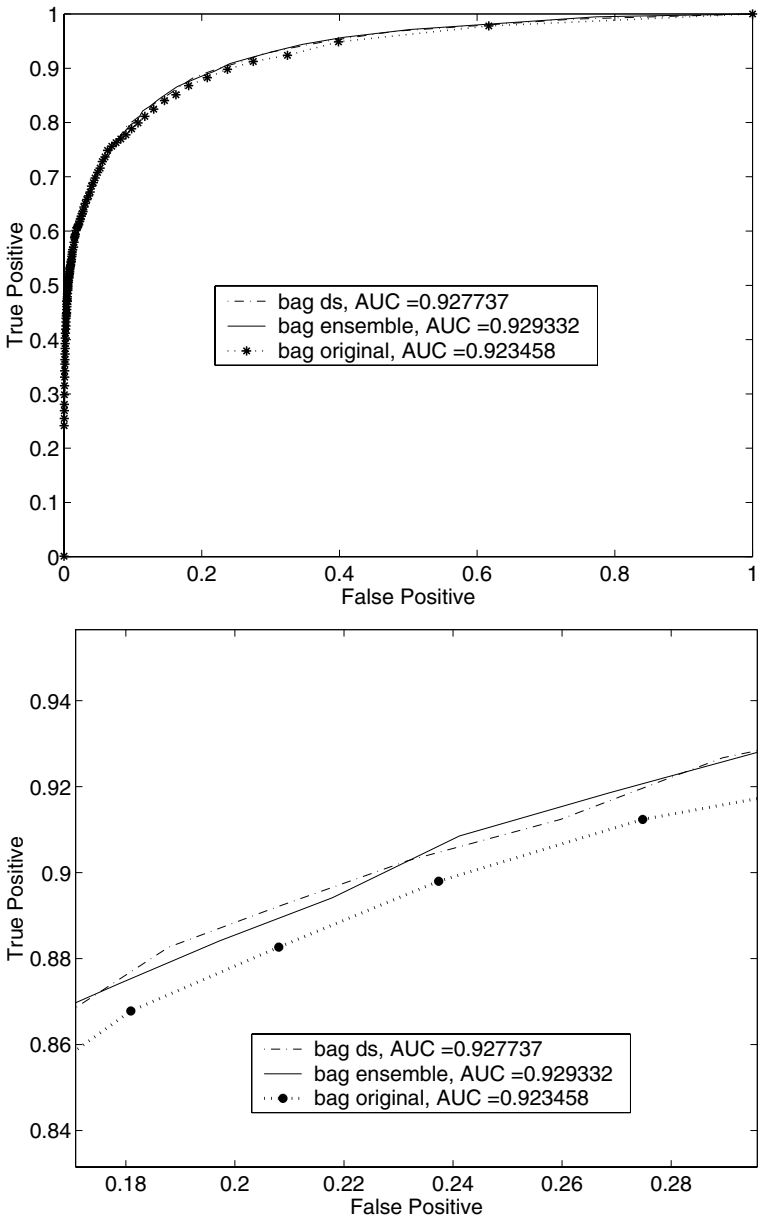

Fig. 5. ROC curves and their AUCs for the decision trees when bagging is used on the downsampled training set (bag-ds), the ensemble downsampled training sets (bag-ensemble), and the original training set (bag-original).

percentage of the SU boundaries, and the recognition word error rate (WER) on the test set. WER is determined using the recognition output from SRI's recognizer used in the 2003 NIST evaluation (Stolcke et al., 2000). As can be seen from Table 6, the amount of the CTS training data used in this study is about 300% greater than the CTS training set used in the pilot study.

Table 6
Information about CTS and BN corpora for the SU detection task

	CTS	BN
Training set	480 K	170 K
Test set	71 K	24 K
SU percentage (%)	13.6	8.1
Recognizer WER (%)	22.9	12.1

The speech recognition output is from SRI's recognizer (Stolcke et al., 2000).

Systems are evaluated on both the reference human transcriptions (REF) and the output of the speech recognition system (STT). Results are reported using the official NIST SU error rate metric. An important reason for choosing this metric is that we are evaluating on both the reference transcriptions and the recognition output, which makes using the classification error rate difficult. This choice of the metric is also useful for comparison with other systems for this N1ST benchmark test.

