Combination of variations of pairwise classifiers applied to multiclass ToBI pitch accent recognition

César González-Ferreras, Carlos Vivaracho-Pascual, David Escudero-Mancebo, Valentín Cardeñoso-Payo

Departamento de Informática, Universidad de Valladolid, Spain

{cesargf,cevp,descuder,valen}@infor.uva.es

Abstract

In this paper we present some experiments on multiclass ToBI pitch accent classification. The system is based on the fusion of pairwise classifiers, which are specialized in the distinction of pairs of prosodic labels. Several machine learning techniques, including neural networks, decision trees and support vector machines, are combined in different ways in order to find the best overall combination. Variations of pairwise classifiers are introduced in order to take into account the influence of the samples of the remaining classes during the training of the binary classifiers. The use of these techniques allowed us to improve the results, both the overall classification accuracy and the balance across the different ToBI pitch accent classes.

Index Terms: automatic prosodic labeling, ToBI, classifier combination, pairwise classifiers

1. Introduction

Automatic multiclass pitch accent classification remains a challenging problem in computational prosody. There is a high perceptual similarity between some ToBI labels and some classes are more difficult to identify than others. On the other hand, some prosodic events are more frequent than others, which causes the corpora used in experiments to be clearly imbalanced, and, therefore, the classification performance is negatively affected.

In our previous work we reported a classification strategy based on pairwise classifiers which provided good performance [1]. Pairwise classifiers are specialized in the distinction of the prosodic labels in pairs. Basically, the multiclass classification problem is divided into a set of binary classification subproblems. The distinction of classes in pairs is an easier problem than the distinction between multiple classes and the combination of binary decisions provides improved classification results [2, 3].

In this paper we evaluate two variations of the pairwise strategy: *training with remaining classes* and *correcting classifiers* (to be described in sections 3.3.1 and 3.3.2 respectively). These variations try to avoid the problem that a binary classifier trained to distinguish between two particular classes l and m, might provide unreliable estimations for instances which belong neither to class l nor to class m. We experimented with the fusion of different configurations of the pairwise classifiers based on these variations and on different types of classifiers: neural networks, decision trees and support vector machines. Different types of classifiers appear to behave differently when they attempt to discriminate different classes and their outputs can be complementary.

The use of these machine learning techniques for prosody recognition allowed us to improve the results in multiclass pitch accent classification. As a conclusion, it is difficult to improve at the same time the total classification accuracy and the accuracy rate of each individual class. Thus, we selected two different configurations of the final system: one which improves the total classification accuracy and one which provides more balanced rates among all the prosodic classes.

The structure of the paper is as follows. First, we review the state of the art on automatic prosodic labeling. Then, the classification procedure and the experimental setup are described. Finally, we analyze the results and present some conclusions.

2. State of the art

Automatic detection and classification of ToBI events have been performed using different machine learning techniques: decision trees [1, 4, 5, 6, 7, 8, 9], Markov models [4, 10, 11], maximum entropy models [12], neural networks [1, 7, 8, 13, 14], GMM [13, 15, 16, 17], n-grams [10, 13, 18], Bayesian networks [19], conditional random fields [7, 8] and support vector machines [7, 8, 9, 14]. In most of those works, a combination of these techniques was used.

A common finding of previous work is that accuracy rates are highly dependent on the task: the identification of boundary tones and breaks is easier than the identification of pitch accents. Besides, the results were significantly better in prosodic event detection than in classification. The most efficient classifiers use morpho-syntactic features in conjunction with prosodic acoustic features (F0, intensity and duration) and their temporal evolution. Accuracy rates over 90% are reported in the detection of pitch accents [8]. Nevertheless, accuracy rates in classification are lower, 70.8% in [1], showing a high dependence on the number of classes and speakers, as shown in table 1. Although results can be improved by reducing the number of classes, we decided to keep the original set of classes in this work, since they convey linguistic meaning as defined in the standard [20], which should be preserved.

