VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematics, Statistics and Computer Science
Computer Science

Genetic Programming for Automatic Stress Detection in Spoken English

Huayang Xie, Mengjie Zhang, Peter Andreae

Technical Report CS-TR-06/4
January 2006

School of Mathematics, Statistics and Computer Science
Victoria University
PO Box 600, Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematics, Statistics and Computer Science
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341, Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

Genetic Programming for Automatic Stress Detection in Spoken English

Huayang Xie, Mengjie Zhang, Peter Andreae

Technical Report CS-TR-06/4
January 2006

Abstract

This paper describes an approach to the use of genetic programming (GP) for the automatic detection of rhythmic stress in spoken New Zealand English. A linear-structured GP system uses speaker independent prosodic features and vowel quality features as terminals to classify each vowel segment as stressed or unstressed. Error rate is used as the fitness function. In addition to the standard four arithmetic operators, this approach also uses several other arithmetic, trigonometric, and conditional functions in the function set. The approach is evaluated on 60 female adult utterances with 703 vowels and a maximum accuracy of 92.61% is achieved. The approach is compared with decision trees (DT) and support vector machines (SVM). The results suggest that, on our data set, GP outperforms DT and SVM for stress detection, and GP has stronger automatic feature selection capability than DT and SVM.

Keywords Speech recognition, stress detection, genetic programming, decision trees, support vector machines

Author Information

All authors are academic staff members and postgraduate students in computer science in the School of Mathematics, Statistics and Computer Science, Victoria University of Wellington, New Zealand.

1 Introduction

Stress is a form of prominence in spoken language. Usually, stress is seen as a property of a syllable or of the vowel nucleus of the syllable. There are two types of stress in English. *Lexical stress* refers to the relative prominences of syllables in individual words. *Rhythmic stress* refers to the relative prominences of syllables in longer stretches of speech than an isolated word. When words are used in utterances, their lexical stress may be altered to reflect the rhythmic (as well as semantic) structure of the utterance.

As English becomes more and more important as a communication tool for people from all countries, there is an ever increasing demand for good quality teaching of English as a Second Language (ESL). Learning English well requires lots of practice and a great deal of individualised feedback to identify and correct errors. Providing this individualised feedback from ESL teachers is very expensive, therefore computer software that could help ESL learners to speak as a native speaker is highly desirable. Properly placing rhythmic stress is one of the important steps for teaching ESL students to have good speech production. Thus to be able to automatically detect the rhythmic stress patterns in students' speech becomes a really important functionality in this kind of computer software.

There are a number of prosodic (sometimes referred to as 'suprasegmental') features that relate to stress. Thus the perception of a syllable as stressed or unstressed may depend on its relative duration, its amplitude and its pitch. Duration is simply how long the syllable lasts. Amplitude relates to the perceived loudness of the syllable, and is a measure of its energy. Pitch is the perceptual correlate of the fundamental frequency (F_0) of the sound signal, i.e. the rate of vibration of the vocal folds during voiced segments.

A further correlate of stress is the quality of the vowel in a syllable. Vowels are split into *full* vowels and *reduced* vowels in terms of the quality based on the configuration of the tongue, jaw, and lips [1]. Full vowels tend to be more peripheral, and appear in both stressed syllables and unstressed syllables [2]. Reduced vowels, including /@/ and /I/ in New Zealand English, tend to be more central, and are only associated with unstressed syllables. Therefore, vowel quality is not a completely reliable indicator of stress [2].

In order to automatically detect rhythmic stress, prosodic features and vowel quality features as two main sets of features have been studied by many researchers using machine learning algorithms.

Waibel [3] used duration, amplitude, pitch, and spectral change to identify rhythmically stressed syllables. A Bayesian classifier, assuming multivariate Gaussian distributions, was adopted and 85.6% accuracy was achieved. Jenkin and Scordilis [4] used duration, energy, amplitude, and pitch to classify vowels into three levels of stress — primary, secondary, and unstressed. Neural networks, Markov chains, and rule-based approaches were adopted. The best overall performance was 84% by using Neural networks. Rule-based systems performed worse with 75%. Van Kuijk and Boves [5] used duration, energy, and spectral tilt to identify rhythmically stressed vowels in Dutch — a language with similar stress patterns to those of English. A simple Bayesian classifier was adopted, on the grounds that the features can be jointly modelled by a N-dimensional normal distribution. The best overall performance achieved was 68%. Our previous work [6] used duration, amplitude, pitch and vowel quality to identify rhythmically stressed vowels. Decision trees and support vector machines were applied and the best accuracy, 85%, was achieved by support vector machines.

