Automatic Pitch Accent Prediction for Text-To-Speech Synthesis

Ian Read and Stephen Cox

School of Computing Sciences, University of East Anglia, Norwich, UK
{ihr, sjc}@cmp.uea.ac.uk

Abstract

Determining pitch accents in a sentence is a key task for a text-to-speech (TTS) system. We describe some methods for pitch accent assignment which make use of features that contain information about a complete phrase or sentence, in contrast to most previous work which has focused on using features local to a syllable or word. Pitch accent prediction is performed using three different techniques: N-gram models of syllable sequences, dynamic programming to match sequences of features, and decision trees. Using a C4.5 decision tree trained on a wide range of features, most notably each word's orthographic form and information extracted from the syntactic parse of the sentence, our feature set achieved a balanced error rate of 46.6%. This compares with the feature set used in [11] which had a balanced error rate of 55.55%.

1. Introduction

This paper presents a number of algorithms that predict the intonation pattern for an utterance. In text-to-speech synthesis, this is accomplished by predicting the appropriate pitch accents for each syllable and then using these to derive the target fundamental frequency (F0) contour.

The objective of this study is to investigate a number of techniques that classify a syllable into one of four groups of pitch accents: high, downstepped, low and unaccented [9]. The approaches to predicting pitch accents in this paper draw on techniques we have previously presented for predicting phrase breaks [5], [6], [7]. The general approach in these papers was that long-range information across several words of an utterance was beneficial in determination of phrase breaks, and we have taken a similar approach in this paper to the problem of predicting pitch accents. By predicting phrase breaks *prior* to predicting pitch accents, we are able to utilize prosodic phrases [3] in our system, which are typically much shorter than whole sentences, and achieve a good balance between local information around a syllable and information across a complete utterance. We also make use of syntactically-derived features, which are shown to be useful for pitch accent in the same way that they were useful for phrase break prediction [7].

This paper first documents the feature set used for predicting pitch accents (Section 2). Subsequent sections outline the techniques and algorithms used, which include N-gram models of syllable sequences, dynamic programming to match sequences of features, and C4.5 decision trees. Section 5 contains details of the experimental methods used and the results. Finally, Section 6 summarizes the techniques and findings of this research.

2. Features

In previous work [7], we have shown that the choice of features is at least as important as the choice of classifier for prosody prediction. Table 1 summarises some of the available features based on what can be automatically derived from the text analysis stage of a text-to-speech synthesis systems.

Part-of-speech tags were automatically derived using the Brill POS tagger [1]. The syntactically-derived features (K0 through DNW in Table 1) were made using the Collins parser [ref] and detailed descriptions of these features are available in [6], [7] and [8].

3. Algorithms

3.1. N-grams

The first experiment undertaken for pitch accent prediction involved using an N-gram model of syllable sequences. This is the only technique that uses only local features: it attempts to predict pitch accents using features associated with a small "window" of syllables around the target syllable.

Each pitch accent in the data is characterized by a sequence of M syllables preceding the accent and $L - M$ syllables following the accent (including the syllable that is directly associated with the accent that is being predicted). Each syllable is represented by one or more of the features POS, CSWP, CLS—if more than one feature is used, they are concatenated into a sequence of values. Given the context, S', the most likely pitch accent, a^*, is chosen where a^* maximizes the following equation:

$$a^* = \underset{a}{argmax} \begin{cases} P(a = unaccented|S') \\ P(a = high|S') \\ P(a = downstepped|S') \\ P(a = low|S') \end{cases} \quad (1)$$

Although the CLS feature is binary-valued and the CSWP feature takes on only a small number of values, the POS feature can take on over 40 values using the Brill tag-set, and hence data sparseness becomes an issue. To alleviate this problem we used the algorithm introduced in [5] to reduce the number of PoS tags by combining several tags together.

3.2. Dynamic Programming (DP)

In [5], we developed a technique for prediction of phrase breaks by using dynamic programming to measure the similarity between each of the training-set sentences and the test sentence. Phrase breaks were then predicted in the test sentence at the points where it was aligned to the best-matching training-set sentence. We used a similar technique in this work for predicting pitch accents: features are associated with each syllable in the training- and test-set sentences, the sentences are segmented into appropriate units, and DP is used to compare units. The