3. Classification method

In this section we describe the classification procedure used in the experiments, which is an evolution of the system presented in [1]. First we describe the strategy of multiple classifier combination and the base classifiers used in the experiments. Then, two variations of pairwise classification are explained. Finally, we present the experimental setup.

Table 1: Accuracy of pitch accent tone classification for different mappings of the ToBI labels, as reported in the state of the art. All the experiments used the Boston University Radio News Corpus.

Mapping	H*	H*	H*	H*	high	high	high
	L+H*	L+H*	L+H*	L+H*	high	high	high
	!H*	!H*	H*	!H*	downstepped	downstepped	downstepped
	H+!H*	H+!H*	H+!H*	ignored	high	high	high
	L+!H*	L+!H*	L+H*	ignored	downstepped	downstepped	downstepped
	L*	L*	L*	L*	low	low	low
	L*+H	L*+H	L*+H	ignored	low	low	low
	no label	none	ignored	ignored	unaccented	unaccented	unaccented
	#Classes	8	5	4	4	4	4
	Reference	[1]	[21]	[18]	[10]	[22]	[6]
	Level	word	word	word	syllable	syllable	syllable
	#Words/Syllables	27,767	29,578	28,300	14,599	14,599	14,377
	#Speakers	6	6	6	1	1	1
	Accuracy	**70.8%**	**63.99%**	**56.4%**	**80.17%**	**81.3%**	**87.17%**

3.1. Multiple classifier combination

The pairwise coupled approach basically divides a given multiclass classification problem into a number of binary classification subproblems, whose results must be combined to obtain the final classification result [2, 3]. According to this approach, let us refer by $\hat{P}(l|x, \lambda^k_{l,m})$ to an estimation of the probability $P(y = l|x, \quad y = l \vee m)$, where l and m are two different prosodic labels; x is the input of the classifier (in our case, the prosodic features); y is the class label; and $\lambda^k_{l,m}$ is a pairwise classifier of type k that is trained to separate classes l and m (neural network, $k = 1$; decision tree, $k = 2$; support vector machine, $k = 3$).

From these estimators, we build $\hat{P}(l|x, \lambda^k)$, which is obtained with classifiers of type k by:

$$\hat{P}(l|x, \lambda^k) = \prod_{\substack{m=1..C \\ l \neq m}} \hat{P}(l|x, \lambda^k_{l,m}) \quad (1)$$

where C is the number of classes, or prosodic labels.

Then, the results of K different types of classifiers are combined, so that the final estimation of $P(l|x)$, $\hat{P}(l|x)$, is computed as follows:

$$\hat{P}(l|x) = \prod_{k=1..K} \hat{P}(l|x, \lambda^k) \quad (2)$$

For each classifier type, there are as many classifiers as there are combinations of pairs of C classes: $\frac{C \cdot (C-1)}{2}$. Each classifier, $\lambda^k_{l,m}$, provides the posterior probability estimates $\hat{P}(l|x, \lambda^k_{l,m})$ and $\hat{P}(m|x, \lambda^k_{l,m})$.

Since the labeling of a given word depends on the context in which the word has been uttered, we introduce language model dependence. Experiments reported in [4, 13, 23] showed an improvement in results when a model of the sequence of labels was used. A detailed description of the process can be found in [1, 4, 13]. To search for the most likely prosodic label sequence, we applied the Viterbi algorithm [24]. The SRILM toolkit was used to build trigram prosodic language models [25], with Katz backoff for smoothing. The training data was used to build these models.

3.2. Base classifiers

We used three different types of classifiers in this work: decision trees (DT), neural networks (NN) and support vector machines (SVM). The reason for using different types of classifiers is that different classifiers behave differently on the discrimination of prosodic labels [1, 26].