The pilot study results suggested that bagging is beneficial for generating a robust classifier, and the two best approaches are ensemble bagging and bagging on the original training set. Hence, we will evaluate these two approaches, alone and in combination with the LM. Since we used a downsampled training set in our prior work (Shriberg et al., 2000), we include this as a baseline. In addition to investigating sampling and bagging, we evaluate the impact of the recognition errors and speaking style by using different corpora. In contrast to the pilot study, we preserve all the prosodic features (a total of 101 features), expecting that bagging could generate different trees that might make use of different features. Since language model training requires a Large corpus, and the size of the SU-annotated training data is quite limited for BN, we also train a word-based event N-gram LM using available extra text data, which is not annotated precisely according to the LDC annotation guideline, but has punctuation information that can be used to approximate SUs. The two LMs are then interpolated with weights 0.8 and 0.2 (for the N-gram LM that is trained from the LDC-annotated training data). We will refer to the knowledge source provided by the interpolated LMs as LM in the following.

6.2. Results

Table 7 shows the results using each of the prosody models, and their combination with the LM on both the CTS and BN SU task, using both the reference transcriptions (REF) and the recognition output (STT). Overall, there is a performance degradation when using the speech recognition output, as one might expect. Recognition errors affect both the LM and the prosody model, with less impact on the latter; the prosody model is more robust than the LM in the face of speech recognition errors. The gain from bagging and sampling techniques in the REF condition seems to transfer well to the STT condition. Given the increase in data set size relative to the pilot study, we find that applying bagging techniques still yields a substantial win compared to non-bagging conditions. When the prosody model is used alone, applying bagging on the original training set achieves significantly better results (at $p < 0.01$) than ensemble bagging, on both corpora. When the prosody model is combined with the LM, the difference between using bagging on the original

Table 7
Experimental results (SU error rate in %) for both CTS and BN tasks on the REF and STT conditions, using the prosody model alone, and in combination with the LM

Approaches	BN		CTS	
	REF	STT	REF	STT
Prosody-ds	85.67	85.67	68.77	70.98
Prosody-ens-bag	72.94	72.09	61.23	64.35
Prosody-bag-original	67.65	67.77	59.19	62.98
LM	68.16	72.54	40.56	51.85
LM + prosody-ds	53.61	59.69	35.05	45.30
LM + prosody-ens-bag	50.03	56.17	32.71	43.71
LM + prosody-bag-original	49.57	55.14	32.40	43.81

Three conditions are evaluated for the prosody model, random downsampling (prosody-ds), bagging on ensemble downsampling (prosody-ens-bag), and bagging on the original training set (prosody-bag-original).

training set and bagging on the ensemble of balanced training sets is diminished (the gain is not significant). It is worth mentioning that there is a computational advantage of using downsampled training sets (random downsampling or ensemble downsampling) compared to using the original training set for decision tree training.

We observe similar patterns for CTS and BN, even though the extent of the imbalance differs. We also observe some differences across the two corpora, which have different class distributions because of their different speaking styles. The SU error rate is relatively higher for BN than for CTS, partly due to the small percentage of SUs in BN. Table 7 shows that the LM alone performs much better relative to the prosody model alone on CTS than on BN, suggesting that textual cues are more effective on CTS than BN. This can be attributed to backchannel words and pronouns being frequent in CTS, and the sparse data problem for the LM being more severe for BN. The error rate reduction from the combination of the LM and the prosody model compared to the LM alone is greater on BN than CTS. The degradation on the STT condition compared to the REF condition is smaller on BN than on CTS, mostly because of better recognition accuracy.

7. Conclusions

We have addressed the imbalanced data set problem that arises in the SU boundary detection task by using sampling and bagging approaches when training our prosody model. Empirical evaluations in a pilot study show that downsampling the data set works reasonably well, while requiring less training time. This computational advantage might be more important when processing a very large training set. Oversampling with replication increases training time without any gain in classification performance. An ensemble of multiple classifiers trained from different downsampled sets yields a performance improvement when using the prosody model alone. We have also found that the performance of the prosody model alone may not always be correlated with results obtained when the prosody model is combined with the language model; for example, SMOTE outperforms the downsampling approach when the prosody model is used alone, but not when the prosody model is combined with the language model. Using the original training set

achieves the best classification error rate among the sampling methods. However, if ROC or AUC is used for performance measurement, then using a balanced training set yields better results than the original training set, especially if the minority class is of greater interest. This is especially important if soft decisions from the SU detection system are desired by the subsequent language processing modules.

