Recognition of Tones in YorÙbÁ Speech: Experiments With Artificial Neural Networks

Ọdẹ́túnjí Àjàdí ỌDẹ́lọBí

Ọbáfẹ́mi Awólọ́wọ̀ University, Ilé-Ifẹ̀, Nigeria,
oodejobi@oauife.edu.ng, oodejobi@yahoo.com

Summary. The speech recognition technology has been applied with success to many Western and Asian languages. Work on African languages still remains very limited in this area. Here a study into automatic recognition of the Standard Yoruba (SY) language tones is described. The models used fundamental frequency profile of SY syllables to characterize and discriminate the three Yoruba tones. Tonal parameters were selected carefully based on linguistic knowledge of tones and observation of acoustic data. We experimented with Multi-layered Perceptron (MLP) and Recurrent Neural Network (RNN) models by training them to classify feature parameters corresponding to tonal patterns. The results obtained exhibited good performances for the two tone recognition models, although the RNN achieved accuracy rates which are higher than that of the MLP model. For example, the outside tests for the H tone, produced a recognition accuracy of 71.00 and 76.00% for the MLP and the RNN models, respectively. In conclusion, this study has demonstrated a basic approach to tone recognition for Yoruba using Artificial Neural Networks (ANN). The proposed model can be easily extended to other African tone languages.

1 Introduction

The application of Artificial Neural Networks (ANN) to the processing of speech has been demonstrated for most Western and Asian languages. However, there is limited work reported on African languages. In this chapter we provide the background to the application of Artificial Neural Networks (ANN) to the processing of African language speech. We use the problem of tone recognition for the Standard Yorùbá (SY) language as a case study. This presentation has two aims (1) to provide background materials for research in tone language speech recognition, (2) to motivate and provide arguments for the application of ANN to the recognition of African languages. To achieve the stated aims, we demonstrate the design and implementation of Multilayer Perceptron (MLP) and Recurrent Neural Network (RNN) in the recognition of SY tones.

Ọdẹ́túnjí Àjàdí ỌDẹ́lọBí: *Recognition of Tones in YorÙbÁ Speech: Experiments With Artificial Neural Networks*, Studies in Computational Intelligence (SCI) **83**, 23–47 (2008)
www.springerlink.com

Tone languages, such as Yorùbá and Mandarin, differ from non-tone languages, such as English and French [20, 71]. In non-tone language, lexical items are distinguished by the stress pattern on the syllables that constitute an utterance. For example, the English words ***re***cord (verb) and re***cord*** (noun) differ in syntactic class and meaning because of the stress pattern on their component syllables. In the verb ***re***cord the first syllable is stressed. In the noun re***cord*** the second syllable is stressed.

In tone languages, tone, rather than stress, is used to distinguish lexical items. The tones are associated with the individual syllables in an utterance. For example, the following mono-syllabic Yorùbá words: *bí́* (H) [to give birth], *bi* (M) [to ask], *bì* (L) [to vomit] differ in meaning because of the tone associated with each syllable. A high tone is associated with the first monosyllabic word, i.e. *bí*. The second and third words carry the mid and low tones, respectively. Most tone languages have distinguishable tones and the number of tones vary for different languages, e.g. two for Hausa and Igbo, four for Mandarin (plus a neural tone), five for Thai and nine for Cantonese [35]. Standard Yorùbá has three phonological tones [1].

Two important features of tone languages make them an interesting subject of research in the area of speech recognition. First, tones are associated with syllables which are unambiguous speech units. Second, each tone, in isolated utterance, has a unique fundamental frequency (f_0) curve. Although the f_0 curves are affected by their context in continuous speech, they can still be easily recognised in speeches articulated at moderate or slow speaking rates. The complexities of speech recognition for tone languages can be reduced considerably if these two features are exploited in the design of speech recognition system. It is important to note that the recognition of tones is a major step in the recognition of speech in tone languages [15].

In speech signal, the timing, intensity and the fundamental frequency (f_0) dimensions contribute, in one way or the other, to the recognition and perception of tones. However, the fundamental frequency (f_0) curve has been shown to be the most influential acoustic correlate of tone in tone languages [54]. The possible application of tone recognition system such as the one presented here include the following:

- Recognition of SY monosyllabic utterances
- A component in a system for the automatic segmentation of continuous speech into syllables
- A component in a system for the automatic syllable level speech database annotation
- Application in automatic stylisation of tone f_0 curves

In Sect. 2 we give a brief description of the Standard Yorùbá language. Section 3 presents background to Automatic Speech Recognition (ASR) technology. The Hidden Markov Model (HMM) is the most popular method applied in ASR, hence we provide a detailed review of work on HMM in Sect. 4. Section 5 contains a literature review on the application of ANN to

speech and tone recognition. The data used for developing the models presented in this chapter is presented in Sect. 6. The tone recognition framework developed in this work is presented in Sect. 7. Experiments, results and discussion on the developed tone recognition models are presented in Sect. 8. Section 9 concludes this Chapter.

2 A Brief Description of the Standard Yorùbá Language

Yorùbá is one of the four major languages spoken in Africa and it has a speaker population of more than 30 million in West Africa alone [17, 62]. There are many dialects of the language, but all speakers can communicate effectively using Standard Yorùbá (SY). SY is used in language education, mass media and everyday communication. The present study is based on the SY language.

The SY alphabet has 25 letters which is made up of 18 consonants (represented by the graphemes: *b, d, f, g, gb, h, j, k, l, m, n, p, r, s, ṣ, t, w, y*) and seven vowels (*a, e, ẹ, i, o, ọ, u*). Note that the consonant *gb* is a diagraph, i.e. a consonant written in two letters. There are five nasalised vowels in the language (*an, en, in, ọn, un*) and two pure syllabic nasals (*m, n*). SY has three phonologically contrastive tones: High (H), Mid (M) and Low (L). Phonetically, however, there are two additional allotones or tone variants, namely; rising (R) and falling (F) [2, 14]. A rising tone occurs when an L tone is followed by an H tone, while a falling tone occurs when an H tone is followed by an L tone. This situation normally occurs during assimilation, elision or deletion of phonological object as a result of co-articulation phenomenon in fluent speech.

A valid SY syllable can be formed from any combination of a consonant and a vowel as well as a consonant and a nasalised vowel. When each of the eighteen consonants is combined with a simple vowel, we will have a total of 126 CV type syllables. When each consonant is combined with a nasalised vowel, we have a total of 90 CVn type syllables. SY also has two syllabic nasals n and m. Table 1 shows the distribution of the components of the phonological structure of SY syllables.

It should be noted that although a CVn syllable ends with a consonant, the consonant and its preceding vowels are the orthographic equivalent of a

Table 1. Phonological structure of SY syllables

Tone syllables (690)[1]				
Base syllables (230)				Tones (3)
ONSET (18)	*RYHME* (14)			
	Nucleus		Coda	
Consonant	Vocalic	Non-Vocalic		
C	V(7)	N(2)	n(1)	H, M, L

[1] The numbers within a parenthesis indicates the total number of the specified unit.

nasalised vowel. There is no closed syllable and there is no consonant cluster in the SY language.