A multilayer perceptron (MLP) was used, trained by means of the standard Error Backpropagation learning algorithm. Nonlinear sigmoid units were used in the hidden and output layers. A single hidden layer was used and a total of 100 training epochs. In the output layer we used as many units as classes, one per each class to classify. The POS feature was transformed into quantitative values by using a binary coding of the 33 values, using 6 bits. Normalization techniques were applied, using Z-Norm normalization across the same speaker.

The Weka toolkit [27] was used to build C4.5 decision trees (J48 in Weka). Different values for the confidence threshold for pruning have been tested, although the best results were obtained with the default value (0.25). The minimum number of instances per leaf was also set to the default value (2). This classifier was trained with qualitative POS features and unnormalized data. To obtain better class probability estimates, we turned off pruning, turned off *collapsing* and calculated class probabilities with the Laplace correction, as described in [28].

We used the Weka machine learning toolkit [27] implementation of the support vector machines. We tested different kernels and selected the polynomial kernel. To obtain probability estimates, logistic regression models were used at the output of the support vector machine. This classifier was trained with qualitative POS features and unnormalized data.

3.3. Variations of pairwise classification

In the canonical pairwise classification scheme, each pairwise classifier is trained to distinguish between two particular classes l and m. Then, only samples of this two classes are used in the learning stage. In the classification stage, each individual pairwise classifier, $\lambda^k_{l,m}$, is coupled with the others in order to get the final output for each test sample x. Given that x can belong to any class, the input of a particular classifier can belong to its target classes (l or m) or not. In this last case, the problem, observed in our work and in the literature [2, 29], is that the

Table 2: *Accuracy of the base classifiers (DT: Decision Tree; NN: Neural Network; SVM: Support Vector Machine; RC: training with Remaining Classes; CC: Correcting Classifiers).*

	DT	DT-RC	DT-CC	NN	NN-RC	NN-CC	SVM	SVM-RC	SVM-CC
H*	61.6%	74.0%	76.0%	64.0%	61.0%	72.2%	44.6%	61.2%	63.5%
L+H*	30.7%	21.2%	19.1%	31.8%	40.4%	34.1%	48.6%	41.9%	36.1%
!H*	35.1%	32.4%	32.7%	36.3%	45.4%	36.3%	44.2%	54.6%	52.0%
H+!H*	17.1%	13.1%	13.8%	18.1%	23.0%	10.1%	36.2%	11.8%	17.7%
L+!H*	7.4%	4.9%	3.9%	14.1%	14.3%	3.0%	29.6%	0.2%	1.3%
L*	18.6%	13.5%	10.3%	16.1%	29.6%	7.2%	45.5%	24.6%	30.2%
L*+H	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
none	86.4%	91.1%	91.6%	88.1%	85.9%	90.8%	82.9%	85.3%	86.5%
Total	66.2%	70.7%	71.2%	68.1%	67.9%	71.2%	63.5%	67.9%	68.6%

classifier might provide an unreliable output. Moreover, this unreliable output could be a high value, which might cause the instance to be incorrectly assigned to class l or to class m.

In order to cope with the described problem, we propose the use of two techniques: training with remaining classes (RC) and correcting classifiers (CC). Therefore, for each classifier type, we have three different classifier configurations: original pairwise classifiers, training with remaining classes and correcting classifiers.

3.3.1. Training with remaining classes

During the training phase, each classifier $\lambda^k_{l,m}$, is trained with examples of three classes: l, m and $\neg lm$ (this last is composed by the training examples of the rest of classes).

In the case of NN, the output layer is composed by two cells, $\{O_1, O_2\}$, assigning each cell at a certain target class, e.g., O_1 to l and O_2 to m. In the standard training method, the desired outputs are fixed to $\{1.0,0.0\}$ for l class samples and $\{0.0,1.0\}$ for m class samples. In the test stage, the input, x, is assigned to the class with the corresponding higher output, i.e, if the higher is O_1 x is assigned to l and if the higher is O_2 x is assigned to m. In the *training with remaining classes* method, the desired outputs in the learning stage are fixed at: $\{1.0,0.0\}$ for the l class training examples, $\{0.0,1.0\}$ for the m class training examples and $\{0.5,0.5\}$ for the $\neg lm$ class training examples. That is, the MLP is trained to provide high outputs when the input belongs only to the l or m classes.

