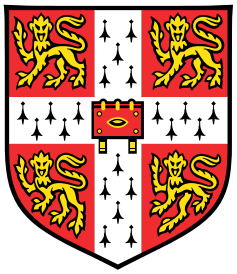
Deep Learning for Automatic Assessment and Feedback of Spoken English



Konstantinos Kyriakopoulos

Supervisor: Prof. Mark J.F. Gales

Department of Engineering
University of Cambridge

This thesis is submitted for the degree of Doctor of Philosophy

Queens' College October 2021

To my mother Elena

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Konstantinos Kyriakopoulos
October 2021

Acknowledgements

I must begin by acknowledging my supervisor, Mark Gales, whose vision, mentorship and guidance made this work possible. I thank him for everything he has taught me, for his unwavering support, and for his wisdom, leadership, understanding and patience. I am also very grateful to Kate Knill, for all her support, guidance and advice.

This thesis forms part of the work of the ALTA Institute and builds on the research and systems developed by other members of the ALTA team. In addition to Mark Gales and Kate Knill, I would like to express my gratitude to Andrew Caines, Rogier van Dalen, Gilles Degottex, Calbert Graham, Yiting Lu, Andrey Malinin, Anton Ragni, Vyas Raina, Linlin Wang, Yu Wang and Xixin Wu, whose work, help and feedback I have relied on in the course of this research. I also thank Cambridge Assessment who provided the funding and data for this project.

Next, I would like to thank Andrew Gee, for his mentorship and support throughout my time at Cambridge. Finally, my thanks go to the people who kept me sane over the years, in particular my parents, Panos and Elena, Alkisti, and Violeta.

Abstract

Growing global demand for learning a second language (L2), particularly English, has led to considerable interest in automatic spoken language assessment, whether for use in computer-assisted language learning (CALL) tools or for grading candidates for formal qualifications. This thesis presents research conducted into the automatic assessment of spontaneous non-native English speech, with a view to be able to provide meaningful feedback to learners. One of the challenges in automatic spoken language assessment is giving candidates feedback on particular aspects, or views, of their spoken language proficiency, in addition to the overall holistic score normally provided. Another is detecting pronunciation and other types of errors at the word or utterance level and feeding them back to the learner in a useful way.

It is usually difficult to obtain accurate training data with separate scores for different views and, as examiners are often trained to give holistic grades, single-view scores can suffer issues of consistency. Conversely, holistic scores are available for various standard assessment tasks such as Linguaskill. An investigation is thus conducted into whether assessment scores linked to particular views of the speaker's ability can be obtained from systems trained using only holistic scores.

End-to-end neural systems are designed with structures and forms of input tuned to single views, specifically each of pronunciation, rhythm, intonation and text. By training each system on large quantities of candidate data, individual-view information should be possible to extract. The relationships between the predictions of each system are evaluated to examine whether they are, in fact, extracting different information about the speaker. Three methods of combining the systems to predict holistic score are investigated, namely averaging their predictions and concatenating and attending over their intermediate representations. The combined graders are compared to each other and to baseline approaches.

The tasks of error detection and error tendency diagnosis become particularly challenging when the speech in question is spontaneous and particularly given the challenges posed by the inconsistency of human annotation of pronunciation errors. An approach to these tasks is presented by distinguishing between lexical errors, wherein the speaker does not know how a particular word is pronounced, and accent errors, wherein the candidate's speech exhibits consistent patterns of phone substitution, deletion and insertion. Three annotated corpora

of non-native English speech by speakers of multiple L1s are analysed, the consistency of human annotation investigated and a method presented for detecting individual accent and lexical errors and diagnosing accent error tendencies at the speaker level.

Table of contents

List of figures xv

List of tables xxv

Nomenclature xxxi

1 Introduction 1

2 Spoken Language Proficiency Assessment 7

2.1 Views of Proficiency . . . 8
2.2 Speech Processing . . . 9
2.3 Single-view grading . . . 13
2.3.1 Text . . . 14
2.3.2 Pronunciation . . . 15
2.3.3 Tempo . . . 18
2.3.4 Stress . . . 19
2.3.5 Rhythm . . . 22
2.3.6 Intonation . . . 27
2.4 Holistic Grading . . . 31
2.5 Comparison of approaches . . . 33
2.6 Chapter Summary . . . 34

3 Pronunciation Error Detection 37

3.1 Native speaker similarity methods . . . 38
3.2 ASR confidence methods . . . 41
3.3 Extended Recognition Networks . . . 44
3.4 Phone Recognition . . . 47
3.5 Supervised methods . . . 49
3.6 Data . . . 50

3.7 Pronunciation Feedback . . . 52
3.8 Comparison of Approaches . . . 52
3.9 Chapter Summary . . . 54

4 Deep Learning 55
4.1 Neural networks . . . 55
4.2 Neural Network Architectures . . . 57
4.2.1 Feed-forward networks . . . 57
4.2.2 Recurrent Neural Networks . . . 59
4.2.3 Attention Mechanisms . . . 63
4.2.4 Siamese Networks . . . 68
4.3 Training Criteria . . . 70
4.4 Optimisation . . . 73
4.4.1 Gradient descent methods . . . 73
4.4.2 Normalisation . . . 75
4.4.3 Parameter Initialisation . . . 77
4.4.4 Learning Rate Schedules . . . 78
4.5 Chapter Summary . . . 79

5 Deep Learning for Spoken Language Proficiency Assessment 81
5.1 Multi-view grading . . . 82
5.2 Single-view graders . . . 85
5.2.1 Pronunciation . . . 85
5.2.2 Rhythm . . . 93
5.2.3 Intonation . . . 95
5.3 Grader Combination . . . 100
5.4 Chapter summary . . . 103

6 Experiments on Spoken Language Proficiency Assessment 105
6.1 Data . . . 106
6.2 Baseline Systems . . . 108
6.3 Performance of single-view graders . . . 111
6.3.1 Pronunciation . . . 112
6.3.2 Rhythm . . . 116
6.3.3 Intonation . . . 117
6.4 Relationship between two-stage and end-to-end graders . . . 120
6.5 Validity of single-view graders . . . 128

6.6 Grader combination . . . 133
6.7 Chapter Summary . . . 134

7 Experiments on Pronunciation Error Detection 137
7.1 Accent and Lexical Errors . . . 138
7.1.1 Generating candidate accent errors . . . 140
7.1.2 Generating lexical errors . . . 142
7.2 Accent and Lexical Error Detection . . . 144
7.3 Corpora and Annotations . . . 149
7.4 Experiments . . . 154
7.4.1 Error annotation . . . 154
7.4.2 Candidate error generation . . . 155
7.4.3 Detection Performance . . . 160
7.4.4 Relationship to proficiency grade . . . 166
7.5 Chapter Summary . . . 167

8 Discussion 169

9 Conclusions 173

References 181

Appendix A Speech Feature Extraction 201
A.1 Filterbank Features . . . 201
A.2 Mel Frequency Cepstral Coefficients . . . 203
A.3 Perceptual Linear Prediction Coefficients . . . 204
A.4 Bottleneck features . . . 205

Appendix B HMM Acoustic Modelling 207

Appendix C Automatic Speech Recognition 211

Appendix D Phonetic alphabets 215

Appendix E Baseline Rhythm Features 217

Appendix F Intonation Annotation 219

Appendix G Equivalence of DCT-II to least-squares approximation 225

Appendix H ALTA Baseline Features **227**

Appendix I Phone Distance Features and L1 **231**

Appendix J Experiments on grader architecture **233**
J.1 Speech features 233
J.2 Phonetic alphabet 234
J.3 Sequence modelling 235
J.4 Attention 237
J.5 Batch normalisation 237
J.6 Learning rate schedule 238

Appendix K Accent Error Types **239**

Appendix L Error annotation interfaces **245**

List of figures

1.1 Structure of this thesis 5

2.1 Categorisation of views of spoken language proficiency used in this thesis 8
2.2 French accented (left) and native (right) speakers saying the word *red*, each narrowly and broadly transcribed. In broad notation, both are [red] (as the broad alphabet does not contain the non-English phone [ɣ]), failing to capture the difference in pronunciation. 10
2.3 Illustration of word sequence $w_{1:2}$, phone sequence $\boldsymbol{\phi}_{1:6}$ and state sequence $s_{1:18}$ (encoding the word and phone at each frame) for 18-frame recording $\boldsymbol{o}_{1:18}$ of phrase *the cat*. 11
2.4 Illustration of a lattice of possible alignments of a realisation of the phrase *the cat*, allowing two possible pronunciations of the word *the* and two possible durations of the final phone [ax] in *the*. The path in red corresponds to the word, phone and state sequences illustrated in Fig. 2.3. 13
2.5 Illustration of phone distance feature concept. Representations of phones are obtained in acoustic space. Each phone (violet point) is characterised by its pair-wise distance to every other phone (blue points). . . 16
2.6 Illustration of extraction of r-PVI features from sample phrase 'on the mat' 24
2.7 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for question *Did Maria throw the ball?* (Fig. F.5). Equal emphasis on *Maria* and *ball* using lower pitch (L* pitch stress). Pitch rises at the end (H-H%) to indicate a yes/no question. 29
2.8 Illustration of pitch contour and features used to describe it. Reproduced from [14] 30

3.1 Lattice used in forced alignment with a canonical dictionary (left) and *extended* lattice allowing both canonical and select errorful pronunciations for use in ERNs. 44
3.2 Illustration of task of free phone recognition 48

4.1 Illustration of a simple neural network mapping a length-3 input vector to a scalar output with one hidden layer of size 2. Training involves tuning the values of the weights and biases in order to capture the relationship between x and y. 59
4.2 Illustration of a a two-hidden-layer DNN mapping a vector to a vector (left), a uni-directional two-hidden-layer RNN mapping a length-3 input sequence to a length-3 output sequence (middle), and a uni-directional encoder-decoder RNN with one-hidden layer each encoder and decoder mapping a length-3 input sequence to a length-3 output sequence (right) 60
4.3 Illustration of a two hidden layer bidirectional RNN for a length-3 input sequence 61
4.4 Illustration of a two-hidden layer LSTM for a length-2 sequence. Bold lines indicate application of weights, then bias and a non-linearity where the arrows meet. 63
4.5 Illustration of a simple attention mechanism. Each bold line indicates a fully connected layer. 64
4.6 Two forms of sequence to vector transformation using a bi-directional RNN: projecting the hidden state vectors at the final positions of the last hidden layer in each direction (left) and attending over all vectors on the final hidden layer (right). Dotted lines indicate attention, the red circles represent attention weights normalised against each other. . 65
4.7 Sequence-to-sequence mapping using a bi-RNN (top left), single-head self-attention (top right) and multi-head self-attention (bottom). Dotted lines of the same colour represent an attention mechanism. Coloured circles on the dotted lines represent attention weights. Solid lines leading to circles indicate the key used together with the value at the source of the dotted line to derive attention weights. 67
4.8 Illustration of Siamese bidirectional RNNs to classify whether a pair of sequences is a match. 69

5.1 Illustration of the phone distance concept 86
5.2 Illustration of proficiency grading using phone distance features. 88

5.3 Illustration of Siamese bi-directional RNN 89
5.4 Illustration of Deep Phone Distance Feature architecture (right) compared to Phone Distance Features (left - reproduced from Fig. 5.2) . . 92
5.5 Illustration of words (black), vocalic intervals (red), intervocalic intervals (blue) and sub-segments (square brackets) in the phrase 'on the mat' 93
5.6 Illustration of extraction of deep rhythm features from sample phrase 'on the mat' (right) compared to the original PVI (left) 95
5.7 Illustration of words, phones, emphases, ToBI annotations and per-frame f_0 and p_v (black and red dots respectively) for a simple realisation of the statement *Maria threw the ball* (Fig. F.1). Equal emphasis is placed on words *Maria* and *ball* by pronouncing the stressed vowels of each with a higher pitch (H* pitch stress). Pitch drops at the end of *ball* (L-L%) to indicate the end of the statement. f_0 is only extracted for voiced regions (i.e. $p_v \geq 0.5$). 96
5.8 Illustration of intonation assessment for an utterance consisting of the word *threw* using overall f_0 statistics through DNN (left) and phone-wise f_0 statistics through bi-directional RNN (right). Score is predicted starting from per-frame f_0 (black dots) and per-frame probability of voicing (red dots). Note $\boldsymbol{Q} = [Q_{.25}, Q_{.5}, Q_{.75}]$. 97
5.9 Illustration of intonation assessment for an utterance consisting of the word *threw* using cosine fitting (left) and multi-head sequence-to-vector attention (right). Intonation grade is predicted from per-frame f_0 (black dots) and per-frame probability of voicing (red dots). 99
5.10 Illustration of text, rhythm, pronunciation, and intonation grading. . . 101

6.1 Confusion matrices and overall % matches for DNN classifiers using x-vectors extracted based on speaker classification (top left), grade prediction (top right) and L1 classification (bottom) criteria, after recognition with the TD-gr ASR and alignment with the GH-ph acoustic model, trained with five different random seeds on BL_GRD_M1 and evaluated on BL_EVL_M. 110
6.2 Confusion matrix and overall % matches for DNN classifier using ALTA baseline features, after recognition with the TD-gr ASR and alignment with the GH-ph acoustic model, trained with five different random seeds on BL_GRD_M1 and evaluated based on average of predictions on BL_EVL_M. 111

6.3 Average (ensemble) predicted scores from deep phone distance grader (dp) and DNN phone distance feature (log K-L divergence) grader (lpron), trained with five different random seeds on BL_GRD_M1 (left) and LS_GRD_M (right) and evaluated on corresponding evaluation sets, plotted against expert human scores. 114
6.4 Confusion matrices and overall % matches for DNN L1 classifier using x-vectors extracted based on speaker classification (top left), DNN L1 classifier using phone distance (log KL divergence) features (top right) and deep phone distance L1 classifier (bottom), after recognition with the TD-gr ASR and alignment with the GH-ph acoustic model, trained with five different random seeds on BL_GRD_M1 and evaluated on BL_EVL_M. 115
6.5 Expert human scores plotted against ensemble predictions from deep rhythm grader, trained on BL_GRD_M1 (left) and LS_GRD_M (right) and evaluated on corresponding test sets. 117
6.6 Expert human scores vs predictions by AttLSTM deep intonation grader, trained on BL_GRD_M1 and evaluated on BL_EVL_M (top) and LS_EVL (bottom). 119
6.7 Histogram of correlations between score and phone distance for each phone-pair extracted from MFCC13 features after recognition with TD-gr and alignment with GH-ph 120
6.8 Plot of expert human assigned score for speakers in BL_EVL_M against select log-one-plus K-L divergences extracted from MFCC13 features after recognition with TD-gr and alignment with GH-ph acoustic model 121
6.9 Plot of expert human assigned score for speakers in BL_EVL_M against the number of phone distances that have no instances in each speaker's speech 122
6.10 Histograms of attention weight values over instances for 3 phones for select speakers in BL_EVL_M, from deep phone distance grader (with AttLSTM instance embedding) trained on BL_GRD_M1, with batch norm and exponential learning rate 124
6.11 Overall distribution overall of the entropy ratio of attention weights over instances of each phone of each speaker in BL_EVL_M, when evaluating a deep phone distance grader (with attention LSTM phone instance embedding) trained on BL_GRD_M1, with batch normalisation and exponential learning rate schedule 125

6.12 Distribution by phone of the entropy ratio of attention weights over instances of each phone of each speaker in BL_EVL_M, when evaluating an attLSTM deep phone distance grader trained on BL_GRD_M1, with batch normalisation and exponential learning rate schedule 126
6.13 Correlation between phone distance and deep phone distance across all speakers in BL_EVL_M for selection of phone pairs. 127
6.14 Correlation between phone-pair phone distance (PD) and deep phone distance (DP) across speakers in BL_EVL_M plotted against absolute correlation of phone distance (PD) with score. 128
6.15 True holistic score plotted against error between holistic and each single-view predicted score for all speakers in BL_EVL_M 130
6.16 Error between holistic scores and single-view predictions with each of the end-to-end rhythm, text, pronunciation (pron), and intonation graders adjusted for mis-calibration, plotted across score ranges for speakers in BL_EVL_M 131
6.17 Performance of the TD-gr ASR on speakers in BL_EVL_M of each CEFR level (shown on the score axis) evaluated by word error rate (WER). 132
6.18 Absolute error between holistic scores and single-view predictions with each of the end-to-end rhythm, text, pronunciation (pron), and intonation graders adjusted for mis-calibration, plotted across score ranges for speakers in BL_EVL_M 132
6.19 Mean adjusted error (left) and mean absolute error (right) between expert score and that predicted by combined grader (attention method) for different score ranges for all speakers in BL_EVL_M 134

7.1 Illustration of accent and lexical errors. A speaker that pronounces *the* as [d ax] is likely to also pronounce *this* as [d ih s]. A speaker that pronounces *fruit* as [f r uw ih t] is unlikely to also pronounce *boot* as [b uw ih t]. 139
7.2 Illustration of process for generating candidate accent errors from recognised word, canonical pronunciation dictionary and phonetic rules. . . . 141
7.3 Illustration of process for generating candidate lexical errors 143

7.4 Illustration of an example lattice for the problem of error detection in spontaneous speech. The speaker is saying *and a cat sat*, *and a hat sat*, *and the cat sat* or *and the hat sat*, the word *the* is pronounced as either the canonical pronunciation [dh ax] or the accent error [d ax] (corresponding to error type $\mathscr{E} = [dh] \rightarrow [d]$), and, if the third word is *hat*, there are two different time stamps that could be the boundary between the second and third words. 145
7.5 Illustration of all possible paths through the lattice in Figure 7.4. The red and orange paths satisfy $w_1 = the$, of which the red paths satisfy $\boldsymbol{\phi}_{1:M}^{(w_1)} \notin \mathscr{D}_{w_1}^{(can)}$. The posterior probability of an error per Eq. 7.18 is given by the sum of the likelihoods of the red paths normalised by the sum of all paths. An estimate of confidence in the word *the* per Eq. 7.20 is given by the sum of the red and orange paths normalised by the sum of all paths. 146
7.6 Lattice for force alignment after ASR on Fig. 7.4 has yielded the 1-best word sequence *and the hat sat*. There are now only four paths. The posterior probability of an error is obtained by summing the likelihoods of the two paths through [d], normalised by the sum of the likelihoods of all paths, multiplied by an estimate of ASR word confidence (Eq. 7.25) 147
7.7 Cumulative frequency of the ranking of identified accent errors (blue) and remaining errors (red) in BLT (top left), SELL (top right) and LPINT (bottom) among the 50-best outputs of a G2P system trained on the corresponding canonical dictionary 156
7.8 Cumulative frequency of the ranking among 50-best G2P outputs of a sample of types of identified accent errors in BLT (top), SELL (middle) and LPINT (bottom) 157
7.9 Overlap of accent and lexical error candidate pronunciations in the dictionaries generated for the words in BULATS (top left), SELL (top right) and LeaP (bottom) 158
7.10 Proportions of words annotated as errors and identified as accent and lexical. 158
7.11 Median ranking of accent and other errors among the 50-best G2P predictions for different context window sizes L on BLT (top left), SELL (top right) and LPINT (bottom) 159

7.12 Precision-recall curves for accent error detection on BLT (top), SELL (middle) and LPINT (bottom) using posterior thresholding, repeated using ASR output, manual transcription (MAN) and, the case of BLT, crowd-sourced transcriptions (CS) 162
7.13 Precision-recall curves for accent error detection on BLT (top), SELL (middle) and LPINT (bottom) using 1-best log ratio thresholding repeated with manual transcription (MAN), ASR output - for BLT and SELL -, and crowd-sourced transcriptions (CS) - for BLT 163
7.14 F1 scores on each dataset for detecting accent errors ('All'), specific types of accent errors and lexical errors (top) and the presence of one or more of the above in a particular utterance, using the lower bound (middle) and upper bound (bottom) methods. 164
7.15 Detected expected number of errors (sum of word-level posteriors) against annotated errors for each speaker in BLT (top left), SELL (top right) and LPINT (bottom). 165
7.16 Relationship between proficiency score and the numbers of annotated (left) and detected, at the F1-maximising threshold, (right) errors in BLT 166
7.17 Relationship between proficiency grade and the numbers of annotated (left) and detected, at the F1-maximising threshold, (right) errors in LPINT 166
7.18 Relationship between number of annotated (left) and detected (right) lexical errors in BLT (top) and LPINT (bottom) 167

A.1 Illustration of the source filter model. Adapted from [76] 202

B.1 Illustration of a Hidden Markov Model (HMM) 207

F.1 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for a simple rendering of the statement *Maria threw the ball*. Equal emphasis is placed on the words *Maria* and *ball* by pronouncing the stressed vowels of each with a higher pitch (H* pitch stress). Pitch drops at the end of *ball* (L-L%) to indicate the end of the statement. . 220
F.2 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for question *Did Maria throw the ball?*. Equal emphasis is placed on *Maria* and *ball* using lower pitch (L* pitch stress). These play the same role as H* in Fig. F.3 but are now low to contrast with the high pitch at the end. Pitch rises at the end of *ball* (H-H%) to indicate a yes/no question. 220

F.3 Illustration of words, phones, emphases, ToBI annotations and f_0 for a rendering of the statement *Maria threw the ball* with an emphasis on *Maria* (i.e. Maria is the one that threw the ball as opposed to someone else). Particular emphasis is placed on the word Maria by lowering the pitch before increasing it to form the pitch stress H*, forming a 'scoop' pitch stress L+H*. Pitch drops (L-L%) to indicate the end of the statement. 221

F.4 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for a rendering of the question *Did Maria throw the ball?* with an emphasis on the word *Maria* (i.e. asking whether Maria was the one that threw the ball as opposed to someone else). Particular emphasis is placed on the word Maria by sharply raising pitch immediately after having reduced it to form the pitch stress L* on the stress syllable [iy], thus the combined scoop stress L*+H. Pitch rises at the end of *ball* (H-H%) to indicate a yes/no question. 221

F.5 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for a rendering of the question *Did Maria throw the ball?*, such that *Did Maria* forms a separate intermediate intonational phrase to *throw the ball*? A small pitch rise after *Maria* separates the two intermediate phrases and signals that the first intermediate phrase is part of a question, such that the whole utterance sounds like *Did Maria*$^{(?)}$*throw the ball?*. Equal emphasis is still placed on the words *Maria* and *ball* by using lower pitch (L* pitch stress). The meaning is the same as in Fig. F.2 but the rendering is more spaced out. Pitch rises again at the end of *ball* (H-H%) to signal the end of the yes/no question. 222

F.6 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for statement *Maria threw the ball, then Paul caught it* with emphasis on *Maria*, *ball*, and *Paul*. The clauses before and after the comma are separate intonational phrases, both in the form of statements. The end of the first is signalled by a drop L to indicate the end of a statement followed by a continuation rise H to signal that another related intonation phrase is coming, together making the boundary tone L-H%. The final phrase ends with a standard pitch drop L-L%. Emphases on *Maria* and *ball* are marked by low pitch stresses, contrasting the continuation rise, while the emphasis on *Paul* is marked by a high pitch stress as normal. 222

F.7 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for a rendering of the statement *Maria threw the ball to Paul*, with emphasis on *Maria*, *ball* and *Paul*, marked by high pitch stresses (H*), an intermediate phrase break (L-) before *to Paul*, marked by a small pitch drop after *ball*, and overall falling pitch (downstep) across the entire intonational phrase, signalled by each pitch stress !H* being lower than its predecessor. A final pitch drop at the end of *Paul* (L-L%) marks the end of the statement. 223

F.8 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for the statement *Maria brought the ball, Paul, and more* as three intonational phrases. The ends of the first two intonational phrases are marked by level boundary tones L-H% after *ball* and *Paul*, signalling the progression to the next element of the list, while the end of the statement is marked with a standard pitch drop L-L%. Emphasis on *Maria*, *ball*, *Paul*, and *more* is marked by high pitch stresses (H*). Overall falling pitch (downstep) across the entire utterance is signalled by each pitch stress !H* being lower than its predecessor. 223

J.1 Illustrations of standard (top, left) and attention (top, right) bi-directional RNNs, and single-head (bottom, left) and multi-head (bottom, right) attention, reproduced from Figures 4.6 and 4.7 235

J.2 Expert human scores against predictions of deep phone distance grader with regular (left) and attention (right) LSTM trained on BL_GRD_M1 and evaluated on BL_EVL_M1. 236

L.1 AMT Hit collecting data on pronunciation errors 245

L.2 AMT Hit collecting data on stress errors 246

List of tables

2.1 Proficiency graders compared by input/structure views (T=Tempo, P=Pronunciation, R=Rhythm, I=Intonation, X=Text, H=Holistic), tunability of feature extraction (Tun. FE), independence to reference speakers (Ref. Ind.), independence to additional annotation (Ann. Ind.), application to spontaneous speech (Spont.), and ground-truth grade views. 33

3.1 Non-native speech corpora used in reviewed papers on pronunciation error detection 50

3.2 Error detection systems from the literature compared to novel system from Chapter 7 evaluated based on not needing training data (Train-Free) or native reference (Ref. Free), diagnosing the type of each error (Diag.), and use on spontaneous speech (Spont.) or speakers of multiple L1s without separate training data (L1 Ind.) 53

6.1 Mapping from BLT/LS proficiency scores to CEFR levels (adapted from [38]) 106

6.2 BLT and Linguaskills (LS) datasets used for training and evaluating automatic proficiency graders in this thesis. Datasets annotated by the original operational graders are used for training while those annotated by experts are used for evaluation. L1 Key: Ar. = Arabic, Fr. = French, Du. = Dutch, Hi. = Hindi, Viet. = Vietnamese, Pr. = Portuguese, Jp. = Japanese., Span. = Spanish, Pol. = Polish 107

6.3 BLT sets used for training and evaluating L1 classifiers. L1s are Tamil, Telugu, Malayalam, Kannada, Gujarati, Hindi, Bengali, Marathi, Spanish, French, Portuguese, Italian 107

6.4 Performance of the DNN grader with ALTA baseline features (base) trained using exponential learning rate with five different random seeds on BL_GRD_M1 and evaluated on BL_EVL_M. Features are derived from MFCC-13 vectors from the output of TD-gr aligned with GH-ph. Accuracy of mean predictions is evaluated using Pearson correlation coefficient (PCC), mean squared error (MSE), mean absolute error (MAE) and by the percentages of predictions with an error below 0.5 (%<0.5) and 1.0 (%<1.0). Sensitivity to random initialisation is measured by the standard deviation of each metric across the five runs (±). 108

6.5 Performance of DNN graders using x-vectors extracted based on speaker classification, L1 classification, and grade prediction criteria, compared against the baseline ALTA features (base), after recognition with the TD-gr ASR and alignment with the GH-ph acoustic model, trained on BL_GRD_M1 and evaluated on BL_EVL_M. 109

6.6 Performance of deep phone distance grader (DP) and each of DNN and Gaussian Process (GP) phone distance graders, trained on BULATS and Linguaskills datasets and evaluated on matched and non-matched evaluation sets, after recognition with TD-gr and alignment with GH-ph.113

6.7 Performance of deep rhythm grader (DR) and DNN rhythm grader (DNN), trained on BL_GRD_M1 with five random seeds and evaluated on BL_EVL_M and LS_EVL (matched and unmatched respectively), after recognition with TD-gr and alignment with GH-ph. 116

6.8 Performance of attention LSTM (AttLSTM) and 8-head attention (8HA) deep graders, f_0 statistics RNN and DNN graders and cosine fitting DNN grader (CF), trained on BL_GRD_M1 and evaluated on BL_EVL_M and LS_EVL (TD-gr ASR and GH-ph to align) 118

6.9 Performance, measured by mean and standard deviation of % accuracy and correlation of distance metric to phone distance of Siamese LSTM phone instance pair classifier trained using binary classification (siam_bin) and phone distance prediction (siam_kl) criteria, on 100,000 pairs of matched phone instances and 100,000 pairs of unmatched phone instances randomly sampled from BL_GRD_M1 and evaluated on 10,000 matched and 10,000 unmatched pairs sampled from BL_EVL_M (input is MFCC-13 from TD-gr aligned with GH-ph). 123

6.10 Performance of single-view graders and their pair-wise averages on BL_EVL_M. Single-view grader performances are performances of ensemble averages of predictions and are reported together with the standard deviations of their underlying ensembles (note the means of the underlying ensembles are smaller than the ensemble performances and are not reported). Pair-wise average performance average figures are marked in italics if they are worse than the ensemble average performance of either of their constituent graders and in bold if they are more than one ensemble standard deviation better than the ensemble average performances of both of their constituent graders. 129
6.11 Kendall's τ between single-view grader predictions 130
6.12 L1 (top) and country of origin of Spanish speakers (bottom) detection rate on BLT_EVL_M2, broken down by CEFR level, of 3-way phone distance DNN country of origin classifier using phone distance features, trained on BLT_GRD_M2 133
6.13 Performance of end-to-end pronunciation, rhythm, intonation, and text graders combined using each of score averaging (mean), concatenating intermediate representations (concat) and attention mechanism over scores (att), compared to performance of DNNs trained on ALTA baseline features (hand) and x-vectors trained with grading criterion (xvec), trained on BL_GRD_M1 and evaluated on BL_EVL_M 133