3 Background

Generally, there are two approaches to the problem of tone recognition: (1) rule base and (2) data driven. In the rule-based approach to tone recognition, the acoustic properties of tones are studied using established theories: e.g. acoustic, phonetics, and/or linguistic theories. The knowledge elicited from such study is coded into rules. Each rule is represented in the form **IF** {*premise*} **THEN** {*consequence*}. The *premise* specifies a set of conditions that must be true for the {*consequence*} to be fired.

The strength of this approach to tone recognition is that the description of the properties of speech are defined within the context of established theories. This facilitates the extension and generalisation of the resulting tone recogniser. The resulting model is also easy to understand making it useful as a tool for the explanation of phenomena underlying tone perception particularly in tone languages, e.g. the tone *sandhi* [46]) phenomenon.

A major weakness of the rule-based approach to speech recognition is that its development requires a collaboration between a number of experts, e.g. linguists, phoneticians, etc. These experts are not readily available for most African languages. The information available in the literature is not sufficient as many phenomena that occur in speech are yet to be described definitively [3,29]. The development of practical speech recogniser requires the coding of speech information into rules. Such rules usually results in a large rule-base. It is well known that, the maintenance and organisation of such a large rule-base can be very complicated [50].

Another limitation of the rules-driven approach is that it is difficult to generate and represent all the possible ways in which rules are interdependent. Therefore, it is inevitable that rules compete with each other to explain the same phenomenon while others are in direct contradiction [39]. Although tools for managing large databases are available, such tools are not primarily designed for speech databases. It is also very difficult to extend such tools to speech database manipulation. These weaknesses are responsible for the poor performances of rule-based speech recognisers and motivated the application of the data-driven approach to speech recognition [19, 61, 70].

In data-driven (also called machine learning) approach to tone recognition, the aim is to develop a model that can learn tonal patterns from speech data. In this approach, a well designed statistical or computational model is trained on carefully selected data. The most commonly used models include; the Classification and Regression Trees (CART) [8, 34, 58, 59], the Hidden Markov Model (HMM) [7,31,56,69] and the Artificial Neural Networks (ANN) [4,26,42,53]. The aim of training data-driven models is to "store" the pattern to be recognised into their memories. The stored pattern is then used to

recognise new patterns after the training phases. A few works have applied the CART in speech recognition. In the following section we review work on the Hidden Markov Model (HMM) and the ANN in more details as they have become more popular in tone language speech recognition.

4 The Hidden Markov Model (HMM) and Artificial Neural Networks (ANN)

HMMs and their various forms (discrete, continuous, and semi-continuous) have been applied to speech recognition problems in general and tone recognition in particular [10,19,38,69]. For example, Liu et al. [43], and later Wang et al. [66], demonstrated a left-to-right continuous HMM with two transitions in the recognition of Mandarin continuous speech with very large vocabulary. They are able to achieve a recognition accuracy of 92.20%.

McKenna [45] applied the HMM approach to the recognition of 4 Mandarin tones. In that work, various HMM systems were trained using a single speaker, and isolated monosyllabic words. f_0 and energy related data were the features used for developing the HMM. McKenna [45] reported an accuracy of about 98.00%. In Peng and Wang [54] an approach to tone and duration modelling was developed within the framework of HMMs. Their experiment showed that the HMM approach reduced the relative word error by as much as 52.90 and 46.00% for Mandarin and Cantonese digit string recognition tasks, respectively. HMM are particularly good in the recognition of non-stationary continuous speech signals.

A major weakness of HMMs, which limits their application in the context of tone recognition systems, is that HMM tries to modify the data to fit the model. This weakness of the HMM limits its effectiveness at modelling tone. In fact, Bird [5] has shown that HMM are poor at the modelling of tone in the context of tone languages. A more appropriate approach is to use a model which incorporates the concept of modelling segments of speech, rather than individual frames as done by HMM. Artificial Neural Networks, such as the MLP, provide a potential solution to this problem. This is because unlike HMMs, ANNs do not need to treat features as independent, as they can incorporate multiple constraints and find optimal combinations of constraints for classification [28].

Modern speech recognition systems uses a hybrid of ANN and HMM models to achieve better performance [60]. In such cases, the ANN is used for estimating the state-dependence observation probability for the HMM. The process of generating such a model is complex and the efforts required may not justify its application to the recognition of SY tones. In this chapter, therefore, we address the problem of ANN for SY tone recognition.

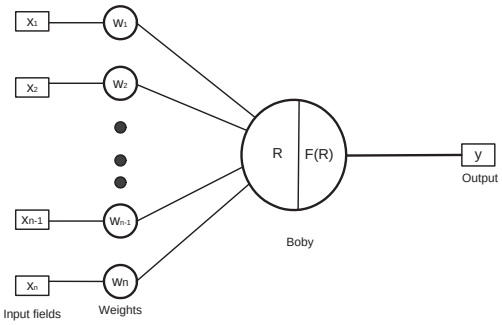

Fig. 1. A model of ANN processing unit

4.1 Introduction to Neural Networks

Artificial Neural Network (ANN), also called *connectionist* model, *neural net*, or *parallel distributed processing* (PDP) model are interconnection of simple, non-linear computational elements. These elements are called nodes or neurons (see Fig. 1). A neuron typically consists of three components: (1) a group of weights, (2) a summation function, and (3) a nonlinear activation function $f(R)$. An ANN can contain a large number of neurons. These neurons are connected by links with variable weights.

The computational behaviour of ANN models is determined by the values of each of the weight in it. To derive the behaviour of an ANN, it is usually trained on sample data. Given an input vector X with a set of n fields represented as $X = \{x_1, x_2, \ldots, x_n\}$, the operations of the processing unit consist of a number of steps [24]. First, each input field x_i is multiplied by a corresponding weight w_i, $w_i \in W = \{w_1, w_2, \ldots, w_n\}$. The product of each input field and its corresponding weight are then summed to produce the cumulative weighted combination R, as shown in (1)

$$R = w_1x_1 + w_2x_2, \ldots, w_nx_n = \sum_{1}^{n} w_ix_i = \mathbf{W.X} \tag{1}$$

The summation is further processed by an activation function **f(R)** to produce only one output signal y. We can express the computation of y mathematically as:

$$y = f(\sum_{1}^{n} w_ix_i) = f(\mathbf{W.X}) \tag{2}$$

The function $f(.)$ can take many forms, e.g. linear, sigmoid, exponential, etc. In this work we used the sigmoid function. The computed value of y can serve as input to other neurons or as an output of the network. Each node is responsible for a small portion of the ANN processing task.

The tone recognition problem being addressed in this work is an example of supervised classification. In this type of problem, decision making process

requires the ANN to identify the class to which an input pattern belongs. The ANN meant to achieve such task is first adapted to the classification task through a learning process using training data. Training is equivalent to finding the proper weight for all the connections in an ANN such that a desired output is generated for a corresponding input [32]. Several neural network models have been developed and used for speech recognition. They include: (1) the Kohonen's Self-organising Map (SOM) model, (2) the Hopefield Model, and (3) the multilayer perceptron [18]. In the next subsection, we review the literature on the application of ANN to tone language speech recognition in general and tone recognition in particular.

5 ANN in Speech and Tone Recognition

ANNs have a number of interesting applications including spatiotemporal pattern classification, control, optimization, forecasting and generalization of pattern sequences [55]. ANNs have also been applied in speech recognition and synthesis as well as in prosody modelling [9, 16, 22, 41, 57, 63, 65].