Feature Name	Description
POS	Current POS Tag
PPOS	Previous POS Tag
NPOS	Next POS Tag
OF	Orthographic Form (OF)
VI	Vowel Identity
CLS	Current Lexical Stress
PLS	Previous Syllable Lexical Stress
NLS	Next Syllable Lexical Stress
CSWP	Current Syllable Word Position
PSWP	Previous Syllable Word Position
NSWP	Next Syllable Word Position
CWNS	Current Word Number of Syllables
PWNS	Previous Word Number of Syllables
NWB	Number of Words From Beginning
NWE	Number of Words To End of sentence
BI	ToBI Break Index
NSCP	Number of Syllables in the Current Phrase
NWCP	Number of Words in the Current Phrase
NSSP	Number of Syllables from the Start of the Phrase
NSNP	Number of Syllables to the Next Phrase
NWSP	Number of Words from the Start of the Phrase
NWNP	Number of Words to the Next Phrase
K0	size of largest phrase ending at this word
K0A	as K0, but counting nodes
K1	true if a major phrase
K2	K0 for the *next* phrase at this level
K2A	K0A for the *next* phrase at this level
K3	true if the phrase is a conjoined S-phrase an (SBAR)
LPS	The largest phrase type ending at the current word
DNW	The distance (in nodes) to the next word

Table 1: Features that can be used for predicting pitch accents in TTS.

pitch accent of a syllable in a test sentence is determined using the top k matching sentences. If there is disagreement about the accent, it is assigned using either a majority vote or by the relative ranks of the matching sentences. The training sentences can be divided into a number of different units of varying size: sentences, prosodic phrases, words and individual syllables. The results of [8] showed that as the sequences grows in length, the likelihood of finding a close DP match decreases.

The algorithm was first evaluated using the same small set of features as were used for the N-gram experiments, configured in the same way as described in section 3.1. Initially, each feature was used individually, and the results of this experiment were used to guide a best-first search to construct and evaluate different combinations of features. The optimum value of k was found by direct search using values in the range $1, \ldots, 20$.

3.3. Decision Trees

Pitch accent prediction has previously been shown to be successful when using decision tree classifiers [3], [9], [11]. In these experiments, we used the same approach discussed in Section 3.2 in which each feature was first used on its own and then a best-first search approach used to find the optimal feature set. This process has the benefit of enabling us to analyse the contribution from each feature used in the process. The Weka implementation [12] of the C4.5 algorithm was used for these experiments, which allows control over a range of parameters for optimizing the feature combinations.

One of the algorithms presented in [11], which predicts pitch accents using text-based features, forms the foundation of the research presented here using the C4.5 classifier. Sun's classifier used the following features: VI, CLS, PLS, NLS, CSWP, PSWP, NSWP, CWNS, PWNS, POS, PPOS, NPOS, NWB and NWE. The C4.5 decision trees presented here extended this feature-set by incorporating the wider set of features outlined in Section 2.

4. Data and Evaluation Procedure

The algorithms were trained and tested using speaker F2B from the Boston Radio News Corpus [4]. This corpus contains utterances annotated with phonetic alignments (including lexical stress), POS, as well as prosodic phrasing, phrasal prominence, and boundary tones labelled using the ToBI system [10]. Initial algorithm optimization experiments were performed on just the training portion of the data, which was divided into a development training set and a development testing set. The final evaluations whose results are presented here were trained and tested on the full data set.

The *Balanced Error Rate* (BER) is the mean of the error-rates for each class, and is defined as

$$BER = \frac{1}{C}\sum_{i} \frac{(\sum_{j} M_{ij}) - M_{ii}}{\sum_{j} M_{ij}}. \qquad (2)$$

In equation 2, C is the number of classes and M is the $C \times C$ confusion matrix, i.e. $M(i, j)$ is the number of times that the true pitch accent of class c_i is mis-recognised as class c_j. Low and downstepped tones occur significantly less frequently in the data than unaccented and high syllables, and previous studies have shown that these two tones are usually harder to predict than high and unaccented tones [11]. The BER gives a more sensitive indication of an algorithm's performance as it gives equal weighting to each of the classes of pitch accent, regardless of the number of examples in each class.

Some pitch accents were labelled as uncertain in the Boston corpus when the annotators were unable to decide on which pitch accent to associate with a given syllable. These were not considered during the evaluation of these algorithms as they introduce noise into the evaluation process.

5. Results

5.1. N-grams

Firstly, experiments were conducted to find the optimal values for L and M as described in section 3.1. Using only the current syllable ($L = 1, M = 0$) achieved a BER of 57.43%, not significantly worse than that obtained using the current and previous syllable (BER = 57.16%) After optimising the set of POS tags used, this result was improved to 52.16%.

	Unit Level			
	Sentence	**Phrase**	**Word**	**Syllable**
Min BER	*63.3%*	*55.2%*	*54.5%*	*57.8%*

Table 2: The best BERs of the different levels of units used for predicting pitch accents with dynamic programming.