However, the accuracies of the automatic stress detection from the literature are not high enough to be useful for a commercial system. The automatic rhythmic stress detection remains a challenge to speech recognition.

Genetic programming (GP) has grown very rapidly and has been studied widely in many areas since the early 1990s. Conrads et al. [7] demonstrated that GP could find programs that were able to discriminate certain spoken vowels and consonants without pre-processing speech signals. However, there are only a few studies using GP in the automatic speech recognition and analysis area. Most current research on automatic rhythmic stress detection uses other machine learning algorithms rather than GP.

1.1 Goals

This paper aims to use GP to develop an approach to automatic rhythmic stress detection in spoken New Zealand (NZ) English. The approach will be examined and compared with other machine learning techniques such as decision tress (DT) and support vector machines (SVM) on a set of NZ English utterances. Specifically, we investigate:

- how GP can be used to construct an automatic rhythmic stress detector,
- whether GP outperforms DT and SVM on the automatic problem, and
- whether GP has a stronger capability of handling irrelevant features than DT or SVM.

The remainder of the paper is organised as follows: section 2 describes the GP approach; section 3 presents the experiment design; section 4 provides experiment results, and section 5 draws conclusions and discusses possible future work.

2 GP Adapted to Stress Detection

A linear-structured GP system [8] is adopted to construct an automatic rhythmic stress detector in this study. This section addresses: 1) the feature extraction and normalisation; 2) the terminal sets; 3) the function set; 4) the fitness function; and 5) the genetic parameters and termination criteria.

2.1 Feature Extraction and Normalisation

As prosodic features and vowel quality features are recognised as the two main sets of features for automatic stress detection, we also use both of them in the approach. For each of the prosodic parameters (duration, amplitude, pitch), there are many alternative measurements that can be extracted, and also many ways of normalising the features in order to reduce variation due to differences between speakers, recording situations or utterance contexts. Vowel quality features are more difficult to extract. The subsections below describe the details of the feature extraction and normalisation.

2.1.1 Duration Features.

The absolute duration of a vowel segment is easily calculated directly from the hand labelled utterances since the start and end points of the vowel segment are clearly marked. Three different levels of normalisation are applied to the directly calculated absolute duration of a vowel segment. The first level normalisation aims to reduce the impact of the different speech rate of speakers. The second level normalisation aims to reduce the effects of the intrinsic duration properties of the vowel. Both narrow and broad methods are considered. The narrow method is to normalise the vowel segment duration by the average duration for that vowel type, as measured in the training data set. The broad method is to cluster the 20 vowel types into three categories (short vowel, long vowel and diphthong) and to normalise vowel segment durations by the average duration of all vowels in the relevant category. The third level normalisation aims to reduce the effects of the local fluctuations in speech rate within the utterance. Based on the three levels of normalisation, we have five duration features for each vowel segment:

- *D_1:* the absolute duration normalised by the length of the utterance.
- *D_2:* D_1 further normalised by the average duration of the vowel type.
- *D_3:* D_1 further normalised by the average duration of the vowel category.
- *D_4:* D_2 further normalised by a weighted average duration of the immediately surrounding vowel segments.
- *D_5:* D_3 further normalised by a weighted average duration of the immediately surrounding vowel segments.