7.1 Cohen's κ for intra-annotator agreement between successive phonetic annotations of an utterance from LeaP by each of five annotators (leading column) and inter-annotator agreement between each pair thereof, from Tables 2 and 3 in [103] 151
7.2 Statistics of human annotations of pronunciation errors on BLT corpus 152
7.3 Annotation statistics for the BLT and LeaP corpora. † This value is calculated on the data in LPINT as the held out utterance used in [103] was not available 152
7.4 Number of detectable annotated errors in each dataset, using original manual (MAN), ASR and crowd-sourced (CS) [271] transcriptions. . . . 153
7.5 Mean pairwise Cohen's κ and proportion of words annotated as errors on different annotation tasks † Error % is calculated on LPINT but κ on a held-out utterance 155

C.1 Description of acoustic models used in this thesis, the data they are trained on and their word error rates (WER), when used together with a K-N trigram language model, evaluated on BLT_EVL_M (from §6.1). The COMBILEX pronunciation dictionary is used throughout. 213

D.1 Phones used in this project in Arpabet and IPA [219, 281, 283]. In CMU, phones with * are merged with phone above and those with † are merged with that to their left. 216

H.1 ALTA baseline grader features extracted from audio and ASR output . 228
H.2 Definition of terms in Table H.1 229

I.1 Percentage of speakers of Romance (top), Indo-Aryan (middle) and Dravidian (bottom) L1s in BLT_EVL_M2 classified as other languages in same group by phone distance DNN trained on BLT_GRD_M2 from MFCC-13 after decode and alignment with GH-ph 231
I.2 Detection rate, by country of origin, on Spanish speakers in BLT_EVL_M2, of phone distance DNN 3-way country classifier, trained on Spanish speakers in BLT_GRD_M2 232

J.1 Performance of DNN graders with phone distances from MFCC-13, PLP-13 and PLP-39 observations after ASR by TD-gr and alignment with the GH-ph acoustic model. Each is trained on BL_GRD_M1 and evaluated on BL_EVL_M. 233
J.2 Performance, measured by Pearson correlation (PCC), of Gaussian Process graders using ALTA baseline features (Base) and phone/grapheme distance features (Pron) obtained from PLP-39 observation vectors, after recognition (ASR) and alignment (Align/Grd) with phonetic (DH-ph and GH-ph) and graphemic (DH-gr and GH-gr) acoustic models. Feature extraction for each of Base and Pron follows the same phonetic alphabet used for alignment. 234
J.3 Performance of end-to-end graders trained on BL_GRD_M1 and evaluated on BL_EVL_M with MFCC-13 from TD-gr aligned with GH-ph. . 236
J.4 Performance of deep pronunciation and rhythm graders with additive (add) and scaled dot-product (sdp) attention, in att-LSTM configuration, trained with CLR on BL_GRD_M1 and evaluated on BL_EVL_M, starting from MFCC-13 observation vectors. 237

J.5 Performance of deep phone distance grader with and without batch normalisation trained using clr with five different random seeds on BL_GRD_M1 and evaluated on BL_EVL_M, with MFCC-13 from TD-gr aligned with GH-ph. 237
J.6 Performance of deep pronunciation and rhythm graders trained using constant, decaying, and exponential learning rate schedules with five seeds on BL_GRD_M1 and evaluated on BL_EVL_M, starting from MFCC-13 observation vectors derived from TD-gr ASR aligned with the GH-ph acoustic model. 238

K.1 Deletion accent error types 239
K.2 Insertion accent error types 240
K.3 Vowel substitution accent error types (Vietn. = Vietnamese, All = French, Chinese, Vietnamese, Thai, Spanish, Russian, Dutch, Arabic, Polish and German) 241
K.4 Consonant pairwise confusion substitution error types (Vietn. = Vietnamese, All = French, Chinese, Vietnamese, Thai, Spanish, Russian, Dutch, Arabic, Polish and German) 242
K.5 Other consonant substitution error types (Vietn. = Vietnamese, All = French, Chinese, Vietnamese, Thai, Spanish, Russian, Dutch, Arabic, Polish and German) 243

Nomenclature

Roman Symbols

$\boldsymbol{v}^{(n)}$	Features extracted to represent the holistic proficiency of a speaker n
$\boldsymbol{v}_j^{(n)}$	Features extracted to represent view j of the proficiency of a speaker n
$\mathscr{D}_w^{(can)}$	Dictionary entry containing canonical pronunciations (i.e. phone sequences) for a given word w_i
$\mathscr{L}()$	Likelihood — usually equivalent to $\log p()$
$\mathscr{S}_{train}$	Training data set
$\mathscr{V}$	The set of all vowels e.g. $\phi_m \in \mathscr{V}$ represents all phone instances in the sequence $\phi_{1:M}$ that are vowels.
$\mathbf{o}_{1:T}$	Sequence of spectral feature vectors $\mathbf{o}_t$ (usually MFCCs or PLPs) for each frame t from 1 to T
f_0	Fundamental frequency
$f_0^{(t)}$	Fundamental frequency at a frame t
h_t	Hidden state of an RNN at timestep t
I	Usually represents the number of words within an utterance
K_C	The number of inter-vocalic intervals within an utterance
K_V	The number of vocalic intervals within an utterance
M	Usually represents the number of a phone instances m within an utterance or word
p_v	Probability of voicing

$p_v^{(t)}$ Probability of voicing at a frame t

T Usually represents the number of frames t within an utterance

w_i ith word in a sequence of I words in an utterance

$w_{1:I}$ Sequence of I words in an utterance

d() Duration of a certain interval, segment or phone

Greek Symbols

$\boldsymbol{\lambda}$ The parameters of a neural network

$\boldsymbol{\phi}_{1:M}^{(w_i)}$ Sequence of M phones corresponding to word w_i

ϕ_m mth phone in a sequence of M phones in an utterance

$\phi_{1:M}$ Sequence of M phones in an utterance

$\tau_k^{(C)}$ The kth intervocalic interval (i.e. segment of speech consisting exclusively of consonants and silences) within an utterance or set of utterances

$\tau_k^{(V)}$ The kth vocalic interval (i.e. segment of speech consisting exclusively of vowels) within an utterance or set of utterances

Superscripts

(n) Pertaining to the nth speaker in a dataset

T Transpose

Terminology

accent the particular manner in which a non-native speaker tends to pronounce the phones of a language [1]

accent error Instance of a systematic pattern of phone insertion, deletion or substitution errors made by a speaker across words e.g. a speaker that tends to realise [dh] as [d] pronouncing *the* as [d ax].

acoustic model Model that predicts the likely sequence of phones given the acoustic properties of the audio

[1] Not to be confused with the second meaning of the word *accent* relating to prosodic stress, which will not be used in this work to avoid confusion.

ASR confidence methods Pronunciation error detection methods whereby the confidence of an ASR in a particular word or phone is taken to be indicative of the intelligibility of the way the speaker realised it.

auto-marking Application that automatically marks candidates in an examination (in the context of this thesis an oral language examination)

automatic speech recognition recognising the sequence of words spoken from audio

classification Determining which of one or more pre-defined categories a new observation belongs to e.g. determining whether a certain syllable in a word is stressed

dimensionality reduction Reducing the number of variables needed to represent an observation with minimum loss of salient information e.g. deriving features to represent rhythm given a sequence of segment durations

end-to-end A paradigm in which transformations that would otherwise be performed by separate systems (e.g. feature extraction and grading) are performed by a single multi-stage model, all stages of which are trained simultaneously

expert feature See handcrafted feature

extended recognition networks Method of pronunciation error detection by aligning utterances with candidate canonical and errorful phonetic pronunciations (usually in broad transcription)

forced alignment recognising the time-aligned sequence of phones spoken from audio and the word sequence

handcrafted feature Feature extracted by a non-trainable, deterministic algorithm, inspired by theoretical insights to measure a particular property of an example of audio or text data (e.g. rate of speech, PVI)

intelligibility the ease with which an utterance is comprehensible to a human listener

intonation The variation of pitch during speaking

language model Model that predicts the prior likelihood of a sequence of words occurring in a sentence

lexical error Instance of a speaker mispronouncing a word because they do not know its correct phonetic pronunciation e.g. pronouncing subtle as [s ah b t el] because they do

not know the b is silent. Also referred to in the literature as letter-to-sound conversion errors.

lexical stress The relative stress of one of the syllables of a word, such that stressing a different syllable changes or removes the word's meaning

multi-view proficiency assessment Assessing the proficiency of a speaker by assigning them both a holistic grade on their overall proficiency and a series of single-view grades on individual views of their proficiency such as pronunciation, rhythm etc.

native speaker comparison methods Pronunciation error detection by comparing acoustic properties of candidate's phones to those of native speakers in similar context

overall pronunciation assessment grading how proficiently a speaker generally pronounces words in their recorded speech

phone The smallest unit of analysable sound in a word regardless of whether replacing it changes or removes the word's meaning.

phone recognition methods Pronunciation error detection by freely recognising narrowly transcribed phones in speech and comparing recognised to canonical phone sequence

phoneme The smallest unit of sound in a language replacing which can change the meaning of a word

phonemic pronunciation Representation of a word as a sequence of phonemes

phonetic alphabet Set of symbols each representing a phone

phonetic pronunciation Representation of a word as a sequence of phones

proficiency Unless otherwise specified refers to spoken language proficiency

proficiency assessment The task of assigning a grade to quantify the level of proficiency of a non-native speaker based on recorded audio of their speech

pronunciation The realisation of a word as a sequence of distinct sounds

pronunciation dictionary Expert-compiled list of the canonical pronunciations of each word in a vocabulary, transcribed using a phonetic alphabet

pronunciation error detection identifying words that have been pronounced non-canonically

pronunciation utterance error detection determining whether a particular utterance contains word-level pronunciation errors, either in general or of a particular type e.g. inserting vowels between adjacent consonants, pronouncing [th] as [t]

prosody Variation of pitch, loudness, and duration over a speaker's utterances

regression Predicting one or more quantities based on the values of one or more others e.g. predicting human-assigned proficiency score based on extracted features

rhythm Statistical properties of phone, syllable and word-level durations salient to the proficient sound of a language

sentence stress The relative stress of one or more of the words in an utterance to communicate grammar, punctuation, emphasis or other semantic information

spoken language proficiency A speaker's level of ability at effectively speaking a language (in this thesis English) as would be evaluated by an expert human grader listening to their recorded speech following a standard set of guidelines

spontaneous speech Utterances in response to an prompt that are not being read out nor were otherwise previously prepared

stress The relative emphasis of particular syllables and words through increase in loudness and duration and variations in pitch

supervised learning Learning the relationship between input and output variables based on a dataset of input-output pairs such that the output can be predicted from just the input in the future

tempo Statistical properties of the speed with which phones and words are uttered

unsupervised learning Automatically discovering structure in data without the help of pre-existing labels

utterance A continuous piece of speech with a beginning and an end, considered as an entity for the purposes of processing - usually consists of the response to a single prompt or otherwise communicates a self-contained idea

variety A dialect, register or other system of expression within a language (in this thesis generally English) governed by situational variables, including its associated grammar, accent and other properties e.g. General American English, Northern English, formal English

Acronyms / Abbreviations

ALTA Automated Language Teaching and Assessment - a virtual institute spanning the Cambridge University Engineering, Computer Science and Linguistics Departments

CALL Computer Aided Language Learning

CAPT Computer Assisted Pronunciation Training

CRF Conditional Random Field

DNN Deep Neural Network

DTW Dynamic Time Warping

ERN Extended Recognition Network

G2P grapheme-to-phoneme [converter]

GOP Goodness of Pronunciation

HMM Hidden Markov Model

IPA International Phonetic Alphabet

POS Part of Speech

ReLU Rectified Linear Unit

RMS root-mean-square

RNN Recurrent Neural Network

SAT Speaker Adaptive Training - technique for automatically adapting a trained ASR system to the identity of each speaker

SI Speaker Independent - to describe a speech recognition system, as opposed to speaker dependent systems such as SAT

SVM Support Vector Machine

WER Word Error Rate

Chapter 1

Introduction

Over a billion people are learning English around the world [99] and millions take assessments every year. Rising demand is causing a growing shortage of qualified educators and assessors which, combined with the increasing availability of effective online platforms, is leading to rapid growth in the market for Computer Assisted Language Learning (CALL) [263]. Central to effective CALL systems is the ability to automatically assess the user's language proficiency and provide useful feedback. It is therefore unsurprising that there has been considerable interest in the development of speech processing and machine learning techniques with which to improve tools, as well as in automating the expensive and time-consuming process of spoken language proficiency assessment [100].

The scope of this work is to investigate novel statistical techniques to automatically assess the proficiency of non-native English speakers based on recordings of their speech and provide useful feedback which could be used to help them improve it. As a statistical approach is taken, a speaker is considered proficient for the purposes of this thesis if they would be perceived as such by listeners. A system is thus considered better at evaluating proficiency the closer it matches the feedback that would be given by human annotators, who have themselves demonstrated consistency with themselves and other listeners. Feedback on proficiency based on recorded speech is considered more useful the more representative the input speech is of speech the speaker would be expected to produce in normal communication, such that spontaneous speech is preferred to read speech. The main forms of feedback investigated are holistic proficiency grades, grades with respect to particular views (i.e. aspects such as pronunciation, rhythm etc.) of proficiency, and feedback on the types and locations of pronunciation errors.

Given the highly complex, non-linear, and largely unknown nature of the precise relationship between input audio and a concept such as proficiency, the investigation concentrates mainly on the application of deep learning techniques to enable representational learning

based on training on human-annotated data (see discussion in Chapter 4). Finally, this work primarily focuses on those aspects of proficiency which are particular to speech over writing, namely pronunciation and prosody.

One of the main challenges to statistical automatic assessment of spoken proficiency is the limited quantity and variability of publicly available data. Almost all publicly available data is based on read speech which has been shown to differ considerably from spontaneous speech that is more representative of normal conversation (see discussion in introduction of Chapter 2 and in §3.6). Available data sources with word-label annotations of pronunciation and other types of errors suffer problematically low inter-annotator agreement (see discussion in §3.6 and §7.3).

Corpora of overall graded data do exist but, as they come primarily from the context of examinations and/or language teaching, it is almost never possible to make such data public. For the purposes of this research, it was possible to obtain access to such a non-public corpus of cross-L1 annotated data from Cambridge Assessment. As is predominantly the case with corpora of its kind, speakers have been graded holistically and not with respect to individual views of proficiency (see discussion in §2.1) and there is a relatively low level of inter-annotator agreement between operational graders.

Given the above constraints, methods in the literature for error detection have focused on read speech rather than spontaneous speech, have been limited in their ability to distinguish different types of errors in a way that gives useful feedback to learners, and have been constrained by the underlying inconsistency of the data on which they are trained and/or evaluated, the causes of which were not fully explored (see discussion in Chapter 3).

Approaches to grading speakers overall (see discussion in Chapter 2) have similarly focused on read speech rather than spontaneous speech. Single-view grading has been limited by the availability of single-view human annotations, which suffer issues of inconsistency due to the lack of generally accepted single-view grading standards, and has been focused on methods based on handcrafted features, which are interpretable but assumption laden, extracting information in a way that can't be tuned to different tasks and coarsely discards potentially useful information. Holistic grading has been approached using either handcrafted features corresponding to different single views or end-to-end approaches, which can be flexibly tuned to individual tasks but lack interpretability and have difficulty generalising to data different to that on which they are trained.

To tackle the challenges described, this thesis presents contributions in two main areas. First, in the area of proficiency grading (see Chapters 5 and 6), a novel approach is introduced to grade speakers on single-views based on their spontaneous speech by leveraging and designing end-to-end networks incorporating domain knowledge to constrain the information

extracted to characterise only the desired view. This approach combines the advantages of handcrafted features with those of the black-box end-to-end approach. The systems also are designed so that they can be trained on holistic grades, yet yield single-view predictions, thus addressing the data availability and consistency issue. Systems based on this approach are presented for the views of pronunciation, rhythm and intonation. Novel research is conducted comparing the performance of these systems to other approaches and examining whether the single-view grades they predict when trained on holistic grades have the properties that would be expected of measures of single-view proficiency. A novel approach to holistic grading based on combining these end-to-end single-view systems is also presented and compared to baselines.

In the area of pronunciation error detection (see Chapter 7), novel research is first conducted into the quality of human word-level annotations and the phenomenon of under-annotation in error-annotated versus phonetically transcribed datasets. A novel approach is presented for pronunciation error detection in spontaneous speech based on dividing pronunciation errors into lexical errors and different types of accent errors and using a modification of a common approach from the literature for read speech (Extended Recognition Methods). Common accent errors in English for a large number of L1s are also collected from the literature to create a framework that can predict an exhaustive list of candidate errorful pronunciations of any word in English. The novel system can detect the probability of a particular type of error at a particular word and yield separate feedback at the word-level for lexical errors, but at utterance or speaker level for accent errors. Novel research is conducted into the performance of this system as well as into the relationship between the number of detected pronunciation errors and overall proficiency grade.

In making the contributions discussed above, the work in this thesis specifically answers nine main research questions (see Chapters 5, 6 and 7):

1. Do deep learning approaches offer superior accuracy and generalisability to alternative machine learning approaches (specifically Gaussian Processes) on the task of grading the proficiency of non-native speakers?

2. Can single-view end-to-end neural graders (i.e. end-to-end neural systems constrained by their input and structure to grade on the basis of specific views) offer superior accuracy and generalisability at the task of single-view proficiency grading to methods based on hand-crafted features?

3. Can single-view end-to-end neural graders be interpretable as to their reasons for assigning grades?

4. Can single-view end-to-end neural graders still validly grade on the basis of those views when trained on holistic grades?

5. Do systems based on combining multiple single-view end-to-end neural graders offer superior accuracy at the task of holistic grading to systems based on concatenating single-view handcrafted features and neural systems trained end-to-end on holistic grades?

6. Does the approach of phonetic transcription (asking annotators to exhaustively transcribe the way a speaker pronounced each word) capture the pronunciation errors made by non-native speakers in their spontaneous speech than the approach of pronunciation error annotation (asking annotators to mark which words in recorded speech contain pronunciation errors)?

7. Are accent errors (errors caused by the speaker systematically inserting, deleting or substituting phones across their speech) distinct and capable of being separately detected to lexical errors (errors caused by a speaker not knowing the correct pronunciation of specific words based on their spelling)?

8. Can a system based on calculating word-level probabilities from lattice path likelihoods obtained using force alignment of spontaneous non-native utterances with multiple candidate pronunciations be used to accurately detect individual lexical errors made by the speaker as well as the overall tendency of the speaker to make different types of accent errors?

9. Is the number of pronunciation errors detected by an error detection system predictive of their holistic proficiency grade?

The original work reported in this thesis continues from the work documented in Kyriakopoulos et al. (MEng final report) [151] regarding two-stage pronunciation assessment using a novel form of feature extraction based on phone distances (which is cited as previous work in §2.3.2). Parts of this thesis' original work have been reported in a number of publications authored during the course of the research, most of which are cited for reference in the corresponding chapters of this thesis [147, 152, 277, 153, 154].

The work reported in this thesis falls within the framework of the Engineering Department branch of the Cambridge University ALTA (Automatic Language Teaching and Assessment) Institute, the goal of which is to develop techniques for automatically assessing and providing feedback to non-native English speakers based on both spontaneous and non-spontaneous

spoken utterances. Data and funding were provided by Cambridge Assessment, University of Cambridge.

The structure of this thesis is illustrated in Figure 1.1. Chapters 2 and 3 review the literature on spoken language grading and pronunciation error detection respectively. Chapter 4 then reviews deep learning techniques from the broader literature which are to be applied to the automatic assessment field in this thesis. Using these techniques, Chapter 5 presents a novel framework for single-view and holistic grading, experiments conducted on which are reported in Chapter 6. Finally, Chapter 7 presents the novel work undertaken in the area of pronunciation error detection. The implications and limitations of the results of Chapters 6 and 7 as well as avenues for future work are discussed in Chapter 8, while conclusions, as they relate to the research questions enumerated above, are summarised in Chapter 9.

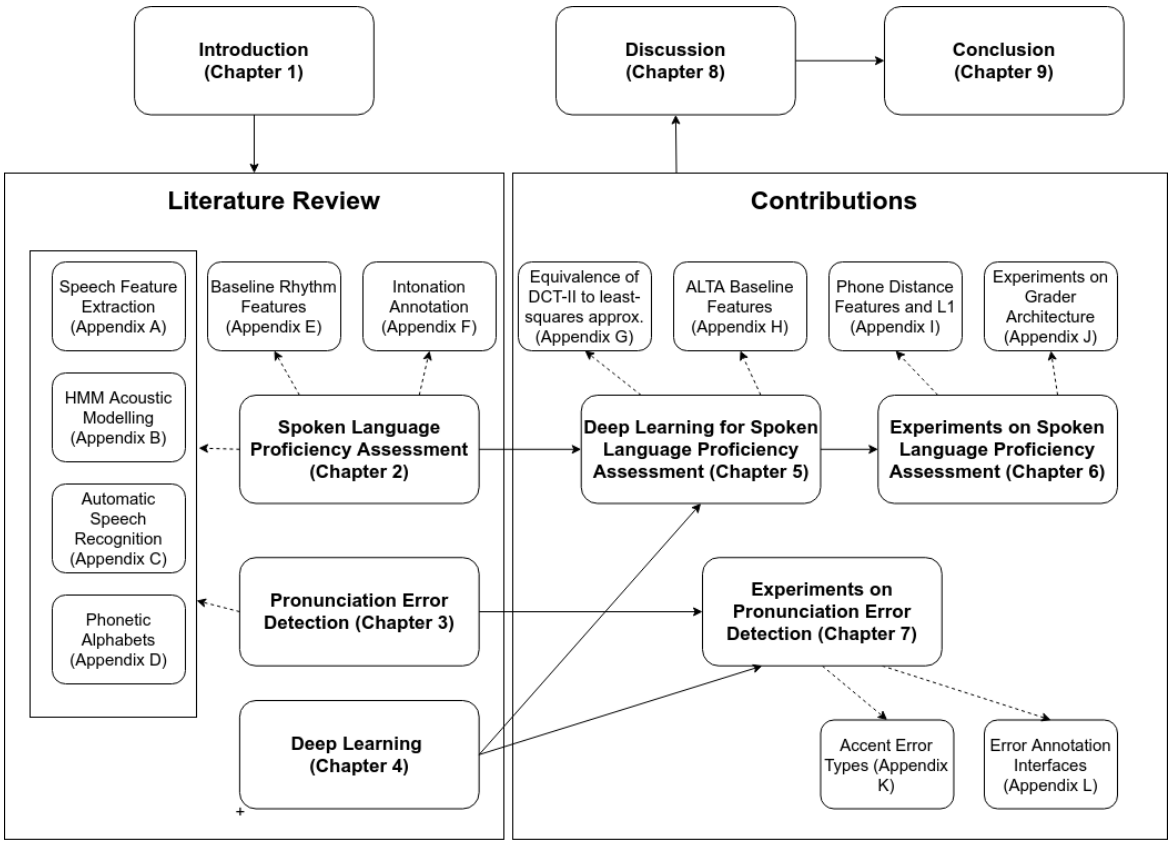


Fig. 1.1 Structure of this thesis

Chapter 2

Spoken Language Proficiency Assessment

This chapter reviews the literature on automatic spoken language proficiency assessment. For the purposes of this thesis, proficiency assessment refers to the task of assigning a grade to quantify the level of proficiency of a non-native speaker based on recorded audio of their speech. The investigation is motivated by Computer Assisted Language Learning (CALL) and auto-marking applications. Techniques should thus be feasible using realistically obtainable data, closely predict the grade that would be given by a human expert, and avoid bias to speaker attributes that are irrelevant to proficiency.

A speaker can be assessed on spontaneous speech (e.g. from interview-style questions) or read speech i.e. recordings of them reading aloud a provided text. Spontaneous speech is more representative of the conversational speech a learner will be called on to use in day-to-day life and in which they will need to be proficient. Read speech has been shown to differ considerably from spontaneous speech acoustically [182, 182], prosodically [189, 192], phonetically [4], in terms of fluency [62], and in the numbers and types of pronunciation errors non-native speakers make [162]. Assessing proficiency using only read speech therefore risks providing misleading or incomplete feedback. For this reason, another important criterion when evaluating assessment techniques is how applicable they are to spontaneous speech.

The concept of language proficiency and its constituent aspects, or views, are discussed in §2.1. The speech audio processing methods which form the basis of most approaches to assessing them, namely automatic speech recognition (ASR) and forced alignment, are then reviewed in §2.2. Approaches to grading speakers on the basis of individual views (specifically text, pronunciation, tempo, stress, rhythm and intonation) are then reviewed in §2.3, while approaches to holistic grading are examined in §2.4. The systems reviewed are compared to the novel systems introduced later in this thesis in §2.5.

2.1 Views of Proficiency

While language proficiency is a contentious concept, there are accepted standards for grading the proficiency of non-native speakers [122], notably those described in the CEFR [207]. These generally do not involve separating proficiency into facets, but rather ask graders to make holistic assessments of communicative competence e.g. *the candidate can interact with a degree of fluency and spontaneity that makes regular interaction with native speakers quite possible* (from CEFR B2). It has nevertheless been demonstrated that grades assigned using such holistic criteria do have a componential structure, being partitionable into individual facets, or views, of proficiency (e.g. pronunciation, intonation, vocabulary and grammar), human-assigned scores for each of which correlate strongly with holistic grade [67].

CALL applications also distinguish between these different views of proficiency during teaching, with different systems used to separately teach facets such as pronunciation [266], prosody [184], and vocabulary [106]. It follows that separately assessing a learner's progress in terms of each of these views (single-view grading) should be useful for feedback to the learner and to inform further teaching adaptively.

A key challenge in such grading is the difficulty in obtaining a reliable ground-truth. Guidelines to annotate speakers on single-view proficiency necessarily deviate from the generally accepted holistic standards. Such scores are thus harder to obtain and have been shown to be subjective and have relatively low inter-annotator agreement [201, 224, 217], compared to the strong agreement usually found with holistic scoring [66, 119].

The names, definitions and categorisation of different views of proficiency for single-view assessment vary across the literature. For the purposes of this thesis, the framework illustrated in Figure 2.1 is used to classify the approaches considered.

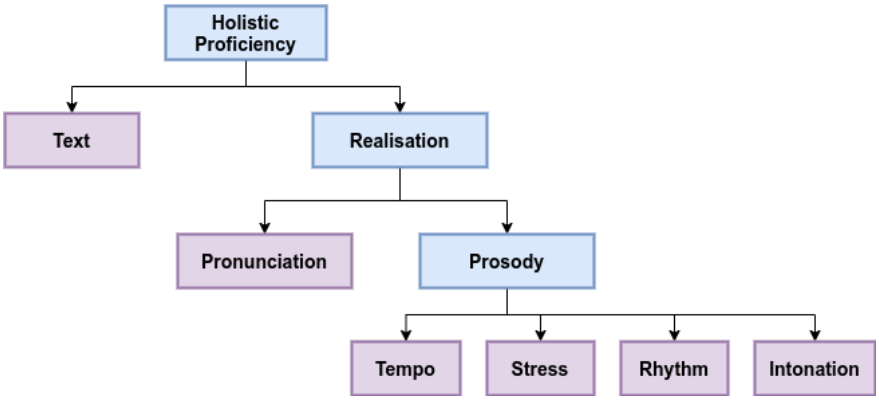


Fig. 2.1 Categorisation of views of spoken language proficiency used in this thesis

Approaches that assess the proficiency of spontaneous speech based on the sequence of words recognised by the speech recogniser are termed text assessment. Approaches that take

the sequence of words as a given (i.e. assess the speaker's realisation of the text) are further divided into pronunciation and prosody assessment. Pronunciation assessment includes those approaches that divide each word into discrete units of sound and use the acoustic properties of those sounds to characterise the speaker's proficiency. Prosody assessment approaches instead use the variation of pitch, loudness, and duration over the speaker's utterances. [61]

Prosody is itself partitioned into four main views. Tempo is the overall speed and consistency with which speech is rendered. More proficient speakers tend to speak faster and more consistently [67]. Stress is the relative emphasis of syllables within words or words within utterances through increased loudness and duration. Every English word has a canonical stressed syllable and there are rules for the stressing of words within sentences in order to convey grammatical, syntactic and semantic information. Deviating from these rules has been shown to considerably decrease the intelligibility and perceived correctness of non-native speech [104]. Rhythm is the pattern of phone, syllable and word durations in a person's speech. Native English speech has a distinctive rhythm, the nature of which is controversial among linguists, but which nonetheless varies between languages and between proficient and non-proficient non-native speakers [98, 117]. Finally, intonation is the variation of pitch during speaking. In English, pitch contours occur at the syllable, word and utterance level and mark emphasis, grammar and other forms of meaning [241, 139] (e.g. indicating a question). Typical and acceptable-sounding pitch contours differ between languages. Matching the intonational patterns of English is thus an important component of proficiency.