In the case of DT and SVM, a similar method is applied. We extended the binary pairwise classifiers and built classifiers that can distinguish between three classes: l, m and $\neg lm$. Thereby, the probability estimates $\hat{P}(l|x, \lambda^k_{l,m})$ and $\hat{P}(m|x, \lambda^k_{l,m})$ provide high values only when the input belongs to classes l or m.

3.3.2. Correcting classifiers

For each pairwise classifier $\lambda^k_{l,m}$, separating class l from class m, an additional classifier is trained, $\phi^k_{l,m}$, separating classes l and m from all the other classes [29]. This additional classifier generates $\hat{Q}(lm|x, \phi^k_{l,m})$, an estimation that sample x belongs to either class l or class m, and can be included in equation (1), which becomes:

$$\hat{P}(l|x, \lambda^k) = \prod_{\substack{m=1..C \\ l \neq m}} \hat{P}(l|x, \lambda^k_{l,m}) \hat{Q}(lm|x, \phi^k_{l,m}) \quad (3)$$

The drawback of this technique is the cost of training $\frac{C \cdot (C-1)}{2}$ additional classifiers for each classifier type.

3.4. Experimental setup

We used the Boston University Radio News Corpus (BURNC) [30]. The experiments were performed using the word as the reference unit. All utterances in the corpus with ToBI labels from all the speakers were used. Pitch accents considered in this paper (and the number of samples of each) were: *H** (7,587), *L+H** (2,383), *!H** (2,144), *H+!H** (586), *L+!H** (638), *L** (517), *L*+H* (44) and *none* (13,868). We used oversampling in order to reduce the negative impact of imbalanced data on the final result [1, 9, 26]. Ten-fold cross-validation was applied in all the experiments.

We used similar features to the ones used in other experiments [13]. *Frequency features*: within-word F0 range, difference between maximum and average within-word F0, difference between average and minimum within-word F0, difference between within-word F0 average and utterance average F0. *Energy features*: within-word energy range, difference between maximum and average within-word energy, difference between average and minimum within-word energy. *Vowel nucleus duration*: we used the maximum normalized vowel nucleus duration from all of the vowels of the word. *Part of speech*: we used the POS tags that come with the BURNC corpus, which were automatically obtained and were hand-corrected [31].

In order to model the temporal evolution of the pitch contour along the unit of reference, we included additional features: Tilt and Bézier parameters. *Tilt* is probably the most widely applied technique for parameterizing the pitch contours [32]. Tilt has been explicitly used in the state of the art of prosodic event detection [9, 18]. *Bézier stylization* is based on the approximation of the pitch contours with Bézier functions [33]. The minimum square fitting approximation technique is used to represent the shape of the F0 contour along a given reference unit. In this work, we use 4 control points of the spline as parameters.

The use of context features can improve the classification results [1, 9, 21, 22, 34]. We decided to select the features to model the context using the Correlation-based Feature Selection (CFS) algorithm [35]. Without the use of context, for each word, we use 18 features. The CFS algorithm selected 8 features to be used as context features. We used 2 previous words and 2 following words as context [1].