5.1 Multilayered Perceptron in Tone and Speech Recognition

The main difference between various models of ANN is how the neurons are connected to each other. Perhaps the most popular model is the multilayer perceptron trained using the backward propagation (back propagation) algorithm. A number of work has been reported on the application of MLP to Asian tone languages recognition. Cheng et al. [13] implemented a speaker-independent system for Mandarin using a Multilayer Perceptron (MLP). They used ten input features which included: energies, f_0 slopes, normalised f_0 curves as well as the duration of the voiced part of the syllables. They are able to get a recognition accuracy above 80.00%. Chang et al. [11] reported the application of MLP to the problem of recognition of four Mandarin isolated syllables tones. To achieve the tone recognition task, ten features extracted from the fundamental frequency and energy contours of monosyllables are used as the recognition features. In that work, the back-propagation algorithm was used to train the MLP. They are able to achieve a recognition rate of 93.80% in a test data set. The work also confirmed that the MLP outperforms a Gaussian classifier which has a recognition rate of 90.60%.

Similarly, Thubthong and Kijsirikul [64] described the application of a 3 layer MLP to the recognition of Thai tones. The MLP has an input layer of 6 units, a hidden layer of 10 units, and an output layer of 5. Each of the 5 units correspond to the 5 Thai tones. The MLP was trained by the back-propagation algorithm for a maximum of 1000 epochs. They obtained the recognition rates of 95.48 and 87.21% for the data and test sets respectively.

5.2 RNN in Tone and Speech Recognition

The Recurrent Neural Networks (RNN) is another type of ANN model. Its configuration is similar to that of the MLP except that the output of some hidden or output layers are feed back to the input layer. A number of researchers have used the RNN in the recognition of tones and speech. Hunt [30] described a novel syllabification technique using a series of RNNs. Several optimisation for the RNN training algorithms are also employed. The technique was developed and tested using the TIMIT database, an American, speaker-independent, continuous-speech, read-sentence database. It was reported that the system places the start and end points with an accuracy within 20 $msec$ of the desired point. This is a very high accuracy considering the fact that this results will allow a system to find syllable at 94.00% accuracy.

In Hane, et al. [27] an RNN was used for acoustic-to-phoneme mapping. This problem is similar to speech recognition in that acoustic data are used to classify phonemes. The RNN was trained using standard back-propagation method. They are able to obtain 90.00% accuracy in consonant recognition and 98.00% for vowel recognition.

RNN has also been applied to the modelling of Mandarin speech prosody recognition [68]. In that work, the RNN is trained to learn the relationship between the input prosodic feature of the training utterance and the output word-boundary information of the associated text. Accuracy rate of 71.90% was reported for word-tag detection and the character accuracy rate of speech-to-text conversion increased from 73.60 to 74.70%.

The results obtained from the above studies revealed that the MLP and RNN are powerful tools for the processing and modeling of speech data. The results also demonstrated that, when carefully constructed using the appropriate tools, the ANN can provide a robust solution to speech recognition for tone languages. However, RNNs can perform highly non-linear dynamic mappings and thus have temporally extended applications, whereas multi-layer perceptrons are confined to performing static mappings [40].

The ANN approach to tone recognition is particulary useful for our purpose for three reasons. First, the tone recognition problem involves discriminating short input speech patterns of a few, about 100, frames in length. It has been shown that the ANN approach are good at handling this type of problem [21, 26, 27]. Second, ANN techniques provide a framework that makes speech recognition systems easier to design and implement. Third, and most importantly, the availability of free software and tools [6] for modeling and implementing ANN. This makes it a feasible approach to African language speech processing applications for economic reasons.

6 Data

The data for this study was collected as part of a project to construct a language resource for Yorùbá speech technology. There are five types of syllable configurations in SY [52]. These are CV, CVn, V, N and Vn. Based on

a careful analysis of SY newspapers and language education textbooks, we collected 150 syllables. Some of the criteria used for selecting the syllables include: ease of articulation, frequency of occurrence in the text as well as the consistency of the recorded speech sound pattern. In addition, the syllables were collected with a view to have a proper representation and distribution of phonetic items in the database.

6.1 Speech Data Recording

Four native male speakers of SY read the syllables aloud. They are undergraduate Yorùbá students of the Ọbáfẹ́mi Awólọ́wọ̀ University, Ilé-Ifẹ̀. Their ages are between 21 and 26 years. Their voices were recorded using the *Wavesurfer* software on a Pentium IV computer running the *Window XP* operating system. The speech sound was captured in a quiet office environment using the head mounted *Stereo-Headphone* by *LightWave*.

The parameter for recording the speech sound is listed in Table 2.

In order to achieve good quality recording, recorded speech corresponding to a syllable was examined for the following defects:

- distortion arising from clippings,
- aliasing via spectral analysis,
- noise effects arising from low signal amplitude-resulting in evidence of quantization noise or poor signal-to-noise ratio (SNR),
- large amplitude variation,
- transient contamination (due to extraneous noise).

Recorded speech samples that have one or more of the above listed defects are discarded and the recording repeated until the resulting speech signal is satisfactory. To prepare the recorded waveform for further processing, we replayed each speech sound to ensure that they are correctly pronounced. Artefacts introduced at the end and beginning of each recording was edited out.

All the speech files are hand labelled. Figure 2 shows the syllable files annotations together with the spectrograph of the syllable Bá. The voiced

Table 2. Speech Sound Recording Parameter

Ser. No.	Parameter	Specification
1	Sampling rate	16 kHz
2	Frame size	10 ms
3	Window type	Hanning
4	Window length	32 ms (512 samples)
5	Window overlap	22 ms (352 sample)
6	Analysis	Short time spectrum
7	Set no. of channel	1
8	Waveform format	.wav (Microsoft)

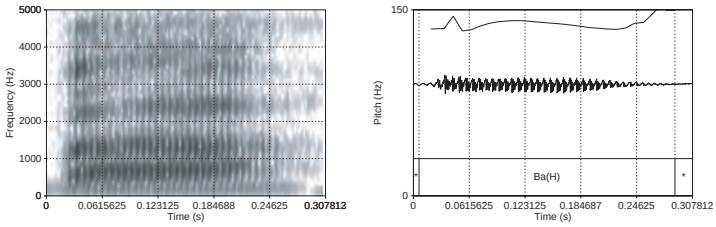

Fig. 2. SY syllable Bá (get to)

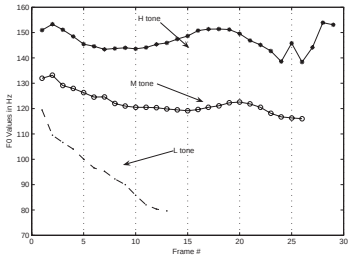

Fig. 3. f_0 curves for H, M & L tone on syllable ẹ, spoken in isolation

part of each speech file is annotated with the corresponding orthography of the syllable. The silence at the beginning and ending of each syllable speech file are annotated with the asterisk, i.e. $*$. The tone data of each syllable is the f_0 curve associated with the RHYTHM part of the speech waveform.

The *Praat* software was used to load and extract the numerical data corresponding to each syllable's f_0 curve. The data were then exported into an ASCII text file, formatted and exported as a *MATLAB mfile*. Third degree polynomials were interpolated using tools available in the *MatLab* environment, such as *polyfit*.