After discussing the speech processing techniques that form the basis of most approaches to assessment in §2.2, approaches for grading speakers on the basis of each of the views of text, pronunciation, tempo, stress, rhythm and intonation are reviewed in §2.3, while approaches for holistic grading are reviewed in §2.4.

2.2 Speech Processing

In traditional phonology, sound is analysed in terms of *phonemes* and *phones*. A *phoneme* is commonly defined as the minimal unit of sound within the system of a language, such that changing one phoneme to another in the same context can change the meaning of a word. For example, switching the [p] in [pet] to [b] makes it a different word *bet*, so the sounds [p] and [b] are each instances of two different phonemes /p/ and /b/. No pair of English words exist where the only difference is switching [p] to [p^h], however, so [p] and [p^h] belong to the same phoneme /p/. In Hindi, by contrast, switching the [p] in [pal] (पल — *moment*) to [p^h] turns it into the word [p^hal] (फल — *fruit*), so /p/ and /p^h/ are said to be different phonemes.

A *phone* is defined as a perceptible discrete segment of sound. In the above examples, [p], [p^h] and [b] are all phones, regardless of the language being discussed. [61]

Phonemes are an abstract linguistic concept [190] subject to controversy over their precise definition. The literature is divided as to the number and identity of phonemes in the English language [23]. As the work in this thesis involves statistically processing speech and characterising it in terms of its constituent sounds (e.g. when evaluating pronunciation), the term *phone* is used to describe the sound segments into which words can be divided for recognition and processing, while discussion of phonemes is avoided.

Phones can be defined and transcribed *narrowly*, giving detail on place and manner of articulation (e.g. transcribing *pen* as [p^hɛn] to specify that the speaker aspirated the [p] and used the standard open-mid front unrounded pronunciation of e), or *broadly*, giving only the minimal detail required to identify the word (i.e. [pen]), and therefore more closely corresponding to the word's phonemes [61]. Broad transcriptions define phones in a way that relies more heavily on the rules of the language being spoken and so are less powerful than narrow transcriptions in precisely transcribing accented non-native pronunciations, as illustrated in the example in Fig. 2.2.

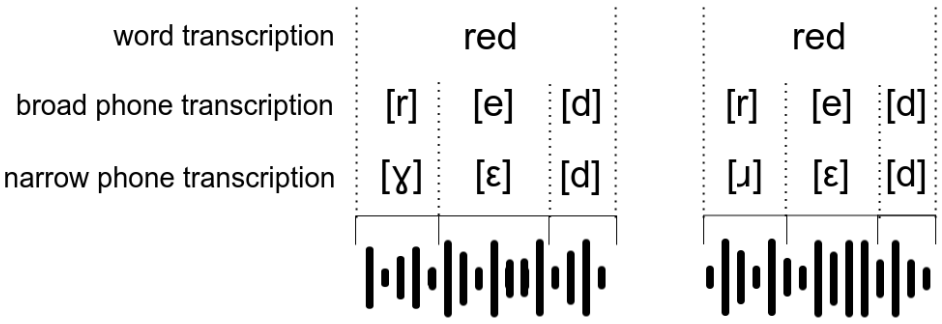


Fig. 2.2 French accented (left) and native (right) speakers saying the word *red*, each narrowly and broadly transcribed. In broad notation, both are [red] (as the broad alphabet does not contain the non-English phone [ɣ]), failing to capture the difference in pronunciation.

The first step in processing speech is extracting salient information from the audio into a compact and informative format. Audio is usually considered in 10-25ms frames, with feature vectors $\boldsymbol{o}_t$ extracted to characterise each frame t. Standard features inspired by the information that the human auditory system uses to convert sound into meaning include MFCCs and PLPs (see Appendix A).

A T-frame recording can thus be represented as a sequence of feature vectors $\boldsymbol{o}_{1:T}$, which in turn can be used to determine the sequence of I words being spoken $w_{1:I}$, the corresponding sequence of M phones $\phi_{1:M}$, and the sequence of states $s_{1:T}$ describing to which phone and

word each frame t corresponds (Fig. 2.3). Aligning to syllables instead of phones is also useful in some applications [25] but is not focused on in this thesis.

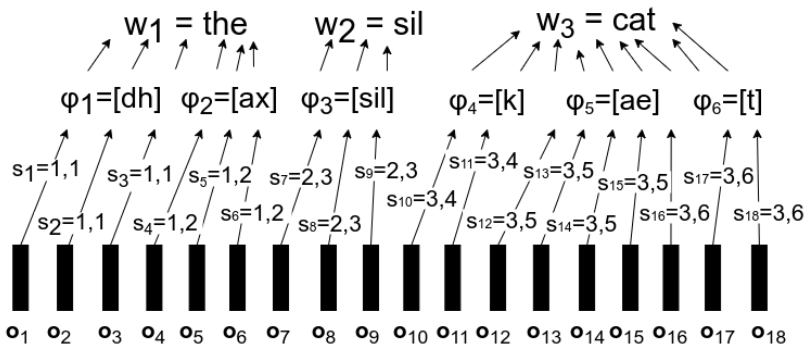


Fig. 2.3 Illustration of word sequence $w_{1:2}$, phone sequence $\phi_{1:6}$ and state sequence $s_{1:18}$ (encoding the word and phone at each frame) for 18-frame recording $\boldsymbol{o}_{1:18}$ of phrase *the cat*.

Automatic Speech Recognition (ASR) finds the most likely word sequence $\hat{w}_{1:I}$ given $\boldsymbol{o}_{1:T}$:

$$\hat{w}_{1:I} = \arg\max_{w_{1:I}} P(w_{1:I}|\boldsymbol{o}_{1:T}) \tag{2.1}$$

$$\hat{w}_{1:I} = \arg\max_{w_{1:I}} \left\{ P(w_{1:I}) \sum_{\phi_{1:M} \in \mathscr{D}_{w_{1:I}}} P(\phi_{1:M}|w_{1:I}) \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \right\} \tag{2.2}$$

where:

- the *acoustic model* $p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M})$ models the realisation of phones as audio
- the *pronunciation dictionary* $\mathscr{D}$ contains the possible sequences of phones (*phonetic pronunciations*) for each word, in broad or narrow transcription, as needed, e.g.

$$\mathscr{D}_{man} = \{[\text{m ae n}]\}) \tag{2.3}$$

  such that $\mathscr{D}_{w_{1:I}}$ represents all possible phone sequences corresponding to $w_{1:I}$, e.g.

$$\mathscr{D}_{\{the\ man\}} = \{[\text{dh ax sil m ae n}], [\text{dh iy sil m ae n}]\}) \tag{2.4}$$

- $P(\phi_{1:M}|w_{1:I})$ reflects any prior information on the likelihood of the different candidate pronunciations in the dictionary entry for each word (by default uniform)
- the *language model* $P(w_{1:I})$ represents the prior likelihood of sequences of words

Acoustic models for ASR are typically based on Hidden Markov Models (HMMs) [295] of which $s_{1:T}$ are the hidden states. HMM-based acoustic models are further discussed in Appendix B. The ASR systems used in this thesis and their choices of acoustic model, pronunciation dictionary and language model, as well as the methods used to extract the observations used to train them, are outlined in Appendix C.

Given a recognised (or otherwise known) word sequence $\hat{w}_{1:I}$, forced alignment finds the most likely phone sequence $\hat{\phi}_{1:M}$:

$$\hat{\phi}_{1:M} = \underset{\phi_{1:M} \in \mathscr{D}_{\hat{w}_{1:I}}}{\arg\max}\ P(\phi_{1:M}|\boldsymbol{o}_{1:T}, \hat{w}_{1:I}) = \underset{\phi_{1:M} \in \mathscr{D}_{\hat{w}_{1:I}}}{\arg\max}\ P(\phi_{1:M}, \boldsymbol{o}_{1:T}|\hat{w}_{1:I}) \tag{2.5}$$

$$\hat{\phi}_{1:M} = \underset{\phi_{1:M} \in \mathscr{D}_{\hat{w}_{1:I}}}{\arg\max} \left\{ P(\phi_{1:M}|\hat{w}_{1:I}) \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \right\} \tag{2.6}$$

followed by the most likely state sequence $\hat{s}_{1:T}$:

$$\hat{s}_{1:T} = \underset{s_{1:T}|\hat{\phi}_{1:M}}{\arg\max}\, p(\boldsymbol{o}_{1:T}, s_{1:T}|\hat{\phi}_{1:M}) \tag{2.7}$$

From $\hat{s}_{1:T}$ it is in turn possible to determine:

- The start and end frames $t_1^{(\hat{\phi}_m)}$ and $t_2^{(\hat{\phi}_m)}$ of each recognised phone $\hat{\phi}_m$ and thus the corresponding acoustic observation sequence (used for assessing how each phone is pronounced):

$$\boldsymbol{o}_{t_1:t_2}^{(\hat{\phi}_m)} = \boldsymbol{o}_{t_1^{(\hat{\phi}_m)}:t_2^{(\hat{\phi}_m)}} \tag{2.8}$$

  and duration (used when assessing rhythm):

$$d(\hat{\phi}_m) = t_2 - t_1 \tag{2.9}$$

- The start and end frames $t_1^{(\hat{w}_i)}$ and $t_2^{(\hat{w}_i)}$ of each word w_i and thus:

$$\boldsymbol{o}_{t_1:t_2}^{(\hat{w}_i)} = \boldsymbol{o}_{t_1^{(\hat{w}_i)}:t_2^{(\hat{w}_i)}} \tag{2.10}$$

- the phones $\hat{\phi}_{m_1:m_2}^{(\hat{w}_i)}$ corresponding to each word $\hat{w}_i$

As forced alignment takes $\hat{w}_{1:I}$ as a given, it requires an acoustic model and a pronunciation dictionary but no language model. Standard dictionaries of *canonical pronunciations* (i.e. pronunciations a listener would recognise as correct) using broad transcription are available for this purpose and generally used [281, 81]. When recognising non-native speech, it is

possible to supplement canonical pronunciations with candidate errorful pronunciations (e.g. *the* as [d ah]), which can be made particularly detailed if narrow transcriptions are used and the L1 is known (e.g. *man* as [mʲen] for a Russian speaker).

In the implementations of ASR used in this thesis, the Viterbi algorithm is used to determine the most likely path through the states, thus estimating $\hat{w}_{1:I}$, $\hat{\phi}_{1:M}$ and $\hat{s}_{1:T}$ simultaneously in the ASR stage. Running forced alignment as a separate task is still expedient, however, as it allows more possible alignments to be considered by limiting the choice of word sequence. The acoustic model is trained on word-transcribed data using a pronunciation dictionary to infer phones. Rather than returning the 1-best alignment, the Viterbi decoder can instead be configured to return a lattice of the most likely paths, as in the example illustrated in Fig. 2.4. Each possible path π through the lattice represents a possible $\{s_{1:T}, \phi_{1:M}, w_{1:I}\}$ and comes with its likelihood $p(\boldsymbol{o}_{1:T}, \pi)$. [296]

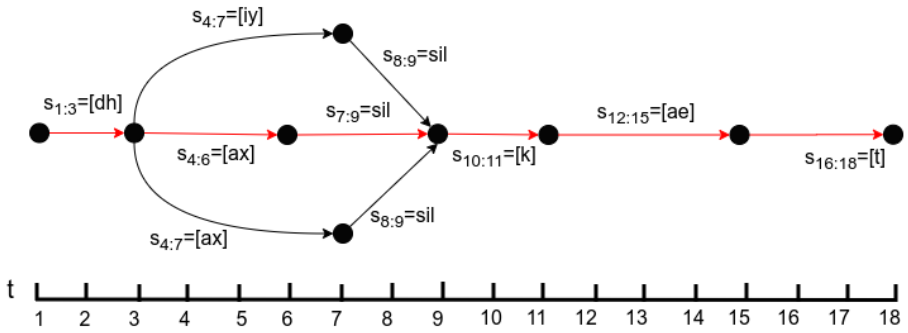


Fig. 2.4 Illustration of a lattice of possible alignments of a realisation of the phrase *the cat*, allowing two possible pronunciations of the word *the* and two possible durations of the final phone [ax] in *the*. The path in red corresponds to the word, phone and state sequences illustrated in Fig. 2.3.

Approaches to ASR confidence estimation can further be used to obtain estimates for the confidence in the predicted word sequence $P(\hat{w}_{1:I}|\boldsymbol{o}_{1:T})$ and the confidence of each word given its aligned location $P(\hat{w}_i|\boldsymbol{o}_{t_1:t_2}^{(\hat{w}_i)})$ [34, 286, 130, 150]. Given an output lattice, path likelihoods can be summed and normalised to obtain more advanced confidence metrics as well (see §7.2).

2.3 Single-view grading

In automatic L2 speech assessment, input sequential data $\boldsymbol{x}_{1:T}^{(n)}$ from a speaker n is used to predict a holistic grade $\hat{y}^{(n)}$ (holistic grading) and/or a grade $y_j^{(n)}$ representing proficiency with respect to a particular view j (single-view grading). The input $\boldsymbol{x}_{1:T}^{(n)}$ may consist, as needed, of acoustic features $\boldsymbol{o}_{1:T}^{(n)}$, recognised words $w_{1:I}^{(n)}$, phones $\phi_{1:M}^{(n)}$ and/or time-alignment

information $s_{1:T}^{(n)}$, obtained as discussed in §2.2, or other information, such as fundamental frequency extracted from audio. This section reviews approaches in the literature to single view grading.

Most approaches extract sets of expert features $\boldsymbol{v}_j^{(n)}$ to capture their chosen view j, using pre-defined, non-trainable, feature extractors $\mathscr{F}_j$:

$$\boldsymbol{v}_j^{(n)} = \mathscr{F}_j(\boldsymbol{o}_{1:T}^{(n)}) \tag{2.11}$$

The features are then fed into (usually parametric) graders $\mathscr{G}_j$ to predict single-view scores $\hat{y}_j^{(n)}$:

$$\hat{y}_j^{(n)} = \mathscr{G}_j(\boldsymbol{v}_j^{(n)}, \boldsymbol{\lambda}_j) \tag{2.12}$$

with $\mathscr{G}_j$ trained on human-annotated single-view scores $y_j^{(n)}$:

$$\hat{\boldsymbol{\lambda}}_j = \arg\min_{\boldsymbol{\lambda}_j} \sum_{n=1}^{N} (\hat{y}_j^{(n)} - y_j^{(n)})^2 \tag{2.13}$$

Alternatively, $\mathscr{F}_j$ can take the form of a parametric model trained either to extract features using a cost function other than grade prediction (e.g. as an unsupervised dimensionality reduction task) or in an end-to-end configuration, together with $\mathscr{G}_j$, such that the combined system learns to predict grade directly from the inputs.

The following sub-sections review approaches according to the classification of views set out in §2.1, namely text (§2.3.1), pronunciation (§2.3.2), tempo (§2.3.3), stress (§2.3.4), rhythm (§2.3.5), and intonation (§2.3.6).

2.3.1 Text

Text assessment is defined in this thesis as assigning a grade to a speaker based on the sequence of words recognised by an ASR for a sample of their speech. There is a broad literature on automatic grading of written material, particularly essays, including by extracting linguistically-inspired handcrafted features on grammar, topic relevance, coherence and syntactic complexity, by detecting grammatical errors, or by training end-to-end neural systems to predict grade directly from the tokenised word sequences [288, 70, 245, 7, 123, 19]. Spoken language differs considerably from written language in terms of grammar, syntax, vocabulary use and standards of proficiency, however [6, 72]. There are also challenges unique to spoken text assessment, including the prospect of ASR errors and the need to

deal with hesitations (words such as um, er) and disfluencies (including repetitions and false starts).

There has therefore been interest in text assessment for the spoken context, including extracting handcrafted features to capture the frequencies of hesitations and disfluencies [270] (usually grouped into the category of *fluency*) and to measure lexical complexity in the spoken context [24]. The application of deep learning methods to off-topic response detection [179], grammatical error correction [175] and end-to-end text grading [223] in spontaneous speech was investigated by other members of ALTA in parallel to the investigations into pronunciation and prosody described in this thesis. The end-to-end text grader from Raina et al. [223] is used in experiments on grader combination in Chapter 5.

2.3.2 Pronunciation

Pronunciation assessment was defined in §2.1 as characterising the proficiency of a speaker based on the way they realise words as sequences of discrete units of sound. It therefore includes those methods which force align candidate utterances to sequences of phones (or syllables), as discussed in §2.2, and then use the recognised sequence of phone instances $\hat{\phi}_{1:M}$, the acoustic features corresponding to each instance $\boldsymbol{o}_{t_1,t_2}^{(\hat{\phi}_m)}$ and/or measures of the confidence of the acoustic model in each phone instance as inputs to assess proficiency.

Most approaches in the literature act locally, identifying individually mispronounced words or phones or utterances containing them (pronunciation error detection). This is variously achieved by comparing $\boldsymbol{o}_{t_1,t_2}^{(\hat{\phi}_m)}$ to realisations produced by native speakers [37, 205, 187, 136, 132], using acoustic model confidence in each word as indicative of its intelligibility [128, 280, 289, 180, 73], aligning with an expanded dictionary containing non-canonical pronunciations, then assessing whether the canonical or non-canonical pronunciations of each word are more likely [238, 107, 134, 133], or training a supervised neural system to classify each word or phone instance as correct or errorful based on the acoustic observations corresponding to each phone instance [71, 80].

These approaches are reviewed in more detail in Chapter 3. The main advantage of error detection approaches over overall grading is their ability to provide rich feedback on the types of pronunciation errors that a learner is making to help adaptively drive further learning (§3.7). However, they have a number of disadvantages related to their data requirements and likely sources of error and bias. In the case of spontaneous speech, all approaches suffer from the issue that some of the words will be recognised incorrectly, making predictions as to their errorfullness meaningless. Native speaker comparison methods (§3.1) are also sensitive to the voice qualities and accents of the native speaker training corpus which may lead to biased evaluation of speakers in the non-native corpus. Confidence measures (§3.2)

are sensitive to a large number of irrelevant factors (background noise, speaker voice quality) that may affect ASR confidence at a particular time, as well as to the initial determination of the boundaries of each word and phone by the acoustic model. Alignment methods (§3.3 and §3.4) are sensitive to the candidate non-canonical pronunciations included in the dictionary. Finally, supervised methods (§3.5) require labelled error data, which is difficult to obtain and has been shown to be unreliable [172, 103].

Having run error detection, pronunciation features $\boldsymbol{v}_{pron}^{(n)}$ for a speaker can then be obtained by simply counting the numbers of detected errors or aggregating the word- or phone-level errorfullness metrics. Metallinou et al. [183] count the numbers of words and phones where confidence measures fall below a certain threshold and average the phone-level native to non-native acoustic model likelihood ratio, weighted by the duration of each phone. In Lee et al. [159], DTW distance metrics between native and non-native speech samples (see §3.1) are compiled into a similarity matrix, which is used as a feature to predict human-assigned scores. These features can then be passed through graders $\mathscr{G}_{|}$ to predict speaker grades.

The main family of approaches for overall grading without first locally characterising individual words or phones is that of phone distances. Features are extracted to represent the way the speaker realises phones across all their recorded speech and used to predict the speaker's grade. The first step is to aggregate the feature sequences $\boldsymbol{o}_{t_1,t_2}^{(\hat{\phi}_m)}$ corresponding to instances $\hat{\phi}_m = \psi$ of each phone ψ, to obtain a representation of how the speaker realises ψ overall (a form of speaker-specific acoustic model). These representations are then characterised relative to each other (phone distances), with the aim of compressing the representation and eliminate the effect of acoustic properties that do not vary between phones. These relative phone representations then act as features to predict grade.

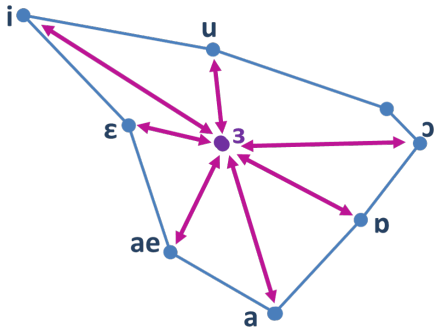


Fig. 2.5 Illustration of phone distance feature concept. Representations of phones are obtained in acoustic space. Each phone (violet point) is characterised by its pair-wise distance to every other phone (blue points).

Alignment should be performed using broad transcriptions, canonical pronunciation dictionaries, and acoustic models trained on non-native speakers, such that each $\hat{\phi}_m$ is as likely as possible to represent the canonical phone that the speaker was trying to realise (or should have realised). The phone-level representation should then capture the distribution of the actual sounds that these broad phones were realised as by the speaker, including any incorrect realisations, to be used to determine their overall pronunciation proficiency.

This approach does not require native speakers and is expected to be more robust to distortions introduced by incorrectly recognised words, as they are more likely to average out over multiple phones. No assumptions are made about the nature of good pronunciation; instead, criteria are learned statistically from human-assigned grades. Such grades are easier to obtain and have been shown to be more internally consistent than annotations of individual errors [172].

An early use of distances between representations of phone instances (segments) to characterise pronunciation in an unbiased way was presented by Huckvale [120] in the context of accent clustering. Chen and Evanini [47] introduced vowel space features introduced by, which measure the overall range of coverage of the vowel space (specifically overall ranges, overall area, overall dispersion and individual dispersion) based on the first two formants of each phone. Graham [100] showed that human-assigned grades can be better predicted by calculating Euclidean distances between formant features of pairs of vowels.

Minematsu et al. [10, 185] trained monophone acoustic models $\mathscr{M}_\psi$ to represent each vowel phone ψ based on all its instances in a speaker's aligned utterance $\boldsymbol{o}_{1:T}$:

$$\hat{\mathscr{M}}_\psi = \arg\max_{\mathscr{M}_\psi} \left\{ \sum_{m|\hat{\phi}_m=\psi} p(\boldsymbol{o}_{t_1,t_2}^{(\hat{\phi}_m)}, \hat{s}_{t_1:t_2}^{(\hat{\phi}_m)} | \hat{\phi}_m, \mathscr{M}_\psi) \right\} \tag{2.14}$$

where $\hat{\phi}_m$ is the mth phone of the 1-best recognised sequence, $\hat{s}_{t_1:t_2}^{(\hat{\phi}_m)}$ is the corresponding state sequence and $\boldsymbol{o}_{t_1,t_2}^{(\hat{\phi}_m)}$ is the corresponding segment.

They then computed the Bhattacharyya distances between the distributions of each pair of vowels ψ_1, ψ_2:

$$\Delta_B(\psi_1,\psi_2) = \int \sqrt{p(\boldsymbol{o}_{1:\tau}|\mathscr{M}_{\psi_1}) p(\boldsymbol{o}_{1:\tau}|\mathscr{M}_{\psi_2})}\, d\boldsymbol{o}_{1:\tau} \tag{2.15}$$

to act as features for pronunciation grading.

A generalisation of vowel distances to all pairs of phones using symmetric K-L divergence instead of Bhattacharyya distance was introduced in Kyriakopoulos et al. [151]:

$$\Delta_{\mathrm{SKL}}(\psi_1||\psi_2) = \frac{1}{2}\Delta_{\mathrm{KL}}(\psi_1||\psi_2) + \frac{1}{2}\Delta_{\mathrm{KL}}(\psi_2||\psi_1) \tag{2.16}$$

where:

$$\Delta_{\mathrm{KL}}\left(\psi_1||\psi_2\right) = \int p(\boldsymbol{o}_{1:\tau}|\mathscr{M}_{\psi_1})\log\left(\frac{p(\boldsymbol{o}_{1:\tau}|\mathscr{M}_{\psi_1})}{p(\boldsymbol{o}_{1:\tau}|\mathscr{M}_{\psi_2})}\right)d\boldsymbol{o}_{1:\tau} \tag{2.17}$$

with all pair-wise K-L divergences passed through a Gaussian Process to predict grade.

These features capture the speaker's pronunciation of each phone relative to each of the others compactly and effectively and were shown to predict grade with high accuracy. Their main weaknesses lie in the inability of the acoustic model training stage to account for the different salience of different phone instances and different frames within each phone instance to proficiency, as it is disconnected from the grading stage. To deal with these issues, phone distance features are built on to develop a tunable, end-to-end deep pronunciation grader in §5.2.1.

2.3.3 Tempo

The statistics of speed and hesitations in a person's speech can help predict proficiency in a direct and straightforward manner, as better speakers speak more quickly and hesitate less. The main features used to characterise tempo extracted from $\hat{w}_{1:I}$, $\hat{\phi}_{1:M}$ and $\hat{s}_{1:T}$ include:

- Rate of speech: The number of words spoken per second

$$r_s = \frac{I}{T} \tag{2.18}$$

- Articulation rate: The number of phones spoken per second

$$r_a = \frac{M}{T} \tag{2.19}$$

- Mean and standard deviation of the duration of disfluencies $\mathscr{H}$. This includes words such as 'um' and 'eh', false starts, repetitions, and other pauses and sounds classified as disfluencies or hesitations by the speech recogniser and its post-processing stages :

$$\mu_h = \frac{1}{I}\sum_{w_i\in\mathscr{H}}(t_2^{(w_i)} - t_1^{(w_i)}) \tag{2.20}$$

$$\sigma_h = \sqrt{\frac{1}{I}\sum_{w_i\in\mathscr{H}}(t_2^{(w_i)} - t_1^{(w_i)})^2 - \mu_h^2} \tag{2.21}$$

- Mean and standard deviation of duration of silences:

$$\mu_h = \frac{1}{I} \sum_{w_i = \text{sil}} (t_2^{(w_i)} - t_1^{(w_i)}) \tag{2.22}$$

$$\sigma_h = \sqrt{\frac{1}{I} \sum_{w_i = \text{sil}} (t_2^{(w_i)} - t_1^{(w_i)})^2 - \mu_h^2} \tag{2.23}$$

These features have been shown to be consistent within the speech produced by the same individual and vary among individuals [95], suggesting they are valid characterisations of the speech of individual speakers. They have been shown to be strong predictors of proficiency score [227, 117, 270], mainly characterising the speaker's general fluency (how fast they speak, how long their words and sentences are, how often and for how long they hesitate etc.). They are each prone to considerable bias due to individual voice quality, which may be somewhat reduced by using them all in combination. They are extremely commonly used in the literature, especially as baseline features [255, 117, 227, 270].

Given the definition of tempo as the overall speed and consistency of speech, which these features directly measure, the strong grading performance they already have, the bias inherent in them, and the fact that their main raw input, duration, is also the main input of rhythm, the grading of tempo is not the subject of further investigation in this thesis. Tempo features are, however, included in the baseline feature set (Appendix H) against which the systems developed in Chapter 5 are evaluated.

2.3.4 Stress

Every word in the English language has a canonical syllable on which it is stressed. These can be found in widely available pronunciation dictionaries (e.g. CMU [281]). Stressing words on the wrong syllable (e.g. 'CORR-ect' instead of 'corr-ECT') sounds unnatural and non-proficient.

Speakers also stress particular words within sentences in order to convey grammatical, syntactic and semantic information. For example, in the phrase "I prefer red wine to white wine", a proficient speaker of English would normally stress the words 'red' and 'white' relative to the two instances of 'wine' to indicate contrastive information. The word 'I' or the word 'prefer' might also be stressed for emphasis, however stressing the word 'wine' or the word 'to', not stressing any of the words, or stressing all of the words would sound unnatural. Such errors have been shown to considerably decrease the intelligibility and perceived correctness of non-native speakers [104].

Given the above, stress, as with pronunciation, is usually assessed in an error detection configuration. Pitch, duration and energy features are first used to determine which syllables of each word are stressed and the results compared against canonical rules for lexical or sentence stress. Grading is then performed using aggregated likelihood or frequency of detected errors.

A syllable by definition consists of a single vowel phone as its nucleus, with optional consonants before and after it [171]. It is the properties of this vowel nucleus that mainly determine whether or not the syllable is stressed [262]. Features used across the literature [14, 264, 262, 49] to detect which syllables are stressed include:

- Duration of the vowel, normalised by dividing by average vowel duration for that speaker (so as to compensate for the speaker's speaking rate):

$$d^{(norm)}(\phi_m) = \frac{t_2^{(\phi_m)} - t_1^{(\phi_m)}}{\frac{1}{\sum_{\phi_n \in \mathscr{V}} 1} \sum_{\phi_n \in \mathscr{V}} (t_2^{(\phi_n)} - t_1^{(\phi_n)})} \tag{2.24}$$

  Tepperman [264] additionally applies a series of fixed transformations based on the identity of the next phone.