Table 3: *Accuracy of the fusion of classifiers, with and without applying the Viterbi algorithm.*

	without Viterbi	with Viterbi
DT + NN + SVM	70.85%	71.29%
DT + NN + SVM-RC	71.26%	71.49%
DT + NN + SVM-CC	71.46%	71.70%
DT + NN-RC + SVM	70.65%	71.47%
DT + NN-RC + SVM-RC	71.07%	71.61%
DT + NN-RC + SVM-CC	71.12%	71.74%
DT + NN-CC + SVM	71.91%	72.07%
DT + NN-CC + SVM-RC	72.02%	72.15%
DT + NN-CC + SVM-CC	72.08%	72.19%
DT-RC + NN + SVM	72.01%	72.23%
DT-RC + NN + SVM-RC	72.05%	72.22%
DT-RC + NN + SVM-CC	72.24%	72.40%
DT-RC + NN-RC + SVM	72.10%	72.46%
DT-RC + NN-RC + SVM-RC	72.17%	72.37%
DT-RC + NN-RC + SVM-CC	72.20%	72.55%
DT-RC + NN-CC + SVM	72.38%	72.51%
DT-RC + NN-CC + SVM-RC	72.46%	72.58%
DT-RC + NN-CC + SVM-CC	72.56%	72.61%
DT-CC + NN + SVM	72.28%	72.45%
DT-CC + NN + SVM-RC	72.33%	72.57%
DT-CC + NN + SVM-CC	72.43%	72.59%
DT-CC + NN-RC + SVM	72.43%	72.51%
DT-CC + NN-RC + SVM-RC	72.25%	72.41%
DT-CC + NN-RC + SVM-CC	72.37%	72.54%
DT-CC + NN-CC + SVM	72.60%	72.64%
DT-CC + NN-CC + SVM-RC	72.62%	72.62%
DT-CC + NN-CC + SVM-CC	72.62%	72.54%

4. Experimental results

Table 2 shows the classification results of the base classifiers, before the fusion. The total accuracy of the different classifiers ranges from 63.5% for the SVM classifier to 71.2% for the DT-CC and NN-CC classifiers. The strategies RC and CC improve the results of their baseline counterparts: for instance, DT improves from 66.2% to 70.7% and 71.2% respectively.

Another important result in table 2 is that some classifiers are more effective in the identification of a given class than others. This justifies the improvements achieved with the classifier fusion strategy. For example, the SVM classifier is the most efficient in identifying class *L**, with a rate of 45.5%. For this class, DT classifiers only obtain 18.6% at most.

Table 3 shows the results of the fusion of classifiers, with and without applying the Viterbi algorithm. A first conclusion from these results is that the fusion improves the results achieved with the base classifiers. The best global results are achieved when the Viterbi algorithm is used, because it allows to search for the most likely prosodic label sequence, instead of considering the accents in isolation. However, this global improvement is mainly due to the improvement of classes *H** and *none* (the most frequent ones), as shown in table 4.

As we are interested in multiclass classification, higher classification rates in each of the classes are also important. Table 4 compares two alternative combinations with the baseline of our previous work. In the third column, the classifier DT-CC+NN-CC+SVM+Vit provides higher total accuracy rate, but is clearly specialized in the *H** and *none* classes, with accuracies of 78.0% and 91.8% respectively. In the second column, the classifier DT+NN-RC+SVM provides a better balance in accuracy across the different pitch accent classes. This classifier obtains the highest rates of all configurations for classes *L+H**, *!H**, *H+!H**, *L+!H** and *L**. These classes proved to be very difficult to recognize.

Table 4: *Rate of ToBI labels for different combinations of classifiers. We show the combination which provides more balanced results among classes and the combination which provides higher accuracy rate (to select the most balanced configuration we calculated the geometric mean of the classification rate of all classes except class L*+H. DT: Decision Tree; NN: Neural Network; SVM: Support Vector Machine; RC: training with Remaining Classes; CC: Correcting Classifiers; Vit: Viterbi).*

	Previous work [1]		More Balanced		Higher rate	
H*	72.5%	DT+NN+Vit	66.8%	DT+NN-RC+SVM	78.0%	DT-CC+NN-CC+SVM+Vit
L+H*	25.3%		37.3%		25.9%	
!H*	35.2%		46.9%		36.5%	
H+!H*	12.1%		25.3%		10.4%	
L+!H*	6.0%		11.4%		2.2%	
L*	11.4%		32.1%		9.1%	
L*+H	0.0%		0.0%		0.0%	
none	91.0%		88.4%		91.8%	
Total	**70.8%**		**70.7%**		**72.6%**	

Table 4 also shows that with the experiments reported in this paper we have outperformed the results of our previous work [1].

