The University of Washington, Department of EE Technical Report Series

Darby Wong, Mari Ostendorf, Jeremy G. Kahn
{darby,mo,jgk}@ssli-mail.ee.washington.edu

Dept of EE, University of Washington
Seattle WA, 98195-2500

UWEE Technical Report
Number UWEETR-2005-0003
January 2005

Department of Electrical Engineering
University of Washington
Box 352500
Seattle, Washington 98195-2500
PHN: (206) 543-2150
FAX: (206) 543-3842
URL: http://www.ee.washington.edu

The University of Washington, Department of EE Technical Report Series

Darby Wong, Mari Ostendorf, Jeremy G. Kahn
{darby,mo,jgk}@ssli-mail.ee.washington.edu

Dept of EE, University of Washington
Seattle WA, 98195-2500

University of Washington, Dept. of EE, UWEETR-2005-0003
January 2005

Abstract

Automatic annotation of prosodic events could help improve speech understanding and synthesis. However, little annotated data is available for training prosody models because hand-labeling is prohibitively expensive. To address this issue, we explore weakly supervised learning techniques (EM, co-training, and self-training with bagging) that use only a small amount of hand-labeled data in combination with a large unlabeled data set with syntactic parses. Experiments on conversational speech show improved performance of decision trees on labeling symbolic prosodic events, specifically break indices and pitch accents.

1 Introduction

With humans, prosody helps communicate information beyond the words that are spoken, such as meaning, emphasis, and intent. For example, the prosody features duration and intonation can help humans resolve ambiguity in speech [11]. It is desirable to have a large corpus of prosodically labeled data for purposes of improving speech understanding and synthesis, but hand-labeling is prohibitively expensive. Our work focuses on using weakly supervised techniques to improve the accuracy of automatic prosody labeling.

In this section, we introduce the prosodic features we seek to label and previous work done to improve automatic labeling of related features. We give an overview of weakly supervised learning, list the methods we explore, show how we leverage syntactic features, and discuss the impact of a heavy data skew. Finally, we note the additional challenges labeling conversational speech presents over labeling read speech.

1.1 Prosody & Prior Work

Prosody can be divided into two categories: acoustic and symbolic. Acoustic prosody consists of prosodic features that can be obtained directly from acoustic waveforms, for example, timing and fundamental frequency (f0); symbolic prosody consists of prosodic features that require more sophisticated processing to determine. Our work seeks to improve automatic labeling of the symbolic prosody features *break index level (BL)* and *pitch accent (prominence)*.

Acoustic prosody has already been shown to be useful for tasks relevant to speech understanding, such as tagging dialog acts [14]. It is believed that symbolic prosody information can also make significant contributions, but the lack of a large corpus of annotated speech has hindered efforts to study its utility. Adding symbolic prosody information to a large corpus by hand-labeling is prohibitively expensive, as it requires specially trained labelers; furthermore, unlike acoustic prosody features, there are currently no systems that can label symbolic prosody from acoustic waveforms with a high degree of accuracy.

Automatic labeling of symbolic prosody has been studied in the past. Wang and Hirschberg [18] described a method of automatically labeling intonational phrase boundaries of task-specific utterances with 90% success, using acoustic prosody and syntactic features. Wightman and Ostendorf [19], working with acoustic prosody from speech

read by professional FM radio announcers, achieved 67% and 84% success rates in predicting ToBI-style break index and pitch accent labels, respectively. More recently, Chen et al. [3] described models that achieved 68.2% and 77.3% success rates at intonational phrase boundary and pitch accent recognition using acoustic prosody for the same data as in Wightman and Ostendorf [19], and 93.1% and 84.2% success rates for those same tasks using syntactic features in addition to acoustic prosody.

1.2 Our Approach

The lack of a large corpus of symbolically labeled data is a major limitation to improving automatic labeling through traditional supervised learning techniques: we need more training examples to improve classifier accuracy, but we need better classifiers to label more training examples. We therefore employ *weakly supervised learning* techniques. In weakly supervised learning, an initial classifier is trained on a small amount of labeled or partially labeled data and used to predict labels for unlabeled data. The predicted labels are then used with the hand-labeled data to train a new classifier, and the process iterates. The idea is that, in each iteration, the benefit from having a larger training set outweighs the cost of less accurate labels. We explore three weakly supervised learning methods: expectation maximization (EM), co-training, and self-training with bagging. These methods were chosen, in part, due to two significant characteristics of our work.