- RMS of the energies $E_n^{(t)}$ of the samples s in the frame:

$$E_t = \sqrt{\sum_{s=1}^{S_t} E_s^{(t)^2}} \tag{2.25}$$

  normalised over all frames t in the utterance:

$$E_t^{(norm)} = \frac{E_t - \frac{1}{T}\sum_{t=1}^{T} E_t}{\sqrt{\frac{1}{T}\sum_{t=1}^{T}(E_t - \frac{1}{T}\sum_{t=1}^{T} E_t)^2}} \tag{2.26}$$

  then characterised by statistics over all frames within the vowel including mean:

$$\mu_E^{(\phi_m)} = \frac{1}{t_2^{(\phi_m)} - t_1^{(\phi_m)}} \sum_{t=t_1^{(\phi_m)}}^{t_2^{(\phi_m)}} E_t^{(norm)} \tag{2.27}$$

  median $Q_{E0.5}^{(\phi_m)}$, max $max_E^{(\phi_m)}$ and lower and upper quartiles $Q_{E0.25}^{(\phi_m)}$ and $Q_{E0.75}^{(\phi_m)}$ [49]).

- Pitch (f_0) extracted for each frame (see §2.3.6), normalised in the same way as RMS energy and represented by the same statistics $\mu_f^{(\phi_m)}$, $Q_{f0.5}^{(\phi_m)}$, $max_f^{(\phi_m)}$, $Q_{f0.25}^{(\phi_m)}$ and $Q_{f0.75}^{(\phi_m)}$.

Having obtained a normalised feature vector with the above metrics for each vowel ϕ_m, Tepperman [264] then trains a binary classifier to recognise whether a given vowel is stressed or not. If two vowels are recognised as stressed, the one with the highest probability is selected as the actual stressed syllable for that word. Chen and Jang [49] take Tepperman's method one step further by feeding the features for all the vowels of each word into a word-length dependent GMM classifier. Thus, for a word with N vowels, the system acts as an N-way classifier of which vowel the speaker has stressed. Shahin et al. [243] instead use a DNN classifier, with the variable number of syllables accounted for by zero padding the input. The detected stressed syllable in each case is then compared to the entry in the canonical dictionary.

Training such systems requires either stress-annotated non-native speech or a training set of speech known to be stressed correctly, in which lexical stress can be assumed to follow the canonical dictionary. The latter approach risks introducing biases if the voice qualities of the speakers of the sample are not representative of those likely to use the system (e.g. if they are native speakers or non-natives of a different L1, or if they are reading set text when the system is to be used on spontaneous speech).

Imoto et al. [126] approach sentence stress detection as an extension of lexical stress detection. Since the stressing of a word mainly manifests through its stressed syllable, each vowel in each word is classified as either not stressed (NS), secondary stressed (SS) — meaning it is the stressed vowel of its word but not of the sentence — or primary stressed (PS) — meaning it is the stressed vowel of its word which is in turn stressed within the sentence. Three-way classification is then performed, using an HMM to take context within the utterance into account. Minematsu et al. [186] extend this to six-way classification, also distinguishing between sentence stresses marking the beginning and end of a phrase. Lee et al. [160] perform lexical stress detection first, then combine the stress features of the stressed syllable of each word with lexical and syntactic features, specifically its identity w_i, a part of speech (POS) tag and class tag (function word vs. content word) obtained from a sentence analyser [35] and the number of vowels and syllables it contains. The combined feature vector for each word is passed, along with the vectors for the two preceding and three following words, through a linear chain Conditional Random Field (CRF) classifier, trained on a stress-annotated corpus, to detect whether each word is sentence-stressed. A second CRF classifier is then trained on a native speaker corpus to predict correct sentence stress position from the lexico-syntactic features only. Lee's combined system was therefore able to detect the sentence stressed words in any arbitrary utterance of spontaneous non-native speech, then automatically determine where the correct stress positions should have been and compare the two to provide feedback to the learner. The thresholds for the magnitude of

the difference between predicted and detected probability values to be identified as an error for feedback were optimised to maximise correlation between number of errors detected and human assigned proficiency score.

Considering the limits on feasibility placed by the requirements for detailed stress-annotated training data and the low weight placed on stress as a component of proficiency [57, 289, 122, 67], stress is not the subject of further investigation in this thesis, with the exception of the experiments on annotation in §7.4.1. The implementation of overall stress graders and/or deep stress detectors is discussed as part of future work in Chapter 8.

2.3.5 Rhythm

Traditionally, the natural rhythm of languages was believed to be governed by a principle known as isochrony, first introduced by Pike [211]. In languages such as French, known as syllable-timed, every syllable takes an equal amount of time to pronounce, while in languages such as English, known as stress-timed, it is the time between the stressed syllables of adjacent words which remains constant. The duration of individual syllables in English is therefore highly variable, depending on where they are relative to the stress of the current and adjacent words.

Part of what sounds strange about non-native speech under this theory is a failure to match the stress-timing rhythm of English [2]. This would suggest that the standard deviation of stress-to-stress intervals should be indicative of English proficiency. On the basis of this theory, Honig et al. [116] introduced *isochrony features*:

1. mean and standard deviation of length of time between consecutive stressed syllables
2. mean and standard deviation of length of time between consecutive unstressed syllables
3. ratios of above two means and above two standard deviations

These features are extracted based on the start and end times obtained from the output of syllable-based forced alignment (or phone-based forced alignment followed by grouping of phones into syllables).

The main problem with this approach arises from issues with the underlying theory. Firstly, not all varieties[1] of English are stress-timed and those that are are stress-timed to different extents [68]. This could lead to bias based on the variety of native English speech the learner is trying to emulate. In addition, the paradigm of isochrony itself is highly

[1] *Variety* is used here to refer to the different dialects, accents, registers and other systems of expression used by native speakers of the English language of different backgrounds or in different situations [61]

controversial, due to lack of direct empirical evidence of the phenomenon and the failure to classify many languages [65, 9].

The problems with simple isochrony features led Ramus et al. [226] to develop three new features which could be more reliably used to classify languages, based on the properties of adjacent vowels and the intervals between them. The phone sequence obtained from forced alignment is used to group speech into vocalic and intervocalic intervals $\tau_{1:K_V}^{(V)}$ and $\tau_{1:K_C}^{(C)}$, the former consisting of adjacent vowels, and the latter of consonants and silences, and obtain the start and end time of each. These are used to compute the following statistics:

1. The proportion of time devoted to vocalic intervals in the sentence, disregarding word boundaries:
$$\%V = \frac{\sum_{k=1}^{K_V} d(\tau_k^{(V)})}{\sum_{k=1}^{K_V} d(\tau_k^{(V)}) + \sum_{k=1}^{K_C} d(\tau_k^{(C)})} \tag{2.28}$$
where $d(\tau_k^{(V)})$ is the duration of the kth vocalic interval and $d(\tau_k^{(C)})$ is the duration of the kth intervocalic interval

2. The standard deviation of the duration of vocalic intervals:
$$\Delta V = \sqrt{\frac{1}{K_V}\sum_{k=1}^{K_V} d(\tau_k^{(V)})^2 - \left(\frac{1}{K_V}\sum_{k=1}^{K_V} d(\tau_k^{(V)})\right)^2} \tag{2.29}$$

3. The standard deviation of the duration of consonantal intervals:
$$\Delta C = \sqrt{\frac{1}{K_C}\sum_{k=1}^{K_C} d(\tau_k^{(C)})^2 - \left(\frac{1}{K_C}\sum_{k=1}^{K_C} d(\tau_k^{(C)})\right)^2} \tag{2.30}$$

In a language such as English in which vowels are routinely shortened depending on their position within a word, ΔV and ΔC are very high, while %V is very low (in fact they were respectively the highest and lowest of all languages tested). A low-proficiency non-native speaker with an L1 in which this is not the case is likely to fail to shorten vowels correctly and should therefore fall more closely to their L1 on these three axes. Honig named these features (together with normalised versions of the latter two) *Global Interval Proportions* (GIP) and used them with limited success to predict proficiency [116].

Grabe and Low [173, 98] generalised this concept to develop a more robust metric of rhythm based on the pairwise variability index (PVI), which measures the variability between successive intervals. PVI is applied to the duration of vowels as well as of inter-vocalic intervals.

Raw PVI, for each of vocalic and intervocalic intervals, is defined as:

$$\texttt{rPVI}^{(V)} = \frac{1}{K_V - 1} \sum_{k=1}^{K_V - 1} |d(\tau_k^{(V)}) - d(\tau_{k+1}^{(V)})| \tag{2.31}$$

$$\texttt{rPVI}^{(C)} = \frac{1}{K_C - 1} \sum_{n=1}^{K_C - 1} |d(\tau_n^{(C)}) - d(\tau_{k+1}^{(C)})| \tag{2.32}$$

The extraction of rPVI is illustrated in Figure 2.6.

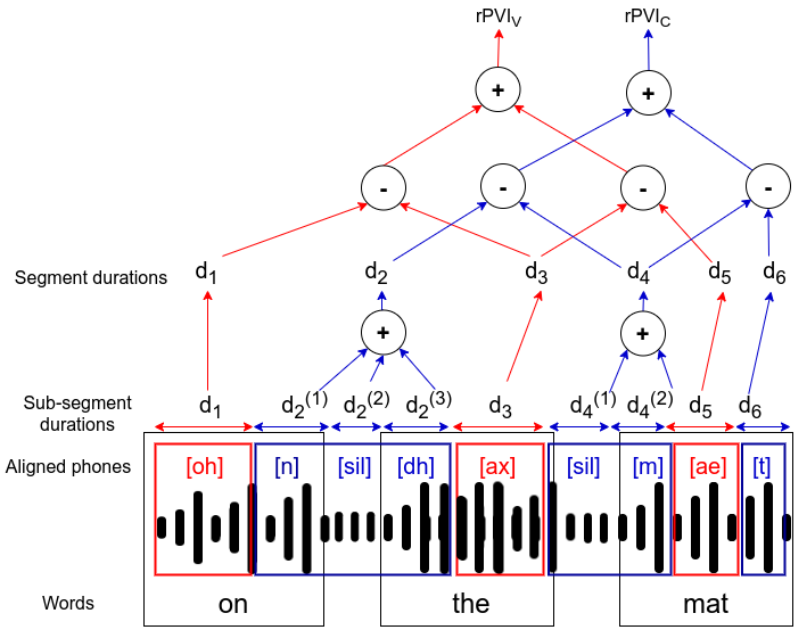


Fig. 2.6 Illustration of extraction of r-PVI features from sample phrase 'on the mat'

A normalised version of PVI (nPVI) is also defined:

$$\texttt{nPVI}^{(V)} = \frac{1}{K_V - 1} \sum_{k=1}^{K_V - 1} \frac{|d(\tau_k)^{(V)} - d(\tau_{k+1})^{(V)}|}{(d(\tau_k)^{(V)} + d(\tau_{k+1})^{(V)}/2} \tag{2.33}$$

$$\texttt{nPVI}^{(C)} = \frac{1}{K_C - 1} \sum_{k=1}^{K_C - 1} \frac{|d(\tau_k)^{(C)} - d(\tau_{k+1})^{(C)}|}{(d(\tau_k)^{(C)} + d(\tau_{k+1})^{(C)}/2} \tag{2.34}$$

Normalised PVI was found to improve on rPVI as it adjusts for the speaker's articulation rate and the duration of the particular syllables in question. The authors found that both rPVI and nPVI significantly outperform the Ramus metrics as well as other isochrony metrics at classifying languages based on their rhythmic properties.

Bertinetto et al. [22] modified PVI based on the idea that it is the lengths of individual vowels and consonants, rather than vocalic and consonantal intervals, the variation of which

is key to the rhythmic properties of languages. They therefore divided the duration of each interval by the number of phones it contained to yield a measure which they term the Control Compensation Index (CCI):

$$\texttt{CCI}^{(V)} = \frac{1}{K_V - 1} \sum_{n=1}^{K_V - 1} \left| \frac{d(\tau_k)^{(V)}}{l_k^{(V)}} - \frac{d(\tau_{k+1})}{l_{k+1}^{(V)}} \right| \tag{2.35}$$

$$\texttt{CCI}^{(C)} = \frac{1}{K_C - 1} \sum_{k=1}^{K_C - 1} \left| \frac{d(\tau_k)^{(C)}}{l_k^{(C)}} - \frac{d(\tau_{k+1})}{l_{k+1}^{(C)}} \right| \tag{2.36}$$

where $l_k^{(V)}$ is the number of sub-segments (phones and silences) in the kth vocalic interval and $l_k^{(C)}$ the number of sub-segments in the kth intervocalic interval.

Languages such as English are in their analysis termed 'compensation' languages, in that the sizes of adjacent vowels and adjacent consonants vary to compensate for each other, resulting in them having high CCIs. Speakers of 'control' languages such as Italian, try to keep phones at a constant length and so have low CCI.

Based on the above work, Honig et al. [116] define six PVI-based features for use in proficiency assessment, namely rPVI, nPVI and CCI for each of vocalic and consonantal intervals. Support Vector Machine (SVM) regression is then used to predict human judgments of the acceptability of subjects' rhythm and melody using these and the previous features. The PVI-based features outperform both isochrony and GIP features, but the combination of PVI and GIP performs even better, suggesting that they each contribute different information about the speaker's rhythm. Honig's six features are combined with the three GIP features, mean vocalic and consonant interval durations and the ratio of mean to standard deviation of each of vocalic and consonant interval durations to form a 13-feature baseline set used in §6.3.3.

Gharsellaoui et al. [92] defined optimised PVI (oPVI) as a generalisation of rPVI, nPVI and CCI:

$$\texttt{oPVI}^{(V)} = \frac{\alpha}{K_V - 1} \sum_{k=1}^{K_V - 1} \frac{\left| \frac{1}{l_k^{\theta}} d(\tau_k^{(V)}) - \frac{1}{l_{k+1}^{\varepsilon}} d(\tau_{k+1})^{(V)} \right|}{\left(\frac{1}{2} d(\tau_k)^{(V)} + d(\tau_{k+1})^{(V)} \right)^{\beta}} \tag{2.37}$$

$$\texttt{oPVI}^{(C)} = \frac{\alpha}{K_C - 1} \sum_{k=1}^{K_C - 1} \frac{\left| \frac{1}{l_k^{\theta}} d(\tau_k^{(C)}) - \frac{1}{l_{k+1}^{\varepsilon}} d(\tau_{k+1})^{(C)} \right|}{\left(\frac{1}{2} d(\tau_k)^{(C)} + d(\tau_{k+1})^{(C)} \right)^{\beta}} \tag{2.38}$$

where $1 \leq \alpha \leq 100$, $0 < \beta \leq 1$, $0 < \varepsilon \leq 1$, and, $0 < \theta \leq 1$ are parameters.

It can be seen that oPVI collapses to PVI when $\theta = \varepsilon = 0$, specifically rPVI when $\beta = 0$ and $\alpha = 1$ and nPVI when $\beta = 1$ and $\alpha = 2$, and to CCI when $\theta = \varepsilon = \alpha = 1$ and $\beta = 0$. The values of the parameters can be trained for the desired task of L1 detection and/or proficiency assessment, allowing the feature extraction to be tuned for optimal prediction of the ground-truth. This approach was shown to significantly improve performance on language classification tasks.

However, as with all previous approaches, this approach still treats all successive interval pairs identically, whereas different pairs would in practice be expected to have different effects and levels of salience for characterising rhythm and predicting proficiency. For example, a proficient speaker would be expected to give stressed syllables a larger duration than adjacent unstressed syllables and such contrasts would be more important for proficiency than the contrast between two adjacent unstressed syllables.

To address this issue, Kato et al. [137] use audio of native speakers reading identical text to measure a reference duration $d(\tau_k)^{(R)}$ for each vocalic interval τ_k. They then define:

$$r_k = \begin{cases} \frac{d(\tau_{k+1})}{d(\tau_k)} & d(\tau_k)^{(R)} \leq d(\tau_{k+1})^{(R)} \\ \frac{d(\tau_k)}{d(\tau_{k+1})} & d(\tau_{k+1})^{(R)} \leq d(\tau_k)^{(R)} \end{cases} = \frac{d(\tau_{k+1})}{d(\tau_k)}^{\operatorname{sgn}\left(\ln \frac{d(\tau_{k+1})^{(R)}}{d(\tau_k)^{(R)}}\right)} \tag{2.39}$$

to measure each pairwise duration ratio in the same direction as the equivalent ratio in the reference speaker.

Pairs with a duration ratio of greater magnitude in the reference native speaker are taken to be more important for proficiency and so the log of this magnitude is used as a weight in aggregating the log pairwise r_k scores to produce a speaker-level Referential Vowel Duration Ratio (RVDR) score:

$$\texttt{RVDR} = \frac{\sum_{k=1}^{K_V-1} \left|\ln \frac{d(\tau_{k+1})^{(R)}}{d(\tau_k)^{(R)}}\right| \ln(r_k)}{\sum_{k=1}^{K_V-1} \left|\ln \frac{d(\tau_{k+1})^{(R)}}{d(\tau_k)^{(R)}}\right|} \tag{2.40}$$

$$\texttt{RVDR} = \frac{\sum_{k=1}^{K_V-1} \left|\ln \frac{d(\tau_{k+1})^{(R)}}{d(\tau_k)^{(R)}}\right| \operatorname{sgn}\left(\ln \frac{d(\tau_{k+1})^{(R)}}{d(\tau_k)^{(R)}}\right) \ln \frac{d(\tau_{k+1})}{d(\tau_k)}}{\sum_{k=1}^{K_V-1} \left|\ln \frac{d(\tau_{k+1})^{(R)}}{d(\tau_k)^{(R)}}\right|} \tag{2.41}$$

$$\texttt{RVDR} = \frac{\sum_{k=1}^{K_V-1} \ln \frac{d(\tau_{k+1})^{(R)}}{d(\tau_k)^{(R)}} \ln \frac{d(\tau_{k+1})}{d(\tau_k)}}{\sum_{k=1}^{K_V-1} \left|\ln \frac{d(\tau_{k+1})^{(R)}}{d(\tau_k)^{(R)}}\right|} \tag{2.42}$$

The main issue with this method, as with native speaker comparison methods in pronunciation error detection (§3.1), is its reliance on recordings of native speakers reading identical

text. Obtaining such references beforehand is cumbersome in read speech assessment tasks and impossible in spontaneous speech assessment tasks. The reference utterances also risk introducing biases towards irrelevant attributes of the native speakers (especially if only one is used for each utterance).

In Kitamura et al. [142], the authors address the first of these problems by replacing $\ln \frac{d(\tau_{k+1})^{(R)}}{d(\tau_k)^{(R)}}$ with a weight $u(\boldsymbol{x}_k, \boldsymbol{x}_{k+1})$ to represent the importance and direction of the ratio $\ln \frac{d(\tau_{k+1})}{d(\tau_k)}$, where $\boldsymbol{x}_k$ is a vector representing phonemic, contextual and prosodic information extracted about vocalic interval τ_k. Equation 2.42 thus becomes:

$$\texttt{WVDR} = \frac{\sum_{k=1}^{K_V-1} u(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}) \ln \frac{d(\tau_{k+1})}{d(\tau_k)}}{\sum_{k=1}^{K_V-1} |u(\boldsymbol{x}_k, \boldsymbol{x}_{k+1})|} \tag{2.43}$$

A decision tree is then trained on adjacent vowel pairs from native speaker utterances to predict $\ln \frac{d(\tau_{k+1})^{(R)}}{d(\tau_k)^{(R)}}$ given $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$. The tree is then used on every adjacent pair in each candidate non-native speaker, with the output used as $u(\boldsymbol{x}_k, \boldsymbol{x}_{k+1})$.

This method has the advantage of not requiring matching native speakers for every utterance, though it still requires a native speaker training set and could thus suffer biases to properties of the rhythm of the native speakers which may not be necessary for proficiency. Weighting by the expected magnitude of the duration ratio is also limiting as it is not necessary that a larger pairwise ratio will be more indicative of proficiency. Finally, none of these methods are able to capture relationships beyond the interval-pair level.

In §5.2.2, a further generalisation of duration variability features using deep learning will be presented, to learn to predict human-assigned grade from duration features in a tunable end-to-end fashion, capturing the entire duration pattern across the utterance as well as the relative salience and different effects of different intervals and sub-segments.

2.3.6 Intonation

Intonation in this thesis refers to patterns of variation of pitch over an utterance. Pitch is an auditory sensation of sounds on a scale of *low* to *high*. The variation of pitch over a word or utterance can communicate semantic and grammatical information about the message being rendered. [61]

Pitch contours that carry meaning can be identified and analysed. For example, rising or falling pitch contours over particular words can indicate emphasis, while rising pitch at the end of an utterance can communicate that it is a question. Intonation patterns over sentences vary between languages, and so the way a speaker varies pitch over their utterances can be an important determiner of whether non-native speech sounds proficient.

During vowels and voiced consonants (collectively called voiced phones), when the source of speech is vibration of the speaker's vocal folds, pitch corresponds to the frequency of the vocal fold vibration [61]. This fundamental frequency (f_0) can be identified from the audio and used to characterise pitch. Fundamental frequency extraction tools such as REAPER [260] first detect the probability p_v that each frame is part of a region of voiced speech. In sections of speech more likely to be voiced, the quasi-periodicity caused by vocal fold vibrations is isolated by detecting regular glottal closure instants (GCIs), the time between which is used to deduce the fundamental frequency $f_0^{(t)}$ at each frame t, which in turn serves as a measure of pitch. Features are then extracted to represent the variation of f_0 over the course of the utterance, thereby characterising intonation.

In the simplest case, overall statistics of f_0 over all the frames of a candidate's speech can be computed to characterise that speaker's use of pitch, including mean:

$$\mu_{f_0} = \frac{1}{T}\sum_{t=1}^{T} f_0^{(t)} \tag{2.44}$$

median $Q_{f_0}^{(0.5)}$, maximum $\max_{1:T}(f_0^{(t)})$ and lower and upper quartiles $Q_{f_0}^{(0.25)}$ and $Q_{f_0}^{(0.75)}$. Such features are commonly employed as parts of larger sets to predict overall proficiency with considerable success [227, 270, 117]. However, as they do not take into account the variation of pitch over words and phones, their ability to assess adherence to the rules of phrasal intonation in English is limited. As with similar simple statistical features in other contexts, they also risk incurring bias towards irrelevant aspects of the speaker's voice quality. In particular, they can be biased to overall voice pitch (including gender variation) and the number and proportion of voiced regions.

A second group of methods is based on native speaker comparison, analogously to the corresponding methods for pronunciation assessment (§3.1). Ito et al. [129] compare the frame-by-frame f_0 sequence (normalised by its mean and standard deviation to minimise gender bias) for a non-native rendering of a known word sequence with that for a native speaker reading the same text, using a difference measure between the two as a predictor of correctness. This method allows phrasal annotation to be taken into account, by using the native as reference. As with other native speaker comparison methods, they can only work for read speech, since they require a recording of a native speaker reading identical text. They also risk being biased towards elements of the native speakers' prosody which may represent only one of multiple possible versions of correct intonation.

Kim and Sung [139] rank the syllables in an utterance by their mean pitch and compare these rankings between native and non-native speakers reading the same text. At the syllable-level, they divide each syllable into three subsections, compute the mean f_0 of each and use

them to classify the syllables into ‘rise-then-fall’, ‘rise-rise’, ‘fall-then-rise’, ‘fall-fall’ and ‘constant’. These classifications are made for both native and non-native speakers reading the same text and compared to grade the non-natives.

Most other approaches in the literature rely on using f_0 to detect meaning-carrying pitch contours and comparing them to a representation of the canonical contours for the phrase being assessed. The tone breaks and indices (ToBI) framework [16, 15, 17] provides a set of conventions for annotating pitch contours in English. Contours are separated into *pitch stresses*[2], which indicate word-level emphasis, and *boundary tones* which mark the boundaries of intonational phrases (see Appendix F). An example is seen in Figure 2.7.

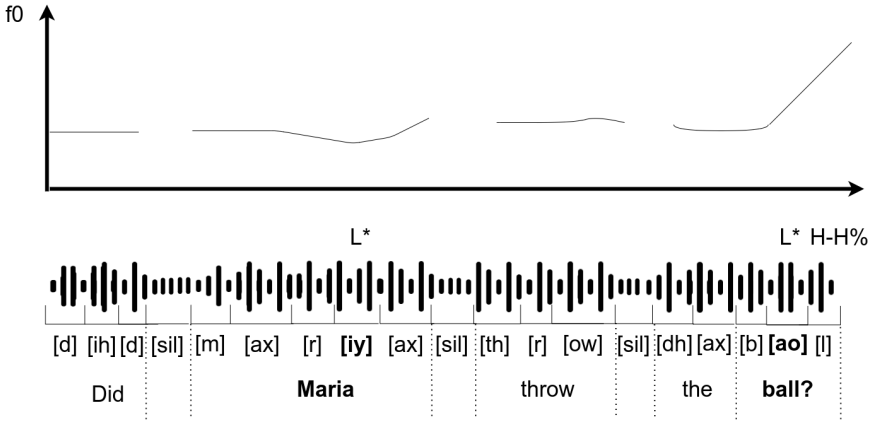


Fig. 2.7 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for question *Did Maria throw the ball?* (Fig. F.5). Equal emphasis on *Maria* and *ball* using lower pitch (L* pitch stress). Pitch rises at the end (H-H%) to indicate a yes/no question.

Li et al. [166] use syllable level pitch statistics to detect ToBI annotations using a hierarchical neural network classifier based on a labeled corpus, then compare the results to a canonical representation.

To avoid the need for such manually annotated canonical contours, Kang et al. [135] trained a model to predict the correct intonation pattern for any given sequence of words, using lexicosyntactic features extracted from the word sequence, following the established practice when generating pitch labels before rendering using f_0 templates in speech synthesis (e.g. the approach in Ronanki et al. [232] using LSTMs). This part of the system was trained using transcriptions and intonation labels from a corpus of native British English speakers. The mean pitch of each syllable in the speech of each non-native candidate, together with its mean MFCCs, are then used as features to predict the pitch contours actually present. Detected pitch contours can then be compared to the predicted canonical contours to evaluate

[2]More commonly called *pitch accents*. The term stress is used in this thesis to avoid confusion with the other meaning of the word accent.

the correctness of the speaker's intonation and detect individual errors. This approach identifies proficiency with similarity to native speakers and thus suffers similar problems and biases to the native speaker training set, as discussed in the case of pronunciation in §3.1. The extraction of lexicosyntactic features is also dependent on the accuracy of the word sequence, which can be an issue with spontaneous speech. Another important weakness of these approaches is that they assume only one correct intonation pattern for each word sequence. As seen in the examples in Appendix F, the same word sequence can be realised with many different intonation patterns, to produce often subtle differences in meaning. Speakers may thus be penalised for using a less common pattern. These approaches also ignore pitch information not captured by the contours.

An alternative approach utilising pitch contours is to detect which of the ToBI contours are present and then compare the actual f_0 trajectory to what would be expected for that contour. Rather than evaluating whether the ToBI contour was correct, this approach assumes that any of the ToBI annotations would constitute correct English intonation and instead focuses on evaluating how well the contours were followed. In Batliner et al. [14], contour-based features are extracted by fitting the f_0 values for each frame in each syllable believed to contain a contour to a line representing the contour trend by regression. The slope of the line together with the statistics of the deviation of actual f_0 from the line are then used to characterise the contour (Figure 2.8).

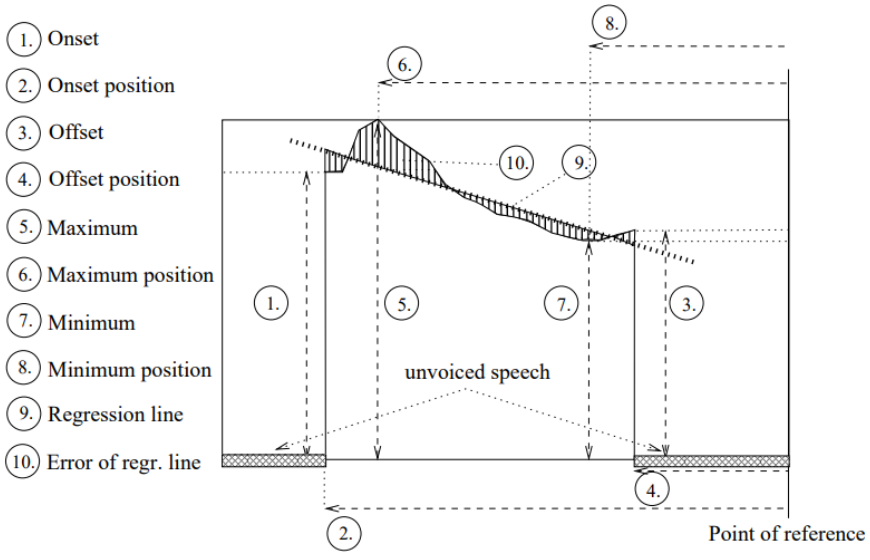


Fig. 2.8 Illustration of pitch contour and features used to describe it. Reproduced from [14]

In Honig et al. [117], the mean of the mean squared error about the regression line of the pitch of syllables immediately following stressed syllables is used as a predictor of

proficiency, since it is believed that these syllables have smooth pitch progression in native speakers and high degrees of f_0 variability in non-natives. In Coutinho et al. [58], contour-extracted features are extracted for each syllable and their means, standard deviations and other statistics used as feature to grade the overall proficiency of the speaker.

