Karlsruhe Institute of Technology

Towards Tonality Detection in Unseen Languages

Master's Thesis
of

Florian Deßloch

at the Department of Informatics
Institute for Anthropomatics and Robotics

Reviewers:	Prof. Dr. Alexander Waibel
	Prof. Dr. Rainer Stiefelhagen
Advisors:	M. Sc. Markus Müller
	Dr. Sebastian Stüker

Time Period: December 1st, 2016 – May 31st, 2017

I hereby declare that I have developed and written the following thesis entirely on my own, and that I have not used sources or means without declaration.

Karlsruhe, May 31st, 2017

Abstract

With half of all spoken languages worldwide in danger of becoming extinct at the end of the century, language preservation is an important challenge that could benefit from machine assistance due to its complexity and current dependency on manual linguistic work. For tonal languages, defining and classifying different tones is a central part of this process, and considering the high number of tonal languages worldwide, a tone recognition system could play an important role in supporting linguists, as well as assisting the development of speech recognition technology. However, previous work on tone recognition was largely developed for the language-specific context of the speech recognition task.

This thesis investigates different approaches to the tone recognition task, with the aim of working towards the long-term goal of developing a universally applicable tone classifier.

Using diverse Cantonese speech data, a variety of classifiers are applied to the frame-wise tone recognition task, indicating that recurrent neural networks, specifically long short-term memory models, are best suited to the challenge. Experiments with numerous network parameters and features lead to a frame error rate of 63.47% for the best-performing system, which is well above chance performance considering classification is conducted across 7 classes. Additionally, the syllable-wise recognition of tone is performed for several neural networks and k-nearest neighbors classifiers, leading to a syllable-error rate of 64.82% for the best-ranking k-nearest neighbor model.

Zusammenfassung

Die Hälfte aller gesprochenen Sprachen weltweit sind davon bedroht, bis zum Ende des Jahrhunderts ausgestorben zu sein. Aus diesem Grund ist die Erhaltung von Sprachen eine wichtige Herausforderung, die aufgrund ihrer Komplexität und derzeitiger Abhängigkeit von linguistischer Handarbeit von maschineller Unterstützung profitieren kann. Für tonale Sprachen ist die Definition und Klassifikation ihrer Töne ein zentraler Bestandteil dieses Prozesses, und in Anbetracht der hohen Anzahl tonaler Sprachen weltweit könnten ein Tonerkennungssystem eine wichtige Rolle in der Unterstützung von Linguisten spielen, sowie eine Bestandteil für die Entwicklung von Spracherkennungstechnologie sein.

Diese Arbeit untersucht verschiedene Ansätze für die Aufgabe der Tonerkennung, mit der Absicht auf das Langzeitziel eines universell einsetzbaren Tonklassifikators hinzuarbeiten.

Auf einem vielseitigen kantonesischen Datensatz werden unterschiedliche Klassifikatoren für die frame-weise Klassifikation von Ton eingesetzt, mit dem Ergebnis, dass neuronale Netze, speziell long short-term memory Modelle, sich am besten für die Herausforderung eignen. Experimente mit einigen Netzparametern und Merkmalstypen führen zu einer Fehlerrate von 63.47% für das beste System - ein Resultat, welches deutlich über dem Zufall liegt, da es sich um eine Klassifikation über 7 Klassen handelt. Zusätzlich wird eine silbenweise Klassifikation von Ton für verschiedene neuronale Netze und k-nearest neighbors Klassifikatoren erprobt, wodurch bestenfalls eine Fehlerrate von 64.82% erreicht wird.

Inhaltsverzeichnis

1. Introduction

1.1 Motivation

Currently, an estimated number of 6900 languages are spoken worldwide. However, with the strong advance of globalization, a large number of languages have decreasing numbers of native speakers, and many are in danger of becoming extinct. There is a consensus among linguists that half of all languages may no longer be spoken by the end of the 21st century.

Knowledge of a language is an important part of understanding a culture's identity, customs and history. To prevent a loss of this information, there is a global effort to preserve these endangered languages. For this to happen, a language must be studied and described by trained linguists, which is a time-consuming and often difficult procedure, involving the collection of data, transcription with the assistance of native speakers, a phonological analysis and the definition of phoneme and grammar systems. Considering the rate of language extinction and the complexity of the problem, language description is certainly a task that could benefit from machine assistance.

One challenging aspect of the language description process is the analysis of *tone*, meaning the use of pitch to influence lexical or grammatical meaning. Estimates state that the majority of languages spoken worldwide use tone to some degree [Yip02], making tone classification an important task. The manual annotation of tone is time-consuming and non-trivial for linguists, as there often exist fine differences that are difficult to distinguish, and strong regional and speaker-specific differences make the clear definition of a tone system even more problematic.

Previous work on tone recognition mostly focuses on integrating tone features into language-specific speech recognition systems. Although this approach has been successful and useful in this limited domain, it is of little use in the field of language documentation, whereas the impact of a language-independent system for tone clas-

sification in the field would be far greater.

In addition to assisting linguistic work, a general approach to tone classification would be a vital piece in the development of speech recognition and translation technologies, which, at the moment, perform very well for only a small subset of languages worldwide. Due to the technological advancement and increasing internationalization, the need to develop such systems for further languages is increasing.

1.2 Goal

With the ultimate perspective of a language-independent tone classification system in mind, the goal of this work is to explore different approaches to the tone recognition task, focusing on features and techniques that are not specific to a certain language. Although this is a complex task, we hope to provide insight regarding the realizability of such a system, and examine which approaches are suitable to the problem.

1.3 Structure

The thesis begins with a chapter on relevant background information, covering different types of neural networks as well as linguistic tone. Next, various related publications are discussed. The following section describes the experimental framework used in this work, including information on the data, the technologies that were used and the types of experiments that were performed. Details regarding the experiments as well as their results are enumerated and evaluated in the next section. Finally, the thesis ends with a conclusion of the results.

2. Related Work

Although extensive work has been published on creating sophisticated speech recognition systems for numerous languages, the majority of spoken languages are still under-resourced and lacking dedicated speech technology. With the exception of Mandarin Chinese, most tonal languages fall into this less researched category. For this reason, most of the existing work on detecting tone in speech is focused on integrating tone features into Mandarin speech recognition systems, along with a small amount of experiments dedicated specifically to other single languages.

[LTGL+93] describes a speaker-dependent real-time dictation system for Mandarin that integrates tone features into classification and represents the first system that includes recognition of the neutral tone in addition to the four lexical tones. Based on hidden Markov models (HMMs), the system first classifies isolated syllables by separating the recognition of phones from the recognition of tone, with a total of 1300 possible syllables. A second subsystem then uses a language model to determine the word sequence.
The tone recognition subsystem was trained and tested on a set of 190 syllables that were specifically chosen to reflect each tone's representation in Mandarin. These 190 utterances were recorded twice by 4 male and 4 female speakers, with one set being used in training and the other for testing. It was found that while pitch contour as a feature worked well in distinguishing the four lexical tones, performance was worse when the neutral tone was included, as this tone is not generally represented by a certain pitch pattern. However, examination of syllable wave forms showed that the neutral tone typically displayed a lower signal amplitude and a shorter duration of voicing than other tones, leading to the following frame-based feature vector:

$$y_t = (log(f_t) + log(f_{t+1}), log(f_t) - log(f_{t+1}), log(e_t) + log(e), a), \tag{2.1}$$

in which f_t and e_t are the pitch and short time energy at frame t, respectively, while e stands for the syllable-wise maximum of e_t and a represents the duration of the voiced part of the syllable. Using these features in a 5-state discrete HMM with a

codebook size of 32, it was possible to achieve an accuracy of 95.5% in tone classification.

[HuQS14] also uses a speech recognition system for Mandarin, and examines the integration of tone features into the deep neural network (DNN) based acoustic model. To represent the tone, the fundamental frequency and its first- and second-order derivates are computed and combined with 39 MFCC features to make up the acoustic model's feature vectors. The speaker-independent system was trained with 66 hours of speech data, consisting of read text such as novels and classical Chinese literature, read by 230 female and 230 male speakers.
Compared to a baseline system for automatic speech recognition (ASR) with the same parameters but using only MFCC features, the integration of tone features achieves a 20% and 23% reduction in the error rate of tonal syllable recognition, for female and male speakers, respectively. The tone error rate is lowered by 32% for female and 35% for male speakers. When investigating the influence of the static F_0 as opposed to its deltas, it was found that the dynamic features are far more important for recognizing tone, as removing the static F_0 value from the features resulted in only a negligible reduction in performance. It was further observed that interpolating F_0 during unvoiced speech segments, as opposed to using raw F_0 values (which are zero in unvoiced segments), did not improve recognition performance.