We have also investigated bagging on a randomly downsampled training set, an ensemble of multiple downsampled training sets, and the original training set. Bagging combines multiple classifiers and reduces the variance of a single classifier, and it improves the generalization of the classifiers. Bagging results in even better performance than the use of more samples (e.g., comparing bagging on a downsampled training set versus the original training set without bagging). Bagging can be parallelized, and thus training can be computationally efficient.

We have evaluated several of the best methods found from the pilot study on the NIST SU detection task across two corpora and across transcription conditions. Different speaking styles result in some differences in the SU detection performance on the CTS and BN data sets. On both corpora, significantly better performance is observed when bagging is used on the original training set than with ensemble bagging when using the prosody model alone, yet most of the gain is eliminated when the prosody model is combined with the LM.

Conclusions may depend in part on the characteristics of the SU detection application, the HMM approach adopted for this task, and the classifier (in our case the decision trees) used for the prosody model. For the SU detection task, the data set is not highly skewed (e.g., compared to the speech disfluency detection task), the prosodic features are quite indicative of the classes, and the annotation consistency is reasonably good. Therefore, it would be informative to investigate the techniques discussed here on other speech classification tasks (e.g., disfluency detection, emotion detection) to study the effectiveness of the various methods as a function of important problem parameters, such as the degree of imbalance and the baseline performance of the prosody model. We have observed some differences (Liu et al., 2004) using the sampling approaches investigated in this paper on the disfluency detection task, which has a more imbalanced class distribution than the SU detection task. We have also found in another study that using a downsampled balanced training set outperforms the original training set when approaches other than the HMM (such as the maximum entropy classifier) are used for knowledge source combination, due to their lesser reliance on the prosodic information (Liu et al., 2004) and the different precision/recall tradeoffs of the various sampling methods. Different classification algorithms may be affected by the imbalance to a different extent; therefore, classification algorithms other than the decision tree learning algorithms (e.g., Naive Bayes, support vector machines) are worth future investigation. Chawla et al. previously reported that SMOTE with decision trees outperformed Naive Bayes by varying the priors, and a rule learning classifier (Ripper) by varying the loss ratio. However, given that SU boundary detection is a very different domain, it will be interesting to evaluate different learning schemes.

Another interesting direction for future work will be investigating a semisupervised learning framework. There is a cost associated with labeling all the training examples precisely as a SU boundary or not. This leaves a large amount of data unlabeled. Using the classifier learned on the labeled data, we can infer labels on the unlabeled data and revise the classifier. The underlying challenge is to formulate a learning task that uses both labeled and unlabeled data such that generalization of the learned model can be improved.

Acknowledgements

The authors thank the anonymous reviewers of this paper for their valuable suggestions, Barbara Peskin at ICSI for her comments on this paper, and Luciana Ferrer at SRI for helping with the prosodic feature computation. This research has been supported by DARPA under Contract MDA972-02-C-0038, NSF-STIMULATE under IRI-9619921, NSF KDIBCS-9980054, NASA under NCC 2-1256, and Purdue Research Foundation. Distribution is unlimited. Any opinions expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA, NSF, or NASA. Part of this work was carried out while the third author was on leave from Purdue University and at NSF.

References

Beeferman, D., Berger, A., Lafferty, J., 1998. Cyperpunc: A lightweight punctuation annotation system for speech. In: Proceedings of the International Conference of Acoustics, Speech, and Signal Processing, 1998.

Bishop, C., 1995. Neural Networks for Pattern Recognition. Cambridge University Press, Cambridge, UK.

Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30 (6), 1145–1159.

Breiman, L., 1996. Bagging predictors. Machine Learning 24 (2), 123–140.

Buntime, W., Caruana, R., 1992. Introduction to IND version 2.1 and Recursive Partitioning. NASA Ames Research Center, Moffett Field, CA.

Campbell, W.N., 1993. Durational cues to prominence and grouping. In: Proceedings of ECSA Workshop on Prosody, Lund, Sweden, pp. 33–41.

Chan, P., Stolfo, S., 1998. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 164–168.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 321–357.

Chawla, N.V., Japkowicz, N., Kolcz, A., (August 2003). Workshop on learning from imbalanced datasets II. In: Proceedings of the 20th International Conference on Machine Learning.

Chawla, N.V., Moore, T.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P., Springer, L., 2003. Distributed learning with bagging-like performance. Pattern Recognition Letters 24 (1–3), 455–471.