These techniques still rely on accurate ToBI annotations (whether labelled or predicted by a system trained on natives) and so suffer from the same associated biases. Another problem across all methods is the inability to extract pitch contours over voiceless consonants and periods of silence. These gaps in the pitch contours distort f_0 statistics calculation, contour extraction, and contour fitting alike. In §5.2.3, a number of novel approaches are considered to predict proficiency directly from f_0, taking unvoiced regions into account and avoiding both the coarseness of f_0 statistics and the loss of information and need for references associated with contour and native speaker methods.

2.4 Holistic Grading

As discussed in §2.1, a speaker n can be graded on their holistic proficiency $y^{(n)}$ or their single-view proficiency $y_j^{(n)}$ with respect to a particular view j. The former has the advantage of more readily available reliable data for training and evaluation, while the latter has the advantage of richer and more useful feedback for CALL purposes.

In §2.3, approaches in the literature to single-view grading for each of the views of text, pronunciation, tempo, stress, rhythm, and intonation were reviewed. Most approaches defined handcrafted algorithms $\mathscr{F}_j$ to extract sets of expert features $\boldsymbol{v}_j^{(n)}$ to represent their chosen view j:

$$\boldsymbol{v}_j^{(n)} = \mathscr{F}_j(\boldsymbol{o}_{1:T}^{(n)}) \tag{2.45}$$

which were then fed into graders $\mathscr{G}_j$ to predict single-view scores $y_j^{(n)}$:

$$\hat{y}_j^{(n)} = \mathscr{G}_j(\boldsymbol{v}_j^{(n)}, \boldsymbol{\lambda}_j) \tag{2.46}$$

To perform holistic grading, a number of authors [195, 60, 170] concatenate view-specific hand-crafted features for multiple views $1..J$ to produce a holistic feature set $\boldsymbol{v}^{(n)}$:

$$\boldsymbol{v}^{(n)} = [\boldsymbol{v}_1^{(n)} ... \boldsymbol{v}_J^{(n)}] \tag{2.47}$$

which is then passed through a grader $\mathscr{G}$, to predict holistic grades :

$$\mathscr{G}(\boldsymbol{v}^{(n)}, \boldsymbol{\lambda}) \rightarrow \hat{\hat{y}}^{(n)} \tag{2.48}$$

This grader can now be trained on human-assigned holistic scores $\bar{y}^{(n)}$:

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}} \sum_{n=1}^{N} (\hat{\bar{y}}^{(n)} - \bar{y}^{(n)})^2 \tag{2.49}$$

It is particularly common for prosodic features to be combined with each other [282, 64, 225, 139, 46, 257] as well as with pronunciation features [201, 200, 86, 85, 159, 183, 54]. A feature set consisting of concatenated tempo, rhythm, intonation and text features (Appendix H) is used as a baseline against which to compare the systems developed in Chapter 5.

If both holistic and single-view annotations are available, these approaches allow a form of multi-view grading, with each single-view feature set used to predict single-view grades and the concatenated set used to predict holistic grades. However, the single view grading is still reliant on difficult to obtain and often inconsistent human-annotated single-view grades. Further, hand-crafted features are, by their nature, inflexible, and each is reliant on a particular set of assumptions, on the basis of which it discards information which may turn out to be relevant. As was discussed in §2.3, the feature extraction processes could be improved if they could be tuned to extract information that is salient to proficiency.

Chen et al. [48] address this issue with a tunable feature extractor $\mathscr{F}$ that projects acoustic observations and recognised words to fixed-length vectors concatenated to a hidden representation $\boldsymbol{v}^{(n)}$, trained end-to-end with a grader $\mathscr{G}$ using $\boldsymbol{v}^{(n)}$ to predict holistic grade.

$$\mathscr{F}(\boldsymbol{x}_{1:T}^{(n)}, \boldsymbol{\lambda}) \rightarrow \boldsymbol{v}^{(n)}; \;\; \mathscr{G}(\boldsymbol{v}^{(n)}, \boldsymbol{\lambda}) \rightarrow \hat{\bar{y}}^{(n)} \tag{2.50}$$

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}} \sum_{n=1}^{N} (\hat{\bar{y}}^{(n)} - \bar{y}^{(n)})^2 \tag{2.51}$$

In contrast to the hand-crafted approach, $\mathscr{F}$ is free to learn any mapping that will extract features predictive of grade. This increases representational capacity and predictive power at the expense of interpretability. Other approaches achieve the same outcome in two stages, with $\mathscr{F}$ taking the form of an unsupervised feature extractor learning to extract compact representations of the speaker's utterances (such as i-vectors and GMM parameters), and $\mathscr{G}$ trained separately [259, 51]. Both the end-to-end and two stage systems can be trained on readily available holistic scores and automatically learn to best extract features predictive of grade rather than relying on the assumptions inherent in hand-crafted features. However, neither can be used for multi-view assessment as no single-view information is encoded.

In Chapter 5, end-to-end single-view graders able to be trained using only holistic scores are investigated, thereby allowing multi-view grading while combining the tunability and training data reliability of the end-to-end holistic graders.

2.5 Comparison of approaches

Table 2.1 compares the grading systems reviewed in this chapter, including previous work (PDF), to the single view (§5.2) and combined (§5.3) graders introduced in this thesis.

Approach	View(s)	Tun. FE	Ref. Ind.	Ann. Ind.	Spont.	Grades
Neumeyer [201, 200]	P	No	Yes	Yes	No	P
Franco [85, 86]	P	No	Yes	Yes	No	P
Cheng [51]	P	No	Yes	Yes	No	P
Muller [195]	P	No	Yes	Yes	No	H
Liu [170]	PX	Yes	Yes	No	Yes	H
Cucchiarini [64]	TPRX	No	Yes	Yes	No	TPRX
Lee [159]	P	No	No	Yes	No	P
Chen [47]	P	No	Yes	Yes	No	P
Honig [117]	PRI	No	Yes	Yes	No	PRI
Kato [137]	R	No	No	Yes	No	R
Kitamura [142]	R	No	Yes	Yes	No	R
Kim [139]	I	No	Yes	No	No	I
Coutinho [58]	TRI	No	Yes	Yes	No	TRI
Strik [255]	T	No	Yes	Yes	Yes	T
Honig [116]	RI	No	Yes	Yes	Yes	RI
van Dalen [270]	TIX	No	Yes	Yes	Yes	H
Bhat [24]	X	No	Yes	Yes	Yes	H
Crossley [60]	TPX	No	Yes	Yes	Yes	H
Metallinou [183]	P	No	Yes	Yes	Yes	H
Graham [100]	P	No	Yes	Yes	Yes	H
Rashid [227]	TIX	No	Yes	Yes	Yes	H
PDF [151]	P	No	Yes	Yes	Yes	H
Chen [48]	H	Yes	Yes	Yes	Yes	H
Takai [259]	H	Yes	Yes	Yes	Yes	H
Raina [223]	X	Yes	Yes	Yes	Yes	H
§5.2	PRI	Yes	Yes	Yes	Yes	H
§5.3	PRIX	Yes	Yes	Yes	Yes	H

Table 2.1 Proficiency graders compared by input/structure views (T=Tempo, P=Pronunciation, R=Rhythm, I=Intonation, X=Text, H=Holistic), tunability of feature extraction (Tun. FE), independence to reference speakers (Ref. Ind.), independence to additional annotation (Ann. Ind.), application to spontaneous speech (Spont.), and ground-truth grade views.

It is seen that most approaches (Neumeyer–Coutinho) are either only designed for read speech or are not able to be trained without the need for additional manual annotation or

references. They also mostly use custom methods for obtaining ground-truth single-view grades which are not connected to the CEFR or other broadly accepted standards. The remainder are divided into single-view handcrafted feature approaches with non-tunable feature extractors (Strik–PDF) and holistic end-to-end systems which cannot give feedback on individual views (Chen and Takai). The novelty of the systems proposed in §5.2 and §5.3, as well as, on the basis of the same work and within the ALTA project by Raina, is that they are end-to-end systems with tunable feature extractors that can still make single-view predictions.

2.6 Chapter Summary

This chapter reviewed the literature on spoken language assessment. In §2.1, the tasks of single-view and holistic grading were distinguished. Single-view grading aids adaptive learning by providing targeted feedback on an individual aspect of proficiency. However, reliable single-view grades to train and test such systems are difficult to obtain, whereas holistic grading follows accepted standards and is both more readily available and more consistent.

Methodologies for converting raw audio into sequences of acoustic features and time-aligned words and phones were discussed in §2.2. Approaches using this information to assign single-view proficiency grades were then reviewed in §2.3, for each of the views of text, pronunciation, tempo, stress, rhythm, and intonation. The approaches reviewed mostly involved extracting hand-crafted features from the original information to represent each view. These features were then fed into a graders trained to predict human-annotated single-view grades. The approaches found to be the most promising were phone distance features to represent pronunciation (§2.3.2), pairwise variability metrics for rhythm (§2.3.5) and various features extracted from f_0 to characterise intonation (§2.3.6). These methods incorporate domain knowledge to ensure they only extract information representative of their respective views. However, this causes them to be inflexible and overly reliant on the assumptions used to define each of them, such that they risk discarding potentially useful information.

Two approaches to holistic grading were then reviewed in §2.4. The first involves concatenating the handcrafted features for various views and feeding them into a grader to predict holistic score. This method suffers the same issues of inflexibility as the single-view graders. The second approach uses neural feature extractors which can be trained in an end-to-end configuration to extract features so as to optimise grade prediction. This allows the feature extraction process to be flexible and tunable to the grading task, such that it thus

learns to preserve the information most representative of proficiency as defined by the human annotators rather than based on the assumptions used in the design of the features. However, it lacks the interpretability and view specificity possible with the first method.

In Chapter 5, a compromise between these two approaches is introduced, aiming to combine the advantages of the end-to-end holistic grader with those of the handcrafted feature single-view graders. The hand-crafted feature extractors for pronunciation, rhythm and intonation are each generalised into hierarchical neural analogues. These continue to limit the information available for grading in a way that exploits domain knowledge to ensure view specificity, while also allowing the parameters of the process to be learned so as to best predict human-assigned grade. The graders are trained using holistic, rather than single-view scores, and their ability to still yield scores linked to their respective views is investigated. Finally, the graders are combined to yield a single holistic grader, which is compared to versions of the two baselines described in §2.4. A comparison of all the systems reviewed in this Chapter with those introduced in Chapter 5 is displayed in §2.5.

Chapter 3

Pronunciation Error Detection

This chapter reviews the literature on pronunciation error detection. Pronunciation, for the purposes of this thesis, refers to the way a speaker realises words as sequences of discrete units of sound. Pronunciation error detection involves evaluating whether each word in a speaker's utterances is pronounced correctly or incorrectly (word-level detection) or whether a particular utterance contains such word-level errors (utterance-level detection). It contrasts with overall pronunciation assessment which involves directly assigning a grade to the pronunciation of the speaker as a whole.

The first step in any pronunciation assessment method as defined above must be to recognise the words and sounds that were spoken and identify the segment of audio corresponding to each. The speech audio processing methods used to achieve this, namely automatic speech recognition (ASR) and forced alignment, were reviewed in §2.2. ASR, which involves recognising the words, is particularly important when dealing with spontaneous speech, where the text being spoken isn't known beforehand. Having identified the acoustic realisations of the sounds making up each word, authors differ on what properties of these constitute proficient pronunciation. Answers include similarity to the way a native speaker would pronounce the same word, intelligibility of the way the word was pronounced to a listener, adherence to a canonical pronunciation defined for the word, and, finally, to make no assumption and learn criteria statistically from human annotations. These different answers lead to different families of approaches, namely native speaker comparison methods, confidence measure methods, recognition methods, and supervised methods respectively.

The first sections of this chapter explore the main categories of approaches for detecting pronunciation errors in the literature, based on these different implicit assumptions about the nature of good pronunciation. Native speaker similarity methods (§3.1) approach proficiency as similarity to the pronunciation of native speakers. ASR confidence methods (§3.2) attempt to assess intelligibility of the word as spoken to a listener. Extended recognition networks

(§3.3) and phone recognition methods (§3.4) define good pronunciation as adherence to the canonical sequence of phones for the word being uttered. Finally supervised methods (§3.5) learn the criteria of good pronunciation implicitly from human annotators.

The availability and quality of read and spontaneous speech corpora for feasibly training and evaluating methods are discussed in §3.6, while §3.7 discusses results from the literature on how to best provide useful types of feedback to learners. Finally, §3.8 compares all reviewed systems to that introduced in Chapter 7.

3.1 Native speaker similarity methods

If proficiency is defined as similarity to native speakers, assessment requires measuring the distance between the way the candidate speaker pronounces the phones of the language and the way reference native speakers pronounce them.

An early approach developed at SRI [21, 86, 201, 200] uses utterances of native speakers $\boldsymbol{o}_{1:T}^{(native)}$ and their transcribed word sequences $w_{1:I}^{(native)}$ to train a parametric acoustic model $p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}, \hat{\mathscr{M}})$ to capture the way native speakers pronounce the phones of English:

$$\hat{\mathscr{M}}^{(native)} = \arg\max_{\mathscr{M}} p(\boldsymbol{o}_{1:T}^{(native)}|w_{1:I}^{(native)}, \mathscr{M}) \tag{3.1}$$

$$\hat{\mathscr{M}}^{(native)} = \arg\max_{\mathscr{M}} \left\{ \sum_{\phi_{1:M} \in \mathscr{D}_{w_{1:I}}^{(native)}} \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}^{(native)}, s_{1:T}|\phi_{1:M}, \mathscr{M}) \right\} \tag{3.2}$$

where $\mathscr{D}_{w_{1:I}}^{(native)} = \mathscr{D}_{w_{1:I}^{(native)}}^{(can)}$ represents all possible canonical pronunciations of $w_{1:I}^{(native)}$.

The likelihood $p(\boldsymbol{o}_{1:T}|w_{1:I}, \hat{\mathscr{M}}^{(native)})$ given the trained native speaker model $\hat{\mathscr{M}}$ of a non-native candidate utterance $\boldsymbol{o}_{1:T}$ with known word sequence $w_{1:T}$ is then used to indicate the degree of nativeness and thus proficiency of the candidate's speech. The idea is that the more similar the candidate's pronunciation is to the native pronunciation, the easier it will be for the native-trained models to recognise the candidate's speech. Utterance-level errors $e(w_{1:I})$ could thus be detected by directly thresholding this likelihood:

$$e(w_{1:I}) = p(\boldsymbol{o}_{1:T}|w_{1:I}, \hat{\mathscr{M}}^{(native)}) < \Theta_{nat} \tag{3.3}$$

$$e(w_{1:I}) = \sum_{\phi_{1:M} \in \mathscr{D}_{w_{1:I}}} \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}, \mathscr{M}^{(native)}) < \Theta_{nat} \tag{3.4}$$

where Θ_{nat} is a tunable threshold.

A more computationally feasible approach is to first use the model to force align each non-native utterance, obtaining the most likely phone sequence $\hat{\phi}_{1:M}$ and state sequence $\hat{s}_{1:T}$:

$$\hat{\phi}_{1:M} = \underset{\phi_{1:M} \in \mathscr{D}_{w_{1:I}}^{(can)}}{\arg\max} \left\{ P(\phi_{1:M}|w_{1:I}) \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}, \hat{\mathscr{M}}^{(native)}) \right\} \tag{3.5}$$

$$\hat{s}_{1:T} = \underset{s_{1:T}|\hat{\phi}_{1:M}}{\arg\max}\, p(\boldsymbol{o}_{1:T}, s_{1:T}|\hat{\phi}_{1:M}, \hat{\mathscr{M}}^{(native)}) \tag{3.6}$$

Utterance-level errors can then be detected by thresholding the likelihood of the 1-best state sequence:

$$e(w_{1:I}) = p(\boldsymbol{o}_{1:T}, \hat{s}_{1:T}|w_{1:I}, \hat{\mathscr{M}}^{(native)}) < \theta_{nat} \tag{3.7}$$

where θ_{nat} is a tunable threshold.

To detect word-level errors, $\hat{s}_{1:T}$ is used to identify the phones $\hat{\phi}_{m_1:m_2}^{(w_i)}$ and frames $\boldsymbol{o}_{t_1:t_2}^{(w_i)}$ corresponding to each word w_i and a threshold η_{nat} applied to the likelihood:

$$e(w_i) = p(\boldsymbol{o}_{t_1:t_2}^{(w_i)}, \hat{s}_{1:T}|\hat{\phi}_{m_1:m_2}^{(w_i)}, \hat{\mathscr{M}}^{(native)}) < \eta_{nat} \tag{3.8}$$

An advantage of this technique is that it does not require the natives and non-natives to have spoken the same text and so can be used on spontaneous speech. Its main weakness is that the likelihood will be affected by factors other than the nativeness of the candidate's pronunciation, including the text being spoken (especially with spontaneous speech), the similarity of other speech attributes between the native and non-native data, and background acoustic conditions. The method thus risks yielding unreliable results, generalising poorly and introducing biases to irrelevant attributes of the speech being assessed.

Another approach, only applicable with read speech, is to have a native reference speaker read out an identical text to the non-native candidate and compare their realisations to each other. Karhila et al. [136] perform free phone recognition on each of the non-native $\boldsymbol{o}_{1:T}$ and a reference native speaker $\boldsymbol{o}_{1:T'}^{\text{native}}$ to obtain the most likely phone sequences for each:

$$\hat{\phi}_{1:M} = \underset{\phi_{1:M} \in \Phi}{\arg\max} \left\{ \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \right\} \tag{3.9}$$

$$\hat{\phi}_{1:M'}^{\text{native}} = \underset{\phi_{1:M} \in \Phi}{\arg\max} \left\{ \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T'}^{\text{native}}, s_{1:T}|\phi_{1:M}) \right\} \tag{3.10}$$

where Φ represents all possible phone sequences (see §3.4).

The Levenshtein distance between the resultant phone sequences is then used to detect errorful utterances:

$$e(w_{1:I}) = \text{lev}(\hat{\phi}_{1:M}, \hat{\phi}^{\text{native}}_{1:M'}) < \theta_{lev} \tag{3.11}$$

where θ_{lev} is a threshold, and errorful words:

$$e(w_i) = \text{lev}(\hat{\phi}^{(w_i)}_{m_1:m_2}, \hat{\phi}^{(w_i)\,\text{native}}_{m_1:m_2}) < \eta_{lev} \tag{3.12}$$

where $\hat{\phi}^{(w_i)}_{m_1:m_2}$ and $\hat{\phi}^{(w_i)\,\text{native}}_{m_1:m_2}$ are the sections of the respective aligned phone sequences corresponding to the word w_i, and η_{lev} is a threshold.

Most other authors following the native reference speaker approach instead force align the native and non-native utterances to the same phone sequence $\phi_{1:M}$ representing canonical pronunciations of every word:

$$\hat{s}_{1:T} = \underset{s_{1:T}|\phi_{1:M}}{\arg\max}\, p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \tag{3.13}$$

$$\hat{s}^{\text{native}}_{1:T'} = \underset{s_{1:T'}|\phi_{1:M}}{\arg\max}\, p(\boldsymbol{o}^{\text{native}}_{1:T'}, s_{1:T'}|\phi_{1:M}) \tag{3.14}$$

using the results to obtain the start and end frames in each of the native and non-native utterance of each phone ϕ_m.

Phone-level errors $e(\phi_m)$ can then be detected by thresholding a distance metric between the two resultant sequences of frames $\boldsymbol{o}^{(m)}_{t_1:t_2}$ and $\boldsymbol{o}^{(m,\text{native})}_{t'_1:t'_2}$:

$$e(\phi_m) = \Delta(\boldsymbol{o}^{(m)}_{t_1:t_2}, \boldsymbol{o}^{(m,\text{native})}_{t'_1:t'_2}) > \theta_d \tag{3.15}$$

where Δ is some distance measure and θ_d is a threshold.

This distance metric is commonly obtained by Dynamic Time Warping (DTW) [37, 187]. Nicolao et al. [205] instead feed the raw pair of observation vector sequences $\boldsymbol{o}^{(m)}_{t_1:t_2}$ and $\boldsymbol{o}^{(m,\text{native})}_{t'_1:t'_2}$ into a binary classifier to detect errors, though this method requires error-annotated training data (see §3.5).

The main disadvantage of these methods is that they assume the native and non-natives pronounced the common text using the same canonical pronunciation. Kamimura et al. [132] resolve this by first freely aligning the native and non-native utterances, then computing the minimum edit distance between them, as in the Karhila et al. Levenshtein method discussed above, and then using DTW to compare features between pairs of phones that are matched between the two sequences.

Native speaker comparison methods in general suffer from high sensitivity to accent and other voice attributes of the reference native speakers which may not be relevant to proficiency. As a result, they risk bias against accents and attributes not represented in the native speaker training data. These approaches also do not allow any richer feedback about the nature of the error being made in each case, other than its location.

3.2 ASR confidence methods

In recent years, there has been increasing agreement that *intelligibility*, the ease with which an utterance is comprehensible to a human listener, is more critical than native speaker similarity as a metric of communicative competence, particularly for speakers at less advanced stages of language learning [128, 289]. As a result, an increasing number of researchers have approached pronunciation assessment without comparison to native speaker trained models [117, 273, 180, 280, 185].

By removing the explicit comparison and only training on non-native speakers, it is possible to avoid many of the problems inherent in native speaker similarity methods, such as the increased data requirements, the tendency to be text-dependent and the sensitivity to the voice attributes of the native speakers, which among other things creates a propensity towards unfairly penalising otherwise fluent and intelligible speakers with accents not present in the native speaker training data.

Defining proficiency as intelligibility avoids the problems with native speakers but opens the question of how intelligibility itself should be quantified. The primary method of characterising intelligibility is to consider the ease with which an utterance can be understood by humans as being represented by the ease with which it can be understood by a machine i.e. by confidence measures derived from an ASR. The idea is that a word or phone in which the ASR has lower confidence is more likely to have been pronounced unclearly or incorrectly.

Given a non-native utterance $\boldsymbol{o}_{1:T}$ corresponding to a known or recognised word sequence $w_{1:I}$, forced alignment is first performed using a canonical pronunciation dictionary $\mathscr{D}^{(can)}$ to yield the most likely phone sequence $\hat{\phi}_{1:M}$ and state sequence $\hat{s}_{1:T}$:

$$\hat{\phi}_{1:M} = \underset{\phi_{1:M} \in \mathscr{D}^{(can)}_{w_{1:I}}}{\arg\max} \left\{ \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \right\} \tag{3.16}$$

$$\hat{s}_{1:T} = \underset{s_{1:T}|\phi_{1:M}}{\arg\max}\, p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \tag{3.17}$$

The confidence in the recognised phones $\hat{\phi}_{1:M}$ at their aligned locations is taken to indicate the intelligibility of the utterance and is thus thresholded to detect utterance errors $e(w_{1:I})$:

$$e(w_{1:I}) = p(\boldsymbol{o}_{1:T}, \hat{s}_{1:T}|\hat{\phi}_{1:M}) < \Theta \tag{3.18}$$

The aligned state sequence $\hat{s}_{1:T}$ is used to determine the frames $\boldsymbol{o}_{t_1:t_2}^{(w_i)}$ corresponding to each word w_i and $\boldsymbol{o}_{t_1:t_2}^{(\phi_m)}$ corresponding to each phone $\hat{\phi}_m$. The confidence of each w_i in its aligned location can then be computed and thresholded to detect word-level errors [54]:

$$e(w_i) = \sum_{\phi_{m_1:m_2} \in \mathscr{D}_{w_i}^{(can)}} \sum_{s_{t_1:t_2}|\phi_{m_1:m_2}} p(\boldsymbol{o}_{t_1:t_2}^{(w_i)}, s_{t_1:t_2}|\phi_{m_1:m_2}) < \theta \tag{3.19}$$

Phone-level errors $e(\hat{\phi}_m)$ can similarly be detected by thresholding the log likelihood of each phone in its aligned position [200, 21, 64, 46]:

$$e(\hat{\phi}_m) = \log(p(\mathbf{o}_{t_1:t_2}^{(\hat{\phi}_m)}|\hat{\phi}_m)) < \eta \tag{3.20}$$

where η is a threshold. In some versions, likelihood is first normalised by phone duration:

$$e(\hat{\phi}_m) = \frac{\log(p(\mathbf{o}_{t_1:t_2}^{(\hat{\phi}_m)}|\hat{\phi}_m))}{t_2^{(m)} - t_1^{(m)}} < \eta \tag{3.21}$$

In a further development of this approach, forced alignment is repeated multiple times, fixing the time stamps $t_1^{(m)}$ and $t_2^{(m)}$ and the rest of the utterance around them, but replacing the phone $\hat{\phi}_m$ by alternative phones $\psi_m \in \Phi$. The likelihood $\log(p(\mathbf{o}_{t_1:t_2}^{(\hat{\phi}_m)}|\psi_m)$ is obtained each time and a normalised log-posterior probability of the original canonical phone $\hat{\phi}_m$, known as Goodness of Pronunciation (GOP), obtained as:

$$GOP(\hat{\phi}_m|\mathbf{o}_{t_1:t_2}^{(\hat{\phi}_m)}) = \log\left(\frac{p(\mathbf{o}_{t_1:t_2}^{(\hat{\phi}_m)}|\hat{\phi}_m)}{\sum_{\psi\in\Phi} p(\mathbf{o}_{t_1:t_2}^{(\psi_m)}|\psi_m))}\right) \Big/ (t_2^{(m)} - t_1^{(m)}) \tag{3.22}$$

which in turn can be estimated as:

$$GOP(\hat{\phi}_m|\mathbf{o}_{t_1:t_2}^{(\hat{\phi}_m)}) = \log\left(\frac{p(\mathbf{o}_{t_1:t_2}^{(\hat{\phi}_m)}|\hat{\phi}_m)}{\max_{\psi\in\Phi} p(\mathbf{o}_{t_1:t_2}^{(\hat{\phi}_m)}|\psi_m))}\right) \Big/ (t_2^{(m)} - t_1^{(m)}) \tag{3.23}$$

and used to detect phone-level errors:

$$e(\hat{\phi}_m) = GOP(\hat{\phi}_m|\mathbf{o}_{t_1:t_2}^{(\hat{\phi}_m)}) < \eta \tag{3.24}$$

GOP is generally used on a non-native adapted ASR, rather than a purely native trained one [290, 257, 180, 73]. Thus, a single phone loop alignment on each phone is required to calculate GOP. Unnormalised versions were used in older papers [86, 85, 54, 140, 84].

An important weakness of this method is its over-reliance on the initial time-alignment, the phone start and end times from which are then fixed during the phone recognition stage. By not allowing different phones to have different lengths, the phone recognition problem is distorted in favour of phones likely to have similar lengths to that originally aligned. On the other hand, relaxing the time constraints would make alignments difficult to compare to each other, as each phone would be based on a different segment of audio.

To address the problem, Cox et al. [59] align the entire utterance twice, once with ASR followed by alignment with a canonical dictionary and the second time by phone recognition:

$$\hat{w}_{1:I} = \arg\max_{w_{1:I}} \left\{ P(w_{1:I}) \sum_{\phi_{1:M} \in \mathscr{D}_{w_{1:I}}} \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}, \mathscr{M}) \right\} \tag{3.25}$$

$$\hat{\phi}^{(1)}_{1:M} = \arg\max_{\phi_{1:M} \in \mathscr{D}_{\hat{w}_{1:I}}} \left\{ \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}, \mathscr{M}) \right\} \tag{3.26}$$

where $\mathscr{D}_{w_{1:I}}$ is a canonical pronunciation dictionary.

$$\hat{\phi}^{(2)}_{1:M'} = \arg\max_{\phi_{1:M} \in \Phi} \left\{ P(\phi_{1:M}|w_{1:I}) \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}, \mathscr{M}) \right\} \tag{3.27}$$

where Φ contains all possible phone sequences.

Each sequence is then decomposed into words based on $\hat{w}_{1:I}$, to obtain each pair of phone sequences corresponding to each word $\hat{\phi}^{(w_i)(1)}_{m_1:m_2}$ and $\hat{\phi}^{(w_i)(2)}_{m_1:m_2}$. The proportion of agreement between the two phone sequences for each word is then used as indicative of confidence, based on the idea that a well pronounced word is more likely to be recognised as easily without the language model and canonical dictionary than a poorly pronounced word. Likelihood ratio or edit distance could also be used for the same purpose:

$$e(w_i) = \Delta(\hat{\phi}^{(w_i)(1)}_{m_1:m_2}, \hat{\phi}^{(w_i)(2)}_{m_1:m_2}) > \theta \tag{3.28}$$

A similar method is used in [54]. If a reliable training set of human-annotated errors exists, a classifier can be trained to detect errors based on the metrics above instead of thresholding them [297]. A number of methods combine ASR confidence approaches with native speaker similarity approaches by computing confidence measures on the candidate speech with both native and non-native trained acoustic models and using log likelihood ratio or other

similarity metrics to place words on a scale from native-resembling to non-native-resembling pronunciation [193] or as features to a pronunciation error classifier [280].