2.1.2 Amplitude Features.

The amplitude of a vowel segment can be measured from the speech signal, but since amplitude changes during the vowel, there are a number of possible measurements that could be made — maximum amplitude, initial amplitude, change in amplitude, *etc.* A measure commonly understood to be a close correlate to the perception of amplitude differences between vowels is the root mean square (RMS) of the amplitude values across the entire vowel. This is the measure chosen as the basis of our amplitude features. Two levels of normalisations are applied to the RMS amplitude value across a vowel segment. The first level normalisation aims to reduce the effects of speaker voice volume differences and recording condition differences. It is done by normalising the RMS amplitude of each vowel segment against the overall RMS amplitude of the entire utterance. The second level normalisation aims to reduce the effects of changes in amplitude across the utterance. We normalise the vowel amplitude against a weighted average amplitude of the immediately surrounding vowel segments.

- *A_1:* the RMS amplitude of each vowel segment normalised by the overall RMS amplitude of the entire utterance.
- *A_2:* A_1 further normalised by a weighted average amplitude of the immediately surrounding vowel segments.

2.1.3 Pitch Features.

Pitch is calculated by measuring F_0 of the speech signal. Five pitch features of a vowel segment are computed, including the mean pitch value of the vowel segment, the pitch values at the start and at the end points of the vowel segment, and the minimum and maximum pitch values of the vowel segment. In order to reduce the effects of speaker differences caused by their different physiologies, we normalise the five pitch features of a vowel segment over the mean pitch of the entire utterance. In addition, based on the five normalised pitch features, we compute five other features that are intended to capture pitch changes over the vowel segment.

- *P_1:* the mean pitch value of the vowel normalised by the mean pitch of the utterance.
- *P_2:* the pitch value at the start point of the vowel normalised by the mean pitch of the utterance.
- *P_3:* the pitch value at the end point of the vowel normalised by the mean pitch of the utterance.

- P_4*:* the maximum pitch value of the vowel normalised by the mean pitch of the utterance.

- P_5*:* the minimum pitch value of the vowel normalised by the mean pitch of the utterance.

- P_6*:* the difference between the normalised maximum and minimum pitch values — a negative value indicates a falling pitch and a positive value indicates a rising pitch.

- P_7*:* the magnitude of P_6, which is always positive.

- P_8*:* the sign of P_6 — 1 if the pitch "rises" over the vowel segment, -1 if it "falls", and 0 if it is "flat".

- P_9*:* a boolean attribute — 1 if the pitch value at either the start point or the end point of the vowel segment cannot be detected, otherwise -1.

- P_{10}*:* a boolean attribute — 1 if the vowel segment is too short to compute meaningful mean, minimum, or maximum values, otherwise -1.

2.1.4 Vowel Quality Features.

Since there is some flexibility in the formation of a vowel, there will in fact be a range of articulator parameter values that correspond to the same vowel. Therefore, vowel quality features are more difficult to extract. Pre-trained Hidden Markov Models (HMMs) phoneme models are used to analyse vowel segments and extract measures of vowel quality [9]. The algorithm is illustrated in figure 1 and outlined below.