5. Conclusions

We have presented a system for the multiclass classification of ToBI pitch accents, which is based on classification by pairwise coupling and is an extension of our previous work [1]. A classifier for each pair of classes is built and the final label is assigned combining all the pairwise predictions. Several machine learning techniques are used to build the base classifiers: neural networks, decision trees and support vector machines.

We have described two different techniques in order to incorporate the samples of the other classes during the training of the pairwise classifiers: training with remaining classes and correcting classifiers. The use of both techniques provided us with various different configurations of the base classifiers, which seemed to be complementary. The combination of these configurations allowed us to improve our previous results [1]. Some combinations improve the overall classification accuracy: the classifier DT-CC+NN-CC+SVM+Vit improves the total rate from 70.8% to 72.6%. Other combinations provide more balanced accuracies among the different pitch accent classes: the classifier DT+NN-RC+SVM doubles the identification rate (or close to double the rate) of the classes *L**, *L+!H** and *H+!H**.

6. Acknowledgements

This work has been partially supported by Ministerio de Ciencia e Innovacion, Spanish Government (Glissando project FFI2011-29559-C02-01).

7. References

[1] C. González-Ferreras, D. Escudero-Mancebo, C. Vivaracho-Pascual, and V. Cardeñoso-Payo, "Improving Automatic Classification of Prosodic Events by Pairwise Coupling," *IEEE Transaction on Audio, Speech and Language Processing*, vol. 20, no. 7, pp. 2045–2058, September 2012.

[2] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *The annals of Statistics*, vol. 26, no. 2, pp. 451–471, April 1998.

[3] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, pp. 975–1005, December 2004.

[4] C. Wightman and M. Ostendorf, "Automatic labeling of prosodic patterns," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 469–481, October 1994.

[5] J.-S. Lee, B. Kim, and G. G. Lee, "Automatic corpus-based tone and break-index prediction using k-tobi representation," *ACM Transactions on Asian Language Information Processing*, vol. 1, pp. 207–224, September 2002.

[6] X. Sun, "Pitch accent prediction using ensemble machine learning," in *International Conference on Spoken Language Processing (ICSLP)*, 2002, pp. 16–20.

[7] C. Ni, W. Liu, and B. Xu, "From English pitch accent detection to Mandarin stress detection, where is the difference?" *Computer Speech and Language*, vol. 26, no. 3, pp. 127 – 148, 2012.

[8] C. Ni, W. Liu, and B. Xu, "Automatic prosodic events detection by using syllable-based acoustic, lexical and syntactic features," in *Proceedings Interspeech*, 2011, pp. 2017 – 2020.

[9] A. Rosenberg, "Automatic Detection and Classification of Prosodic Events," Ph.D. dissertation, University of Columbia, USA, 2009.

[10] K. Ross and M. Ostendorf, "Prediction of abstract prosodic labels for speech synthesis," *Computer Speech & Language*, vol. 10, no. 3, pp. 155–185, 1996.

[11] S. Ananthakrishnan and S. Narayanan, "An automatic prosody recognizer using a coupled multi-stream acoustic model and a syntactic-prosodic language model," in *International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2005, pp. 269–272.

[12] V. Rangarajan Sridhar, S. Bangalore, and S. Narayanan, "Exploiting Acoustic and Syntactic Features for Automatic Prosody Labeling in a Maximum Entropy Framework," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 4, pp. 797–811, May 2008.

[13] S. Ananthakrishnan and S. Narayanan, "Automatic Prosodic Event Detection Using Acoustic, Lexical, and Syntactic Evidence," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 216–228, January 2008.