While ASR confidence approaches to error detection have been shown to be reasonably good predictors of the locations of human-annotated errors under certain experimental conditions, they are prone to considerable bias, as there are multiple factors other than speaker proficiency that might affect the confidence of a speech recogniser, particularly under realistic test conditions when not all candidates will be recorded under identical conditions. Accurate phone-level ASR is particularly difficult to guarantee with spontaneous speech, which is why most of these approaches use read text. As with native speaker comparison methods, these methods cannot generally distinguish different types of errors or provide richer feedback about detected errors beyond their location and likelihood.

3.3 Extended Recognition Networks

If proficiency is defined as adherence to canonical pronunciation, local pronunciation assessment (i.e. pronunciation error detection) becomes a matter of recognising the phones $\hat{\phi}_{1:M}$ actually spoken by the candidate and comparing them to the canonical pronunciation. An error $e(w_i)$ is detected in word w_i if its recognised corresponding phone sequence $\hat{\phi}^{(w_i)}_{m_1:m_2}$ is not one of the pronunciations in the canonical dictionary entry for the word $\mathscr{D}^{(can)}_{w_i}$.

$$e(w_i) = \hat{\phi}^{(w_i)}_{m_1:m_2} \notin \mathscr{D}^{(can)}_{w_i} \tag{3.29}$$

An utterance level error is in turn detected if any word errors are present in the utterance:

$$e(w_{1:I}) = \bigcup_{i=1}^{I} \left(\hat{\phi}^{(w_i)}_{m_1:m_2} \notin \mathscr{D}^{(can)}_{w_i} \right) \tag{3.30}$$

To obtain $\hat{\phi}^{(w_i)}_{m_1:m_2}$ for each word, the entire utterance can be aligned using a modified dictionary $\mathscr{D}^{(ERN)}$ allowing both canonical and errorful pronunciations (see Fig. 3.1).

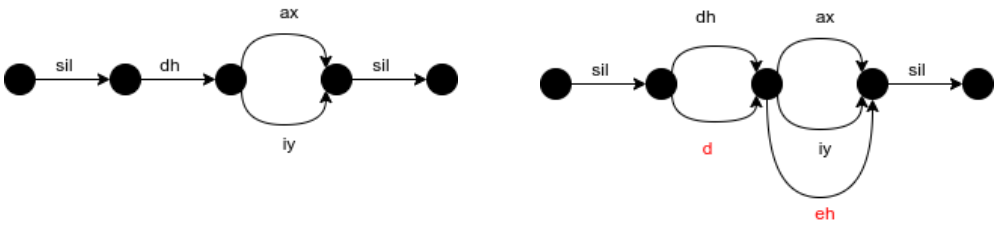


Fig. 3.1 Lattice used in forced alignment with a canonical dictionary (left) and *extended* lattice allowing both canonical and select errorful pronunciations for use in ERNs.

Alignment yields the time-aligned most likely phone sequence for the entire utterance:

$$\hat{\phi}_{1:M} = \underset{\phi_{1:M} \in \mathscr{D}_{w_{1:I}}^{(ERN)}}{\arg\max} \left\{ \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \right\} \tag{3.31}$$

$$\hat{s}_{1:T} = \underset{s_{1:T}|\hat{\phi}_{1:M}}{\arg\max}\, p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \tag{3.32}$$

where:

$$\mathscr{D}_{w_i}^{(ERN)} = \mathscr{D}_{w_i}^{(can)} \cup \mathscr{D}_{w_i}^{(err)} \tag{3.33}$$

for every word w_i, and $\mathscr{D}_{w_i}^{(err)}$ contains candidate errorful pronunciations, which must be generated beforehand for every word.

Most ERN approaches generate the candidate errorful pronunciations using phonological rules, modifying the canonical pronunciation of words based on pre-defined (usually L1-dependent) patterns of phone insertions, substitutions and deletions. Schaden et al. [238] derived sets of rules for French and German learners of English by analysing transcribed non-native speech from small numbers of speakers reading isolated words. Harrison et al. [107] similarly derived a set of rules from linguistic investigations into Cantonese accented English. However, these approaches are only narrowly applicable to the specific L1s they were developed for. In the Broad Phone Groups (BPG) approach of Kane et al., phones are substituted for other phones within the same articulatory group [134, 133]. This approach is more generally applicable but less targeted to actual L1 effects.

The above approaches detect errors caused by systematic phone effects in the speaker's accent (in this work called *accent errors*[1]) — e.g. deleting word-final vowels, confusing [b] and [v] — but cannot model errors caused by not knowing the correct phonetic pronunciation of individual words (in this work called *lexical errors*) — e.g. not knowing that the b in *subtle* is silent.

Schaden et al. [238] supplement their phonological rules with graphemic substitution rules, similarly derived from their analysis of non-native speech. However, this approach only covers one-to-one letter-to-sound substitutions and cannot model more complex relationships between spelling and pronunciation. In [220], a method for generating candidate lexical errors is presented by training a grapheme-to-phoneme converter (G2P) to predict canonical pronunciations given words' spellings and using the non-canonical pronunciations it generates as candidate errors. The intuition is that mistakes made by the G2P when guessing the pronunciation of an unknown word given only its spelling are also likely to be made by a

[1]The word *accent* here is used in the sense of 'the particular manner in which a speaker tends to pronounce the phones of a language' not with its second meaning related to stress and pitch.

non-native human attempting the same. In Li et al. [165], candidate errors were automatically learned from the canonical pronunciation and spelling of each word by a layer trained end-to-end with the error detector, in theory generating both accent and lexical errors. Accent and lexical errors are further discussed in §7.1.

Having obtained the aligned outputs $\hat{\phi}_{1:M}$ and $\hat{s}_{1:T}$, the phones corresponding to each word $\hat{\phi}_{m_1:m_2}^{(w_i)}$ are identified and looked up in the canonical dictionary $\mathscr{D}_{w_i}^{(can)}$ to determine which words were pronounced errorfully, as per Equation 3.29. By restricting the errors included in $\mathscr{D}_{w_i}^{(err)}$ to a particular type (defined as narrowly as necessary e.g. accent errors, substitutions, $dh \rightarrow d$ substitutions) detection can be performed for a single type of error at a time. Richer feedback on the type as well as the presence of errors at both the word and utterance level can thus be obtained.

Extended recognition networks are employed successfully in the literature for both word [133, 107] and utterance level [90, 169] error detection, the latter usually performed by counting or otherwise aggregating the word errors present in each utterance. Since canonical and errorful pronunciations are provided as alternatives during forced alignment, and paths through all of them considered, there is no need to freeze time stamps at their canonical values as in the case of ASR confidence methods. This allows fairer, less distorted comparison between canonical and alternative pronunciations. Another advantage of this approach is that it can be configured to detect only one type of error at a time, by controlling which errorful pronunciations are included in $\mathscr{D}_{w_i}^{(err)}$.

One of the main weaknesses of extended recognition networks is that the number of possible phone sequences $\phi_{1:M} \in \mathscr{D}_{w_{1:I}}^{(ERN)}$ in the utterance increases exponentially with the number of errorful pronunciations per word. This places a limit on the number of candidate errors that can be effectively searched for. One of the consequences of this limitation is that ERNs are generally confined to using broad transcriptions to keep the number of possible phone sequences limited. The larger phonetic alphabets of narrow transcriptions also require difficult to obtain non-standard pronunciation dictionaries as well as ground-truth error annotations in the narrow alphabet on which to test the results. Such human annotations are in turn scarcer and have been shown to be less reliable [246] as they require more precision on the part of the human annotators. The narrower the transcription, the finer the distinction between correct and incorrect pronunciations that need to be made by the annotators and the greater the room for disagreement and inconsistency.

Broad transcriptions, however, are less descriptive and can miss the distinction between correct and incorrect pronunciation, especially in non-native speech where the speaker might use phones not typically found in English [55]. The use of broad transcriptions limits the ability of ERNs to detect accent errors involving narrow phones e.g. substitution of a

canonical phone with a non-English phone. This can be tackled by expanding the phonetic alphabet to also incorporate the broad phone sets of the speakers' L1s, if these are known.

Kawai and Hirose [138] developed an early implementation of this technique, using HMM acoustic models for the phones of the source and target language trained on native speakers of each and defining candidate errorful pronunciations consisting of substituting target language phones with those of the source language. Ito et al. [129] extended this approach with more complex phonological rules, including insertions of L1 phones and deletions of L2 phones. More recently, Duan et al. [74] trained a classifier to detect a list of cross-L1 phone substitutions in a multi-task fashion together with a place of articulation classifier. In Yan et al. [293], the issue of substitution with non-English phones was instead tackled by defining generic 'anti-phone' models, which were added to the phonetic alphabet to capture any non-canonical narrow phone, in a manner agnostic to the speaker's L1.

ERNs are reliant on the procedure for generating candidate errorful pronunciations, and risk reproducing any biases present therein. The methods in the literature use 1-best outputs so, unlike pronunciation scoring, are not probabilistic and so do not provide confidence estimates or allow incorporation of prior probabilities. On the other hand, ERNs do not rely on native speaker data so aren't exposed to associated biases. The comparison of different alignment results to each other mitigates the problem of factors other than proficiency affecting ASR confidence.

3.4 Phone Recognition

To avoid the problem of exploding phone sequence combinations seen with ERNs, an alternative approach to obtain phone sequences for comparison with canonical pronunciations is to align the utterance with a canonical dictionary first:

$$\hat{\phi}_{1:M}^{(can)} = \underset{\phi_{1:M} \in \mathscr{D}_{w_{1:I}}^{(can)}}{\arg\max} \left\{ \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \right\} \tag{3.34}$$

$$\hat{s}_{1:T}^{(can)} = \underset{s_{1:T}|\hat{\phi}_{1:M}^{(can)}}{\arg\max}\, p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}^{(can)}) \tag{3.35}$$

use $\hat{s}_{1:T}^{(can)}$ to obtain the the segment $\boldsymbol{o}_{t_1:t_2}^{(w_i)}$ corresponding to each word w_i and re-align that segment only, for each word in turn:

$$\hat{\phi}_{m_1:m_2}^{(w_i)} = \underset{\phi_{1:M} \in \Phi}{\arg\max} \left\{ \sum_{s_{t_1:t_2}|\phi_{1:M}} p(\boldsymbol{o}_{t_1:t_2}^{(w_i)}, s_{t_1:t_2}|\phi_{1:M}) \right\} \tag{3.36}$$

Since the number of possible phone sequences is no longer constrained as it was when the whole utterance was aligned, it is now possible to omit the finite errorful dictionary $\mathscr{D}_{w_i}^{(err)}$ and instead allow all possible combinations of phones Φ, as illustrated in Fig. 3.2.

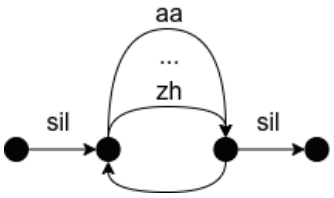


Fig. 3.2 Illustration of task of free phone recognition

As before, errors are detected by comparing each $\hat{\phi}_{m_1:m_2}^{(w_i)}$ to the canonical dictionary entry for w_i (Equation 3.29 — reproduced below):

$$e(w_i) = \hat{\phi}_{m_1:m_2}^{(w_i)} \notin \mathscr{D}_{w_i}^{(can)} \tag{3.37}$$

This technique has the advantage that it does not require the errorful pronunciations being searched for to be specified in advance and any possible errorful pronunciation can be detected. However, this comes at the expense of the distortions caused by fixing the start and end frames of the word to those obtained during the canonical alignment. As a result, detection accuracy using this method is generally low. A compromise between aligning the whole utterance and fixing word boundaries, aiming to avoid the problems with both, based on aligning the whole utterance but only allowing errorful pronunciations on one word at the time, is developed and described in §7.2.

Most phone recognition approaches in the literature use broad transcriptions to keep the complexity of the problem constrained [164, 268]. Approaches that use narrow transcriptions tackle this issue by increasing the information that can be learnt from existing data, including recognising articulatory features instead of phones [265] and training the phone recogniser in a multi-task fashion with recognition tasks in the speakers' native languages (L1s), in order to incorporate L1 effects into the training [279, 73]. These approaches require difficult to obtain narrowly transcribed pronunciation dictionaries and ground-truth evaluation data, in addition to these additional data sources (articulatory feature annotation and L1 knowledge respectively). As discussed above, human narrow phonetic transcriptions are not only difficult to obtain but also tend to yield inconsistent results [246].

The main problem with the canonical adherence approach in general is the difficulty in defining what constitutes canonical pronunciation, especially when using narrow transcriptions. This includes taking into account the variation in acceptable realisation of phonemes between different native accents the candidates might be trying to emulate, which in the case

of English are numerous and vary considerably [283]. It is also unclear whether it should be considered poor pronunciation to mix phones from different native accents or indeed to speak in a consistent non-native accent, but otherwise clearly and intelligibly.

3.5 Supervised methods

Supervised methods for error detection involve training a system in an end-to-end fashion to detect the presence of errors from input features directly. They are trained on an error-annotated training data set and learn the definition of proficient pronunciations implicitly from the annotators.

Some supervised methods build on the techniques discussed above. For example, Feng et al. [80], combine a phone recogniser with a feed-forward layer to predict canonical pronunciations and a classifier to predict the presence of errors based on the difference between them. Diment et al. [71] similarly combine forced alignment and comparison to canonical pronunciations into a single network, trained end-to-end.

Others force align speech with broad transcriptions and then feed the sequence of observation vectors $\mathbf{o}_{t_1^{(m)}:t_2^{(m)}}$ corresponding to each phone ϕ_m directly into a classifier trained to detect pronunciation errors [180, 273, 289]. The narrow phone corresponding to each broad phone and its correctness (whether this is due to phonology, intelligibility, native speaker similarity or other factors) are not assumed beforehand but are expected to be learned automatically from the human-annotated errors.

Supervised methods are entirely reliant on reliable human annotated training corpora, which are scarce and suffer problems of consistency. Such corpora have been shown to suffer reliability problems, as annotators are not consistent in what they consider to be errors, in the case of error-annotation, or which phones they annotate, in the case of phonetic trascription [172, 103] (see discussion in §3.6). Unlike ERNs and phone recognition methods, they cannot generally distinguish different types of errors.

Approaches to instead obtain a more reliable ground truth than human annotation include using motion sensors on speakers' mouths [77] and even MRI scanning [206] to directly measure articulation, but such techniques have limited practical applicability due to scarcity of training data and difficulty assigning physical metrics to pronunciation perceived by a listener.

3.6 Data

One of the major challenges in the area of pronunciation assessment is the availability and reliability of human-annotated non-native speech corpora, particularly containing spontaneous speech. Of the non-native English pronunciation error detection papers reviewed in this chapter, the vast majority collected their own datasets of non-native speech, apparently only for use in the work in question. The largest group of these consisted of small numbers (7–11) of non-native speakers reading pre-defined words or phrases [138, 290, 129, 149, 73]. The articulatory classification approaches of [77] and [206] collected data from only one native speaker each. Data from larger numbers of speakers were collected for use in the papers listed in Table 3.1 below.

Dataset	Paper(s)	Approach(es)	# Speakers	Spontaneous
Custom	[140]	phone recognition	100	no
Custom	[195]	phone recognition	90	no
Custom	[118]	confidence	60	no
Custom	[257]	confidence	60	no
Custom	[84]	confidence	206	no
Custom	[71]	ERN	120	no
ISLE	[265]	phone recognition	46	no
ATR SLT	[54]	native comparison	96	no
iCALL	[158]	native comparison	305	no
C-AuDiT	[117]	confidence	94	no
CU_CHLOE	[164, 107, 165, 220]	phone rec., ERN	210	no
L2-ARCTIC	[80, 293]	phone rec., ERN	24	no
LeaP	[154]	ERN	46	yes

Table 3.1 Non-native speech corpora used in reviewed papers on pronunciation error detection

Most of these corpora are not publicly available and have not been used in pronunciation assessment papers by more than one research group. A recent exception is the openly available L2-ARCTIC corpus [298], consisting of read speech from 24 non-native speakers, which has been used to evaluate the phone recognition approach in [80] and the anti-phone approach in [293]. All the listed large corpora consist entirely of read speech. Some papers collected spontaneous speech from small numbers of speakers [193], while Robertson et al. had 58 speakers repeat spontaneous dialogues from memory (which can be thought of as a compromise between read and spontaneous speech) [230]. An openly available corpus of phonetically annotated spontaneous speech exists in the form of LeaP [102], however no

automatic pronunciation assessment papers on this corpus could be found prior to its use in this work (see Chapter 7 and [154]).

There are a number of reasons why read speech is so overwhelmingly preferred to spontaneous speech in the laboratory environment. First, it allows the researchers to select the precise text being spoken in advance, in order to assess the pronunciation of particular words and phones in a targeted manner. The researchers can ensure the text being spoken is identical among different candidate speakers and between candidates and any native speakers, allowing for fairer comparison. Further, it gives the algorithm used for pronunciation assessment a high degree of confidence in the words being spoken at evaluation time. Finally, it has been shown to yield more reliable results than spontaneous speech in a number of pronunciation scoring tasks [256]. However, as discussed in the beginning of this chapter, spontaneous speech is more representative of the day-to-day speech a learner practically needs to be proficient in and read speech has been shown to differ significantly from it [182, 182, 4], including in terms of the numbers and types of pronunciation errors that non-native speakers make [162].

Another major challenge with manually annotated corpora is reliability. As previously discussed, it has been shown that human annotators called on to phonetically transcribe non-native speech or label pronunciation errors therein suffer from low inter-annotator agreement (percentage agreement of around 60–80% and kappa and cross-correlation values in the range 0.2–0.5) [31]. In Gut et al. [103], it was seen that annotators who agree on word and utterance level annotations disagree on phone-level annotations, while Loukina et al. [172] demonstrated that annotators tend to disagree on pronunciation error detection but agree on overall pronunciation assessment. In Kim et al. [140], it was similarly shown that annotators disagree on the location of individual pronunciation errors at the word-level but agree at the utterance-level on which utterances contain which types of pronunciation errors. While most papers reviewed in the previous sections did not cite inter-annotator agreement, those that did generally showed pictures consistent with these results. Nicolao et al. [205], for example, reported, among the three annotators who identified errors in the corpus of read speech they collected, a mean pair-wise inter-annotator agreement of 0.82 and a mean pair-wise inter-annotator cross-correlation of 0.423.

In addition to being more reliable, data for overall pronunciation assessment are easier to obtain, as it is only necessary to assign a score to the proficiency of the speaker, rather than individually annotate every word. This has allowed larger corpora to be collected, such as the 924-speaker corpus of Takai et al. [259], which includes spontaneous speech, though smaller read speech corpora are also used by some authors [51, 47]. Nonetheless, openly available corpora used by multiple groups of researchers could not be found for this task either.

3.7 Pronunciation Feedback

Reviews of the literature regarding the best paedagogical practices regarding feedback of errors during pronunciation training [199, 78] reflect a general consensus that not every pronunciation error should be fed back to the learner. High frequency of feedback has been shown to be detrimental to candidate learning and self-confidence, so both human and automatic pronunciation teachers should select only the most important errors to point out [199, 78].

It has been shown to be more effective to feed back errors which are repeated and which are common among non-native speakers generally and speakers of the candidate's L1 in particular [199, 78]. In automatic systems, only errors in which the system has the greatest confidence should be fed back. Further, less severe errors should only be fed back if the learner is of a higher proficiency [199, 78]. In terms of the way the errors are fed back, the method of *recast*, in which the error is pointed out and corrected, was found to be the least effective form of feedback, while *elicitation*, in which the student is encouraged to attempt to provide the correct pronunciation again by being assigned a new task, is the most effective [199, 78].

Given the above, it follows that effective pronunciation error detection systems should simultaneously evaluate the speaker's overall pronunciation score and their tendency to make particular types of pronunciation error at the utterance level, using them to inform which detected errors to feed back to the user and, in an adaptive learning fashion, to determine the next exercise to give the learner. Error detection systems should operate in a high precision, low recall configuration, so only high confidence detected errors are fed back to the learner. This suggests the use of an error detection framework that has the ability to separately detect different types of error at the utterance level and provide confidence estimates on its detected errors.

3.8 Comparison of Approaches

Table 3.2 compares the error detection systems reviewed in this chapter with the approach introduced in Chapter 7 of this thesis. It is seen that the majority of approaches in the literature (Karhila–Engwall) require either annotated training data or a native speaker reference making them difficult to deploy in practice. Most of the remaining methods (Neumeyer–Wei) are unable to give feedback on the type of each error. The ERN methods reviewed (van Doremalen–Tepperman) can diagnose error types without training data or references, but they are all designed for and tested on read speech, using a single L1. The work in this thesis

is therefore novel in being designed and tested for the tasks of error detection and diagnosis in spontaneous speech across multiple L1s without the need for training data or references.

System	Train-Free	Ref.-Free	Diag.	Spont.	L1 Ind.
Karhila [136]	No	No	No	No	No
Bugbol [37]	Yes	No	No	No	No
Miodonska [187]	Yes	No	No	No	No
Nicolao [205]	No	No	No	No	No
Honig [117]	No	Yes	Yes	Yes	Yes
Wei [280]	No	Yes	Yes	No	No
Chen [46]	No	Yes	No	No	No
Zhang [297]	No	No	No	No	No
Moustroufas [193]	Yes	No	No	No	No
Lin [169]	No	No	No	No	No
Duan [74]	Yes	No	Yes	No	No
Yan [293]	No	Yes	Yes	No	Yes
Thoth [268]	No	Yes	Yes	No	No
Feng [80]	No	Yes	Yes	No	Yes
Diment [71]	No	Yes	Yes	No	No
Engwall [77]	No	No	Yes	No	No
Neumeyer [201, 200]	Yes	Yes	No	Yes	No
Bernstein [21]	Yes	Yes	No	No	Yes
Kamimura [132]	Yes	Yes	No	No	No
Strik [257]	Yes	Yes	No	No	No
Duan [73]	Yes	Yes	No	No	No
Kim [140]	Yes	Yes	No	No	No
Franco [86, 85, 84]	Yes	Yes	No	No	Yes
Wei [279]	Yes	Yes	No	No	No
van Doremalen [273]	Yes	Yes	Yes	No	No
Witt [290]	Yes	Yes	Yes	No	No
Harrison [107]	Yes	Yes	Yes	No	No
Qian/Li [220, 165, 164]	Yes	Yes	Yes	No	No
Gao [90]	Yes	Yes	Yes	No	No
Kawai/Ito [138, 129]	Yes	Yes	Yes	No	No
Tepperman [265]	Yes	Yes	Yes	No	No
Chapter 7	Yes	Yes	Yes	Yes	Yes

Table 3.2 Error detection systems from the literature compared to novel system from Chapter 7 evaluated based on not needing training data (Train-Free) or native reference (Ref. Free), diagnosing the type of each error (Diag.), and use on spontaneous speech (Spont.) or speakers of multiple L1s without separate training data (L1 Ind.)

3.9 Chapter Summary

This chapter reviewed the literature on pronunciation error detection. Different approaches, based on different perspectives on the nature of good pronunciation were explored. Methods based on native speaker comparison (§3.1) were seen to be biased to irrelevant attributes of the native speaker data, while not providing feedback on types of errors. ASR confidence methods (§3.2) similarly cannot provide error type feedback and were also seen to suffer from over-reliance on the accuracy of the initial ASR output, especially when used with spontaneous speech. ERN (§3.3) and phone recognition (§3.4) methods allow word and utterance level error detection with the ability to separately detect and get feedback on different types of errors in an unsupervised manner. Supervised methods for error detection (§3.5) were seen to require labelled training data that's difficult to obtain in practice. ERNs are limited in the number of candidate errorful pronunciations that can be considered, while phone recognition methods rely excessively on the word boundaries from the initial canonical alignment.

In §3.6, the corpora of non-native speech used in the literature were reviewed. It was seen that there are no openly available corpora of annotated spontaneous speech for any of the three tasks and that the open read speech corpora for error detection that exist are not used by most papers, which instead tend to collect their own data. It was further seen that inter-annotator agreement for human annotations of localised pronunciation errors tends to be low, creating issues not only for training end-to-end error detectors but also evaluating unsupervised methods. Inter-annotator agreement was seen to be higher for utterance error detection and overall pronunciation assessment tasks.

The literature regarding feedback to the learner was briefly reviewed in §3.7. It was concluded that effective CAPT systems should be capable of overall pronunciation scoring as well as pronunciation error detection, with the system used for the latter having the ability to distinguish different types of errors and to provide confidence measures on the detected errors.

Following from this analysis, a modified alignment and canonical pronunciation comparison approach, designed to resolve the issues with ERNs, is used to investigate detection of accent and lexical errors at the word and utterance level in the presence of inconsistent human annotation on three different corpora in Chapter 7. The properties of this system compared to those of each system from the literature reviewed in this chapter are visualised in §3.8.

Chapter 4

Deep Learning

Chapters 1, 2 and 3 introduced the problems of assessing the proficiency of non-native English speakers based on pronunciation, tempo, rhythm and intonation in their spontaneous utterances. This chapter introduces deep learning techniques which will then be used in Chapters 5 and 6 to present solutions to these problems.

An overview of the concept of deep learning and neural networks and of the significance of network architecture, training criteria and optimisation is given in §4.1. In §4.2, the main categories of neural network architectures that will be used later in this thesis are presented. Training criteria that can be used for different types of supervised and unsupervised learning tasks are reviewed in §4.3. Finally, some of the key issues involved in the optimisation process, including the choice of algorithm, initialisation and learning rate modulation, are reviewed in §4.4.

4.1 Neural networks

The object of deep learning is to learn to represent complex relationships between variables involving a hierarchy of concepts from examples of those variables [97]. This is achieved by training parametric models called *neural networks*:

$$\hat{\boldsymbol{y}} = f(\boldsymbol{x}; \boldsymbol{\lambda}) \tag{4.1}$$

where $\hat{\boldsymbol{y}}$ is the model's prediction of the output variable $\boldsymbol{y}$ based on the value of the input variable $\boldsymbol{x}$ and f is a non-linear function parameterised by a set of parameters $\boldsymbol{\lambda}$. In a neural network, the function f is in turn the composition of L non-linear functions $f_{1:L}()$ called *layers*, with the aim of representing the hierarchy of concepts in the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$.

Equation 4.1 can thus be expressed as:

$$\boldsymbol{h}^{(1)} = f_1(\boldsymbol{x}; \boldsymbol{\lambda}^{(1)}) \tag{4.2}$$

$$\boldsymbol{h}^{(2)} = f_2(\boldsymbol{h}^{(1)}; \boldsymbol{\lambda}^{(2)}) \tag{4.3}$$

$$\ldots$$

$$\hat{\boldsymbol{y}} = f_L(\boldsymbol{h}^{(L-1)}; \boldsymbol{\lambda}^{(L)}) \tag{4.4}$$

where $[\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}..\boldsymbol{\lambda}^{(L)}] = \boldsymbol{\lambda}$ and $\boldsymbol{h}^{(1)}...\boldsymbol{h}^{(L-1)}$ are intermediate representations known as *hidden states*. The optimal values $\hat{\boldsymbol{\lambda}}$ of $\boldsymbol{\lambda}$ are to be learned from examples in training data $\mathscr{S}_{train}$ to best capture the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$.

If examples of both $\boldsymbol{x}$ and $\boldsymbol{y}$ are available in the training data set $\mathscr{S}_{train} = \{\boldsymbol{x}^{(1:N)}, \boldsymbol{y}^{(1:N)}\}$ and the object is to learn the relationship between the two so as to be able to predict unknown values of $\boldsymbol{y}$ from known values of $\boldsymbol{x}$ in the future (e.g. training a system to assign a proficiency grade to non-native utterances using examples of utterances that have been graded by humans), the task is called *supervised learning*. If only examples of $\boldsymbol{x}$ are available and the goal is to learn a transformation to a different (e.g. more compact) representation of the data $\boldsymbol{y}$, so as to capture useful properties of the structure of the distribution of $\boldsymbol{x}$, the task is called *unsupervised learning*.

The exact form of the function f and its constituent layers, known as the *architecture* of the neural network, is designed depending on the structure of the data $\boldsymbol{x}$ and to reflect prior knowledge about the nature of the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$.