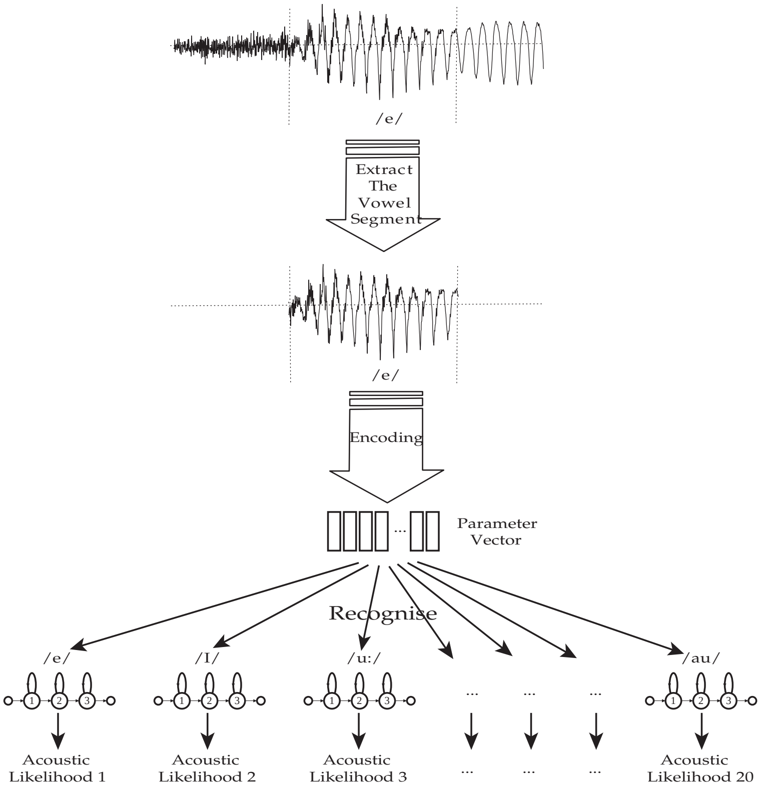

Figure 1: Vowel quality features processing.

Step 1 Extract vowel segments from the hand labelled utterance.

Step 2 Encode each vowel into a sequence of acoustic parameter vectors, using a 15ms Hamming window with a step size (frame period) of 11ms. These parameters consist of 12 MFCC features and the 0'th cepstral coefficient with their first and second order derivatives.

Step 3 Feed the parameter vector sequence into the 20 pre-trained HMM vowel recognisers to obtain 20 normalised acoustic likelihood scores. Each score is the geometric mean of the acoustic likelihoods of all frames in the segment, as computed by the HMM recogniser. The scores are likelihoods that reflect how well the segment matches the vowel type of the HMM.

Step 4 Find the score of the labelled vowel type S_e, the maximum score of any full vowel phoneme S_f and the maximum score of any reduced vowel phoneme S_r from the above 20 scores.

Step 5 We then compare the scores of the best matching full vowel and the best matching reduced vowel to the score of the labelled vowel. We compute four features, two of which measure the difference between the likelihoods, and two measure the ratio of the likelihoods. In each case, we take logarithms to reduce the spread of values:

$$R_d = \begin{cases} -log(S_r - S_e) & \text{if } S_e < S_r \\ 0 & \text{if } S_e = S_r \\ log(S_e - S_r) & \text{if } S_e > S_r \end{cases} \tag{1}$$

$$F_d = \begin{cases} -\log(S_f - S_e) & \text{if } S_e < S_f \\ 0 & \text{if } S_e = S_f \\ \log(S_e - S_f) & \text{if } S_e > S_f \end{cases} \tag{2}$$

$$R_r = \log(S_e/S_r) = \log S_e - \log S_r \tag{3}$$

$$F_r = \log(S_e/S_f) = \log S_e - \log S_f \tag{4}$$

Step 6 We also compute a boolean vowel quality feature, T, to deal with cases where the vowel segment is so short that F or R cannot be calculated. If the vowel segment is less than 33ms, which is the minimum segment duration requirement of the HMM recognisers, then the value of this attribute will be 1. Otherwise, -1. If this value is 1, we set F and R to 0.

2.2 Terminal Sets

All the features introduced in section 2.1 were organised into three terminal sets. Terminal set I consists of 17 prosodic features (five duration features, two amplitude features, and 10 pitch features). Terminal set II consists of five vowel quality features. Terminal set III consists of all features combined from the prosodic and vowel quality features. Features whose values have not been explicitly mentioned in previous sections are floating point numbers with precisions up to seven digits. In each terminal set, we also include real-valued constants in the range $[-1.0, 1.0]$.

2.3 Functions

The function set contains not only the four standard arithmetic functions, but also several other arithmetic and trigonometric functions and conditional functions, as shown in equation 5.

$$FuncSet = \{abs, sqrt, cos, sin, +, -, *, /, iflt, ifpr, ifnr\} \tag{5}$$

Each of the first four mathematical operators takes a single argument. The *abs* function returns the absolute value of the argument. The protected *sqrt* function returns the square root of the argument. The *cos* or *sin* functions return the cosine or sine values of the argument respectively. Each of the $+$, $-$, $*$, and / operators takes two arguments. They have their usual meanings except that the / operator is protected division that returns 0 if its second argument is 0. The three conditional functions each takes two arguments. The *iflt* function returns 1 if the first argument is less than the second one, otherwise 0. The *ifpr* function returns the second argument if the first argument is positive, otherwise does nothing. The *ifnr* function returns the second argument if the first argument is negative, otherwise does nothing. Note that there is a redundancy in that the conditional functions could be expressed in terms of each other. There is a trade off between the increased breadth of the search space resulting from having redundant functions, and the more complex programs (hence a deeper search of the search space) resulting from a minimal set of non-redundant functions. We believe that the smaller programs that are possible with the expanded function set more than compensates for the broader search space.