6.2 ANN Modelling Data Preparation

In most work on syllable and tone recognition, the following acoustic features are commonly used: (1) the fundamental frequency (f_0) of the voiced part, (2) energy, (3) zero crossing rate, (4) (linear Predictive Coefficient) LPC coefficient, (5) cepstrum, and delta cepstrum [19, 48, 54, 67]. In this work we are using only the f_0 curve because it provides enough discrimination of the SY tones due to the simplicity of the f_0 profiles. For example, Fig. 3 shows the f_0 curves for the SY High (H), Mid (M) and Low (L) tones over the syllable *ẹ*. The excursions of these f_0 curves are distinct in that, while the low tone f_0 is low and falling, that of the high tone is high while the f_0 of the mid tone occupies a region between the high and low tone f_0 curves.

To represent the f_0 profile of each syllable we interpolate a third degree polynomial into the f_0 data over the voiced portion of the syllable. Ọdẹ́jọbí,

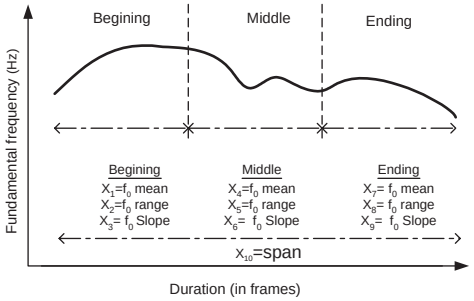

Fig. 4. f_0 feature parameters for the ANN

et al., [51] have shown that the third degree polynomials adequately approximate the f_0 data of SY syllables. To generate the data for training the ANNs, the interpolated f_0 curve for each syllable is partitioned into three equal sections: beginning, middle and ending (see Fig. 4). This partitioning is achieved by dividing the total duration of the f_0 curve into three equal parts. This approach is informed by the findings that the nonlinear pattern of the f_0 curve of each tone type has unique turning points [33, 67]. The location in time and the amplitude of the turning points provide a good discriminating parameter for each of the tones [36].

For each of the three parts, three data values were computed: f_0 range, f_0 slope and f_0 mean. The f_0 range is computed as the difference between the minimum and maximum f_0 value for that part. The f_0 slope is computed as the f_0 range divided by the time span, i.e. change of f_0 with time (i.e. number of frames). The computations here is similar to that used by Lee et al. [36] in the recognition of Cantonese tones. A similar approach was also used by Hanes et al. [27] for generating formant data for acoustic-to-phoneme mapping. Our process produced three data values for each of the three parts of the f_0 curve. These data together with the total duration of the f_0 curve produced a total of 10 data values for each syllable. It is important to note that the beginning and ending sections of the f_0 curve have the tendency of being affected by the phonetic structure of the syllable with which the tone is associated [25]. The middle section of an f_0 curve, on the other hand, is more stable.

The ten data values are mapped into a $3-bit$ binary pattern representing the three tones. Assume the pattern is represented as $b_1b_2b_3$, the most significant bit, b_1 indicates the status of the High tone. The least significant bit, b_3, represents the status of the Low tone while the middle bit, b_2, indicates the status of the Mid tone. When a data set is for a tone, the bit for that tone is set to 1 and that of the others is set to 0. For example, 100 is the output pattern for a High tone data. Sample data generated and used for the modelling are shown in Fig. 5.

	Input fields										Output fields		
	Beginning (f_0 Hz)			Middle (f_0 Hz)			End (f_0 Hz)			X_{10}			
No.	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	Frames	H	M	L
1	125.09	48.41	02.25	131.11	31.24	01.24	141.11	24.09	00.93	45.00	1	0	0
2	105.30	10.58	00.60	104.33	03.09	00.31	104.91	03.11	00.29	35.04	0	1	0
3	119.72	57.01	−0.92	112.73	34.01	−0.71	112.13	27.31	−0.60	23.06	0	0	1
4	128.14	36.17	00.38	129.17	29.12	00.24	130.41	30.09	00.23	38.75	1	0	0
5	118.70	9.18	−0.37	119.31	01.17	−0.32	118.18	02.68	−0.30	33.00	0	1	0
6	126.46	62.01	−2.23	116.58	41.17	−1.98	117.87	39.11	−2.13	23.60	0	0	1
7	122.45	42.01	00.05	128.15	44.73	00.11	126.33	34.17	00.13	42.00	1	0	0
8	113.41	8.01	−0.06	103.41	06.33	−0.12	100.17	01.31	−0.11	34.54	0	1	0
9	121.69	49.11	−2.05	121.77	40.93	−2.11	118.38	32.01	−2.06	23.60	0	0	1
10	124.49	39.71	01.81	124.47	28.11	01.37	128.13	29.37	01.11	44.75	1	0	0

Fig. 5. Sample data for the ANN modelling

Table 3. Syllable statistics for the training set

Tone	Syllable types					% of Total
	CV	V	Vn	CVn	N	
H	59 (36%)	36 (22%)	29 (18%)	28 (17%)	12 (7%)	164 (46%)
M	20 (34%)	16 (27%)	8 (13%)	11 (19%)	4 (7%)	59 (17%)
L	43 (34%)	31 (25%)	22 (17%)	23 (18%)	8 (6%)	127 (36%)
Total	122	83	59	62	24	350

Table 4. Syllable statistics for the test set

Tone	Syllable types					% of Total
	CV	V	Vn	CVn	N	
H	18 (31%)	13 (22%)	10 (17%)	12 (21%)	5 (9%)	58 (46%)
M	10 (37%)	8 (30%)	2 (22%)	6 (22%)	1 (4%)	27 (22%)
L	14 (35%)	9 (22%)	5 (13%)	9 (22%)	3 (8%)	40 (32%)
Total	42	30	17	27	9	125

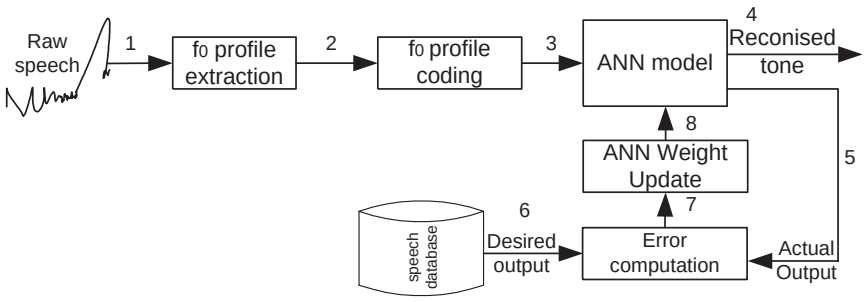

Fig. 6. Overview of the tone recognition system

A total of 475 data items were generated from the computed f_0 data. These data were divided into two disjoint sets: training (350) and test set (125). The distribution of the data is shown in Tables 3 and 4 for the training and test set respectively.

7 Overview of the Tone Recognition System

An overview of the architecture of the proposed SY tone recognition system is shown in Fig. 6. The system is composed of five main parts: f_0 profile extraction, f_0 profile coding, the ANN model, weight update and feedback system, and the speech database. The raw speech signal is applied to the model through the f_0 profile extraction component. The f_0 profile extraction component extracts the f_0 profile from the voiced portion of the speech signal. The f_0 profile coding component computes the data that are used to

model the ANN. To achieve this, a third degree polynomial is first interpolated into the f_0 data. The resulting f_0 curve is divided into three equal parts and the 10 parameters discussed in Sect. 6.2. The extracted feature is then fed into the ANN for the training and evaluation processes. The standard back-propagation algorithm is used for supervised training of the two ANN models [4, 23, 60].