5.2. Dynamic Programming

It was important to choose an appropriate level of representation to use for classification: the possibilities were whole sentences, prosodic phrases, words or syllables. This was investigated by constructing models using each of these units and evaluating them on all the different combinations of features that were used in the N-gram experiments above: CLS, NLS, PLS, CSWP, K1, LPS, POS, NWSP and PSWP. Table 2 shows the results obtained using these four representations. Sentence units are the worst, as the probability of a good match of a test sentence with one from the training-set is low, as noted in [8]. Syllables perform better, but not as well as word or prosodic phrase units, probably due to their lack of any contextual information. Words, short sequences of syllables, work the best as they achieve the best balance of local and longer-range information. The best result of 54.5% was achieved when the syllable sequences were divided into words and represented by the syllable's current lexical stress (CLS) and the reduced POS tag (RPOS). Subsequent optimization experiments utilize phrases and words units.

In the previous section, only a small number of features were used to gain an initial estimate of the usefulness of different unit sizes. Subsequent experiments searched all possible feature sets using a controlled best-first search, in the same way as was described in [5]. The ten best performing feature sets were then used to optimize the value for k. All feature sets achieved the best performance when predictions were based on the two most similar sequences ($k = 2$). The optimum parameters found for this algorithm are summarised in Table 3.

Features	Level	k	**BER**
NSSP, CWNS, CLS, K1	Words	2	48.5%

Table 3: Optimal parameters for predicting pitch accents using DP and k-nn.

5.3. C4.5 Decision Trees

Our previous research [7] found that a C4.5 decision tree was the most successful algorithm for predicting prosodic phrase breaks, and previous techniques [3], [9], [11] have also shown that tree-based classifiers are well suited to this task. Section 3.3 introduced the feature set used in [11] for predicting pitch accents. This feature set was used as a benchmark in our experiments and achieved a BER of 55.55%.

Initial optimization of the algorithm was performed using the Sun feature set. This involved experimenting with a range of different values for the C4.5 parameters which are described and discussed fully in [8]. The best configurations of the algorithm parameters use binary splits for nominal attributes—such as POS tags and vowel identities—and use four-fold reduced error pruning instead of subtree raising to optimize the tree during its creation. The BER of 53.09% was achieved using this algorithm and the Sun feature set. Optimizing the full feature set was performed in the same way as it was for the dynamic programming approach outlined in Section 5.1 Table 4 presents the best feature set found for predicting pitch accents using C4.5 decision trees. The best results from this algorithm made use of a wide range of features that encapsulated information relating to each syllable's phonetic transcription, position (within the sentence, word and phrase), and a significant number of syntactic features. The word's orthographic form became a useful feature because the algorithm was able to make use of correlations between pitch accents and certain words.

Features	**BER**
CLS, CSWP, K0A, K1, K2, K2A, K3, NPOS, NSNP, NSSP, NWB, NWE, NWSP, OF, PLS, POS, PSWP, PWNS, RPOS, VI	46.64%

Table 4: Our optimized feature set for predicting pitch accents using the C4.5 decision trees.

		Classified			
		unacc.	**high**	**low**	**downst.**
Actual	**unacc.**	1987	211	2	97
	high	91	529	5	106
	low	11	38	3	30
	downst.	31	76	2	113

Table 5: Confusion matrix for the C4.5 algorithm using the features CLS, CSWP, K0A, K1, K2, K2A, K3, NPOS, NSNP, NSSP, NWB, NWE, NWSP, OF, PLS, POS, PSWP, PWNS, RPOS & VI and algorithm parameters -R -B -N 4

The confusion matrix for this approach, presented in Table 5, shows the main failing of this algorithm is its poor ability to predict low accents. When the error rate for low accents is ignored, the balanced error for three other classes drops to 30.1%

5.4. Statistical Significance Tests

McNemar's Test [2] was applied separately to results for each pitch accent type to account for the unbalanced distribution of accent classes. Table 6 shows the p-values found from application of McNemar's test to the results [1]. The first row of Table 6 shows the DP/k-nn algorithm is significantly better at predicting low accents than the C4.5 approach, but that the C4.5 method is better at assigning unaccented symbols. Rows two and three show that the parse features and orthographic form aid the C4.5 approach by significantly improving the prediction of downstepped accents. The fourth row compares the difference between the feature set presented here with that used by [11]. Again, there is a clear benefit for predicting downstepped syllables, but the Sun feature set is significantly better at predicting unaccented syllables.

6. Discussion

This paper has presented a number of different techniques for automatically predicting pitch accents from natural text. Table 7 summarizes all of the algorithms developed here for this task. The N-gram approach was the least successful algorithm. As

[1] A p-value of x indicates that the algorithms are significantly different at the $100x\%$ level i.e. it indicates that if there were actually no significant difference between the two algorithms, the results found would have occurred by chance on only $100x\%$ of occasions.