2.4 Fitness Function

Error rate is used as the fitness function to evaluate programs. The classification error rate of a program is the fraction of fitness cases in the training data set that are incorrectly classified by the program. Rhythmic stressed vowel segments and unstressed vowel segments are both treated as important so that neither class is weighted over the other. In our data set, class *stressed* is represented by 1 and class *unstressed* is represented by -1. If a program's output is greater than or equal to 0, then the output is counted as a class *stressed* output. Otherwise, it is counted as a class *unstressed* output.

2.5 Parameters and Termination Criteria

In this GP system the learning process uses the tournament selection mechanism with size four and the crossover, mutation and reproduction operators. It is worth noting that in this GP system, the crossover and mutation operators are independent in that the mutation operator can be applied regardless of whether a tournament winner has also been selected for crossover, so that the sum of the crossover rate and the mutation rate may be more than 100%. The selection of parameter values used in this study is shown in Table 1. These values were obtained through prior empirical research. The unusually high mutation rates were found to be the most helpful for this problem.

The learning/evolutionary process is terminated when either of the following criteria is met:

- The classification problem has been solved on the training data set, that is, all vowel segments in the training set have been correctly classified, with the fitness of the best program being zero.

- The number of generations reaches the pre-defined maximum number of generations without improvement (*max-gwi*). In this study, max-gwi is set at 200, which means

Table 1: Parameters used for GP training for three terminal sets.

Parameter Kind	Parameter Name	I	II	III
Search Parameters	Population size	1024	1024	1024
	max-gwi	200	200	200
Genetic Parameters	Crossover rate	71%	57%	47%
	Mutation rate	97%	87%	83%
Program Parameters	Initial program size	80	80	80
	Max program size	256	256	256

that, if fitness values have had no improvement within 200 generations, the learning process will terminate.

3 Experiment Design

The system uses a data set collected by the School of Linguistics and Applied Language Studies at Victoria University of Wellington. The date set contains 60 utterances of ten distinct English sentences produced by six female adult NZ speakers, as part of the NZ Spoken English Database (`www.vuw.ac.nz/lals/nzsed`). The utterances were hand labelled at the phoneme level, including the time stamps of the start and the end of a phoneme segment and the phoneme label. Further, each vowel was labelled as rhythmic *stressed* or *unstressed*. There were 703 vowels in the utterances, of which 340 are marked as stressed and 363 are marked unstressed. Prosodic features and vowel quality features of each vowel segment are calculated from the hand labelled utterances.

Three experiments were conducted on the three terminal sets respectively. For each terminal set, since the data set was relatively small, a 10-fold cross validation method for training and testing the automatic rhythmic stress detectors was applied. In addition, the training and testing process was repeated ten times, that is, 100 runs of training and testing procedures were made in total for each terminal set. The average classification accuracy of the best program in each experiment is calculated from the outputs of the 100 runs.

In addition, we investigate whether scaling the feature values in the three terminal sets to the range $[-1, 1]$ results in better performance.

We also compare our GP approach with the C4.5 [10] decision tree (DT) system and a SVM system (LIBSVM [11]) on the same set of data. The SVM system uses an RBF kernel and a C parameter of 1.0.

4 Results and Analysis

4.1 Detection Performance

4.1.1 Terminal Set I.

Table 2 shows system performance of Terminal Set I. Based on the average of 100 runs, GP achieved the best accuracy on the test set (91.9%). The accuracy of GP was 11.5% and 12.2% higher than that of DT and SVM respectively on unscaled data, and was 11.0% and 8.4% higher on scaled. There is little evidence showing any impact of using scaled data on GP and DT. However, there is an improvement of 3.5% by using scaled data for SVM.

Table 2: Accuracy(%) for the terminal set I.