Chawla, N.V., 2003. C4.5 and imbalanced datasets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In: Proceedings of the ICML'03 Workshop on Class Imbalances.

Chen, C.J., 1999. Speech recognition with automatic punctuation. In: Proceedings of the European Conference on Speech Communication and Technology, pp. 447–450.

Christensen, H., Gotoh, Y., Renal, S., 2001. Punctuation annotation using statistical prosody models. In: ISCA Workshop on Prosody in Speech Recognition and Understanding.

DARPA, 2003. Information Processing Technology Office, Effective, Affordable, Reusable Speech-to-text (EARS). Available from: <http://www.darpa.mi1/ipto/programs/ears/>.

De Pijper, J.R., Sanderman, A.A., 1994. On the perceptual strength of prosodic boundaries and its relation to suprasegmental cues. Journal of the Acoustical Society of America 96 (4), 2037–2047.

Dietterich, T.G., 2000. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning 40 (2), 139–157.

Drummond, D., Holte, R., 2003. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In: Proceedings of ICML'03 Workshop on Learning from Imbalanced Datasets.

Duda, R.O., Hart, P.E., 1973. Pattern Recognition and Scene Analysis. Wiley, New York.

Ferrer, L., 2002. Prosodic features for the Switchboard database, Tech. rep., SRI International.

Freund, Y., Schapire, R., 1996. Experiments with a new boosting algorithm. In: Machine Learning: Proceedings of the Thirteenth National Conference, pp. 148–156.

Freund, Y., 1996. Boosting a weak learning algorithm by majority. Information and Computation, 256–285.

Gotoh, Y., Renals, S., 2000. Sentence boundary detection in broadcast speech transcripts. In: Proceedings of ISCA Workshop: Automatic Speech Recognition: Challenges for the New Millennium ASR-2000, pp. 228–235.

Hand, D., 1997. Construction and Assessment of Classification Rules. Wiley, Chichester.

Hirst, D., 1993. Peak, boundary and cohesion characteristics of prosodic grouping. In: Proceedings of ECSA Workshop on Prosody, Lund, Sweden, pp. 32–37.

Huang, J., Zweig, G., 2002. Maximum entropy model for punctuation annotation from speech. In: Proceedings of the International Conference on Spoken Language Processing, pp. 917–920.

Japkowicz, N., Stephen, S., 2002. The class imbalance problem: A systematic study. Intelligent Data Analysis 6 (5), 429–450.

Kim, J., Woodland, P.C., 2001. The use of prosody in a combined system for punctuation generation and speech recognition. In: Proceedings of the European Conference on Speech Communication and Technology, pp. 2757–2760.

Kompe, R., 1996. Prosody in Speech Understanding System. Springer.

Kubat, M., Matwin, S., 1997. Addressing the curse of imbalanced training sets. In: Proceedings of the International Conference on Machine Learning, pp. 179–186.

Kubat, M., Holte, R., Matwin, S., 1997. Learning when negative examples abound. In: Proceedings of European Conference on Machine Learning, pp. 146–153.

Laurikkaka, J., 2001. Improving identification of difficult small classes by balancing class distribution, Tech. rep., Department of Computer and Information Science, University of Tampere, Finland.

Lee, S., 2000. Noisy replication in skewed binary classification. Computational Statistics and Data Analysis 34, 165–191.

Lickley, R., Bard, E., 1996. On not recognizing disfluencies in dialog. In: Proceedings of the International Conference on Spoken Language Processing, pp. 1876–1879.

Ling, C., Li, C., 1998. Data mining for direct marketing problems and solutions. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 73–79.

Liu, Y., Shriberg, E., Stolcke, A., 2003. Automatic disfluency identification in conversational speech using multiple knowledge sources. In: Proceedings of the European Conference on Speech Communication and Technology, pp. 957–960.

Liu, Y., Stolcke, A., Shriberg, E., Harper, M., 2004. Comparing and combining generative and posterior probability models: Some advances in sentence boundary detection in speech. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing.

Liu, Y., Shriberg, E., Stolcke, A., Peskin, B., Harper, M., The ICSI/SRI/UW RT04 structural metadata extraction system. In: EARS Rich Transcription Workshop, 2004.

Liu, Y., Shriberg, E., Stolcke, A., Harper, M., 2004. Using machine learning to cope with imbalanced classes in natural speech: Evidence from sentence boundary and disfluency detection. In: Proceedings of the International Conference on Spoken Language Processing.