Having defined the network architecture and depending on the nature of the task, a *training criterion*, consisting of a cost function $C(\boldsymbol{\lambda}, \mathscr{S}_{train})$, must be defined to optimise $\boldsymbol{\lambda}$. The training criterion defines the properties that the learned function $f(\boldsymbol{x}; \hat{\boldsymbol{\lambda}})$ is desired to have. For supervised learning, this usually includes ensuring the outputs $\hat{\boldsymbol{y}}^{(n)} = f(\boldsymbol{x}^{(n)}; \boldsymbol{\lambda})$ predicted by the network for each input example $\boldsymbol{x}^{(n)}$ closely match the supplied outputs $\boldsymbol{y}^{(n)}$. For unsupervised learning, it usually includes $\hat{\boldsymbol{y}}$ capturing as much information about $\boldsymbol{x}$ as possible, often measured by how closely each $\hat{\boldsymbol{y}}^{(n)}$ can be used to reconstruct $\hat{\boldsymbol{x}}^{(n)}$. Another common goal is for the model to generalise i.e. avoid overfitting the training data such that it can't make predictions for unseen inputs [28, 97].

Given a model architecture and cost, a suitable *optimisation* algorithm is used to determine the optimal parameters $\hat{\boldsymbol{\lambda}}$:

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}} C(\boldsymbol{\lambda}, \mathscr{S}_{train}) \tag{4.5}$$

4.2 Neural Network Architectures

The architecture of a neural network defines the form of the function that connects its inputs to its outputs given its parameters. It is chosen given the nature of the task and the structure of the input and output data. The choice of architecture determines the parameters that need to be optimised and may itself be determined by certain hyperparameters. The layered nature of neural networks means that architectural components can be combined sequentially or hierarchically to create larger networks.

This section reviews four categories of neural network architectures designed to deal with different types of tasks which will arise in this thesis. Feed-forward networks (§4.2.1) map unstructured vector inputs to other unstructured vector outputs $\boldsymbol{x} \rightarrow \boldsymbol{y}$. Recurrent neural networks (§4.2.2) map sequences to either fixed-length vectors or other sequences ($\boldsymbol{x}_{1:T} \rightarrow \boldsymbol{y}$ and $\boldsymbol{x}_{1:T} \rightarrow \boldsymbol{y}_{1:T}$). Attention mechanisms (§4.2.3) can be used for these two tasks as well as to compress collections of items of irrelevant order and varying importance to a fixed-length representation ($\boldsymbol{x}_{1:N} \rightarrow \boldsymbol{y}$). Finally, Siamese networks (§4.2.4) project pairs of inputs to scalar values indicative of the distance between them ($\{\boldsymbol{x}_1, \boldsymbol{x}_2\} \rightarrow y$).

4.2.1 Feed-forward networks

In feed-forward networks, each layer f_l of the network defines each hidden state as a non-linear function of its predecessor:

$$\boldsymbol{h}^{(l)} = f_l(\boldsymbol{h}^{(l-1)}; \boldsymbol{\lambda}^{(l)}) \tag{4.6}$$

such that each element of $\boldsymbol{h}^{(l)}$ only depends on $\boldsymbol{h}^{(l-1)}$ and $\boldsymbol{\lambda}^{(l)}$ and not on other elements of $\boldsymbol{h}^{(l)}$ (i.e. there are no cyclic connections) [239].

The simplest form of f_l is the *fully-connected* layer. This approach considers the input $\boldsymbol{x}$, output $\boldsymbol{y}$ and hidden states $\boldsymbol{h}^{(1:L)}$ to be simple vectors with no further internal structure. It consists of pre-multiplication by a matrix of weights $\boldsymbol{W}^{(l)}$, addition of a bias $\boldsymbol{b}^{(l)}$ and application of a non-linear transformation (known as the activation function) σ_a:

$$\boldsymbol{h}^{(l)} = \sigma_a(\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)}) \tag{4.7}$$

such that the parameters to be learned for the layer are $\boldsymbol{\lambda}^{(l)} = \{\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}\}$. In addition to the choice of σ_a, an important *hyperparameter* of the fully-connected layer is the size of $\boldsymbol{h}^{(l)}$, which in turn determines the sizes of $\boldsymbol{W}^{(l)}$, $\boldsymbol{W}^{(l+1)}$, and $\boldsymbol{b}^{(l)}$.

A fully-connected feed-forward neural network with one or more hidden layers is called a multi-layer perceptron (MLP) or a deep neural network (DNN).

The form $\hat{\boldsymbol{y}} = f_{DNN}(\boldsymbol{x}; \{\boldsymbol{W}^{(1:L)}, \boldsymbol{b}^{(1:L)}\})$ of a DNN is given by:

$$\boldsymbol{h}^{(1)} = \sigma_a(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) \tag{4.8}$$

$$\boldsymbol{h}^{(l)} = \sigma_a(\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)}) \quad 1 < l < L \tag{4.9}$$

$$\hat{\boldsymbol{y}} = \sigma_a(\boldsymbol{W}^{(L)}\boldsymbol{h}^{(L-1)} + \boldsymbol{b}^{(L)}) \tag{4.10}$$

Given a sufficiently large hidden layer and sufficient amounts of training data, a one-hidden-layer DNN should be able to learn to model any arbitrary function (i.e. it is a universal function approximator) [29] as long as σ_a is a piecewise continuous, non-polynomial, locally bounded function [161].

DNNs with more hidden layers are generally considered to be more efficient function approximators (the *deep learning hypothesis*) [20]. The number of hidden layers and the size of each hidden layer of a DNN are hyperparameters which must be tuned to optimally solve each task.

In addition to the requirements of non-polynomiality and continuity, the choice of σ_a is primarily constrained by the gradient descent methods used for optimisation (see §4.4.1), which require it to be differentiable and conducive to efficient convergence. Early approaches [156, 29] tended to use the sigmoid and hyperbolic tangent functions:

$$\sigma_a^{(sigmoid)}(x) = \frac{1}{1+e^{-x}} \tag{4.11}$$

$$\sigma_a^{(tanh)}(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{4.12}$$

With the rise of DNNs, it was seen that both these functions suffer from saturation issues, as they both become flat at extreme values, causing convergence to slow down when activations reach these values, a problem that becomes worse as the networks become deeper [94]. This problem was ameliorated with the introduction of the Rectified Linear Unit (ReLU) [197], which continues increasing linearly however extreme positive activations become:

$$\sigma_a^{(ReLU)}(x) = \max(0, x) \tag{4.13}$$

The protection against saturation can be extended to negative values of activation by using modified ReLUs such as the Leaky Rectified Linear Unit (LReLU) [178]:

$$\sigma_a^{(LReLU)}(x) = \max(\alpha x, x) \qquad 0.0 < \alpha < 1.0 \tag{4.14}$$

through ReLUs remain the dominant activation function in the literature [97, 26, 267].

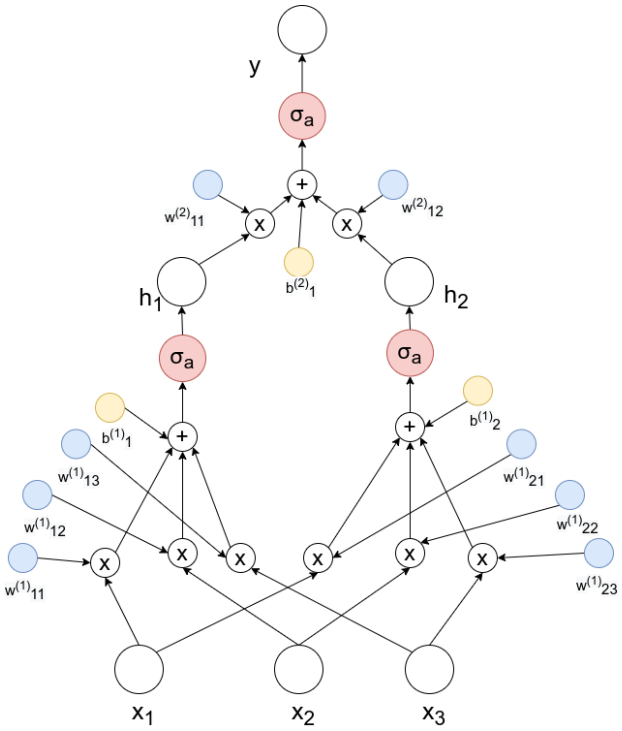


Fig. 4.1 Illustration of a simple neural network mapping a length-3 input vector to a scalar output with one hidden layer of size 2. Training involves tuning the values of the weights and biases in order to capture the relationship between x and y.

In tasks where the input is known to have a two or three dimensional grid-like structure (e.g. image data), *convolutional layers* [156] can be used instead, where each 2D slice is convolved with a set of kernels before passing through the activation function. Feed-forward networks with convolutional layers are known as convolutional neural networks (CNNs).

4.2.2 Recurrent Neural Networks

Recurrent neural networks (RNNs) are employed for tasks where the input $\boldsymbol{x}_{1:T}$ consists of a sequence of vectors $\boldsymbol{x}_t$, such that the meaning of each vector is dependent upon those preceding it (unidirectional) or those both preceding and following it (bidirectional) in the sequence. They can be used for tasks where the output $\boldsymbol{y}_{1:T}$ is also sequential or for sequence-to-vector tasks (seq2vec), where the output $\boldsymbol{y}_{out}$ is a fixed-length vector.

In a recurrent layer, the hidden state $\boldsymbol{h}_t^{(l)}$ at each position t of each layer l is not just dependent on the hidden state of the previous layer $\boldsymbol{h}_t^{(l-1)}$ at the same position, but also of the previous position at the same layer $\boldsymbol{h}_{t-1}^{(l)}$.

$$\boldsymbol{h}_t^{(l)} = f_l(\boldsymbol{h}_t^{(l-1)}, \boldsymbol{h}_{t-1}^{(l)}; \boldsymbol{\lambda}^{(l)}) \tag{4.15}$$

The output layer is usually an exception, each vector depending only on the final hidden layer at the same position. The form $\hat{\boldsymbol{y}}_{1:T} = f_{RNN}(\boldsymbol{x}_{1:T}; \boldsymbol{\lambda}^{(1:L)})$ of a uni-directional RNN (Figure 4.2, middle) is thus:

$$\boldsymbol{h}_t^{(1)} = f_l(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}^{(1)}; \boldsymbol{\lambda}^{(1)}) \tag{4.16}$$

$$\boldsymbol{h}_t^{(l)} = f_l(\boldsymbol{h}_t^{(l-1)}, \boldsymbol{h}_{t-1}^{(l)}; \boldsymbol{\lambda}^{(l)}) \quad 1 < l < L \tag{4.17}$$

$$\boldsymbol{y}_t = f_l(\boldsymbol{h}_t^{(L-1)}; \boldsymbol{\lambda}^{(L)}) \tag{4.18}$$

In cases where a fixed-length output $\boldsymbol{y}_{out}$ is needed instead of a sequential output $\boldsymbol{y}_{1:T}$, the output is predicted from the last position of the last hidden layer $\boldsymbol{h}_T^{(L-1)}$, as it is a function of all previous positions. Equation 4.18 is thus replaced by:

$$\hat{\boldsymbol{y}}_{out} = f_l(\boldsymbol{h}_T^{(L-1)}; \boldsymbol{\lambda}^{(L)}) \tag{4.19}$$

In cases where the output is a sequence which is not necessarily aligned to the input sequence (e.g. neural machine translation) an encoder-decoder sequence-to-sequence RNN can be used instead, mapping from a sequence to a vector and back to a sequence [258], as illustrated in Figure 4.2, right.

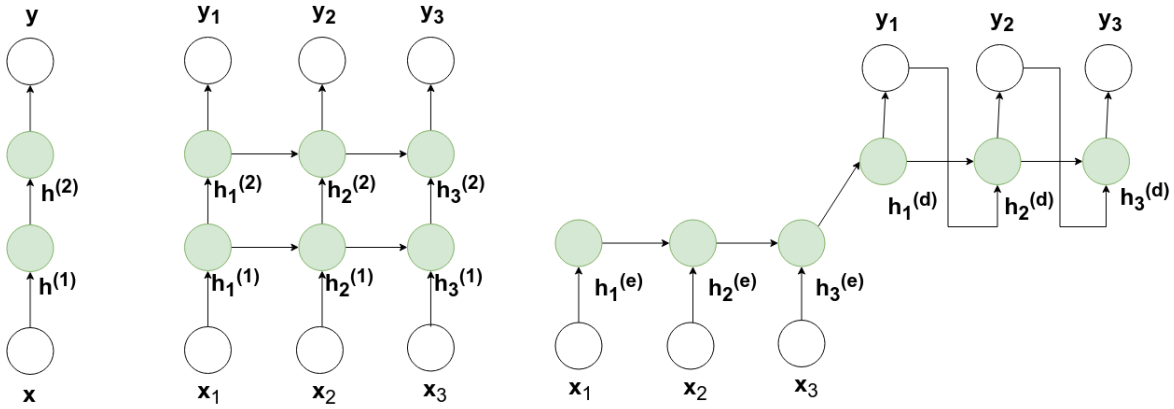


Fig. 4.2 Illustration of a a two-hidden-layer DNN mapping a vector to a vector (left), a uni-directional two-hidden-layer RNN mapping a length-3 input sequence to a length-3 output sequence (middle), and a uni-directional encoder-decoder RNN with one-hidden layer each encoder and decoder mapping a length-3 input sequence to a length-3 output sequence (right)

In a bi-directional RNN, introduced by Schuster and Paliwal [240], two recurrent neural networks are combined: one where each hidden unit is a function of its previous position and one where it is a function of the next position. The input is passed through both and the hidden state vectors of the final layers of each are concatenated to obtain the output sequence. The combined network $\hat{\boldsymbol{y}}_{1:T} = f_{biRNN}(\boldsymbol{x}_{1:T}; \{\boldsymbol{\lambda}^{(1:L)}, \boldsymbol{\lambda}'^{(1:L)}\})$ is thus given by:

$$\boldsymbol{h}_t^{(1)} = f_l(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}^{(1)}; \boldsymbol{\lambda}^{(l)}) \tag{4.20}$$

$$\boldsymbol{h}_t'^{(1)} = f_l(\boldsymbol{x}_t, \boldsymbol{h}_{t+1}'^{(1)}; \boldsymbol{\lambda}'^{(l)}) \tag{4.21}$$

$$\boldsymbol{h}_t^{(l)} = f_l(\boldsymbol{h}_t^{(l-1)}, \boldsymbol{h}_{t-1}^{(l)}; \boldsymbol{\lambda}^{(l)}) \qquad 1 < l < L \tag{4.22}$$

$$\boldsymbol{h}_t'^{(l)} = f_l(\boldsymbol{h}_t'^{(l-1)}, \boldsymbol{h}_{t+1}'^{(l)}; \boldsymbol{\lambda}'^{(l)}) \quad 1 < l < L \tag{4.23}$$

$$\boldsymbol{y}_t = f_l([\boldsymbol{h}_t^{(L-1)}, \boldsymbol{h}_t'^{(L-1)}]; \boldsymbol{\lambda}^{(L)}) \tag{4.24}$$

For a fixed-length output, Equation 4.24 becomes:

$$\hat{\boldsymbol{y}}_{out} = f_l([\boldsymbol{h}_T^{(L-1)}, \boldsymbol{h}_1'^{(L-1)}]; \boldsymbol{\lambda}^{(L)}) \tag{4.25}$$

as $\boldsymbol{h}_1'^{(L-1)}$ is a function of the entire backwards sequence and $\boldsymbol{h}_T^{(L-1)}$ is a function of the entire forward sequence.

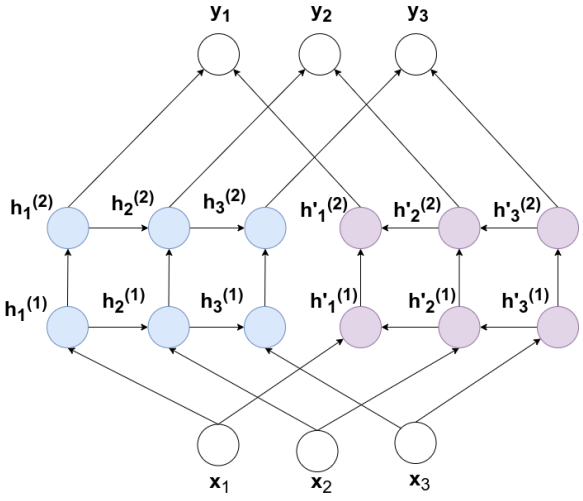


Fig. 4.3 Illustration of a two hidden layer bidirectional RNN for a length-3 input sequence

In traditional RNNs, the form of f_l follows that of fully connected layers:

$$\boldsymbol{h}_t^{(l)} = \sigma_a(\boldsymbol{V}^{(l)}\boldsymbol{h}_t^{(l-1)} + \boldsymbol{W}^{(l)}\boldsymbol{h}_{t-1}^{(l)} + \boldsymbol{b}^{(l)}) \quad \boldsymbol{\lambda}^{(l)} = \{\boldsymbol{V}^{(l)}, \boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}\} \tag{4.26}$$

A key weakness of this method is its difficulty in learning long-term dependencies, as any effect of an early position on a late position will either decay to zero or explode as it is propagated through all the intermediate hidden states [113, 101].

To tackle this issue, Hochreiter et al. [114] introduced a modified recurrent layer called Long Short Term Memory (LSTM) designed to learn what to remember and what to forget at each stage. To that end a second hidden state at each position called the memory cell $\boldsymbol{c}_t$ is introduced. As above, $\boldsymbol{h}_t^{(l-1)}$ and $\boldsymbol{h}_{t-1}^{(l)}$ are used to compute an update state $\tilde{\boldsymbol{h}}_t^{(l)}$:

$$\tilde{\boldsymbol{h}}_t^{(l)} = \sigma_a(\boldsymbol{V}^{(l)}\boldsymbol{h}_t^{(l-1)} + \boldsymbol{W}^{(l)}\boldsymbol{h}_{t-1}^{(l)} + \boldsymbol{b}^{(l)}) \tag{4.27}$$

However, rather than this becoming the new hidden state, three element-wise gating functions $\boldsymbol{i}_t^{(l)}$, $\boldsymbol{f}_t^{(l)}$, and $\boldsymbol{o}_t^{(l)}$ are also computed, respectively representing how much of each element of $\tilde{\boldsymbol{h}}_t^{(l)}$ to *input* to the memory cell, how much of each element of the contents of the memory cell at the previous position $\boldsymbol{c}_{t-1}^{(l)}$ to *forget*, and how much of each element of the contents of the memory cell at the current position $\boldsymbol{c}_t^{(l)}$ to *output* to the current hidden state $\boldsymbol{h}_t^{(l)}$.

The value of each gate is projected via a fully connected feed-forward layer from $\boldsymbol{h}_t^{(l-1)}$ and $\boldsymbol{h}_{t-1}^{(l)}$, using sigmoid activation functions to constrain each element of each gate to be between 0 and 1:

$$\boldsymbol{i}_t^{(l)} = \sigma_a^{(sigmoid)}(\boldsymbol{T}^{(l)}\boldsymbol{h}_t^{(l-1)} + \boldsymbol{U}^{(l)}\boldsymbol{h}_{t-1}^{(l)} + \boldsymbol{c}^{(l)}) \tag{4.28}$$

$$\boldsymbol{f}_t^{(l)} = \sigma_a^{(sigmoid)}(\boldsymbol{R}^{(l)}\boldsymbol{h}_t^{(l-1)} + \boldsymbol{S}^{(l)}\boldsymbol{h}_{t-1}^{(l)} + \boldsymbol{d}^{(l)}) \tag{4.29}$$

$$\boldsymbol{o}_t^{(l)} = \sigma_a^{(sigmoid)}(\boldsymbol{P}^{(l)}\boldsymbol{h}_t^{(l-1)} + \boldsymbol{Q}^{(l)}\boldsymbol{h}_{t-1}^{(l)} + \boldsymbol{e}^{(l)}) \tag{4.30}$$

These gates are then applied by element-wise multiplication:

$$\boldsymbol{c}_t^{(l)} = \boldsymbol{f}_t^{(l)} \odot \boldsymbol{c}_{t-1}^{(l)} + \boldsymbol{i}_t^{(l)} \odot \tilde{\boldsymbol{h}}_t^{(l)} \tag{4.31}$$

$$\boldsymbol{h}_t^{(l)} = \boldsymbol{f}_t^{(o)} \odot \sigma_a(\boldsymbol{c}_t^{(l)}) \tag{4.32}$$

The LSTM network is illustrated in Figure 4.4 below.

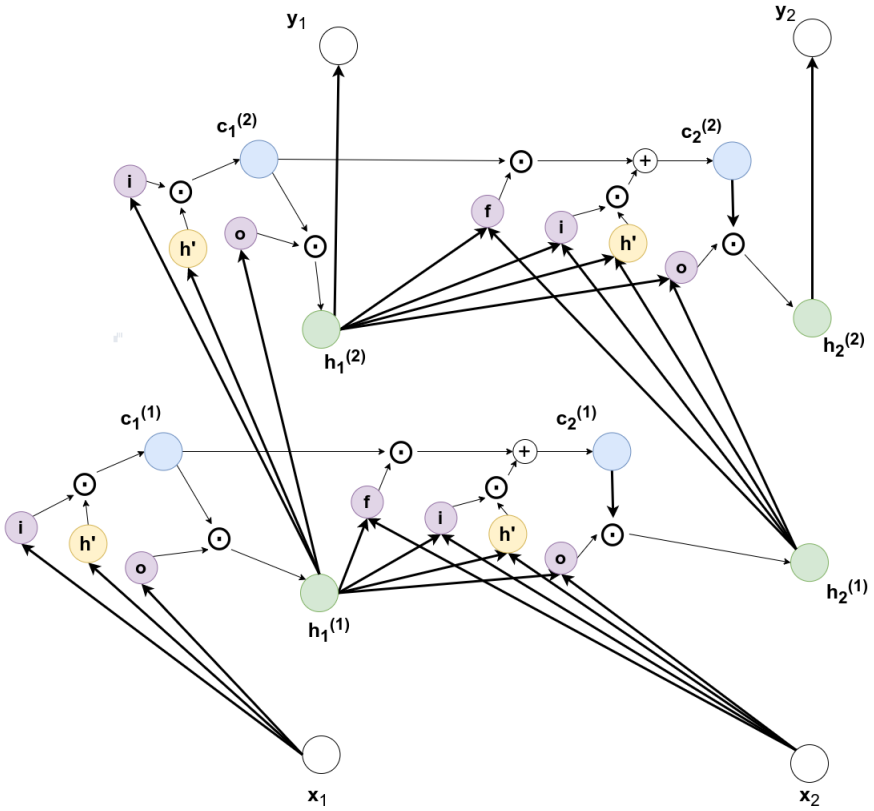


Fig. 4.4 Illustration of a two-hidden layer LSTM for a length-2 sequence. Bold lines indicate application of weights, then bias and a non-linearity where the arrows meet.

A more recent modification to the recurrent layer, proven useful in some applications, is the Gated Recurrent Unit (GRU), introduced by Cho et al. [52], which is not used in this thesis.

4.2.3 Attention Mechanisms

Attention mechanisms are an architectural component used to compress a variable-length input $\boldsymbol{x}_{1:N}$ to a fixed length output $\boldsymbol{y}$, based on assessing the relative salience to $\boldsymbol{y}$ of each $\boldsymbol{x}_n$ [12]. This is achieved by computing a weighted sum of the inputs $\boldsymbol{x}_n$ with corresponding weights a_n, obtained by normalising representations s_n of the salience of each $\boldsymbol{x}_n$ to $\boldsymbol{y}$ so that they sum to one:

$$\hat{\boldsymbol{y}} = \sum_{n=1}^{N} a_n \boldsymbol{x}_n \quad \text{where} \quad a_n = \frac{\exp(s_n)}{\sum_{m=1}^{N} \exp(s_m)} \tag{4.33}$$

In the simplest case, the representations s_n are obtained as a function of their corresponding $\boldsymbol{x}_n$, such that Equation 4.33 becomes:

$$\hat{\boldsymbol{y}} = \sum_{n=1}^{N} \frac{\exp(f_{att}(\boldsymbol{x}_n; \boldsymbol{\lambda}_{att})}{\sum_{m=1}^{N} \exp(f_{att}(\boldsymbol{x}_m; \boldsymbol{\lambda}_{att}))} \boldsymbol{x}_n \tag{4.34}$$

where f_{att} is usually a fully-connected layer:

$$f_{att}(\boldsymbol{x}_n; \boldsymbol{\lambda}_{att}) = \boldsymbol{u}^T \sigma_a(\boldsymbol{V}\boldsymbol{x}_n + \boldsymbol{b}) \qquad \boldsymbol{\lambda}_{att} = \{\boldsymbol{u}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{b}\} \tag{4.35}$$

The network is thus able to learn to focus on the parts of $\boldsymbol{x}_{1:N}$ that have the most salience to the output. Unlike RNNs, attention mechanisms without explicit positional encoding are agnostic to the order of the items in $\boldsymbol{x}_{1:N}$ and would be unchanged if their positions were to be shuffled.

A simple attention mechanism as described by Equations 4.34 and 4.35 above is illustrated in Figure 4.5.

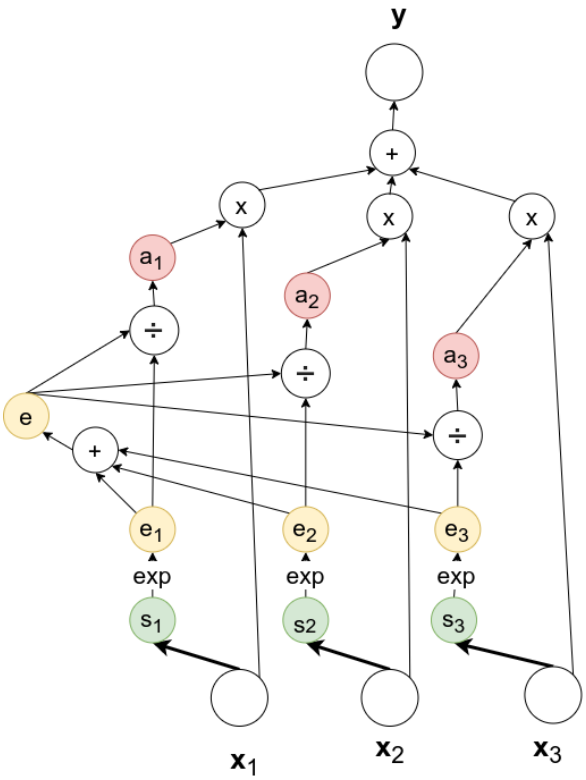


Fig. 4.5 Illustration of a simple attention mechanism. Each bold line indicates a fully connected layer.

One use of this type of mechanism is to map the hidden states of the final hidden layer of an RNN to a fixed-length vector, as an alternative to using the final position, replacing Equation 4.25 with:

$$\hat{\boldsymbol{y}}_{out} = \sum_{t=1}^{T} \frac{\exp(f_{att}(\boldsymbol{h}_t^{(L-1)}; \boldsymbol{\lambda}_{att})}{\sum_{s=1}^{T} \exp(f_{att}(\boldsymbol{h}_s^{(L-1)}; \boldsymbol{\lambda}_{att}))} \boldsymbol{h}_t^{(L-1)} \tag{4.36}$$

This is illustrated, alongside a traditional RNN, in Figure 4.6.

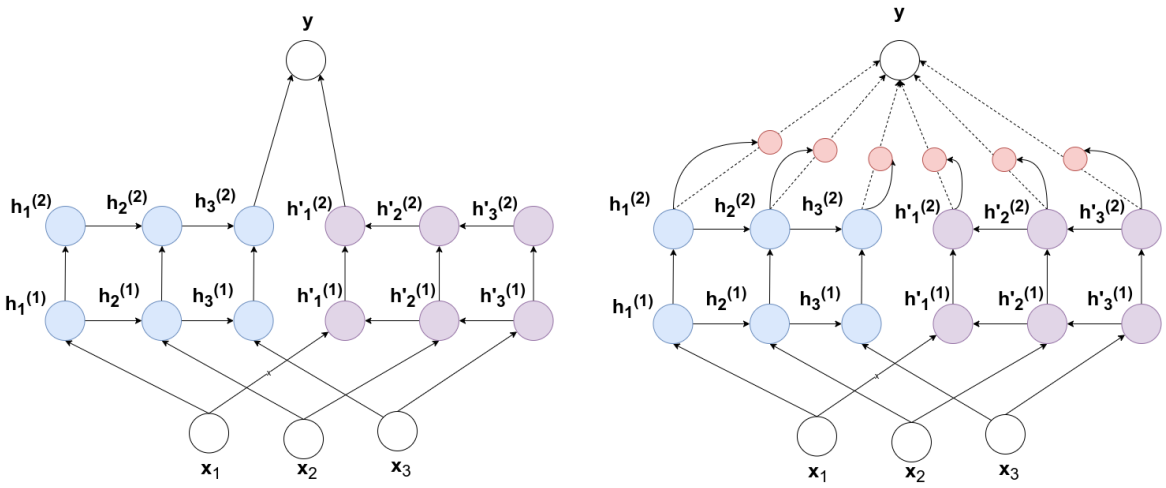


Fig. 4.6 Two forms of sequence to vector transformation using a bi-directional RNN: projecting the hidden state vectors at the final positions of the last hidden layer in each direction (left) and attending over all vectors on the final hidden layer (right). Dotted lines indicate attention, the red circles represent attention weights normalised against each other.