The system operates in two phases: *train phase* and *operation phase*. In the train phase, the ANN is presented with the training data set. After training, it is assumed that the ANN had "learnt" the behaviour embedded in the data. During the operation phase, the ANN is presented with test data and required to produce the corresponding output. The ANN performance is determined by its ability to correctly classify or predict the test data.

The ANN is used to learn the f_0 features corresponding to SY tones. During the training phase of the ANN, the path 1, 2, 3, 5, 6, 7 and 8 is traversed repeatedly. During this training process, the network uses the tone features to compute the connection weights for each node in the network. The feedback error is computed based on the desired output (i.e. output from the database, line 6) and the actual output of the ANN (i.e. output from line 5). During the operation phrase, the path 1, 2, 3 and 4 is traversed. By activating the output corresponding to a tone, the ANNs recognises the lexical tone of the input features. For the ANN models, the input layer consists of 10 neurons. Each of these neurons is responsible for one of the input variables. Various number of hidden layer neurons were experimented with in the range 20–65 units. The output layer consisted of 3 units corresponding to each of the 3 SY tones. The mapping vectors from the input space $\mathfrak{R}^n$ to the output space $\mathfrak{R}^m$ of the ANN can be expressed mathematically as $MLP : \mathfrak{R}^n \rightarrow \mathfrak{R}^m$. In these ANN models $n = 10$ and $m = 3$.

7.1 The MLP Architecture

The bahaviour of an ANN is determined by its structure. The structure of an ANN model is defined by its architecture and the activation functions. A representation of the architecture of the MLP used in this work is shown in Fig. 7. The MLP is composed of three layers: input, hidden and output. The output from each neuron in the input layer is connected to the input of every neuron in the hidden layer. Also, the output of each neuron in the hidden layer is connected to the input of the output layer. The sigmoid activation function was used in implementing the model.

This represents a feed-forward neural network architecture. It is important to note that there is no interconnection between neurons in the same layer. To train the neural network, all the weights in the network are assigned small initial values. These values were generated randomly. In our model we used random values in the range 0.0 to 1.0. After the weight initialisation process, the following steps are taken iteratively:

1. Apply the next training data to nodes in the input layer.

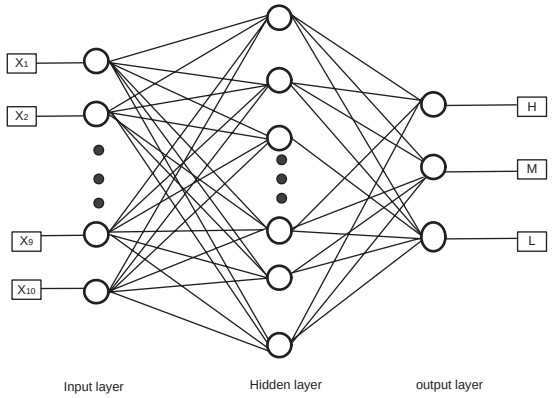

Fig. 7. Architecture of the MLP

2. Propagate the signal computed by each input node forward through the nodes in the others layers of the MLP.
3. Compute the actual output of the network.
4. Compute the error for each processing unit in the output layer by comparing the actual output to the desired output.
5. Propagate the errors backwards from the output layer to the hidden and then the input layer.
6. Modify each weight based on the computed error.

The above process is repeated until all the data set had been processed or the training converges. The MLP training has converged when the error computed is *negligible*. In such cases, the error propagated backwards will not affect the weights in the MLP significantly. The algorithm to implement the process described above is shown in Table 5.

7.2 The RNN Model Architecture

The recognition capability of a feed-forward network is affected by the number of hidden layers and increasing the number of hidden layers increases the complexity of the decision regions that can be modelled. This problem is addressed in the RNN through the delay in the Elman network, identified as Z in Fig. 8, which creates an explicit memory of one time lag [21]. By delaying the output, the network has access to both prior and following context for recognising each tone pattern. An added advantage of including the delay is that the recognition capability of the network is enhanced. This is because the number of hidden layers between any input features and its corresponding output tones is increased. The delay, however, has the disadvantage of making the training process slower requiring more iteration steps than is required in the standard MLP training.

Table 5. Back-propagation algorithm

Begin
Initialisation
Initialise all weights w_{ij} to small random values with w_{ij} being the value of the weight connecting node j to another node i in the next lower layer.
While(MoreTrainingData() and !Converged())
{
ApplyData
Apply the input from class m, i.e. $X_i^m = \{x_1^m, x_2^m, \ldots, x_n^m\}$, to the input layer. Apply the desired output corresponding to the input class n i.e. $Y_i^l = \{y_1^l, y_2^l, \ldots, y_n^l\}$ to the output layer.
Where l is the number of layers.
In our model, $m = 12$*,* $n = 3$ *and we set the desired output to* 0 *for all the output nodes except the* m^{th} *node, which is set to* 1
ComputeOutput
Compute actual output of the j^{th} node in layer n, where $n = 1, 2, \ldots, l$ using the following expression
$y_j^n = F\left(\sum_j x_j^{n-1} W_{ij}^n\right)$
where w_{ij}^n is the weight from node j in the $(n-1)^{th}$ layer to node i in the n^{th} layer.
$F(.)$ is the activation function described in 2.
The set of output at the output layer can then be represented as $X_p^l = \{x_1^l, x_2^l, \ldots, x_n^l\}$.
ComputeErrorTerm
Compute an error term, δ_j, for all the nodes. If d_j and y_j are the desired and actual output, respectively, then for an output node:
$\delta_j = (d_j - y_j) y_j (1.0 - y_j)$
and the hidden layer node,
$\delta_j = y_j (1.0 - y_j) \sum_k^N \delta_k w_{jk}$
AdjustWeights
Adjust weight by
$w_{ij}(n+1) = w_{ij}(n) + \alpha \delta_j y_i + \zeta(w_{ij}(n) - w_{ij}(n-1))$

} **EndWhile**
End

We used an Elman [21] model for our RNN based tone recognition system. An Elman RNN is a network which, in principle, is set up as a regular feed-forward network. Figure 8 shows the architecture of the Elman RNN used for our experiment. It is a three-layered network with all the outputs of the neurons in the hidden layer delayed and fed back into the input layer as additional input. Similar to a regular feed-forward neural network, the strength

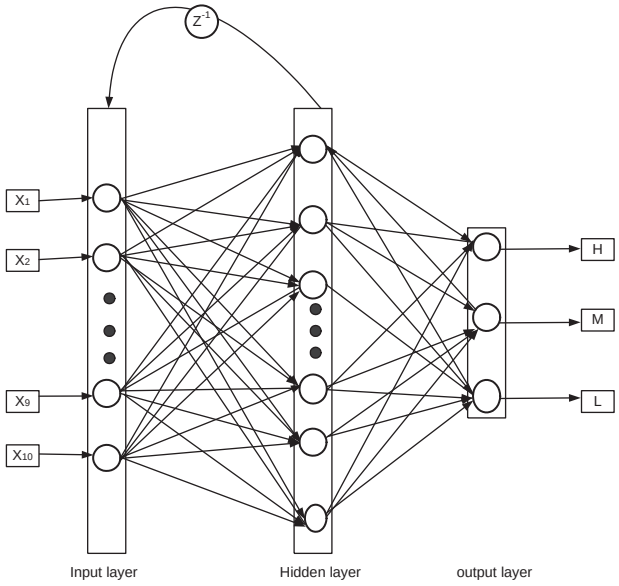

Fig. 8. Architecture of the RNN

of all connections between neurons are indicated with a weight. Initially, all weight values are chosen randomly and are optimized during the training process.