[ScVu16] explores the impact of different types of features on a syllable-based tone recognition system for Vietnamese. A baseline system was defined using absolute pitch from three regions in the syllable, as well as two delta pitch values and the syllable duration as features. System performance was measured as more features were added step by step, beginning with energy and delta energy features, once for the whole syllable and once divided into three equal-length parts, and later including the degree of voicing, spectral tilt, harmonicity, and PaIntE parameters. PaIntE is a model designed for intonation modeling in speech synthesis, and was restricted to be based on a single syllable. Using six parameters that are linguistically motivated, the model defines a function that approximates the F_0 contour.Speech data was taken from the GlobalPhone database, for both female and male speakers.
The classification results, which were obtained using Random Forest and Bagging classifiers, showed that each newly added feature improved the results, most notably the harmonicity and PaIntE parameters. Using the full set of features, it was possible to increase system performance from 57.7% to 71.2% using Random forests and from 65.5% to 72.4% for the Bagging classifier. Experiments with subsets seperated by gender also showed that the algorithm is capable of generalizing between male and female speakers to a degree, while using only female speakers led to higher performance that when using male speakers. Furthermore, performing experiments on dialect-specific speech showed much lower accuracy for cross-dialect models than for those with a single dialect.

[MSWG+13] evaluates different approaches in tone feature modelling when used in ASR systems for two tonal and two non-tonal languages. Apart from F_0, six delta and two double-delta features, as well as the cross-correlation, were used to model pitch contour. Additionally, the fundamental frequency variation features described

in [LaHE08] were applied, resulting in 7 additional coefficients. Different approaches of integrating these features with the ASR system's MFCC features were compared, namely an early integration, resulting in merged features for bottleneck feature training, and a late integration, in which the tone features were added after bottleneck feature extraction. All combinations led an improvement in ASR performance, with the integration of tonal features leading to an accuracy gain of multiple percent points for tonal and non-tonal languages, relative to the previously best-performing bottleneck feature ASR system.

3. Background

3.1 Neural Networks

Artificial neural networks are computational models that have been successfully applied to classification in the area of speech recognition, and are presently used across a wide range of fields, including natural language processing, image recognition and object detection. This section provides an overview of relevant types of neural networks and techniques for their usage.

Inspired by the human brain and its ability of excelling at various cognitive tasks, the core principles of artificial neural networks are derived from the connectionist structure and computational process of their biological counterparts. The idea was to create a model that is capable of learning a representation and adapting it if necessary, using parallelism on a large scale for computational efficiency. Like the human brain, this model should be able to generalize on unseen data and exhibit robustness towards noisy inputs.

3.1.1 Perceptron

While research on artificial neural networks dates back to the description of the first mathematical model of a neuron by McCulloch and Pitts in 1943 [McPi43], the first operational model of a neural network, the *perceptron*, was conceived by Frank Rosenblatt in 1958 [Rose57].

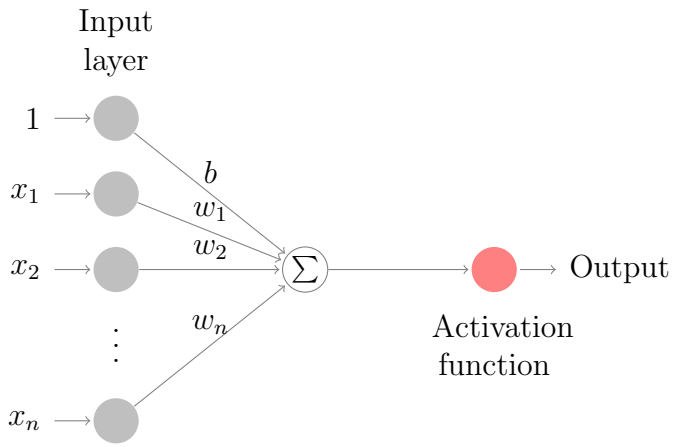

Figure 3.1: A perceptron

The perceptron is a linear classifier in which knowledge is represented by the weight vector $\mathbf{w}$ of the connections between the network's input layer and the single neuron responsible for the classification decision. The network classifies an (n+1)-dimensional input vector $\mathbf{x}$ by passing the weighted sum of its components $(1, x_1, x_2, ..., x_n)$ to an activation function f in the neuron, which computes a binary output $\mathbf{y}$, as described in the following equation:

$$y = f(\sum_{i=0}^{n} w_i x_i) = f(\mathbf{w}^T \mathbf{x}) \tag{3.1}$$

The perceptron training is supervised, on a training set which provides a target output value t for each input vector $\mathbf{x}$. After specifying a learning rate η and threshold γ and initializing the weight vector, training is an iterative process with two steps:

- Compute the current output y of the network
- Update the weight vector $\mathbf{w}$: $w \leftarrow w + \Delta w$, with $\Delta w = -\eta \nabla E(w)$.

$E(w)$ denotes the error criterion, which is defined as

$$E(w) = \frac{1}{2} \sum_{x \in X} (t_x - y_x)^2. \tag{3.2}$$

3.1.2 Feedforward Neural Networks

A perceptron is a very basic network with limited learning ability - one example of this is its inability to learn the XOR function. However, the same concept can be used to create larger, more complex networks with multiple layers of neurons which are capable of learning higher-order functions.

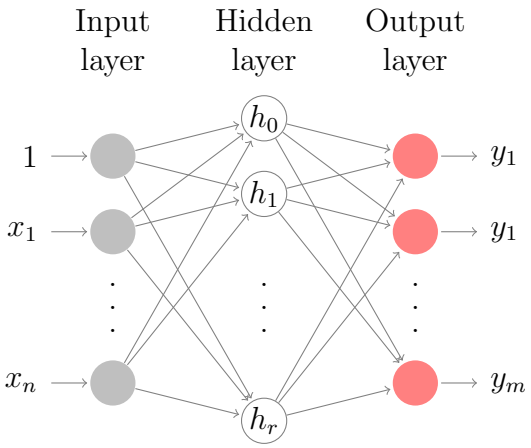

Figure 3.2: A feedforward neural network with a single hidden layer

A feedforward neural network, as depicted in Figure 3.2, is a neural network made up of an input layer, an ouput layer as well as at least one hidden layer of neurons, in which the connections between neurons do not contain cycles - meaning that information only flows in one direction. For classification, the number of neurons in the input layer is at least as high as the feature dimension, while the output layer may typically contain as many neurons as classes in the problem.

As in the perceptron, a neuron's output is computed by its activation function. In classifying networks, typical activation functions are the sigmoid function $\sigma(x)$ and hyperbolic tangent function $\phi(x)$:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.3}$$

$$\phi(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{3.4}$$

Both functions are well-suited to classification due to their nonlinearity and limited value range.

Furthermore, when dealing with classification problems with $K > 2$ classes, the softmax function (3.5) is applied to the output layer neurons, normalizing the output values to a range of $[0, 1]$ with a total sum of 1, which allows the output to be interpreted as class probabilites.

$$\varphi(o)_j = \frac{e^{o_j}}{\sum_{k=1}^{K} e^{o_k}} \tag{3.5}$$

Training the network is accomplished by iteratively updating its weights using the backpropagation algorithm [RuHW88], which allows for the calculation of an error value for each neuron in the network, and uses the gradient descent method to minimize a loss function.

More recently, there has been an improvement in quality of systems in the fields of speech recognition and natural language processing by using networks with multiple hidden layers and a high number of neurons, known as *deep neural networks* [DeHK13]. The increased model complexity along with improved learning procedures such as pre-training techniques for weight initialization and the availability of larger datasets led to a higher system performance, while more powerful hardware and the incorporation of parallel computing made the utilization of large models possible.

3.1.3 Recurrent Neural Networks

In contrast to traditional feedforward neural networks, recurrent neural networks (RNNs) [LiBE15] are artificial neural networks in which information from one layer can flow to previous layers as well as following layers. This is achieved by introducing connections from a neuron to neurons of the previous layer. With these connections, the network's state is not only determined by its weights and input values, but also by its previous states, thus forming a memory of past behaviour.

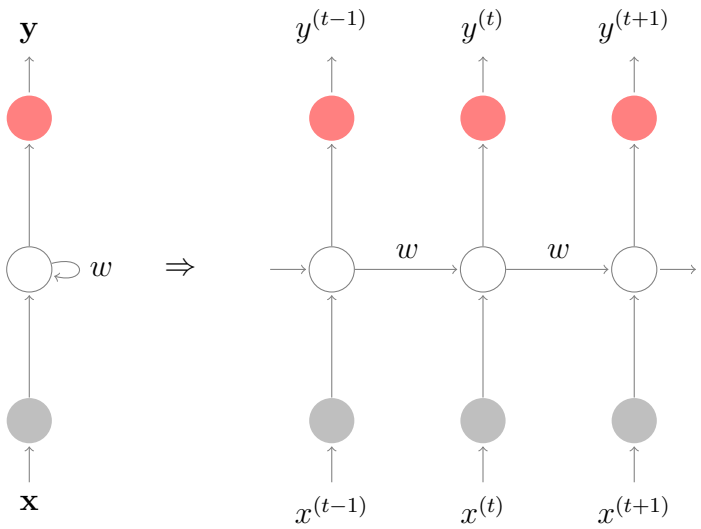

Figure 3.3: A diagram of an RNN, unfolded to show time steps

This sensitivity to context makes RNNs ideal for tasks such as speech recognition and language processing, in which access to previous states provides a significant information benefit. For example, the task of recognizing the semantic meaning of a word in a sentence often requires knowledge of the previous words. With RNNs, it is possible to create a model that processes phrases on a word level but includes information from preceding words in the classification decision.