	GP	DT	SVM
Unscaled	91.9	80.4	79.7
Scaled	91.6	80.6	83.2

4.1.2 Terminal Set II.

Table 3 shows the experiment results of Terminal Set II. GP also achieved the best accuracy of 85.4%. The accuracy of GP was 5.7% and 6.3% higher than that of DT and SVM respectively on unscaled data, and was 5.7% and 4.1% higher on scaled. There is also little evidence to show any impact of scaled data.

Table 3: Accuracy(%) for the terminal set II.

	GP	DT	SVM
Unscaled	85.4	79.7	79.1
Scaled	84.6	78.9	80.5

4.1.3 Terminal Set III.

Table 4 shows the results of Terminal Set III, which combines all the features used in Terminal Set I and Terminal Set II. Again, the best accuracy of 92.6% was achieved by GP. GP outperformed DT and SVM by 12.1% and 10.8% on unscaled data respectively, and by 12.6% and 10.6% on scaled. For all three systems, accuracies on scaled data were invariably higher than those on the unscaled data but the differences were very small.

Table 4: Accuracy(%) for the terminal set III.

	GP	DT	SVM
Unscaled	92.0	79.9	81.3
Scaled	92.6	80.1	82.0

Comparing the results of all three terminal sets, we obtained the following observations.

- On all terminal sets, regardless of whether the data are scaled or unscaled, accuracy of GP is consistently and significantly higher than that of DT and SVM. This indicates that GP is more effective than DT and SVM on the automatic stress detection problem on our data set.

- For all systems, Terminal Set I consistently returns higher accuracies than terminal set II. This indicates that either prosodic features are more accurate than vowel quality features, or that vowel quality feature extraction needs to be further improved.

- Maximising the coverage of features (using Terminal Set III) resulted in some improvement for GP, but not for DT and SVM. Since terminal set III has the most complete set of features, it is likely that not all of them are necessary in detecting stress. Therefore the difference in performance of Terminal Set I and Terminal Set III could be used as an indication of how robust a system is at handling unnecessary and irrelevant features. Except for GP, both DT's and SVM's best accuracy scores dropped on Terminal Set III, therefore GP is the most robust algorithm among the three at handling unnecessary and irrelevant features on our data set.

4.2 Feature Impact Analysis

The top thirty programs in each run were analysed and the average *impact* of each terminal input in programs was computed as a percentage, as shown in Tables 5 and 6. The impact of a terminal input refers to the change of the performance of a program if all occurrences of the terminal input are removed from the program.

Table 5: Impact analysis for prosodic features.

Unscaled		Scaled	
Input	Average Impact(%)	Input	Average Impact(%)
D_5	27.7	D_5	28.2
D_3	27.4	D_3	16.5
D_2	15.1	D_4	13.4
D_4	13.7	D_1	12.9
D_1	8.3	D_2	7.8
A_1	2.3	A_2	1.4
A_2	1.1	A_1	1.4
P_2	1.1	P_5	0.6
P_1	0.7	P_3	0.6
P_3	0.7	P_2	0.6
P_8	0.6	P_4	0.5
P_4	0.4	P_1	0.4
P_5	0.3	P_8	0.4
P_{10}	0.3	P_7	0.2
P_7	0.3	P_{10}	0.2
P_6	0.1	P_9	0.2
P_9	0.1	P_6	0.1

Table 5 shows the impact of the prosodic features. The patterns of the impact of prosodic features are similar on both unscaled data and scaled data. Three broad bands of impact emerged as high (above 5%, including all duration features), medium (1% to 5%, including amplitude features), and low (under 1% including all pictch features), corrsponding exactly with the three feature categories - duration, amplitude and pitch. This indicates duration has a bigger impact than amplitude while pitch has the smallest impact. On both unscaled and scaled data set, D_5 and D_3 are ranked as the first and second, indicating that normalisations of a duration feature over the average duration of a vowel category is better than that over the average duration of a vowel type.

The ranking of duration, amplitude and pitch in terms of impact in this study matches the result in [6]. However, only one experiment was conducted in this study whereas seven experiments with various combinations of the feature sets were conducted in [6], where DT and SVM were used. This suggests that: 1) GP has stronger feature selection ability than DT and SVM on the problem; 2) GP can automatically handle a large number of features; and 3) GP can automatically select features that are only important to a particular domain.