A_1	A_2	$P - value$			
		Downstepped	**High**	**Low**	**Unaccented**
C4.5	DP/k-nn	0.6137	0.0584	0.0352 ($A_1 < A_2$)	0.0000 ($A_1 > A_2$)
C4.5	C4.5 ex. parse features	0.0026 ($A_1 > A_2$)	0.3284	1.0000	0.4709
C4.5	C4.5 ex. orthographic	0.0000 ($A_1 > A_2$)	1.0000	1.0000	0.1750
C4.5	C4.5 (Sun features only)	0.0000 ($A_1 > A_2$)	0.7898	1.0000	0.0261 ($A_1 < A_2$)

Table 6: Statistical significance comparisons of some of the algorithms presented in this paper. Where there is a statistical difference, the better performing algorithm is specified in brackets. For example, $A_1 > A_2$, means that A_1 performed better than A_2.

Algorithm	**Features**	**BER**
N-grams	CLS, CSWP, NWSP, PLS & RPOS	52.6%
DP/2-nn	NSSP, CWNS, CLS & K1	48.5%
C4.5 -R -B -N 4	CLS, CSWP, K0A, K1, K2, K2A, K3, NPOS, NSNP, NSSP, NWB, NWE, NWSP, OF, PLS, POS, PSWP, PWNS, RPOS & VI	46.6%

Table 7: Summary of the pitch accent algorithms presented in this paper.

well as an overall high error rate, this technique also suffered from an inability to predict low accents, a problem previously observed in [11]. Using a reduced POS tagset helped to increase the performance, as well as providing additional features for subsequent models.

A more successful method used DP matching to make predictions by analogy with the samples in the training data. This technique assigned pitch accents to unseen syllables based on the most similar samples in the training data. The best results were achieved when matching was performed on the syllable sequences for each of the words.

C4.5 decision trees emerged as the most successful approach for predicting pitch accents. The best results from this algorithm made use of a wide range of features that encapsulated information relating to each syllable's phonetic transcription, position (within the sentence, word and phrase), and a significant number of syntactic features. The word's orthographic form became a useful feature because the algorithm was able to make use of correlations between pitch accents and certain words.

A significant finding of this study was the benefit that syntactic parse features contributed to the quality of pitch accent prediction algorithms. Just as they were shown to improve phrase break prediction in [7], their ability to model the relationship between syntactic and prosodic phrase structures helped to reduce the number of errors for this task.

None of the algorithms presented here was able to predict low accents successfully. However, given that these do not occur much in speech, it is a moot point whether working on improvements to classifying this accent type is worthwhile. A more interesting line of research at this point might be to deliberately introduce incorrect accent types into synthetic speech and see which errors were perceived as degrading the speech quality, and which were not noticed, or were harmless.

7. References

[1] E. Brill. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[2] L. Gillick and Stephen Cox. Some Statistical Issues in the Comparison of Speech Recognition Algorithms. In *IEEE Conference of Acoustics, Speech and Signal Processing*, pages 532–535, 1989.

[3] Julia Hirschberg. Pitch accent in context: Predicting intonational prominence from text. *Artificial Intelligence*, 63(1–2), 1993.

[4] M. Ostendorf, P. J. Price, and S. Shattuck-Hufnagel. The Boston University radio news corpus. Technical Report ECS-95-001, Boston University, 1995.

[5] I. H. Read and S. J. Cox. Using Part-of-Speech for Predicting Phrase Breaks. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Seoul, South Korea, October 2004.

[6] I. H. Read and S. J. Cox. Stochastic and Syntactic Techniques for Predicting Phrase Breaks. In *Proc. 9th European Conference on Speech Communicaiton and Technology*, Lisbon, Portugal, September 2005.

[7] Ian Read and Stephen Cox. Stochastic and Syntactic Techniques for Predicting Phrase Breaks. *Computer Speech and Language*, 21(3):519–542, July 2007.

[8] Ian Harvey Read. *Approaches to Prosody Prediction for Text-To-Speech Synthesis*. PhD thesis, University of East Anglia, 2007.

[9] K Ross and M Ostendorf. Prediction of abstract prosodic labels for speech synthesis. *Computer Speech and Language*, 10:155–185, 1996.

[10] K. Silverman, M. E. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg. ToBI: a standard for labelling English prosody. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 867–870, Banff, Canada, September 1992.

[11] Xuejing Sun. Pitch Accent Prediction Using Ensemble Machine Learning. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Denver, Colorado, Sept. 16-20 2002.

[12] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, second edition, 2005.