One well-documented problem of RNNs arises in the case of long-term dependencies, meaning that the output at a certain point in time is dependent on information, and thus a network state, from a far earlier point in time. The derivates of the sigmoid and hyperbolic tangent functions which are used in the neurons are close to zero at both ends, and the method of calculating the loss function in the backpropagation algorithm involves a higher number of chain rule applications on neuron outputs for

neurons that are further away in the network. For this reason, the gradient for these neurons from more distant points in time quickly becomes zero, meaning that they have no influence on the training. This is known as the vanishing gradient problem. Similarly, the exploding gradient problem describes the opposite case, in which activation functions with high-valued derivatives lead to an exponential growth of the gradient.

3.1.3.1 Bidirectional Recurrent Neural Networks

Although standard RNNs incorporate information from past time states, information from future points in time, which can be just as relevant to the task, is not captured. One way to achieve this is to delay the output by a fixed number of frames in order to include following input data. However, this method does not perform well for higher delay values.

Bidirectional Recurrent Neural Networks (BRNNs) [ScPa97] provide a solution to this problem by extending a standard RNN's hidden layers with a second, equally large set of neurons in which the connections convey information from future instead of past time states. Since these two sets are disjoint, the network can be unfolded into a feedforward neural network just like a standard RNN, and can be trained in the same manner.

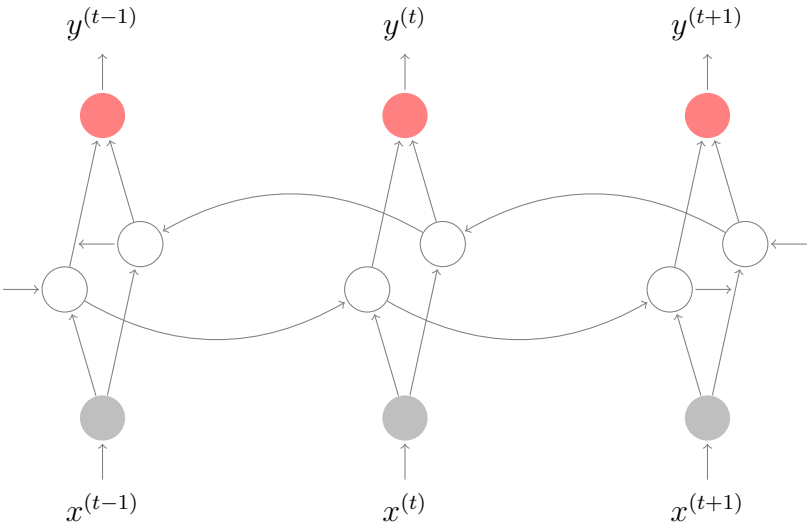

Figure 3.4: A diagram of a bidirectional RNN, unfolded to show time steps

BRNNs were found to often outperform their single-direction predecessors, and the principle of bidirectionality is regularly used in more complex, state-of-the-art variants of the RNN.

3.1.3.2 Long Short-Term Memory

As mentioned before, traditional RNNs fail to successfully represent long-term dependencies due to the vanishing/exploding gradient problem. To provide a solution

to this problem, long short-term memory (LSTM) was introduced in 1997 by Hochreiter and Schmidhuber [HoSc97]. Due to their superiority in performance over standard RNNs in context-sensitive tasks, LSTM networks or a variation of the concept is frequently used in practice.

While a standard RNN has long-term and short-term memory (represented by its weights and the neuron activations with influence from previous layers, respectively), the concept of LSTM is to add an additional type of memory by replacing each node in the network's hidden layer with a more complex model, referred to as the memory cell. This additional memory is stored in the cell state, which is passed forward in time through the network, from one memory cell to the next. Gates inside the cell determine which part of the cell state should be output, and which information should be added or removed from the cell state.

The memory cell can be described fully with the following elements:

- An input node g_c, which receives activation from the input layer $\mathbf{x}^{(t)}$ and the previous time step's hidden layer $\mathbf{h}(^{(t-1)}$.
- An input gate i_c, which receives the same activation as the input node, and whose value is multiplied with the value of the input node to determine which information to pass on.
- The internal state s_c, which has a self-connected recurrent edge with a constant weight, thus allowing for an unhindered error flow across time steps and preventing the gradient from vanishing or exploding during gradient descent.
- A forget gate f_c, which allows for information to be removed from the internal state. The forget gate was proposed as a variation of LSTM in [GeSC99] and was found to significantly improve performance.
- An output gate o_c, whose value is multiplied with the value of the internal state to produce the output of the cell.

There are numerous variants of the standard LSTM model that are used in practice. One model that has shown success in practice is bidirectional LSTM, which combines the LSTM's memory cell concept with the structure of a bidirectional RNN. A more recent modification aimed at computational efficiency is the GRU.

3.1.3.3 Gated Recurrent Units

A commonly used variation of LSTM is the gated recurrent unit (GRU) [CGCB14]. The GRU simplifies traditional LSTM by combining the input and forget gates into a single update gate, while also merging the cell state and the hidden state and removing the output gate. This reduction of parameters makes a GRU computationally more efficient, while results show that their performance is comparable to that of LSTM.

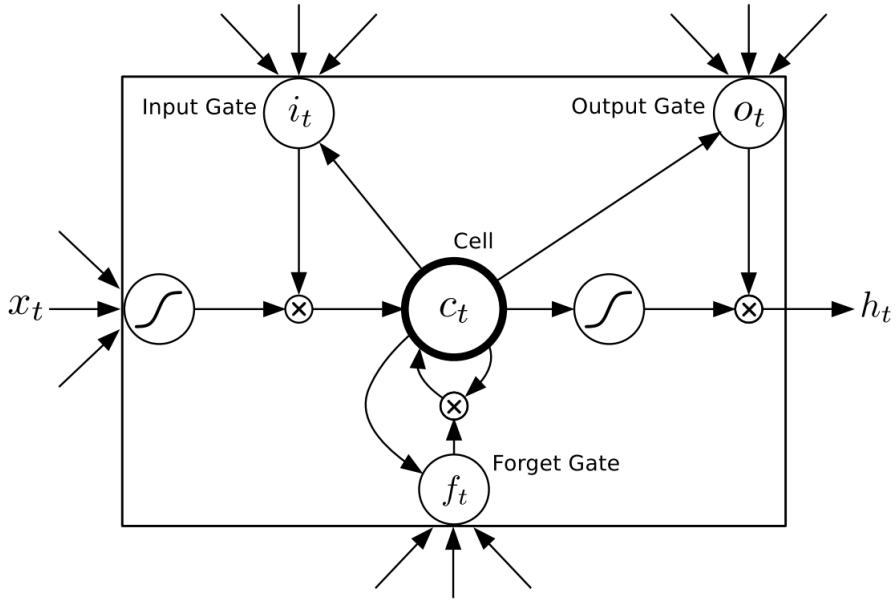

Figure 3.5: An LSTM cell. [http://blog.otoro.net/2015/05/14/long-short-term-memory]

3.1.4 Training Techniques for NNs

Minibatch gradient descent

There are different strategies to applying the backpropagation algorithm when training a neural network. One option is to compute the error and update the network weights for each training sample seperately. The stochastic gradient descent algorithm applies this method by sampling from training points and adjusting the weights accordingly. However, this approach can be slow to converge, as noisy data will quickly result in weight updates that don't reflect the data. Another approach, known as batch gradient descent, is to calculate the error for the complete training set and then adjust the weights based on the average error. This approach unfortunately makes iterations very time-consuming for larger datasets, which is why *minibatch gradient descent* is used. This method divides the training set into smaller batches of samples and then iterates over these subsets, calculating a weight update for each minibatch. Ideally, the minibatches are large enough to be representative of the training data distribution and robust to noisy data points, while being small enough to allow for fast iterations and to fit completely into the memory of the device used, which reduces the iteration time by simplifying the computation.

Newbob scheduling

Newbob scheduling is an algorithm to control the learning rate during the training of a neural network. The training starts with a fixed learning rate, which allows for a quick training, and continues until the change of the validation error falls under a predefined threshold. From this point on, the learning rate is exponentially decreased with each step, fine-tuning the weights to a conversion. Training is completed when the change of validation error falls below a second predefined threshold.

3.2 Tone

In the field of linguistics, prosody is the general term for phenomena in a language's sound system that span across multiple phonetic units. Instead of being properties of a single phoneme, these features are attributes of larger units such as syllables, phrases or sentences. Among them are phenomena such as tone, intonation, and rhythm, all of which can add different levels of meaning to the speech. For example, a speaker can use prosodic elements to emphasize certain words or to define an utterance as a question, but prosodic features can also convey information about the speaker's emotional state.