As shown in Table 6, R_d and R_r have a much larger impact than F_d, F_r, and T on both unscaled and scaled data. On unscaled data R_d's impact (31.9%) is larger than R_r's impact (20.7%), whereas on scaled data the two features display a similar impact. The results suggest that the reduced vowel quality features are far more useful than full vowel quality features, regardless of whether differences or ratios are used.

Table 6: Impact analysis for vowel quality features.

Unscaled		Scaled	
Input	Average Impact(%)	Input	Average Impact(%)
R_r	31.9	R_d	21.0
R_d	20.7	R_r	19.9
F_r	2.8	T	4.5
T	1.5	F_d	0.76
F_d	0.34	F_r	0.38

5 Conclusions and Future Work

The goal of this paper was to develop an approach to using GP for automatic rhythmic stress detection in spoken NZ English. A range of prosodic and vowel quality features were calculated, normalised and/or scaled from vowel segments in speech. The approach was tested on 60 female adult utterances. A maximum average accuracy of 92.61% was achieved by our GP system.

The results strongly support the use of GP to construct a more effective automatic rhythmic stress detector than DT and SVM. Furthermore, according to our data set, GP is more robust at handling large numbers of unnecessary features and maintaining high performance than DT and SVM. GP also has a stronger automatic feature selection ability than DT and SVM.

In addition, prosodic features appear to be more accurate in detecting stress than vowel quality features, duration features being specially identified as the most important features. If using vowel quality, reduced vowel quality features are more useful than full vowel quality features.

In future work, we will further analyse the GP programs to understand the specific relationship amongst the feature terminals and the perceived stressed and unstressed vowels in order to determine whether the generated GP program with/without adapting can be applied to any other kind of data sets. We are also planning to investigate the possibility of having GP automatically perform higher level normalisations of the prosodic features and calculate vowel quality features directly from acoustic likelihoods in order to erase the limitation of the manual pre-process of the features.

Acknowledgment

The data set used in the paper was provided by Dr Paul Warren of the School of Linguistics and Applied Language Studies at Victoria University of Wellington.

References

[1] Ladefoged, P.: Three Areas of experimental phonetics. Oxford University Press, London (1967)

[2] Ladefoged, P.: A Course in Phonetics. third edn. Harcourt Brace Jovanovich, New York (1993)

[3] Waibel, A.: Recognition of lexical stress in a continuous speech system - a pattern recognition approach. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Tokyo, Japan (1986) 2287–2290

[4] Jenkin, K.L., Scordilis, M.S.: Development and comparison of three syllable stress classifiers. In: Proceedings of the International Conference on Spoken Language Processing, Philadelphia, USA (1996) 733–736

[5] van Kuijk, D., Boves, L.: Acoustic characteristics of lexical stress in continuous speech. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Volume 3., Munich, Germany (1999) 1655–1658

[6] Xie, H., Andreae, P., Zhang, M., Warren, P.: Detecting stress in spoken English using decision trees and support vector machines. Australian Computer Science Communications (Data Mining, CRPIT 32) **26** (2004) 145–150

[7] Conrads, M., Nordin, P., Banzhaf, W.: Speech sound discrimination with genetic programming. In: Proceedings of the First European Workshop on Genetic Programming. (1998) 113–129

[8] Francone, F.D.: Discipulus owner's manual (2004)

[9] Xie, H., Andreae, P., Zhang, M., Warren, P.: Learning models for English speech recognition. Australian Computer Science Communications (Computer Science, CRPIT 26) **26** (2004) 323–330

[10] Quinlan, J.: C4.5: Programs for machine learning. Morgan Kaufmann (1993)

[11] Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. `http://www.csie.ntu.edu.tw/$^\sim$cjlin/papers/libsvm.pdf` (2003)

[12] Koza, J.R.: Genetic Programming — On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)

[13] Dy, J.G., Brodley, C.E.: Feature selection for unsupervised learning. Journal of Machine Learning Research **5** (2004) 845–889