Let $O_k^{(i)}(n)$ denote the output of the k^{th} neuron in the i^{th} layer and the connection weight from the j^{th} in the i_1^{th} layer to the k^{th} neuron in the i_2^{th} layer be denoted by $W_{k,j}^{i_2,i_1}$, where n is the time index. Then,

$$O_k^{(3)} = f(\sum_j W_{k,j}^{(3,2)} O_j^{(2)}(n)) \tag{3}$$

$$= f(\sum_j W_{k,j}^{(3,2)} f(net_j^{(2)}(n))) \tag{4}$$

$$= f(net_k^{(3)}(n)) \tag{5}$$

$$net_j^{(2)}(n) = \sum_l W_{j,l}^{(2,1)} I_l(n) + \sum_{l'} W_{j,l'}^{(2,2)} O_l'^{(2)}(n-1) \tag{6}$$

Where $I_l(n)$ represents the input signals and $f()$ is the sigmoid function. An RNN of this type has been shown to possess a good ability of learning the complex relationship [68]. The model was trained on same data used for the MLP.

7.3 Model Implementation

The two ANN described above were simulated using the MATLAB 7.0 software package with neural networks toolbox [49]. This problem requires that the networks read 10 single value input signal and output three binary digits, at each time step. Therefore, the networks must have ten input element, and three output neuron. The input fields are stored as two dimensional array IP and the output are stored in the one dimensional array OP. The input vector to the input layer is same as those for the MLP. There are generally five steps in the ANN implementation process:

- Assemble the training data.
- Create the network object.
- Initialise the network.
- Train the network.
- Simulate the network response to new inputs.

For the MLP simulation, the network reads 10 single value input data and output three binary digits, at each time step. The network was created using *newf()* command. The network was trained using the *train()* command and simulated using the *sim()* command. We experimented using various number of neurons for the hidden layers. Thirty seven hidden layer neurons, i.e. $n = 37$, produced the best result.

For the RNN network, we want the input fields and target outputs to be considered a sequence, so we make the conversion from the matrix format using the command *con2seq()*. These data are used to train the Elman network. The network were created using the *newelm()* command. It was trained using the *train()* command and simulated using the *sim()* command. We experimented using various number of neurons for the hidden layer. Thirty two hidden layer neurons, i.e. $n = 32$, produced the best result.

The recognition performances of the MLP and RNN using training data set (inside data) and test set (outside test) was obtained. The reason for evaluating with the inside test is to determine how well the model represent the data. The outside test was used to determine how well the model can extrapolate to unknown data. In the RNN, for example the 350 data items (from three speakers) were used for training and the 125 data items from forth speaker, were used for testing. Training of the RNN was done in 1500 epochs since the values of mean square error (MSE) converged to small constants at that value. However, the MLP converged in 1200 epochs. Thus, the convergence rate of the RNN presented in this study was found to be lower than that of the MLP.

8 Results and Discussions

We have experimented with MLP and RNN for SY tone recognition. The experiments show that the adequate functioning of neural networks depends on the sizes of the training set and test set. For the evaluation, we used percent

recognition rate. This is computed, for a tone type, as the ratio of the total number of correctly recognised tone to that of the total number of tones of that type multiplied by 100. For example, if there are 30 H tone in the test sample and 15 H tones are recognised correctly, then the recognition rate for the H tone is $(15.00/30.00) \times 100.00 = 50.00\%$. We are using this approach because the occurrence of the three tone types is not equal. The data for the experiments were divided into two disjoint set: the *training set* and the *test set*. The training set is used to develop the model. The test set is used to obtain an objective evaluation the model.

The results for the two ANN models are shown in Table 6. Generally, the inside test produced a higher accuracy than the inside test. This implies that although the ANN models the data relatively well, they do not extrapolate to new data at the same level of accuracy. For example, while the MLP produced an accuracy of 87.50% for the inside test for H tone, it produces 71.30% for the outside test. Similarly, the RNN model produced an inside test of 89.50% for the H tone while it produces 76.10% for outside test. Another thing that the results in Table 6 indicates is that the RNN produces a better recognition accuracy than the MLP. However, a student *t-test* revealed that there is no statistically significant ($p > 0.05$) difference between the accuracy of recognition for the two models.

The tone confusion matrix of the ANN models were generated as shown in Table 7 and 8 for the MLP and RNN respectively. This result showed that tones H and M are more easily confused by the two models. For example, the

Table 6. Tone recognition results

Models		Tone recognition Rate % H	M	L	Mean
MLP	Inside test	87.50	92.71	73.50	82.30
	Outside test	71.30	85.50	81.20	75.32
RNN	Inside test	89.50	97.07	85.50	87.52
	Outside test	76.10	86.15	83.42	79.11

Table 7. Confusion matrix for MLP

Tones	H	M	L
H	78.00	17.00	5.00
M	12.00	82.00	6.00
L	6.00	17.00	77.00

Table 8. Confusion matrix for RNN

Tones	H	M	L
H	91.00	7.00	2.00
M	2.00	98.00	0.00
L	2.00	3.00	95.00

MLP recognises the H tone as such 73.00% of the time. It also recognises the M and L tone as such in 82.00 and 75.00% of the time. The RNN on the other hand recognises the H tone as such 91.00% of the time. It also recognises the M and L tone as such in 98.00% and 95.00% of the time. These recognition results are encouraging despite the simplicity of the input parameters used for the modelling.

The general conclusion from these results is that the M tone has the best recognition rate. Our results agree with those reported in the literature. For example, the f_0 curve of Mandarin Tone 1 is similar to that of the SY Mid tone (cf. [12]). The results of Mandarin tone recognition showed that Tone 1 is the least difficult to recognise with recognition accuracy as high as 98.80% while the neural tone (Tone 5) has recognition accuracy of 86.40%. A similar result was obtained by Cao et al. [10] although the HMM was used in that work.

Also Tone 1 and Tone 7 in Cantonese have similar f_0 curve as SY Mid tone although at difference f_0 ranges. Lee [35, 37] implemented a MLP for Cantonese. He found that the overall recognition accuracy for training and test data are 96.60 and 89.00% respectively. Tone 1 and 7, which have remarkably high pitch level, show the best recognition rates. We speculate that, a reason for the high recognition accuracy of the SY Mid tone is related to the relatively stable excursion of its f_0 curve when compared with those of the High and Low tones.

We did not use a linguistic model, such the Finite State Automaton (FSA), to further clarify the SY tone data. This because linguistic categories are not required for the recognition of isolated tones as will be the case in continuous speech [10, 44, 69]. Moreover the SY tones have simple f_0 signatures when compared with other tone languages such as Mandarin, Thai and Mandarin. This simple signature does not require that the f_0 curves of syllables be model linguistically before accurate recognition can be obtained. The fewer number of SY tones also reduces the complexities of the classification problem.