The definition of *tone* is the usage of pitch to change the lexical or grammatical meaning of a word. It stands in contrast to intonation, which refers to the use of pitch on the utterance level. The term pitch describes the human perception of the fundamental frequency F_0 of speech, which is defined as the rate of opening and closing of the speaker's vocal chords. It can be determined in the spectral representation of an audio signal as the frequency of the first harmonic. The fundamental frequency range is determined by the vocal tract length, making it speaker-dependent, with the average fundamental being about 150 Hz for male speakers and 200 Hz for female speakers.

The categorization of languages into tone languages and languages without tone is not always obvious, as tone is used in different degrees of frequency and distinctiveness - it is however recognized that a tone language is one that uses tone regularly, and not just in a few rare cases. Some estimates state that 60 to 70 percent of the world's languages may be tonal, and some tone languages, such as Mandarin and Vietnamese, are among the most spoken worldwide. A high density of tonal languages can be found in Sub-Saharan Africa, East and Southeast Asia, the Pacific region, and Central America.

There are different mechanisms to create a system of different tones in a language, based on the pitch and its contour. For one, it is possible to define a set of tones with a level contour, but with different relative heights in pitch to each other. This is exhibited frequently in the Bantu language family of Africa, in which most languages distinguish between three or more levels of pitch. Another possibility is to differentiate between types of pitch contour, meaning the change of pitch over time during the syllable. The perhaps most well-studied case of this is found in Mandarin Chinese, which uses four different tones, as well as a neutral tone, to state the meaning of a word. An example of this for the syllable *[ma]* can be seen in Table 3.1.

Word (pinyin)	contour	translation
mā	high level	mother
má	rising	hemp
mǎ	falling-rising	horse
mà	falling	scold
ma	neutral	(question particle)

Table 3.1: Example for the influence of tone in Mandarin on the syllable *[ma]*

Some languages also use a combination of contours and different level tones, such as Cantonese, which defines six different types of tone, demonstrated in Table 3.2 for the syllable *[yau]*:

Word (pinyin)	contour
high level	worry
high rising	paint (noun)
mid level	thin
low level	again
very low level	oil
low rising	have

Table 3.2: Example for the influence of tone in Cantonese on the syllable *[yau]*

Tones can be defined to mean or modify different things. As seen in the case of Mandarin and Cantonese, one option is the difference in lexical meaning for contrasting tones. However, a change in the grammatical meaning of the word can also be induced by the tone, as is often the case in African tonal languages. As an example, the Edo language, which is spoken in Nigeria, uses tone to define the tense of a word, as well as the aspect (meaning if the action is completed or ongoing).

Due to the relative nature of tones and the speaker-dependency of the fundamental frequency, it is not always possible to classify a tone from an isolated speech segment. Additionally, the sentence intonation must be taken into account and distinguished from the syllable-based tone. These difficulties, along with the strong variance across similar dialects, often make the description of a tone system and the classification a challenging task for linguists.

4. Experimental Setup

4.1 Frameworks

This section gives an overview of frameworks that were used for the experimental part of the thesis.

4.1.1 Janus

The JANUS Recognition Toolkit (JRTk) [WAWBC+94] was developed by ISL to provide tools for the development of speech recognition systems for research as well as for applications. The toolkit is created in C and can be programmed using Tcl. Modules include extensive pre-processing options, acoustic modeling, and the Ibis decoder. JRTk was used in this work to create a speech recognition system for Cantonese, upon which phoneme labels were extracted to align the corpus transcription to frames. Additionally, pre-processing and extraction of tone features was also accomplished using JRTk.

4.1.2 Lasagne

Lasagne [DSRO+15] is a Python library which provides modules for building and customizing various types of neural networks. It uses the Theano framework [Thea16], which is responsible for lower-level optimization of training processes on GPUs, to provide efficient functionality for training. Lasagne includes numerous neural network architectures and layer types, including a variety of activation functions and different scheduling techniques. In this work, the different types of recurrent neural networks were realised using Lasagne.

4.1.3 detl

A second framework that was used is detl, which is a Python library for deep learning that supplies functionality for creating and training neural networks. Detl provided the layer models and training algorithms to create the feed-forward neural networks used in this work.

4.2 Data

4.2.1 Dataset

Cantonese was chosen as a tonal language to perform experiments on. The term Cantonese is sometimes used to describe all language varieties of the Yue Chinese language family, which are spoken in the Guangdong and Guangxi provinces of China as well as in Hong Kong. However, the more precise definition, sometimes also known as Cantonese 'proper', describes the form spoken in the cities of Guangzhou and Nanning on the Chinese mainland, along with Hong Kong and Macau. When the broader definition is applied, Cantonese is a language with an estimated 80 million speakers. It is generally considered to have nine distinctive tones, and each morpheme, which is typically one syllable in length, has a tone. However, three of these tones are categorized as 'checked syllables' or 'entering tones' and are observed on syllables that end abruptly with a stop consonant such as 'p', 't' or 'k'. These tones are typically treated as allotones of the three level tones, meaning that the language can be described with six tones.
Data was obtained from the IARPA Babel Cantonese Language Pack [Aeal16], which consists of transcribed speech data spoken by a variety of speakers from five different dialect groups of Cantonese, all of which are spoken in the Guangdong and Guangxi provinces only. These dialect groups were defined with respect to phonological variation, as there are pronunciation differences between regions, as well as geographic location and cultural differences, which lead to differences in lexical choice. The dataset consists of 176 hours of conversational audio recorded from telephone conversations in an 8 kHz 8-bit encoding, of which about 40-50% is speech data [CCRK+13]. Also included are annotations in simplified Chinese characters as well as in a romanized form, including a designation of the tone. In total, 1495 unique syllables were present in the dataset.
In contrast to the Pinyin system used for Mandarin, a standardized romanization scheme for Cantonese does not exist, and representing the language in alphabetic form is recognized as a challenging task in itself. [Aeal16] uses the Yale romanization system [fHKo58]. The Yale system defines seven contrasting tones, while Matthews and Yip [BaBe97] recognize only six, conflating the Yale system's high level and high falling tones. On inspecting the data however, it was found that the high falling tone was only annotated a total of 3 times, leading to the assumption that high level and high falling tones were not distinguished. For this reason, six tone classes were used in all experiments. Table 4.1 lists the six different tones that were defined in the dataset.

Class	Tone
1	high level
2	high rising
3	mid level
4	low falling
5	low rising
6	low level

Table 4.1: The tone classes present in the dataset

The dataset was divided into a large training set with 80% of the data, and validation and test sets with 10% each, with each set using completely different speakers. Table 4.2 shows the distribution of tones to syllables in the dataset.

Tone	Syllables
1	28.48%
2	12.77%
3	16.37%
4	9.32%
5	12.71%
6	20.34%

Table 4.2: Distribution of tones to syllables in the dataset

4.2.2 Features

The first step was the extraction of frame-wise tone features. A frame distance of 10ms was used, with a windows size of 32ms. For one, the absolute fundamental frequency F_0 was extracted, along with three delta and three double-delta values over a range of 1, 2 and 3 frames in both directions. Secondly, the fundamental frequency variation (FFV) features [LaHE08] were computed. FFV provides a continuous measure for frequency variation that is instantaneous, meaning it is computed without context from adjacent frames. It makes use of all harmonics in the speech signal (as opposed to only the first harmonic) and evaluates a vanishing-point product to obtain seven parameters that determine the variation of the fundamental frequency. Additionally, the frame energy was computed, along with three delta and three double-delta values in the same scheme as the F_0 features. This led to a total feature dimension of 21.

4.2.3 Data Extraction

To label the data, each frame had to be annotated with one of the six tone classes present in Cantonese. Tone annotations were provided by the dataset on a syllable basis. Using the transcript and pronunciation dictionary, a reference phoneme sequence was created for each utterance. An existing JANUS speech recognition system trained on the same dataset was then used to provide frame-wise phoneme sequences of the utterances, which were then aligned to the reference phoneme sequences, allowing each frame to be assigned to a syllable and thus also a tone.
Frames containing silence, noises, or phonemes that could not be aligned to the reference were annotated with a seventh class with the label '0'. The distribution of tones to frames in the training set is presented in Table 4.3.

Tone	Number of frames
0	8032872
1	2910107
2	962970
3	1482438
4	596194
5	879093
6	2006673

Table 4.3: Distribution of tones to frames the dataset

As the utterances frequently contained long durations of silence, there is a high number of frames with label '0'. These frames can be important as context for frames containing speech, which is the reason why this class is included in the classification. However, long periods of silence contain many frames that are too far away from the nearest speech segment to contain useful information, and were deemed to be of little use to tone recognition. Therefore, it was decided create a modified dataset that does not include class '0' frames that are more than 7 frames away from the nearest frame annotated with a tone, significantly reducing the number of class '0' frames.