[14] J. Jeon and Y. Liu, "Automatic prosodic events detection using syllable-based acoustic and syntactic features," in *International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2009, pp. 4565–4568.

[15] Y. Ren, S. Kim, M. Hasegawa-Johnson, and J. Cole, "Speaker-independent automatic detection of pitch accent," in *Proceedings of Speech Prosody*, 2004, pp. 521–524.

[16] K. Chen, M. Hasegawa-Johson, A. Cohen, and J. Cole, "A Maximum likelihood Prosody Recognizer," in *Proceedings of Speech Prosody*, 2004, pp. 509–512.

[17] K. Chen, M. Hasegawa-Johson, and A. Cohen, "An automatic prosody labeling system using ANN-based syntactic-prosodic model and GMM-based acoustic-prosodic model," in *International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2004, pp. 509–512.

[18] S. Ananthakrishnan and S. Narayanan, "Fine-grained pitch accent and boundary tone labeling with parametric F0 features," in *International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2008, pp. 4545–4549.

[19] M. Hasegawa-Johnson, K. Chen, J. Cole, S. Borys, S.-S. Kim, A. Cohen, T. Zhang, J.-Y. Choi, H. Kim, T. Yoon, and S. Chavaria, "Simultaneous recognition of words and prosody in the Boston University Radio Speech Corpus," *Speech Communication*, vol. 46, no. 3–4, pp. 418–439, 2005.

[20] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, "ToBI: A standard for labelling English prosody," in *International Conference on Spoken Language Processing (ICSLP)*, 1992, pp. 867–870.

[21] A. Rosenberg, "Classification of Prosodic Events using Quantized Contour Modeling," in *HLT/NAACL*, 2010, pp. 721–724.

[22] G. Levow, "Context in Multi-lingual Tone and Pitch Accent Recognition," in *Annual Conference of the International Speech Communication Association (Interspeech)*, 2005, pp. 1809–1812.

[23] A. Rosenberg, "Symbolic and Direct Sequential Modeling of Prosody for Classification of Speaking-Style and Nativeness," in *Annual Conference of the International Speech Communication Association (Interspeech)*, 2011.

[24] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[25] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *International Conference on Spoken Language Processing (ICSLP)*, 2002, pp. 901–904.

[26] C. González-Ferreras, C. Vivaracho-Pascual, D. Escudero-Mancebo, and V. Cardeñoso-Payo, "On the automatic ToBI accent type identification from data," in *Annual Conference of the International Speech Communication Association (Interspeech)*, 2010, pp. 142–145.

[27] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[28] F. Provost and P. Domingos, "Tree induction for probability-based ranking," *Machine Learning*, vol. 52, no. 3, pp. 199–215, 2003.

[29] M. Moreira and E. Mayoraz, "Improved pairwise coupling classification with correcting classifiers," in *European Conference on Machine Learning (ECML)*, ser. Lecture Notes in Computer Science, vol. 1398. Springer, 1998, pp. 160–171.

[30] M. Ostendorf, P. Price, and S. Shattuck, "The Boston University Radio News Corpus," Boston University, Tech. Rep., 1995.

[31] M. Marcus, M. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: The Penn Treebank," *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[32] P. Taylor, "Analysis and synthesis of intonation using the Tilt model," *Journal of Acoustical Society of America*, vol. 107, no. 3, pp. 1697–1714, 2000.

[33] D. Escudero and V. Cardeñoso Payo, "Applying data mining techniques to corpus based prosodic modeling," *Speech Communication*, vol. 49, no. 3, pp. 213–229, 2007.

[34] A. Rosenberg and J. Hirschberg, "Detecting Pitch Accent at the Word, Syllable and Vowel Level," in *HLT/NAACL*, 2009.

[35] M. A. Hall, "Correlation-based feature subset selection for machine learning," Ph.D. dissertation, University of Waikato, Hamilton, New Zealand, 1998.