9 Conclusion

We have presented the Multi-layered Perceptron (MLP) and the Recurrent Neural Network (RNN) models for Standard Yorùbá (SY) isolated tone recognition. The neural networks were trained on SY tone data extracted from recorded speech files. Our results led to three major conclusions:

1. SY tone recognition problem can be implemented with MLP and RNN;
2. The RNN training converges slower than the MLP using the same training data;
3. The accuracy rates achieved using the RNN was found to be higher (although not significantly) than that of the MLP on the inside and outside test data sets;
4. the SY Mid tone has highest recognition accuracy.

However, the efforts required for building the RNN is relatively more than those required for building the MLP. An extension of this work could be to apply the ANN models to continuous SY speech recognition. However, it is well known that ANNs have difficulty in modelling the time-sequential nature of speech. They have therefore not been very successful at recognising continuous utterances. In addition, the training algorithm is not guaranteed to find the global minimum of the error function since gradient descent may get stuck in local minima. Therefore, the ANN approach does not generalise to connected speech or to any task which requires finding the best explanation of an input pattern in terms of a sequence of output classes.

To address this problem, there has been an interest in approaches which are a hybrid of HMMs and neural networks to produce hybrid systems. The aim of this type of approach is to combine the connectionist capability provided by neural networks with the ability of HMM to model the time-sequential nature of speech sound [7]. One approach has been to use MLPs to compute HMM emission probabilities [47] with better discriminant properties and without any hypotheses about the statistical distribution of the data. An alternative approach [4] is to use a neural network as a post-processing stage to an *N-best* HMM system [31, 39, 47, 47, 60].

References

1. L. O. Adéwọlé. *The categorical status and the function of the Yorùbá auxiliary verb with some structural analysis in GPSG*. PhD thesis, University of Edinburgh, Edinburgh, 1988.
2. A. Akinlabí. Underspecification and phonology of Yorùbá /r/. *Linguistic Inquiry*, 24(1):139–160, 1993.
3. A. M. A. Ali, J. Spiegel, and P. Mueller. Acoustic-phonetic features for the automatic classifcation of stop consonants. *IEEE Transactions on Speech and Audio Processing*, 9(8):833–841, 2001.
4. S. Austin, G. Zavaliagkos, J. Makhoul, and R. Schwartz. Continuous speech recognition using segmental neural nets. In *IEEE ICASSP*, 625–628, San Francisco, 2006.
5. S. Bird. Automated tone transcription. http://www.idc.upenn.edu/sb/home/papers/9410022/941002.pdf, May 1994. Visited: Apr 2004.
6. P. Boersma and D. Weenink. *Praat*, doing phonetics by computer. http://www.fon.hum.uva.nl/praat/, Mar 2004. Visited: Mar 2004.
7. H. Bourlard, N. Morgan, and S. Renals. Neural nets and hidden Markov models: Review and generalizations. *Speech Communication*, 11:237–246, 1992.
8. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Tree*. Wadworth, CA, 1984.
9. T.-L. Burrows. *Trainable Speech Synthesis*. PhD thesis, Cambridge, Mar 1996.
10. Y. Cao, S. Zhang, T. Huang, and B. Xu. Tone modeling for continuous Mandarin speech recognition. *International Journal of Speech Technology*, 7:115–128, 2004.
11. P. C. Chang, S. W. Sue, and S. H. Chen. Mandarin tone recognition by multi-layer perceptron. In *Proceedings on IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 517–520, 1990.

12. S.-H. Chen, S.-H. Hwang, and Y.-R. Wang. An RNN-based prosodic information synthesiser for Mandarin text-to-speech. *IEEE Transactions on Speech and Audio Processing*, 6(3):226–239, 1998.
13. P.-C. Cheng, S.-W. Sun, and S. Chen. Mandarin tone recognition by multi-layer perceptron. In *IEEE 1990 Internertional Conference on Acoustics, Speech and Signal Processing, ICASSP-90*, 517–520, 1990.
14. B. Connell and D. R. Ladd. Aspect of pitch realisation in Yorùbá. *Phonology*, 7:1–29, 1990.
15. B. A. Connell, J. T. Hogan, and A. J. Rozsypal. Experimental evidence of interaction between tone and intonation in Mandarin Chinese. *Journal of Phonetics*, 11:337–351, 1983.
16. R. Córdoba, J. M. Montero, J. M. Gutiérrez, J. A. Vallejo, E. Enriquez, and J. M. Pardo. Selection of most significant parameters for duration modelling in a Spanish text-to-speech system using neural networks. *Computer Speech and Language*, 16:183–203, 2002.
17. D. H. Crozier and R. M. Blench. *An Index of Nigerian Languages*. Summer Institute of Linguistics, Dallas, 2nd edition, 1976.
18. E. Davalo and P. Naïm. *Neural Networks*. MacMillan, Hong Kong, 1991.
19. T. Demeechai and K. Mäkeläinen. Recognition of syllable in a tone language. *Speech Communication*, 33:241–254, 2001.
20. S. J. Eady. Difference in the f_0 patterns of speech: Tone languages versus stress languages. *Language and Speech*, 25(Part 1):29–41, 1982.
21. J. L. Elman. Finding structure in time. *Cognitive Science*, 12(2):179–211, 1990.
22. J. W. A. Fackrell, H. Vereecken, J. P. Martens, and B. V. Coile. Multilingual prosody modelling using cascades of regression trees and neural networks. *http://chardonnay.elis.rug.ac.be/papers/1999 0001.pdf*, 1999. Visited: Sep 2004.
23. A. K. Fernando, X. Zhang, and P. F. Kinley. Combined sewer overflow forecasting with feed-forward back-propagation artificial neural network. *Transactions On Engineering, Computing And Technology*, 12:58–64, 2006.
24. S. C. Fox and E. K. Ong. A high school project on artificial intelligence in robotics. *Artificial Intelligence in Engineering*, 10:61–70, 1996.
25. J. Gandour, S. Potisuk, and S. Dechnonhkit. Tonal co-articulation in Thai. *Phonetica*, 56:123–134, 1999.
26. N. F. Gülera, E. D. Übeylib, and I. Gülera. Recurrent neural networks employing Lyapunov exponents for EEG signals classification. *Expert Systems with Applications*, 29:506–514, 2005.
27. M. D. Hanes, S. C. Ahalt, and A. K. Krishnamurthy. Acosutic-to-phonetic mapping using recurrent neural networks. *IEEE Transactions on Neural Networks*, 4(5):659–662, 1994.
28. W. Holmes and M. Huckvale. Why have HMMs been so successful for automatic speech recognition and how might they be improved? Technical report, Phonetics, University Colledge London, 1994.
29. M. Huckvale. 10 things engineers have discovered about speech recognition. In *NATO ASI workshop on speech pattern processing*, 1997.
30. A. Hunt. Recurrent neural networks for syllabification. *Speech Communication*, 13:323–332, 1993.
31. B. H. Juang and L. R. Rabiner. Hidden Markov models for speech recognition. *Technometrics*, 33:251–272, 1991.