The examination of the data also shows that the tone classes are not equally represented. This is in part due to the uneven distribution of syllables shown in Table 4.2, and is further caused by the fact that syllables with certain tones were longer on average. This can be seen in Table 4.4.

Tone	Average syllable length
1	40
2	27
3	34
4	24
5	23
6	37

Table 4.4: Average syllable length for the different tones

As a consequence, the a balanced training set was created by undersampling the data until all six tone classes were included with the same number of frames.

4.3 Experimental Approach

The experiments that were performed can be divided into two seperate groups that take different approaches.

The experiments in the first grouped perform classification on single frames, returning a predicted tone class for each frame separately. Initial experiments were performed on a feedforward neural network and compared different types of features

as well as the impact of including an additional class for nno speechïnto the classification. Afterwards, experiments moved on to recurrent neural networks, including the comparison of different network types and architectures as well as various feature types and context lengths. The best-performing configuration was finally tested on the test set.

The second group consists of experiments following the approach of a syllable-based classification. After explaining how the data points were modified to allow for this tactic, two different classifier types are described - a k-nearest neighbors classifier as well as two different neural networks.

5. Experimental Results

This chapter describes in detail the experiments that were performed on the dataset, and the results that they produced. Initial experiments were performed on feedforward neural networks, which are presented in the first section. The second section describes the work with different types of recurrent neural networks and the testing of various parameters. In the third section, experiments with syllable-based features show an alternative approach. The chapter closes with a final evaluation on the test set.

5.1 Feedforward Neural Networks

First classification experiments were performed on the unaltered dataset with a feedforward deep neural network architecture that had previously been used successfully in speech recognition setups. To compare the influence of the different feature types, three models were trained: one model using only pitch and FFV features, one model based on energy features, and a third model which utilized all features. A context of 7 was used, meaning that the input consisted of the frame along with the 7 previous and 7 following frames. This brought the input dimension to 210 for tone features, 105 for energy features and 315 for the combination. All three nets consisted of 4 hidden layers with 1000 neurons each, and an output layer with 7 neurons. Table 5.1 shows the performance of the three models on the validation set.

Features	Frame Error Rate
Pitch + FFV	47.26%
Energy	47.92%
Combination	46.19%

Table 5.1: Results for different features in a feedforward neural network

Although the error rate is significantly lower than expected for all three models, a closer examination of the results showed that the network simply classified most frames as belonging to the '0' class. As frames with this label make up almost 50% of

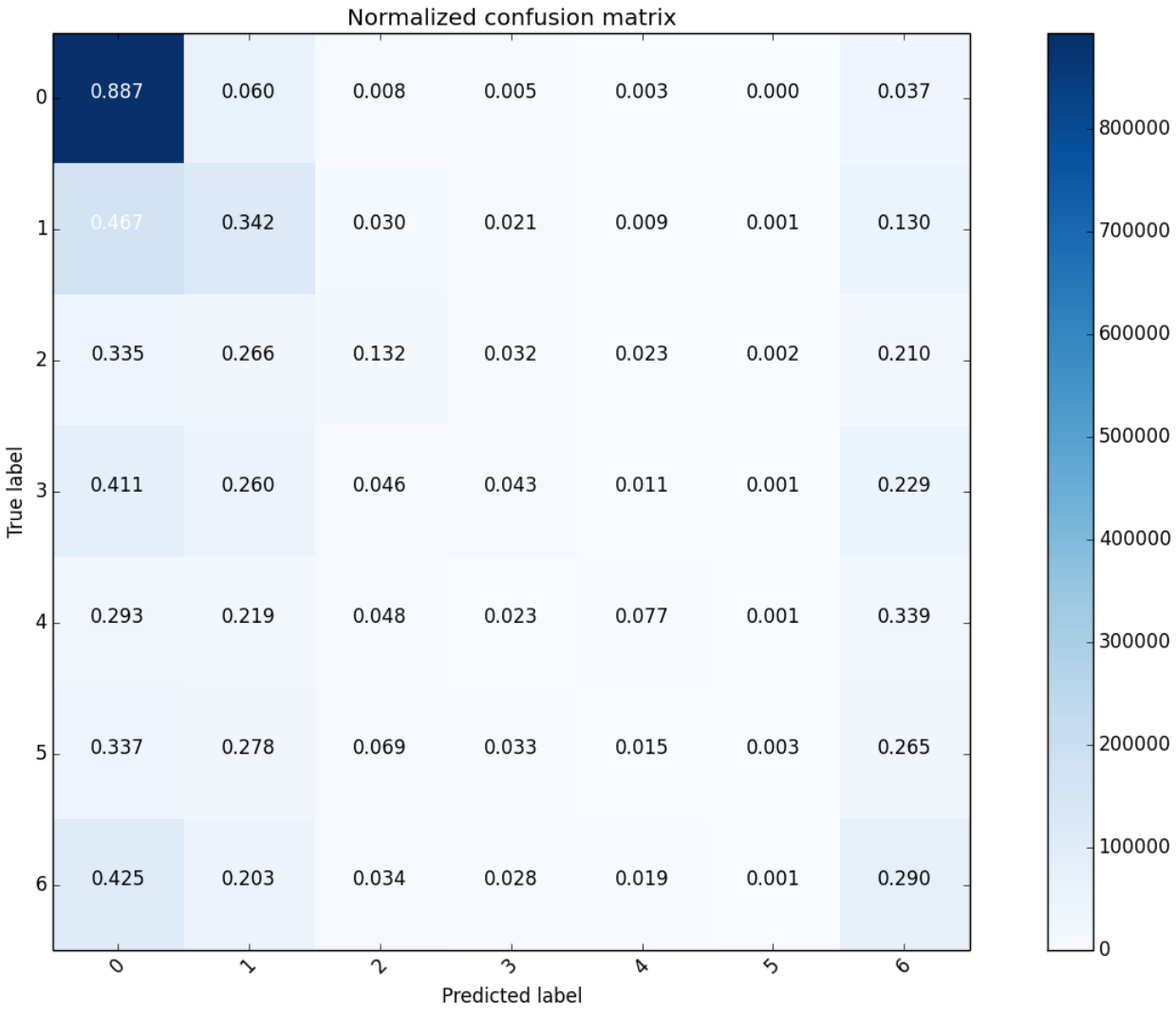

Figure 5.1: Confusion matrix for combined features in a feedforward neural network. Most frames were predicted to belong to the '0' class, and the good frame error rate values can be explained by the fact that this class made up the largest portion by far.

the dataset, the error rates are not surprising. Of the remaining frames, the largest share was labeled with one of the classes that were represented more strongly in the dataset, namely the '1' and '6' tones. It appeared that the uneven distribution was influencing the classifier, and for this reason, only the balanced version of the training set, with the irrelevant '0' frames removed, was used in further experiments.

In the next step, the same net architecture was applied to the modified training set, this time only once for the combination of all features. Performance dropped to an error-rate of 89.75%, which is significantly below chance. A confusion matrix of the predicted labels for the validation set showed that the network was still mainly outputting 3 classes, while some class labels were not even output a single time.

To investigate the influence of including the '0' class in classification, a neural net that distinguishes between 6 classes instead of 7 was trained. In order to preserve

the context information that the '0' frames include, the dataset for this case was further modified by giving these frames the labels of the neighbouring tones for which they provide context. Unfortunately, this approach proved to be unsuccessful, as the model classified every single input frame from the validation set as having tone '5'.

5.2 Recurrent Neural Networks

5.2.1 Comparison of different RNN types

After feed-forward neural networks were shown to be unsuited to the task, several experiments were performed with recurrent neural networks. Due to the temporal nature of speech, information from previous and future time states is often relevant to recognition tasks. This is especially the case with tone recognition, as the process relies strongly on the change of pitch over time. Since recurrent neural networks are designed to integrate information from surrounding time states, it was a logical next step to try this type of network.

As a first step, three different types of RNN were compared: a classic RNN, a standard LSTM model, and a GRU model. All models were bidirectional in order to integrate preceding frames as well as succeeding frames. The architecture was kept simple and was shared across all three RNNs - a single hidden layer with 150 units. Context was included from the 7 neighbouring frames in either direction, and only pitch and FFV features were used in order to keep the input dimension smaller.

Model Type	Frame Error Rate
Basic RNN	77.40%
LSTM	68.06%
GRU	67.37%

Table 5.2: Results for different recurrent neural networks

The results, presented in Table 5.2 show that all three models are clearly superior to the feed-forward architecture, performing with a better-than-chance accuracy. The two LSTM variants outperform the classic RNN by a significant margin, indicating that their ability to represent long-term dependencies may make LSTM models the most appropriate RNN variant to the tone recognition task. For this reason, further experiments focus on LSTM networks.