The first characteristic is our use of *syntactic features* to boost weakly supervised learning. Classifiers using syntactic features generally outperform those that do not, as evidenced in Chen et al. [3]; syntactic features can't always be inputs to a final classifier, however, since one of the goals in training symbolic prosody classifiers is to improve syntactic parsing. Here, we use syntactic features to improve intermediate classifiers, training all but the final classifier on both syntactic and acoustic features. The final classifier can thus benefit from syntactic information without having been trained on it. The presence of syntactic information also provides a second view of the data, which motivates our examination of co-training.

The second characteristic is the presence of a large data skew. As in previous prosody work [19, 18, 13], we use decision trees as classifiers. However, in Kubat et al. [5], it was shown that *data skew* can be harmful to the success of decision tree systems such as C4.5; regions of the instance space will tend to be dominated by the majority class, causing minority class instances to be classified as the majority class. In the context of weakly supervised learning, the potential contribution of these minority class instances are lost for future iterations. These effects have severe implications for our work, as there is a large skew in break index labels, with one class comprising approximately 70%. EM mitigates this by allowing partial counts; by avoiding a hard single-class decision, the presence of minority class instances is preserved for future iterations. Our variant of self-training addresses this issue as well, by down-sampling majority classes and up-sampling minority classes, such that each class is equally represented in the training data.

As a final note, in contrast to previous work, done on read speech, our work focuses on improving the accuracy of automatic symbolic prosody labeling of *conversational speech*. Conversational speech is spontaneous, which has been recognized to contain many differences from read speech. One key difference is an increased presence of disfluencies (filled pauses, repetitions, repairs, and false starts); such factors make modeling conversational speech significantly harder than modeling read speech.

The rest of this paper is organized as follows. In Section 2, detailed descriptions are given for the EM, co-training, and self-training algorithms. In Sections 3 and 4, experiment setup and results are given, respectively. Finally, in Section 5, we discuss the experiment results and present future directions to explore.

2 Methodology

Several methods of weakly supervised learning were tried: EM, co-training, and self-training with bagging. These methods all share our same basic algorithm of using acoustic prosody and syntactic features to train intermediate classifiers, while training final classifiers on acoustic prosody alone. They differ however, in whether a single classifier or a committee of classifiers is used at each iteration, and in the criteria used to select and label unlabeled data. The remainder of this section contains details for each method. For the following descriptions, let us represent the collections of labeled, weakly labeled, and unlabeled word boundaries as $\mathcal{D}^l$, $\mathcal{D}^w$, and $\mathcal{D}^u$, respectively.

Algorithm 1 Selection and labeling for $\mathcal{D}^w$.

```
if h_a < t_sure, h_s < t_sure then
    if d < t_agree then
        add to D^w with common label
    end if
else if h_a < t_sure, h_s > t_unsure then
    add to D^w with label from θ̂_a
else if h_a > t_unsure, h_s < t_sure then
    add to D^w with label from θ̂_s
else
    do nothing
end if
```

2.1 Expectation Maximization

Expectation maximization (EM) [4] is a popular unsupervised learning algorithm used to obtain maximum likelihood estimates of missing data, such as model parameters. EM can be used as a single-view weakly supervised learning algorithm by using estimated component membership of unlabeled data to iteratively train classifiers [9]. An advantage of EM for our application is that it provides a means of mitigating the effects of data skew by avoiding hard classifications.

In EM, an initial classifier is trained on acoustic prosody and syntactic features. This classifier is then used to generate class probability distributions for the data in $\mathcal{D}^u$. For each class, each word boundary is added to $\mathcal{D}^w$ with a weight corresponding to classifier certainty of membership in that class. Data labels with classifier certainty lower than 0.02 are pruned in order to avoid large file issues. The classifier is then re-trained on $\mathcal{D}^l$ and $\mathcal{D}^w$ using acoustic prosody and syntactic features. This process iterates until a local maxima of classifier accuracy is reached, at which point a final classifier is built solely on acoustic prosody features.