32. A. Khotanzak and J. H. Lu. Classification of invariant images representations using a neural netwrok. *IEEE Transactions on Speech and Audio Processing*, 38(6):1028–1038, 1990.
33. D. R. Ladd. Tones and turning points: Bruce, pierrehumbert, and the elements of intonation phonology. In M. Horne (ed.) *Prosody: Theory and Experiment – Studies presented to Gösta Bruce*, 37–50, Kluwer Academic Publishers, Dordrecht, 2000.
34. S. Lee and Y.-H. Oh. Tree-based modelling of prosodic phrasing and segmental duration for Korean TTS systems. *Speech Communication*, 28(4):283–300, 1999.
35. T. Lee. *Automatic Recognition Of Isolated Cantonese Syllables Using Neural Networks*. PhD thesis, The Chinese University of Hong Kong, Hong Kong, 1996.
36. T. Lee, P. C. Ching, L. W. Chan, Y. H. Cheng, and B. Mak. Tone recognition of isolated Cantonese syllables. *IEEE Transactions on Speech and Audio Processing*, 3(3):204–209, 1995.
37. W.-S. Lee. The effect of intonation on the citation tones in Cantonese. In *International Symposium on Tonal Aspect of Language*, 28–31, Beijing, Mar 2004.
38. Y. Lee and L.-S. Lee. Continuous hidden Markov models integrating transition and instantaneous features for Mandarin syllable recognition. *Computer Speech and Language*, 7:247–263, 1993.
39. S. E. Levinson. A unified theory of composite pattern analysis for automatic speech recognition. In F. F. and W. A. Woods (eds) *Computer Speech Processing*, Prentice-Hall International, London, 1985.
40. Y.-F. Liao and S.-H. Chen. A modular RNN-based method for continuous Mandarin speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):252–263, 2001.
41. C.-H. Lin, R.-C. Wu, J.-Y. Chang, and S.-F. Liang. A novel prosodic-information synthesizer based on recurrent fuzzy neural networks for Chinese TTS system. *IEEE Transactions on Systems, Man and Cybernetics*, B:1–16, 2003.
42. R. P. Lippman. Review of neural networks for speech recognition. *Neural Computing*, 1:1–38, 1989.
43. F.-H. Liu, Y. Lee, and L.-S. Lee. A direct-concatenation approach to training hidden Markov models to recognize the highly confusing Mandarin syllables with very limited training data. *IEEE Transactions on Speech and Audio Processing*, 1(1):113–119, 1993.
44. L. Liu, H. Yang, H. Wang, and Y. Chang. Tone recognition of polysyllabic words in Mandarin speech. *Computer Speech and Language*, 3:253–264, 1989.
45. J. McKenna. Tone and initial/final recognition for Mandarin Chinese. Master's thesis, University of Edingbrugh, U.K., 1996.
46. N. Minematsu, R. Kita, and K. Hirose. Automatic estimation of accentual attribute values of words for accent sandhi rules of Japanese text-to-speech conversion. *IEICE Transactions on Information and System*, E86-D(3):550–557, Mar 2003.
47. N. Morgan and H. Bourlard. Continuous speech recognition using multilayer perceptrons with Hidden Markov Models. In *Proceedings of IEEE ICASSP*, 413–416, Albuquerque, 1990.
48. R. D. Mori, P. Laface, and Y. Mong. Parallel algorithms for syllable recognition in continuous speech. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(1):56–69, 1985.

49. Y. Morlec, G. Bailly, and V. Aubergé. Generating prosodic attitudes in French: data, model and evaluation. *Speech Communication*, 33:357–371, 2001.
50. S. M. O'Brien. Knowledge-based systems in speech recognition: A survey. *International Journal of Man-Machine Studies*, 38:71–95, 1993.
51. O. A. Ọdẹ́jọbí, A. J. Beaumont, and S. H. S. Wong. A computational model of intonation for Yorùbá text-to-speech synthesis: Design and analysis. In P. Sojka, I. Kopeček, and K. Pala, (eds) *Lecture Notes in Artificial Intelligence*, Lecture Notes in Computer Science (LNAI 3206), 409–416. Springer, Berlin Heidelberg New York, Sep 2004.
52. O. A. Ọdẹ́jọbí, A. J. Beaumont, and S. H. S. Wong. Experiments on stylisation of standard Yorùbá language tones. Technical Report KEG/2004/003, Aston University, Birmingham, Jul 2004.
53. S. M. Peeling and R. K. Moore. Isolated digit recognition experiments using the multi-layer perceptron. *Speech Communication*, 7:403–409, 1988.
54. G. Peng and W. S.-Y. Wang. Tone recognition of continuous Cantonese speech based on support vector machines. *Speech Communication*, 45:49–62, Sep 2005.
55. A. A. Petrosian, D. V. Prokhorov, W. Lajara-Nanson, and R. B. Schiffer. Recurrent neural network-based approach for early recognition of Alzheimer's disease in EEG. *Clinical Neurophysiology*, 112(8):1378–1387, 2001.
56. L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of IEEE*, 77:257–286, 1989.
57. M. Riedi. A neural-network-based model of segmental duration for speech synthesis. In *European Conference on Speech Communication and Technology*, 599–602, 1995.
58. M. D. Riley. Tree-based modelling of segmental durations. In G. Bailly, C. Benoit, and T. R. Sawallis (eds), *Talking Machines: Theories, Models and Designs*, p 265–273. Elsevier, Amsterdam, 1992.
59. S. R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man and Cybernetics*, 21:660–674, 1991.
60. P. Salmena. Applying dynamic context into MLP/HMM speech recognition system. *IEEE Transactions on Neural Networks*, 15:233–255, 2001.
61. J. Sirigos, N. Fakotakis, and G. Kokkinakis. A hybrid syllable recognition system based on vowel spotting. *Speech Communication*, 38:427–440, 2002.
62. C. Taylor. Typesetting African languages. http://www.ideography.co.uk/library/afrolingua.html, 2000. Visited: Apr 2004.
63. P. Taylor. Using neural networks to locate pitch accents. In *Proceedings of EuroSpeech '95*, 1345–1348, Madrid, Sep 1995.
64. N. Thubthong and B. Kijsirikul. Tone recognition of continuous Thai speech under tonal assimilation and declination effects using half-tone model. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(6):815–825, 2001.
65. M. Vainio. *Artificial neural network based prosody models for Finnish text-to-speech synthesis*. PhD thesis, Department of Phonetics, University of Helsinki, Helsinki, 2001.
66. H.-M. Wang, T.-H. Ho, R.-C. Yang, J.-L. Shen, B.-O. Bai, J.-C. Hong, W.-P. Chen, T.-L. Yu, and L.-S. Lee. Complete recognition of continuous Mandarin speech for Chinese language with very large vocabulary using limited training data. *IEEE Transactions on Speech and Audio Processing*, 5(2):195–200, 1997.

67. T.-R. Wang and S.-H. Chen. Tone recognition of continuous Mandarin speech assisted with prosodic information. *Journal of the Acoustical Society of America*, 96(5):2637–2645, 1994.
68. W.-J. Wang, Y.-F. Liao, and S.-H. Chen. RNN-based prosodic modelling for Mandarin speech and its application to speech-to-text conversion. *Speech Communication*, 36:247–265, 2002.
69. Y. R. Wang, J.-M. Shieh, and S.-H. Chen. Tone recognition of continuous Mandarin speech based on hidden Markov model. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(1):233–246, 1994.
70. M. Wester. Pronunciation modeling for ASR knowledge-based and data-driven methods. *Computer Speech and Language*, 38:69–85, 2003.
71. Y. Xu. Understanding tone from the perspective of production and perception. *Language and Lingustics*, 5:757–797, 2005.