5.2.2 Comparison of features

To measure the influence of the different types of features, two LSTM models with the same structure as in the previous experiment were trained separately, once only with the 14-dimensional tone features, consisting of pitch contour and FFV coefficients, and once using the 7-dimensional energy features. The results, which can be seen in Table 5.3, show that they can not match the performance of the LSTM

trained on all available features. The numbers also indicate that tone-related features are the most valuable information for tone recognition, while energy features are certainly beneficial to the system, but not capable of representing tones by themselves.

Features	Frame Error Rate
Pitch + FFV	68.06%
Energy	77.01%
Combination	66.47%

Table 5.3: Results for different features in a LSTM network

5.2.3 Comparison of context lengths

The next step was to compare different context sizes. The previously used context value of 7 means that the tone classification is based on a sequence of 15 frames. However, average syllable lengths for the different tones range between 23 and 40 frames, meaning that a larger context may be more suitable to represent the full pitch contour of a tone.
The same LSTM structure was used, based on the combined 21-dimensional features.

Context length [frames]	Frame Error Rate
11	67.75%
15	66.47%
19	67.99%
23	68.02%
31	68.41%

Table 5.4: Results for different context lengths in a LSTM network

As can be seen from the performance figures in Table 5.4, the system performs slightly worse every time the context size is increased, while choosing a smaller context length also leads to suboptimal results. Overall, there is no strong difference in the error rates for the different context sizes. The results suggest that the choice of context length does not have a strong influence on the system performance, at least not for the given network structure.

5.2.4 Testing LSTM network parameters

In order to find the appropriate network size for the tone recognition task, experiments comparing different network structures were performed.

In the first step, the number of units in the hidden layer was varied from 150 to other values, while the context size remained fixed at 15. This lead to the results in Table 5.5:

Number of hidden units	Frame Error Rate
100	73.92%
150	66.47%
200	67.58%
300	66.03%
500	67.96%

Table 5.5: Results for LSTM networks with different layer sizes

As can be seen from the frame error rate, increasing the size of the hidden layer did not have a strong impact on the system performance. Using 300 hidden units led to a very slight increase in performance, while using 200 or 500 units showed a minimally higher frame error rate. Reducing the number of units to 100 showed a strong decrease in result quality.

Up to this point, context and the number of hidden units had been varied separately, so the possibility remained that a simultaneous increase in context size and layer size could benefit the system. It appears plausible that the increased input dimension would justify a larger network.

To test this possibility, a network with a context length of 31 and a hidden layer size of 300 units was trained. It performed with a frame error rate of 67.32%, which is similar, but not superior to previously tested systems.

A second way to increase the complexity of the network is to add a second layer of hidden units. Several nets featuring two hidden layers were tested with different combinations of context length and layer size. The results can be seen in Table 5.6.

Hidden units per layer	Context length [frames]	Frame Error Rate
150	15	75.95%
150	23	87.42%
250	15	85.94%

Table 5.6: Results for LSTM networks with different dimensions

The results show that the addition of a second hidden layer into the network had a negative impact of the frame error rate. For two of the three combinations, the results were not better than chance.

5.2.5 Analysis of best-performing system

To gain further understanding of the functionality of the best system, a LSTM network with one hidden layer of 300 units, the results on the validation set were examined more closely. A confusion matrix with the different classes is shown in Figure 5.2.

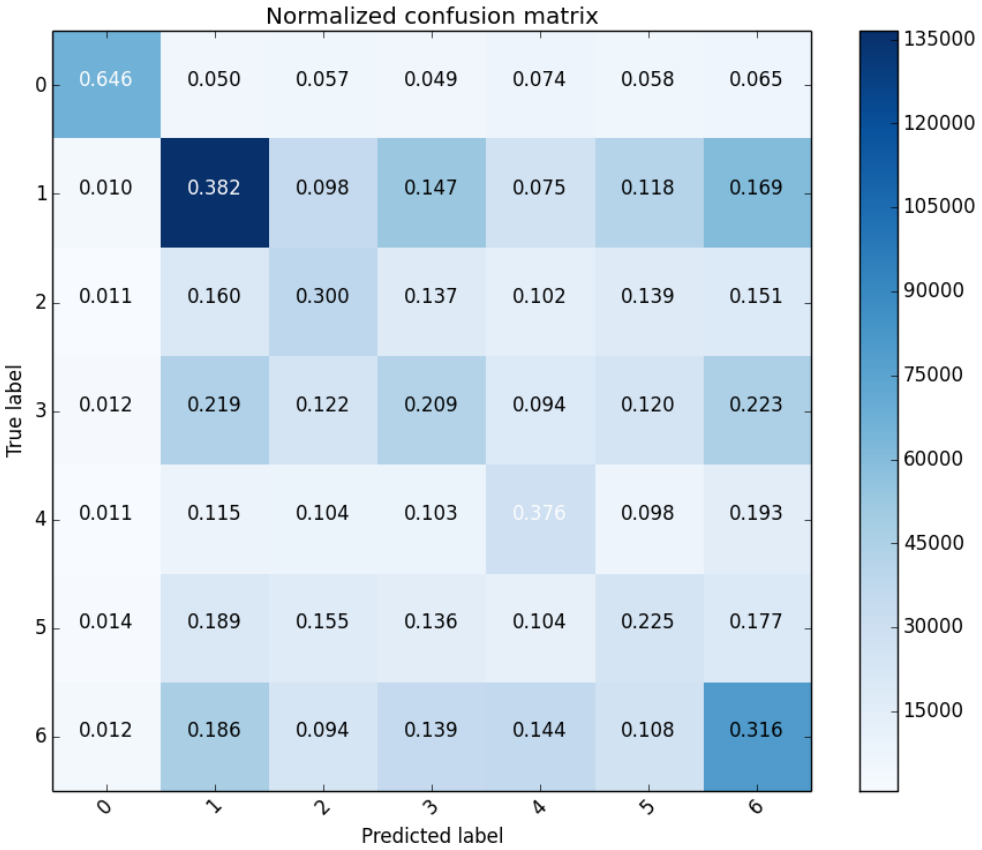

Figure 5.2: Confusion matrix for the best LSTM on the validation set

The confusion matrix shows that not every tone is recognized equally well. The high level (1) and low falling (4) tones were distinguished best from the others, while the mid level (3) and low rising (5) tones were most often not identified correctly. It can also be seen that the three level tones (1), (3) and (6) were frequently confused with other level tones when misclassified, suggesting that the classification relies strongly on contour. This confusion between tones with similar contours is less distinct for the tones with rising contour, (2) and (5), but it is observable that the relative frequency for the false prediction of tone (5) is highest for syllables with tone (2), and vice versa. For tone (4), the only tone with falling contour, the frequency of false predictions is spread relatively even across classes.

Additionally, discriminating non-speech frames from class 0 works relatively well, with a significantly lower error rate than on speech frames.

Closer examination of some classification results showed that certain types of mistakes occurred frequently. For one, boundaries between two syllables with different tones seem to be a challenge. As presented in an example in Table 5.7, frames in these transition sections are often misclassified, with a strong fluctuation between classes. One reason for this could be that the tone is not always recognizable in these parts of syllables due to the nature of connected speech. Since speech is an anticipatory process, the usage of tone on a syllable is dependent on the following syllable and its tone. The fact that the same syllable with the same tone can sound different depending on its context is an aspect of tone recognition that can also be challenging for linguists.

Reference sequence	6	6	6	6	6	6	6	6	6	6	6	6	1	1	1	1
Predicted sequence	6	6	6	2	2	2	2	2	2	1	1	1	1	1	1	1

Table 5.7: Example frame sequence showing errors around syllable boundaries

Examples such as the frame sequence depicted in Table 5.8 show that this issue is also a problem at word boundaries. In this case, the location of the word boundary was recognized correctly, but the final frames of the syllable were classified incorrectly and with strong fluctuation.

Reference sequence	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0
Predicted sequence	2	2	2	2	2	2	6	3	1	6	5	0	0	0	0	0

Table 5.8: Example frame sequence showing errors on borders between syllables and silence

Another source of error that was observed often, although less frequently than errors at syllable boundaries, was single frames (or small groups) inside a larger block of correctly recognized frames being misclassified. An example frame sequence for this case is presented in Table 5.9. The large part of the syllable (which includes more

l	Frame Error Rate
0	66.03
1	65.50
2	65.20
3	64.81
4	64.50
5	64.20
6	64.11
7	63.98
8	63.88
9	63.89
10	63.90
11	63.94
12	64.09
13	64.30
14	65.04
15	65.41

Table 5.10: Frame error rate for application of minimum error criterion with different lengths

frames on both sides not pictured here) was correctly recognized, while multiple small groups of one or two frames were classified incorrectly.

Reference sequence	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Predicted sequence	1	1	1	6	6	1	1	3	1	1	1	1	6	1	1	1

Table 5.9: Example frame sequence showing short errors inside a syllable

5.2.6 Minimum length criterion

Syllables in the data are well above 20 frames in length on average, and the articulation of a syllable in only a few frames is, in most cases, not realistic. Inspired by examples such as the one in Table 5.9, a minimum length criterion for tones was defined, which states that a sequence of frames in the classification output must be at least l frames in length. This criterion was applied as a smoothing technique in certain cases in the validation set results as follows:

Given a label sequence of length n with $f = f[1], ..., f[n]$ and a minimum length l, if $f[i] = f[i + l + 1]$, then $f[j]$ is set to $f[i]$ for $i < j \leq i + l$.