2.2 Co-training

Co-training [1] is a multi-view algorithm in which two redundant but independent views of data are used to improve classifiers. We chose to explore co-training because of the natural split between acoustic prosody and syntactic features. Two classifiers are trained for each intermediate iteration, one on acoustic prosody features, and one on syntactic features.

In our implementation of co-training (Algorithm 1), we add criteria to prevent the addition of data for which the two classifiers have conflicting predictions. We calculate the entropy h of each class probability distribution P for each word boundary $w \in \mathcal{D}^u$,

$$h = -\sum_x P(x|w) \log P(x|w),$$

where $P(x|w)$ is the probability that the word boundary w is in class x. High values of entropy indicate lack of confidence by a classifier, and low values indicate a high degree of certainty. For each word boundary, we calculate two entropies, h_a and h_s, corresponding to the distributions P_a and P_s from the classifiers trained on acoustic prosody and syntactic features, respectively. We also calculate Kullback-Leibler distance (relative entropy) for each class probability distribution P with respect to Q, the corresponding distribution from the other classifier:

$$D(P||Q) = \sum_x P(x) \log \left(\frac{P(x)}{Q(x)} \right).$$

A low KL-distance indicates that the two distributions are similar, whereas a high KL-distance is observed when they disagree.

Since KL-distance is not symmetric, we calculate the distances for both classifiers, $D(P_a||P_s)$ and $D(P_s||P_a)$, and average them:

$$d = \frac{1}{2} \left[D(P_a||P_s) + D(P_s||P_a) \right].$$

We set two thresholds for entropy values, t_{sure} and t_{unsure}. If a distribution has entropy below t_{sure}, it is said to be confident, but if the entropy is above t_{unsure}, then it is said to be uncertain. We set a third threshold for d, t_{agree}. If d is below t_{agree}, then the two distributions are said to agree on a label. Weakly labeled data is only added to $\mathcal{D}^w$ if the two classifiers are confident and agree, or if one classifier is confident and the other is uncertain.

2.3 Self-Training with Bagging

We use the variant of self-training described in Ng and Cardie [7], in which bagging [2] is used. Self-training with bagging is a single-view algorithm in which committees of classifiers built from bootstrap samples, created by sampling with replacement from training data, are used at each iteration. Here, this sampling is done equally from all classes, to avoid the negative effects data skew have on decision tree construction [5].

We create five bootstrap samples, each two-thirds the size of $\mathcal{D}^w$ and $\mathcal{D}^l$ combined. At least three classifiers are required to agree in order to add a label to $\mathcal{D}^w$ for the next iteration. We account for the up-sampling and down-sampling by normalizing the class probability distributions of each classifier against the prior class distribution of labeled data. The final classifier is a single decision tree trained on the acoustic prosody features of the data in $\mathcal{D}^l$ and $\mathcal{D}^w$.

3 Experimental Paradigm

3.1 Tree Design Software

We used a modified version of Quinlan's C4.5 [12] to construct decision trees for use as classifiers. C4.5 constructs decision trees by recursively splitting nodes based on information gain ratio. Pruning is done after a tree has been constructed; subtrees are replaced with leaf nodes if the estimated error of the subtree is greater than in the single leaf node. Modifications were made in order to allow each input datum to be individually weighted, a requirement of the EM algorithm.

3.2 Corpus

We used the Switchboard corpus of telephone conversations with Penn Treebank parses [6] in our experiments. There were a total of 1296 conversation sides, 124 of which had been hand-labeled with break index and prominence using a subset of the ToBI system [10, 16]. The "-" and "?" diacritics indicating labeler uncertainty were removed from the break index and pitch accent presence labeling, respectively. Any break index labels with the 'p' diacritic (1p, 2p, etc.) were mapped to simply 'p' for a general class of disfluent boundaries, including hesitations. Break index level 4 (BL 4) mapped to itself, while BL 0, 1, 2, and 3 were mapped to BL 1. BL 3 was mapped to BL 1 because preliminary 4-class prediction experiments showed that BL 3 was being mis-classified by decision trees as BL 1 in the vast majority of cases. The result of this mapping was a class for full intonational phrase boundaries (BL 4), and a class for all other fluent boundaries (BL 1). This 3-class system is relatively language independent, in that essentially all languages have a notion of fluent and disfluent segmentation. Of the 124 hand-labeled conversation sides, 10 were set aside for evaluating the performance of our classifiers.