Liu, Y., 2004. Structural event detection for rich transcription of speech, Ph.D. thesis, Purdue University.

Nakatani, C., Hirschberg, J., 1994. A corpus-based study of repair cues in spontaneous speech. Journal of the Acoustical Society of America, 1603–1616.

National Institute of Standards and Technology, 2003. RT-03F evaluation, http://www.nist.gov/speech/tests/rt/rt2003/fall/rt03f-evaldisc doc/index.htm.

National Institute of Standards and Technology, (Nov. 2003) RT-03F workshop agenda and presentations, http://www.nist.gov/speech/tests/rt/rt2003/fall/presentations/.

Ostendorf, M., Hillard, D., 2004. Scoring structural MDE: Towards more meaningful error rates. In: EARS Rich Transcription Workshop.

Palmer, D.D., Hearst, M.A., 1994. Adaptive sentence boundary disambiguation. In: Proceedings of the Fourth ACL Conference on Applied Natural Language Processing, pp. 78–83.

Potisuk, S., 1995. Prosodic disambiguation in automatic speech understanding of Thai, Ph.D. thesis, Purdue University.
Price, P.J., Ostendorf, M., Shattuck-Hufnagel, S., Fong, C., 1991. The use of prosody in syntactic disambiguation. Journal of the Acoustical Society of America 90 (6), 2956–2970.
Provost, F., Fawcett, T., 2001. Robust classification for imprecise environments. Machine Learning 42 (3), 203–231.
Rabiner, L.R., Juang, B.H., 1986. An introduction to hidden Markov models. IEEE ASSP Magazine 3 (1), 4–16.
Reynar, J., Ratnaparkhi, A., 1997. A maximum entropy approach to identifying sentence boundaries. In: Proceedings of the Fifth Conference on Applied Natural Language Processing, Washington, DC, pp. 16–19.
Schmid, H., 2000. Unsupervised learning of period disambiguation for tokenization, University of Stuttgart, Internal Report.
Scott, D.R., 1982. Duration as a cue to the perception of a phrase boundary. Journal of the Acoustical Society of America 71 (4), 996–1007.
Shriberg, E., Stolcke, A., Hakkani-Tur, D., Tur, G., 2000. Prosody-based automatic segmentation of speech into sentences and topics. Speech Communication, 127–154.
Sonmez, K., Shriberg, E., Heck, L., Weintraub, M., 1998. Modeling dynamic prosodic variation for speaker verification. In: Proceedings of the International Conference on Spoken Language Processing, pp. 3189–3192.
Stevenson, M., Gaizauskas, R., 2000. Experiments on sentence boundary detection. In: Proceedings of the North American Chapter of the Association for Computational Linguistics annual meeting, pp. 24–30.
Stolcke, A., Shriberg, E., 1996. Automatic linguistic segmentation of conversational speech. In: Proceedings of the International Conference on Spoken Language Processing, pp. 1005–1008.
Stolcke, A., Bratt, H., Butzberger, J., Franco, H., Rao Gadde, V.R., Plauché, M., Richey, C., Shriberg, E., Sönmez, K., Weng, F., Zheng, J., 2000. The SRI March 2000 Hub-5 conversational speech transcription system. In: Proceedings of NIST Speech Transcription Workshop, College Park, MD, 2000. URL http://www.nist.gov/speech/publications/tw00/html/cts80/cts80.htm.
Strassel, S., Walker, C., 2003. Data and annotation issues in RT-03. In: EARS Rich Transcription Workshop.
Strassel, S., 2003. Simple Metadata Annotation Specification V5.0, Linguistic Data Consortium. URL http://www.ldc.upenn.edu/projects/MDE/Guidelines/SimpleMDE_V5.0.pdf.
Swerts, M., 1997. Prosodic features at discourse boundaries of different strength. Journal of the Acoustical Society of America 101 (1), 514–521.
Wang, D., Narayanan, S.S., 2004. A multi-pass linear fold algorithm for sentence boundary detection using prosodic cues. In: Proceedings of the International Conference of Acoustics, Speech, and Signal Processing.
Weiss, G., Provost, F., 2003. Learning when training data are costly: The effect of class distribution on tree induction. Artificial Intelligence Research, 315–354.
Wrede, B., Shriberg, E., 2003. Spotting "hotspots" in meetings: Human judgments and prosodic cues. In: Proceedings of the European Conference on Speech Communication and Technology, pp. 2805–2808.