The frame error rate for different values of l is given in Table 5.10. The application of the minimum length criterion to frame groups bounded by the same label on both sides leads to an absolute performance increase of 2.05%.

5.3 Syllable-based Approach

In previous experiments, classification was performed on single frames, as opposed to entire syllables. This approach is very universal in that it does not depend on

the syllable length and can be used without knowledge of syllable boundaries, and including surrounding frames allows for context to be taken into account. However, it was frequently observed that frames which belong to the same syllable (and therefore share the same tone) were labeled with different tone classes.

In contrast, syllable-based classification leads to one decision for the entire syllable, which eliminates the possibility of this inconsistency. To investigate if focusing on syllables instead of frames can improve the classification performance, tone classification was performed on syllables from the same data using different classifiers.

To achieve a syllable-based representation, the dataset was first split into groups of frames marked by syllable boundaries, which had already been computed as a by-product of the data extraction. As the syllables contained different amounts of frames, making them difficult to compare, all groups were reduced to a 10-point representation using linear interpolation on each of the 21 feature dimensions. Thus, classification was performed on 210-dimensional vectors for the combined feature case. Only the six tones were included in classification; frames containing silence or noises were omitted.

5.3.1 K-Nearest Neighbors

For the purpose of gauging the performance of neural networks in general when applied to the task of tone recognition, the k-nearest neighbors method for classification was tested.

The k-nearest neighbors (k-NN) algorithm classifies a sample by examining the k points that are nearest to the sample in the feature space and assigning a class to the sample based on a majority vote of these points. Multiple values for k (1, 10 and 100) were tested on a small subset, and $k = 100$ was decided upon as the most promising setting. As a distance metric, the Euclidean distance was used, which is defined as

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}. \tag{5.1}$$

In addition to the combined features, separate classifiers for pitch and FFV features as well as energy features were also trained. The performance of the classifiers can be seen in Table 5.11.

Features	Syllable Error Rate
Pitch + FFV	65.34%
Energy	73.91%
Combination	64.82%

Table 5.11: Results for a kNN classifier with different features

It can be seen that the k-NN classifier performs better than chance for all feature inputs, while the combination of all features again performs best, with an error rate

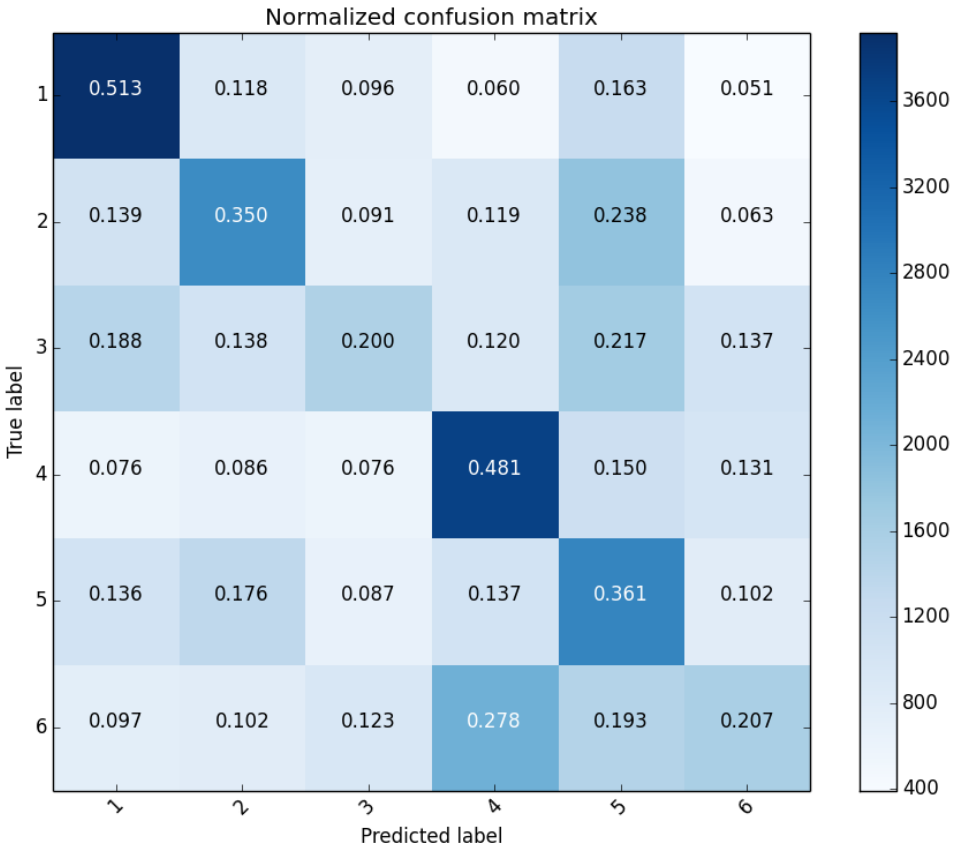

Figure 5.3: Confusion matrix for k-NN classifier with combined features

of 64.82%. In this case, integrating the energy features had a very low impact on performance.

A confusion matrix for a k-NN classifier trained with the combined features is shown in Figure 5.3. The recognition of the high level tone (1) works with a relatively high accuracy. However, the other level tones (3) and (6) are misclassified much more frequently than in the frame-based classifier. In comparison to the frame-based system, the syllable-based classifier performs significantly better on tones with a non-flat pitch contour, with much higher recognition rates for the tones (2), (4) and (5).

Since a single decision is made for a whole group of frames, the error rates are not directly comparable to the results of the frame-based neural network classifiers. They do however prove that the distinction of different tones is possible to some degree.

5.3.2 Neural Networks

The syllable-based features were also tested with two types of neural network: the 4-layer feedforward neural network from section 5.1, as well as one the more successful recurrent networks from section 5.2, a LSTM network with 150 hidden units. Both nets were trained with the full 210-dimensional feature vectors. The results are presented in Table 5.12.

Classifier	Syllable Error Rate
FFNN	83.41%
LSTM	74.55%

Table 5.12: Results for two syllable-based neural networks

The syllable error rates show that this approach was not very successful. The feedforward network performed with chance accuracy, while the performance of the LSTM network was better than chance. However, both networks were outperformed by the simple k-nearest neighbors classifier.

5.4 Final Evaluation

Finally, the best-performing system was evaluated on the test set. As the main focus of this work was on the more universal frame-based classifiers, the evaluation was performed on the most promising system of this type, which was a LSTM network with one hidden layer of 300 units, and a context length of 15. Evaluation was performed both with and without smoothing based on the minimal length criterion. The frame error rates can be seen in Table 5.13.

Classifier	Frame Error Rate
LSTM (without smoothing)	65.61%
LSTM (with smoothing)	63.47%

Table 5.13: Results on test set for the best-performing system

The system performed relatively well with frame error rate of **65.61%**, which is even slightly lower than the error rate on the validation set. With the application of the minimal length criterion, the error rate could be lowered to **63.47%**.

Overall, the results are more than satisfactory, as the classifier performs with an accuracy that is significantly better than chance, which shows that a tone recognition system of this type can already provide a useful contribution to any task that requires the recognition or annotation of tone.

The performance is also notable considering the challenging nature of the data which was used. For one, the recordings originate from a variety of speakers and dialects, both of which are factors that have a strong influence on the the usage of pitch and the definition of tone in speech. Secondly, the audio quality is far from optimal, due to the fact that the data stems from telephone recordings. It is expected that performance can increase when higher-quality speech data is used.