The conversation sides used for training and testing in the break index labeling experiments were rearranged for compatibility in parse experiments; we re-ran the EM break index labeling experiment on this secondary arrangement and mention the results. In addition, the prominence detection experiments use the secondary arrangement. In both arrangements, there were approximately 55,000 word boundaries in the training set and 4000 in the test set; class distributions of the training and test data sets for both arrangements are in Table 1.

4 Experimental Results

4.1 Break Detection

Table 2 contains results and the number of iterations giving the most accurate classifier for each of the 3 training methods. Overall error rate[1], as well as precision and recall for intonation (BL 4) and hesitation (BL p) breaks are

[1] We define error rate as the number of incorrect predictions divided by the total number of predictions.

data set	*relative frequencies*		
	1	4	p
primary training	72.9	17.4	9.7
primary test	70.1	18.3	11.6
secondary training	72.9	17.3	9.8
secondary test	69.9	20.3	9.8

Table 1: Relative class frequencies for training and test data sets.

experiment	*iterations*	*error rate*	*precision*		*recall*	
			4	p	4	p
baseline		17.1	68.0	65.4	56.4	58.2
em	2	14.6	74.3	74.6	60.8	57.9
co-training	1	16.0	71.1	68.9	59.5	56.1
self-training	1	18.1	74.4	79	51.4	35.8

Table 2: Error rate, precision, and recall of different weakly supervised learning methods on break index labeling.

shown. EM performs best overall, with a 14.6% reduction in error rate over the baseline classifier. Both precision and recall improve significantly for BL 4 predictions, and precision of BL p predictions is improved, with only a small decrease in recall.

Co-training shows similar improvements, though lesser in magnitude; the sole exception is a decrease in recall of BL p predictions. Self-training actually worsens the classifier; in the self-training experiment, we can see that precision of BL 4 and BL p improves but at the expense of recall.

As mentioned in Section 3.2, the training and testing sets were rearranged for the experiments in Section 4.2. Metrics for 3 iterations of the EM algorithm for both arrangements are shown in Table 3 for comparison. We see the same general trends as before: precision of BL 4 and BL p predictions greatly improves; the recall of BL 4 predictions show moderate improvement while the recall of BL p predictions shows slight degradation.

arrangement	*iterations*	*error rate*	*precision*		*recall*	
			4	p	4	p
primary	0	17.1	68.0	65.4	56.4	58.2
	1	14.8	74.4	74.0	59.3	57.2
	2	14.6	74.3	74.6	60.8	57.9
	3	15.0	73.8	76.3	59.1	56.8
secondary	0	16.7	71.2	60.1	58.6	54.0
	1	14.2	81.2	71.6	62.0	52.1
	2	14.4	80.6	70.1	61.7	51.2
	3	14.3	79.9	72.0	62.5	50.7

Table 3: Error rate, precision, and recall for break index labeling with different arrangements of data.

4.2 Prominence Detection

Using the EM algorithm, we achieved a modest 4.1% reduction in error for prominence detection. Metrics for the baseline experiment and 1 iteration of EM are shown in Table 4. Precision and recall for labeling of prominence degrade slightly, but these metrics improve for labeling the absence of prominence.

experiment	*error rate*	*precision*	*recall*
baseline	22.2	72.6	62.7
em	21.3	72.4	62.5

Table 4: Error rate, precision, and recall for pitch accent detection.

5 Conclusions and Future Work

We have shown that EM can be as a weakly supervised learning algorithm to improve break index labeling and pitch accent detection for conversational data; for these tasks, error rates on a test set were 14.6% and 21.3%, with 14.6% and 4.1% reductions in error from the baseline classifier, respectively. Co-training shows a lesser improvement, and self-training worsens the classifier.

In Ng and Cardie [7], self-training with bagging outperforms EM using naive Bayes classifiers, but is outperformed by FS-EM, a variant of EM in which features for training the classifier are iteratively ordered and selected in an attempt to improve performance. This suggests that use of decision trees as classifiers played a role in the success of EM in our experiments, since decision trees are iteratively constructed by choosing features, based on information gain ratio.

A problem and potential area for improvement with EM is the tendency to get stuck in local maxima. It is possible that the method presented in this paper would benefit from an examination of ways to avoid this behavior, perhaps through randomization.

One possible reason co-training might not have performed as well as in other contexts is that it assumes conditional independence between views. In the case of acoustic prosody and syntactic features, this not a realistic assumption. In [8], it was shown that the performance of co-training can be hampered by violation of this assumption.

Finally, it would be important to study the effect automatically labeled symbolic prosody has for natural language processing of speech. Two questions to examine are how symbolic prosody can help improve syntactic parses of conversational speech, and whether classifier performance on any specific classes affects parses more than other classes.

References

[1] Blum, A., Mitchell, T. (1998). Combining Labeled and Unlabeled Data with Co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT '98)*. pp. 92-100.

[2] Breiman, L. (1994). Bagging Predictors. *Technical Report 421*. Department of Statistics, University of California Berkeley.

[3] Chen, K., Hasegawa-Johnson, M., Cohen, A. (2004). An Automatic Prosody Labeling System Using ANN-Based Syntactic-Prosodic Model and GMM-Based Acoustic-Prosodic Model. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004)*. 1, pp. 509-512.

[4] Dempster, A.P., Laird, N.M., Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*. 39(1), pp. 1-38.

[5] Kubat, M., Holte, R.C., Matwin, S. (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*. 30, pp. 195-215.

[6] Marcus, M., Santorini, B., Marcinkiewicz, M. (1993). Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*. 19(2), pp. 313-330.

[7] Ng, V., Cardie, C. (2003). Weakly Supervised Natural Language Learning Without Redundant Views. In *Proceedings of Human Language Technology Conference - North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)*. pp. 173-180.

[8] Nigam, K., Ghani, R. (2000). Analyzing the Effectiveness and Applicability of Co-training. In *Proceedings of the 9th ACM International Conference on Information and Knowledge Management (CIKM 2000)*. pp. 86-93.

[9] Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T. (2000). Text Classification from Labeled and Unlabeled Documents Using EM. *Machine Learning*. 39,2/3, pp. 103-134.

[10] Ostendorf, M., Shafran, I., Shattuck-Hufnagel, S., Charmichael, L., Byrne, W. (2001). A Prosodically Labeled Database of Spontaneous Speech. *Proceedings of ISCA Workshop of Prosody in Speech Recognition and Understanding*. pp. 127-131.

[11] 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), pp. 2956-1970.

[12] Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufman.

[13] Shriberg, E., Bates, R., Stolcke, A. (1997). A Prosody-Only Decision-Tree Model for Disfluency Detection. *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH 1997)*. pp. 2383-2386.

[14] Shriberg, E., Bates, R., Stolcke, A., Taylor, P., Jurafsky, D., Ries, K., Coccaro, N., Martin, R., Meteer, M., Van Ess-Dykema, C. (1998). Can Prosody Aid the Automatic Classification of Dialog Acts in Conversational Speech?. *Language and Speech*. 41(3-4), pp. 439-487.

[15] Shriberg, E., Stolcke, A. (2000). Prosody-Based Automatic Segmentation of Speech into Sentences and Topics. *Speech Communication*. 32(1-2), pp. 127-154.

[16] Silverman, H.F., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., and Hirschberg, J. (1992). TOBI: A Standard for Labeling English Prosody. In *Proceedings of the International Conference on Speech and Language Processing (ICSLP 1992)*. 2, pp. 867-870.

[17] Venkataraman, A., Ferrer, L., Stolcke, A., Shriberg, E. (2003). In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*. 1, pp. 272-275.

[18] Wang, M.Q., Hirschberg, J. (1995). Automatic Classification of Intonational Phrase Boundaries. *Computer Speech and Language*. 6(2), pp. 175-196.

[19] Wightman, C.W., Ostendorf, M. (1994). Automatic Labeling of Prosodic Patterns. *IEEE Transactions on Speech and Audio Processing*. 2(4), 469-481.