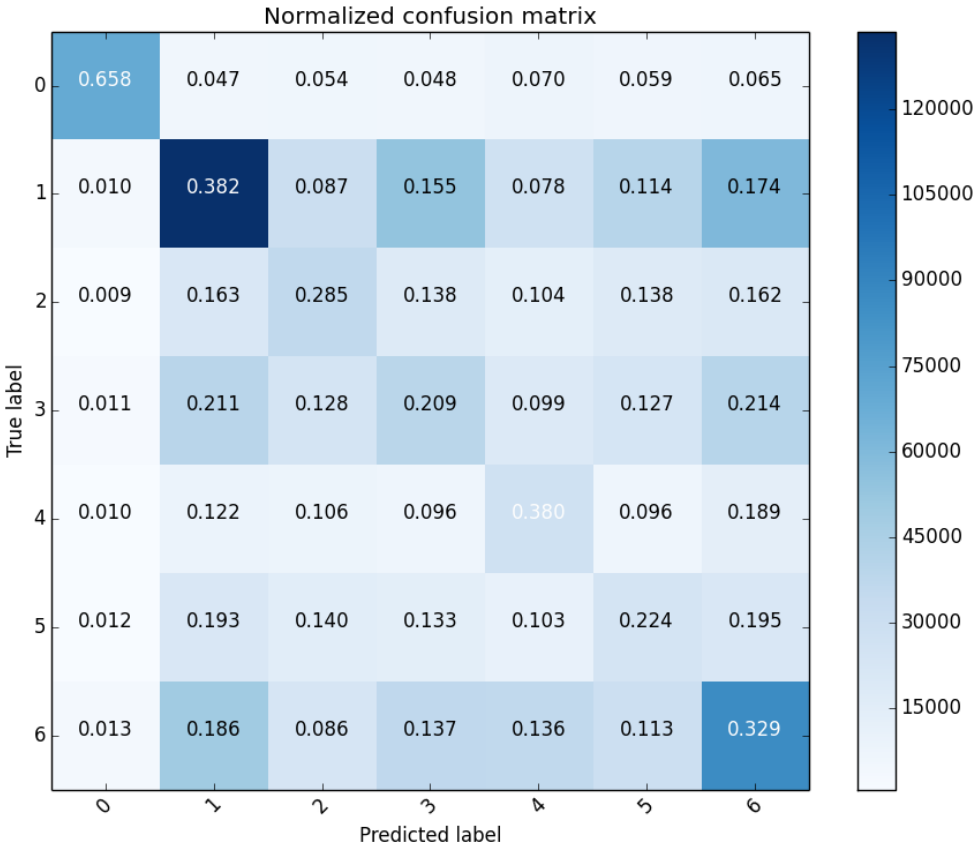

Figure 5.4: Confusion matrix for evaluation of the test set

6. Conclusion

The final chapter provides a short summary of experiments and results of the thesis, and suggests ideas for the further research of the topic.

6.1 Summary

In this thesis, we evaluated different approaches to the language-independent tone recognition task, using a dataset of Cantonese speech recordings from a variety of speakers and regions.

The focus of the work was on developing classifiers for the frame-wise recognition of tone. First experiments were performed using a deep neural network with 4 hidden layers of 1000 units each, which performed with a below-chance accuracy, leading to the decision to switch to recurrent neural networks.

The comparison different RNN types showed a significant performance increase for long short-term memory models. With the goal of optimizing performance for a LSTM network, a number of feature types and system parameters were tested. The highest performance was achieved using a combination of pitch, fundamental frequency variation and frame-wise energy features. As tone recognition relies on pitch contours, neighboring frames were included into classification as context - however, altering the context length had no significant effect on the system performance. A number of classifiers with different numbers of hidden units as well as additional hidden layers were trained. The best-performing model was a LSTM network with one hidden layer of 300 units and a context length of 15 frames, which attained a frame error rate of 65.61% on the test set. With the application of a smoothing technique to reduce the irregularity of classifier output in frame sequences, the error rate was further reduced to 63.47%.

As an alternative to performing classification of single frames, a syllable-based tone recognition was also investigated. Equal-length representations of syllables were

computed using linear interpolation. Performance of DNN and LSTM networks was compared to that of a k-nearest neighbors classifier. The best k-NN model reached a syllable error rate of 64.82%, outperforming both neural networks by a significant margin.

In conclusion, the experiments proved to be successful, as both frame-wise and syllable-wise approaches led to models that perform with an accuracy that is well above chance performance despite the difficulty of the dataset, and are therefore capable of providing beneficial results to the tone recognition task.

6.2 Future Work

Although a high number of experiments were performed with different models and parameters, there are numerous ways in which the described tone recognition system can possibly be improved.

One potential way of improving results is the integration of additional features, such as harmonicity or PaIntE parameters, as these features were shown to improve results of language-specific speech recognition systems.

Furthermore, investigation of frame-wise classification results showed numerous patterns that do not reflect realistic speech processes. The development of constraints or additional smoothing techniques to prevent or correct this behaviour could potentially benefit the quality of results.

Finally, a better evaluation of the system could be attained by performing similar experiments on additional tonal languages and datasets. As tonality is displayed differently across languages, an assessment of which types of tone are easier or more difficult to classify would certainly be noteworthy. Examining the influence of the quality of recordings in the dataset would also allow for better grading of experimental results.

Bibliography

[Aeal16] T. Andrus et al. IARPA Babel Cantonese Language Pack IARPA-babel101b-v0.4c LDC2016S02. Philadelphia: Linguistic Data Consortium, 2016. Web Download.

[BaBe97] R. S. Bauer und P. K. Benedict. *Modern Cantonese Phonology*. Mouton de Gruyter, Berlin/NY. 1997.

[CCRK+13] J. Cui, X. Cui, B. Ramabhadran, J. Kim, B. Kingsbury, J. Mamou, L. Mangu, M. Picheny, T. N. Sainath und A. Sethy. Developing Speech Recognition Systems for Corpus Indexing under the IARPA Babel Program. 2013.

[CGCB14] J. Chung, C. Gulcehre, K. Cho und Y. Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. 2014.

[DeHK13] L. Deng, G. Hinton und B. Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. 2013.

[DSRO+15] S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. K. Sønderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, J. D. Fauw, M. Heilman, D. M. de Almeida, B. McFee, H. Weideman, G. Takács, P. de Rivaz, J. Crall, G. Sanders, K. Rasul, C. Liu, G. French und J. Degrave. Lasagne: First release., August 2015.

[fHKo58] P. P. fei Huang und G. P. Kok. *Speak Cantonese*. Institute of Far Eastern Languages, Yale University. 1958.

[GeSC99] F. A. Gers, J. Schmidhuber und F. Cummins. Learning to forget: continual prediction with LSTM. Band 2, 1999, S. 850–855.

[GrSc05] A. Graves und J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* Band 18, 2005.

[HoSc97] S. Hochreiter und J. Schmidhuber. Long Short-Term Memory. *Neural Computation* Band 9, 1997.

[HuQS14] W. Hu, Y. Qian und F. K. Soong. A DNN-based acoustic modeling of tonal language and its application to Mandarin pronunciation training. 2014.

[Kasa96] N. K. Kasabov. *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. The MIT Press. 1996.

[LaHE08] K. Lasbowski, M. Heldner und J. Edlund. The fundamental frequency variation spectrum. 2008.

[LiBE15] Z. C. Lipton, J. Berkowitz und C. Elkan. A Critical Review of Recurrent Neural Networks for Sequence Learning. arXiv preprint arXiv:1506.00019, 2015.

[LTGL+93] L. shan Lee, C. yu Tseng, H. yan Gu, F. hua Liu, C. hao Chang, Y. hong Lin, Y. Lee, S.-L. Tu, S.-H. Hsieh und C. hung Chen. Golden Mandarin (1) - A Real-Time Mandarin Speech Dictation Machine for Chinese Language with Very Large Vocabulary. *IEEE Transactions on Speech and Audio Processing* 1(2), April 1993.

[MaYi94] S. Matthews und V. Yip. *Cantonese: a comprehensive grammar*. Routledge, London/New York. 1994.

[McPi43] W. S. McCulloch und W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics 5.4*, 1943, S. 115–133.

[MSWG+13] F. Metze, Z. A. W. Sheikh, A. Waibel, J. Gehring, K. Kilgour, Q. B. Nguyen und V. H. Nguyen. Models of Tone for Tonal and Non-Tonal Languages. 2013.

[Rams87] S. R. Ramsey. *The Languages Of China*. Princeton University Press, Princeton, New Jersey. 1987.

[Rose57] F. Rosenblatt. The perceptron - a perceiving and recognizing automaton. Technischer Bericht Report 85-460-1, Cornell Aeronautical Laboratory, 1957.

[RuHW88] D. E. Rumelhart, G. E. Hinton und R. J. Williams. *Neurocomputing: Foundations of Research*. MIT Press, Cambridge, MA, USA. 1988.

[Schu99] K. Schubert. Grundfrequenzverfolgung und deren Anwendung in der Spracherkennung. Diplomarbeit, Universität Karlsruhe, 1999.

[ScKi06] T. Schultz und K. Kirchhoff. *Multilingual Speech Processing*. Elsevier Academic Press. 2006.

[ScPa97] M. Schuster und K. K. Paliwal. Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing* 45(11), 1997.

[ScVu16] A. Schweitzer und N. T. Vu. Cross-Gender and Cross-Dialect Tone Recognition for Vietnamese. 2016.

[Thea16] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints* Band abs/1605.02688, Mai 2016.

[WAWBC+94] M. Woszczyna, N. Aoki-Waibel, F. D. Buo, N. Coccaro, K. Horiguchi, T. Kemp, A. Lavie, A. McNair, T. Polzin und I. Rogina. JANUS 93: towards spontaneous speech translation. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1994.

[Yip02] M. Yip. *Tone*. Cambridge University Press. 2002.

[Yue-91] A. Yue-Hashimoto. The Yue dialect. *Languages and Dialects of China [Journal of Chinese Linguistics Monograph Series Number 3]*, 1991, S. 292–324.

This effort uses the IARPA Babel Program language collection release babel101-v0.4c.