Attention mechanisms are most commonly employed with key vectors $\boldsymbol{k}$, in applications involving alignment of an input and output sequence, such as neural machine translation. At each position in the output, it is desired to focus on parts of the input the context of which is similar to that of the output. A context representation $\boldsymbol{k}_k$ at each output position k is thus used as a key to attend over the input sequence $\boldsymbol{x}_{1:N}$:

$$\hat{\boldsymbol{y}}_k = \sum_{n=1}^{N} \frac{\exp(f_{att}(\boldsymbol{x}_n, \boldsymbol{k}_k, \boldsymbol{\lambda}_{att}))}{\sum_{m=1}^{N} \exp(f_{att}(\boldsymbol{x}_m, \boldsymbol{k}_k, \boldsymbol{\lambda}_{att}))} \boldsymbol{x}_n \tag{4.37}$$

The simplest form for f_{att} is additive attention [12], which uses a fully-connected layer:

$$f_{att}(\boldsymbol{x}_n, \boldsymbol{k}_k, \boldsymbol{\lambda}_{att}) = \boldsymbol{u}^T \sigma_a(\boldsymbol{V}\boldsymbol{x}_n + \boldsymbol{W}\boldsymbol{k}_k + \boldsymbol{b}) \qquad \boldsymbol{\lambda}_{att} = \{\boldsymbol{u}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{b}\} \tag{4.38}$$

Two alternatives to additive attention introduced by Luong et al. [177], known as multiplicative attention, are more constrained to measure the similarity of $\boldsymbol{x}_n$ and $\boldsymbol{k}_k$, and are faster and more space-efficient to implement in practice:

$$f_{att}(\boldsymbol{x}_n, \boldsymbol{k}_k, \boldsymbol{\lambda}_{att}) = \boldsymbol{x}_n^T \boldsymbol{k}_k \tag{4.39}$$

$$f_{att}(\boldsymbol{x}_n, \boldsymbol{k}_k, \boldsymbol{\lambda}_{att}) = \boldsymbol{x}_n^T \boldsymbol{W} \boldsymbol{k}_k \qquad \boldsymbol{\lambda}_{att} = \{\boldsymbol{W}\} \tag{4.40}$$

In Vaswani et al. [274], a variation called scaled dot product attention was introduced to mitigate problems caused by s_n becoming too large when the vectors are too long:

$$f_{att}(\boldsymbol{x}_n, \boldsymbol{k}_k, \boldsymbol{\lambda}_{att}) = \frac{\boldsymbol{x}_n^T \boldsymbol{k}_t}{\sqrt{\dim(\boldsymbol{x}_n)}} \tag{4.41}$$

Vaswani et al. also showed how attention mechanisms could be used for sequence-to-sequence modelling as an alternative to RNNs. First, the position n of each item in $\boldsymbol{x}_{1:N}$ must be encoded and the attention mechanism made dependent on it, so the order of items is taken into account. This is achieved by adding a positional encoding θ_{ni} to each element x_{ni} of each length-I item $\boldsymbol{x}_n$ before passing it through the attention mechanism:

$$\tilde{\boldsymbol{x}}_n = \boldsymbol{x}_n + \boldsymbol{\theta}_n \tag{4.42}$$

where:

$$\theta_{ni} = \begin{cases} \sin\left(n\left((10000^{-\frac{i}{I}}\right)\right) & i \bmod 2 = 0 \\ \cos\left(n\left((10000^{-\frac{i-1}{I}}\right)\right) & i \bmod 2 = 1 \end{cases} \tag{4.43}$$

The modified input $\tilde{\boldsymbol{x}}_{1:N}$ is now aligned to itself (*self-attention*) to obtain output $\boldsymbol{y}_{1:N}$:

$$\hat{\boldsymbol{y}}_k = \sum_{n=1}^{N} \frac{\exp(f_{att}(\tilde{\boldsymbol{x}}_n, \tilde{\boldsymbol{x}}_k, \boldsymbol{\lambda}_{att}))}{\sum_{m=1}^{N} \exp(f_{att}(\tilde{\boldsymbol{x}}_m, \tilde{\boldsymbol{x}}_k, \boldsymbol{\lambda}_{att}))} \tilde{\boldsymbol{x}}_n \tag{4.44}$$

such that $\boldsymbol{y}_k$ is a weighted sum of all $\tilde{\boldsymbol{x}}_{1:N}$ weighted by a measure of their similarity to $\tilde{\boldsymbol{x}}_k$.

The positional encoding means that $\boldsymbol{y}_k$ will be more affected by positions close to k than further away, as with an RNN. However, the attention mechanism is also able to learn dependencies with any other position n, no matter how distant, based on both n itself and the value of $\boldsymbol{x}_n$. By not requiring information to pass through intermediate positions, this approach thus goes further than the LSTM in resolving the long-term dependency problem.

Finally, Vaswani et al. introduced *multi-head attention*, whereby attention is run in parallel H times with different weights and the results concatenated and passed through a

feed-forward layer. Each set of attention weights is thus allowed to specialise on extracting a different type of salience information:

$$\boldsymbol{h}_k^{(h)} = \sum_{n=1}^{N} \frac{\exp(f_{att}(\tilde{\boldsymbol{x}}_n, \tilde{\boldsymbol{x}}_k, \boldsymbol{\lambda}_{att}^{(h)}))}{\sum_{m=1}^{N} \exp(f_{att}(\tilde{\boldsymbol{x}}_m, \tilde{\boldsymbol{x}}_k, \boldsymbol{\lambda}_{att}^{(h)}))} \tilde{\boldsymbol{x}}_n \tag{4.45}$$

$$\hat{\boldsymbol{y}}_k = f_{out}(\{\boldsymbol{h}_k^{(1)} ... \boldsymbol{h}_k^{(H)}\}, \boldsymbol{\lambda}_{out}) \tag{4.46}$$

The combined multi-head self-attention model with scaled dot product attention (Equations 4.41, 4.42, 4.45 and 4.46) is called a *transformer*:

$$\hat{\boldsymbol{y}}_{1:T} = f_{TF}(\boldsymbol{x}_{1:T}, \boldsymbol{\lambda}) \qquad \boldsymbol{\lambda} = \{\boldsymbol{\lambda}_{att}^{(1:H)}, \boldsymbol{\lambda}_{out}\} \tag{4.47}$$

An example is illustrated alongside a bi-RNN and single-head attention in Figure 4.7.

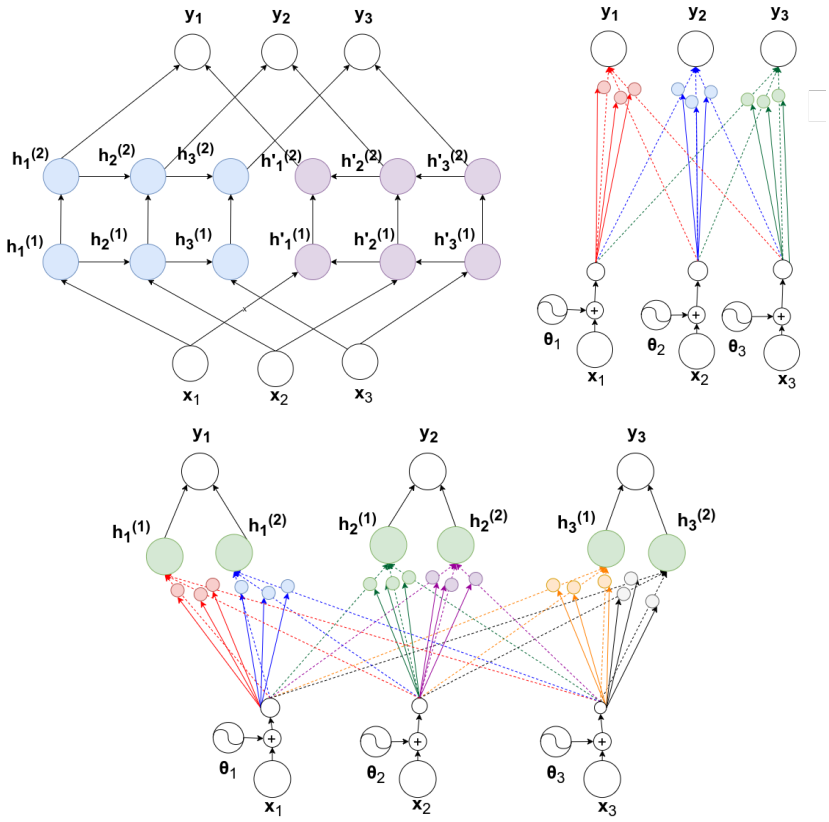


Fig. 4.7 Sequence-to-sequence mapping using a bi-RNN (top left), single-head self-attention (top right) and multi-head self-attention (bottom). Dotted lines of the same colour represent an attention mechanism. Coloured circles on the dotted lines represent attention weights. Solid lines leading to circles indicate the key used together with the value at the source of the dotted line to derive attention weights.

Transformers can themselves in turn be stacked as layers to produce a multi-layer transformer model:

$$\boldsymbol{h}_{1:T}^{(1)} = f_{TF}(\boldsymbol{x}_{1:T}, \boldsymbol{\lambda}^{(1)}) \tag{4.48}$$

$$\boldsymbol{h}_{1:T}^{(l)} = f_{TF}(\boldsymbol{h}_{1:T}^{(l-1)}, \boldsymbol{\lambda}^{(l)}) \qquad 1 < l < L \tag{4.49}$$

$$\hat{\boldsymbol{y}}_{1:T} = f_{TF}(\boldsymbol{h}_{1:T}^{(L-1)}, \boldsymbol{\lambda}^{L}) \tag{4.50}$$

An important recent application of multi-layer transformer models is the Bidirectional Encoder Representations from Transformers (BERT) language model [69]. The long-term dependency, in-built bi-directionality and computational efficiency of self-attention gives this model significant advantages over LSTMs.

In broad terms, the input to BERT is a tokenised word sequence $t_{1:T}$, with one or more words replaced by a missing token and the output is the same word sequence with all words filled in. The system is pre-trained on long sequences of text with words randomly marked as missing, so that it learns to predict missing words given both previous and subsequent words. Once trained, it can be directly used as a language model, to predict the next words in a sequence (by inputting the known words followed by missing tokens). Before use, it can be fine-tuned for this specific task, by training it with examples from the desired domain.

Pre-trained BERT can also be used as a word-embedder, by entering a word sequence as input and retrieving the values of the last layer $\boldsymbol{h}_{1:T}^{(l)}$. As these vectors have been trained to be able to predict their respective and nearby words, they serve as powerful vector representations of the identity and context of the word and word sequence. It is also possible to replace the output layer and fine-tune the combined model for any other NLP task (e.g. sentence classification). BERT can also be trained for question answering or other text pair tasks by feeding the tokenised question and answer concatenated as the input $t_{1:T}$. By learning to predict parts of the question and answer given the rest of both, it learns the relationship between the two and can thus be later fine-tuned to predict the answer given the question, in the same way as in filling in sentences.

4.2.4 Siamese Networks

Siamese networks are used for tasks with pairwise input data $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2\}$, where the output y is dependent on the extent of similarity or difference $\Delta(\boldsymbol{x}_1, \boldsymbol{x}_2)$ between the members of the pair rather than on the individual inputs themselves (e.g. if $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are images of two signatures and the task is to determine whether they belong to the same person) [36].

A Siamese network is composed of two copies of the same neural network, with tied parameters, each fed with one of the elements of a pair of input samples. These identical networks project the samples into a common embedding space:

$$\boldsymbol{h}_1 = f(\boldsymbol{x}_1, \boldsymbol{\lambda}_{emb}) \tag{4.51}$$

$$\boldsymbol{h}_2 = f(\boldsymbol{x}_2, \boldsymbol{\lambda}_{emb}) \tag{4.52}$$

A measure of distance is then computed between the two embeddings, usually Euclidean:

$$d = |\boldsymbol{h}_2 - \boldsymbol{h}_1| \tag{4.53}$$

and used to predict the output, which could be a scalar quantity (for a regression task) or the probability of a match (for a binary classification task):

$$y = f_{out}(d, \boldsymbol{\lambda}_{out}) \tag{4.54}$$

An LSTM Siamese architecture for learning difference metrics between pairs of variable length sequences was presented in [194] for use with pairs of sentences. The concept is illustrated in Figure 4.8

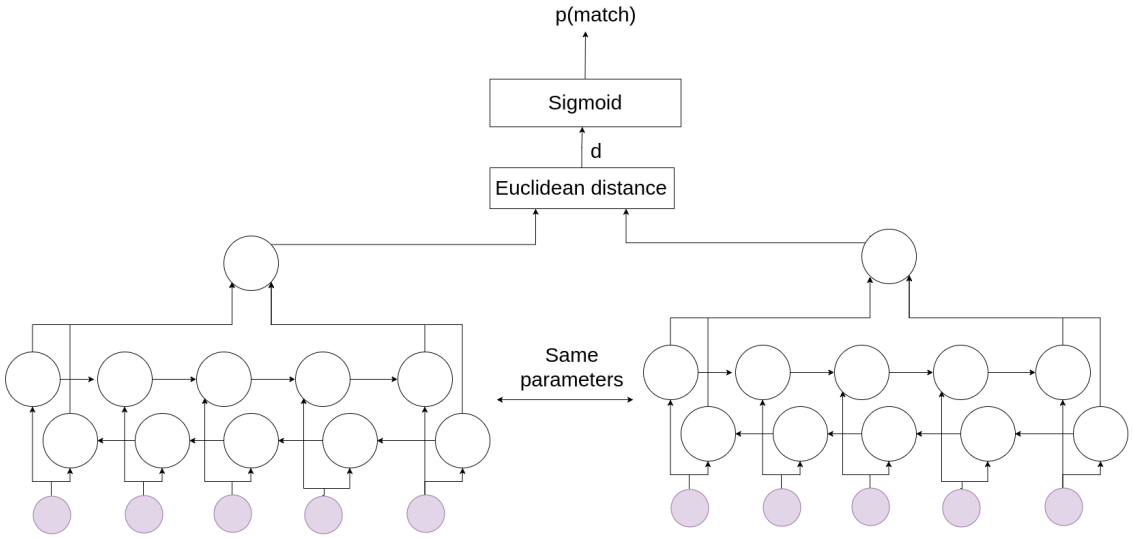


Fig. 4.8 Illustration of Siamese bidirectional RNNs to classify whether a pair of sequences is a match.

4.3 Training Criteria

Neural network training consists in determining the optimal parameters $\hat{\boldsymbol{\lambda}}$ of a neural network $\hat{\boldsymbol{y}} = f(\boldsymbol{x}, \boldsymbol{\lambda})$ given a set of training data $\mathscr{S}_{train}$, to minimise a cost function $C(\lambda, \mathscr{S}_{train})$:

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}} C(\boldsymbol{\lambda}, \mathscr{S}_{train}) \tag{4.55}$$

In supervised learning tasks, the relationship between an input $\boldsymbol{x}$ and an output $\boldsymbol{y}$ is learned from training examples for which both are available:

$$\mathscr{S}_{train} = \{\{\boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}\}, \{\boldsymbol{x}^{(2)}, \boldsymbol{y}^{(2)}\} ... \{\boldsymbol{x}^{(N)}, \boldsymbol{y}^{(N)}\}\} \tag{4.56}$$

In the case of regression, where $\boldsymbol{y}$ is continuous, a common approach, followed in this thesis, is to aim for the network's predictions $\hat{\boldsymbol{y}} = f(\boldsymbol{x}, \boldsymbol{\lambda})$ for likely values of $\boldsymbol{x}$ to deviate as little as possible from the true values $\boldsymbol{y}$. The true globally optimal parameters $\boldsymbol{\lambda}^*$ are thus those that minimise expected squared error:

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \sim p(\boldsymbol{x},\boldsymbol{y})} \left[(f(\boldsymbol{x}; \boldsymbol{\lambda}) - \boldsymbol{y})^T (f(\boldsymbol{x}; \boldsymbol{\lambda}) - \boldsymbol{y}) \right] \tag{4.57}$$

where $p(\boldsymbol{x}, \boldsymbol{y})$ is the true joint distribution of $\boldsymbol{x}$ and $\boldsymbol{y}$ [28, 196].

The extent to which this goal has been achieved can be measured by computing evaluation mean squared error (MSE):

$$MSE_{eval} = \frac{1}{M} \sum_{m=1}^{M} \left(f(\boldsymbol{x}_{eval}^{(m)}; \hat{\boldsymbol{\lambda}}) - \boldsymbol{y}_{eval}^{(m)} \right)^T \left(f(\boldsymbol{x}_{eval}^{(m)}; \hat{\boldsymbol{\lambda}}) - \boldsymbol{y}_{eval}^{(m)} \right) \tag{4.58}$$

where $\mathscr{S}_{eval} = \{\{\boldsymbol{x}_{eval}^{(1)}, \boldsymbol{y}_{eval}^{(1)}\} ... \{\boldsymbol{x}_{eval}^{(M)}, \boldsymbol{y}_{eval}^{(M)}\}\}$ is an evaluation set, which does not overlap with $\mathscr{S}_{train}$, and is assumed to be a representative sample from $p(\boldsymbol{x}, \boldsymbol{y})$.

Assuming $\mathscr{S}_{train}$ to also be a representative sample from $p(\boldsymbol{x}, \boldsymbol{y})$, Equation 4.57 can be approximated by minimising error on the training set:

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}} (MSE_{train}) \tag{4.59}$$

where:

$$MSE_{train} = \frac{1}{N} \sum_{n=1}^{N} \left(f(\boldsymbol{x}^{(n)}; \boldsymbol{\lambda}) - \boldsymbol{y}^{(n)} \right)^T \left(f(\boldsymbol{x}^{(n)}; \boldsymbol{\lambda}) - \boldsymbol{y}^{(n)} \right) \tag{4.60}$$

such that $\hat{\boldsymbol{\lambda}} \to \boldsymbol{\lambda}^*$ as $N \to \infty$.

In the case of classification, where y is a categorical variable representing which of K finite classes a new observation belongs to, a common criterion is cross-entropy. Given a neural network architecture outputting a length-K vector $\boldsymbol{l}$:

$$\boldsymbol{l} = f(\boldsymbol{x}; \boldsymbol{\lambda}) \tag{4.61}$$

a soft-max layer is used to obtain a vector $\boldsymbol{p}$, each element p_k of which represents the probability that y belongs to the category k (i.e. $y = k$):

$$\boldsymbol{p} = \sigma_{softmax}(\boldsymbol{l}) \iff p_k = \frac{\exp l_k}{\sum_{j=1}^{K} \exp l_j} \tag{4.62}$$

The goal is to maximise the probability $\hat{p}_y$ that the network assigns to the correct label y for likely values of $\boldsymbol{x}$:

$$\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}} \mathbb{E}_{\boldsymbol{x}, y \sim p(\boldsymbol{x}, y)} \left[\boldsymbol{y}^T \sigma_{softmax}(f(\boldsymbol{x}; \boldsymbol{\lambda})) \right] \tag{4.63}$$

which is equivalent to:

$$\boldsymbol{\lambda}^* = \arg\min_{\boldsymbol{\lambda}} \mathbb{E}_{\boldsymbol{x}, y \sim p(\boldsymbol{x}, y)} \left[-\log \left(\boldsymbol{y}^T \sigma_{softmax}(f(\boldsymbol{x}; \boldsymbol{\lambda})) \right) \right] \tag{4.64}$$

where $\boldsymbol{y}$ is a one-hot encoding of y (i.e. a length K vector the yth value of which is 1 and all other values of which are zero).

Similarly to regression, the extent to which this goal has been achieved can be estimated with an evaluation data $\mathscr{S}_{eval} = \{\{\boldsymbol{x}_{eval}^{(1)}, y_{eval}^{(1)}\} \ldots \{\boldsymbol{x}_{eval}^{(M)}, y_{eval}^{(M)}\}\}$, using cross-entropy loss:

$$\mathscr{L}_{eval} = -\frac{1}{M} \sum_{m=1}^{M} \log \left(\boldsymbol{y}^{(m)T} \sigma_{softmax}(f(\boldsymbol{x}^{(m)}; \boldsymbol{\lambda})) \right) \tag{4.65}$$

while Equation 4.64 can be approximated by minimising cross-entropy loss on a training set $\mathscr{S}_{train} = \{\{\boldsymbol{x}^{(1)}, y^{(1)}\} \ldots \{\boldsymbol{x}^{(N)}, y^{(N)}\}\}$:

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}} \mathscr{L}_{train} \tag{4.66}$$

where:

$$\mathscr{L}_{train} = -\frac{1}{N} \sum_{n=1}^{N} \log \left(\boldsymbol{y}^{(n)T} \sigma_{softmax}(f(\boldsymbol{x}^{(n)}; \boldsymbol{\lambda})) \right) \tag{4.67}$$

such that, again, $\hat{\boldsymbol{\lambda}} \to \boldsymbol{\lambda}^*$ as $N \to \infty$ [28, 196, 97].

When N is finite, there is a risk of learning a $\hat{\boldsymbol{\lambda}} \neq \boldsymbol{\lambda}^*$ that reduces MSE_{train} or $\mathscr{L}_{train}$ by accurately predicting the individual points $\boldsymbol{y}^{(1:N)}$ but fails to *generalise* to data outside $\mathscr{S}_{train}$, leading to a high MSE_{eval} or $\mathscr{L}_{eval}$. The risk is also increased given that optimisation algorithms can in practice end up returning local minima rather than the required global minimum. As the model's capacity (i.e. how wide a variety of functions it is able to fit) increases compared to N, the chances that there exist $\hat{\boldsymbol{\lambda}}$ that can predict $\boldsymbol{y}^{(1:N)}$ without capturing the underlying relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ also increases. As the cost function becomes rougher, the chances of getting stuck in a local minimum similarly increase. Increasing N for a given model capacity and cost function roughness reduces this risk, as learning $\boldsymbol{x}^{(1:N)} \rightarrow \boldsymbol{y}^{(1:N)}$ without learning $\boldsymbol{x} \rightarrow \boldsymbol{y}$ becomes more difficult. If capacity is too small, both training and evaluation metrics end up too high, as the model can learn neither the training data nor the underlying relationship.

Techniques to improve generalisation by limiting capacity or smoothing the cost function surface are known as *regularisation* [28, 196, 97]. Capacity can be limited by reducing the size and complexity of the network f (e.g. by decreasing the hyperparameters determining the size and number of layers), however this comes at the expense of the network's ability to capture highly non-linear relationships. It is therefore common practice to allow neural networks excess representational capacity and limit effective capacity instead [42, 97].

A common way to limit effective capacity is by stopping the optimisation early [42, 97]. At each iteration of the optimisation algorithm, the training criterion is measured on a third held-out data set, $\mathscr{S}_{val} = \{\{\boldsymbol{x}_{val}^{(1)}, \boldsymbol{y}_{val}^{(1)}\} ... \{\boldsymbol{x}_{val}^{(P)}, \boldsymbol{y}_{val}^{(P)}\}\}$, assumed to be sampled from the same underlying distribution as $\mathscr{S}_{train}$ and $\mathscr{S}_{eval}$. The optimisation is stopped if the validation criterion (MSE_{val} or $\mathscr{L}_{val}$) stops decreasing for a few successive iterations. Thus, optimisation proceeds while training and validation loss are decreasing together and stops training once training loss decreases but validation loss increases. This is based on the intuition that effective capacity is lower at earlier stages of the optimisation and increases as the algorithm zeroes in on the minimum. It has long been known that this is not always strictly the case [218] and, more recently, it has been seen that in certain circumstances there can even be a 'double dip' effect, where validation accuracy improves after first worsening [198, 109].

Another, more direct method of limiting effective capacity is to add a norm penalty term on $\boldsymbol{\lambda}$ (*weight decay*):

$$C(\boldsymbol{\lambda}, \mathscr{S}_{train}) \leftarrow C(\boldsymbol{\lambda}, \mathscr{S}_{train}) + \alpha ||\boldsymbol{\lambda}||_2 \tag{4.68}$$

where α is a hyperparameter [28, 196, 97].

Another regularisation technique is *dropout* [253], whereby a certain percentage of the hidden units in the network are randomly dropped (masked to zero) at each iteration of training. The network preserves its full representational capacity, but effective capacity is significantly limited since the algorithm can never see the whole of $\boldsymbol{\lambda}$ at the same time, so different parts of the network are prevented from fully adapting to each other. The approach can also be thought of as training a large number of smaller networks for only one iteration each and averaging the results

Splitting the data into mini-batches during gradient descent (discussed in §4.4.1) also has the effect of regularisation by limiting effective capacity, while batch and layer normalisation (discussed in §4.4.2) regularise by smoothing the cost function.

4.4 Optimisation

4.4.1 Gradient descent methods

The cost functions discussed in §4.3 for the neural network architectures described in §4.2 do not have closed-form solutions, however, they are differantiable, which means their gradients $\nabla_{\boldsymbol{\lambda}} C(\boldsymbol{x}^{(1:N)}, \boldsymbol{y}^{(1:N)}, \boldsymbol{\lambda})$ can be computed for any value of $\boldsymbol{\lambda}$ and set of N data points $\{\boldsymbol{x}^{(1:N)}, \boldsymbol{y}^{(1:N)}\}$. Optimisation can thus be conducted using a gradient descent method with error backpropagation [234, 108].

At every iteration, parameters are updated by subtracting their gradients multiplied by a hyperparameter η, called the learning rate. Learning ends when the pre-determined maximum number of iterations E is completed or when an early stopping criterion ES (typically based on detecting overfitting by running validation data through the network with the current version of the parameters and seeing if the cost function has stopped decreasing) is satisfied (Algorithm 1) [233].

Algorithm 1 Outline of gradient descent

$\boldsymbol{\lambda}^{(0)} \leftarrow$ initialise;
while $0 < e \leq E$ **and not** $ES(\boldsymbol{\lambda}^{(e)})$ **do**
  $\boldsymbol{\lambda}^{(e+1)} \leftarrow \boldsymbol{\lambda}^{(e)} - \eta \nabla_{\boldsymbol{\lambda}} C(\boldsymbol{x}^{(1:N)}, \boldsymbol{y}^{(1:N)} \boldsymbol{\lambda}^{(e)})$
end while

In stochastic mini-batch gradient descent [112, 97], the algorithm is sped up and an extra measure of regularisation added by partitioning the data into B batches $\{\boldsymbol{x}^{(n_1:n_2)^{(b)}}, \boldsymbol{y}^{(n_1:n_2)^{(b)}}\}$ by shuffling, and updating the parameters based on one batch at a time. Since gradients are only calculated on a small portion of the data each time, the effective capacity of the network

is constrained, hence the regularisation effect. The dependence on shuffling introduces an element of noise, which is why the method is called stochastic. Stochastic mini-batch gradient descent is illustrated in Algorithm 2.

Algorithm 2 Outline of stochastic mini-batch gradient descent

$\boldsymbol{\lambda}^{(0)} \leftarrow$ initialise;
while $0 < e \leq E$ **and not** $ES(\boldsymbol{\lambda}^{(i)})$ **do**
  while $0 < b \leq B$ **do**
    $\boldsymbol{\lambda}^{(e+1)} \leftarrow \boldsymbol{\lambda}^{(e)} - \eta \nabla_{\boldsymbol{\lambda}} C(\boldsymbol{x}^{(n_1:n_2)^{(b)}}, \boldsymbol{y}^{(n_1:n_2)^{(b)}}, \boldsymbol{\lambda}^{(e)})$
  end while
end while

A common issue with stochastic gradient descent is 'zig-zagging' across the loss surface leading to slow convergence, caused by rapidly changing gradients, either due to noise induced by switching between batches or sudden changes in the curvature of the loss surface between different directions.

This is resolved using *momentum methods*, which stabilise gradients by updating parameters using a rolling sum of current and previous gradients (first moment).

The state of the art momentum algorithm is *Adam* [141], illustrated in Algorithm 3, which also uses the square root of the second moment, decelerating learning when the second moment is larger. The Adam algorithm is used throughout this thesis for neural network training.

Algorithm 3 Outline of Adam

$\boldsymbol{\lambda}^{(0)} \leftarrow$ initialise;
while $0 < e \leq E$ **and not** $ES(\boldsymbol{\lambda}^{(i)})$ **do**
  while $0 < b \leq B$ **do**
    $\boldsymbol{g}^{(e)} \leftarrow \nabla_{\boldsymbol{\lambda}} C(\boldsymbol{x}^{(n_1:n_2)^{(b)}}, \boldsymbol{y}^{(n_1:n_2)^{(b)}}, \boldsymbol{\lambda}^{(e)})$
    $\boldsymbol{s}^{(e)} \leftarrow \frac{1}{1-\beta_1^e}\left(\beta_1 \boldsymbol{s}^{(e-1)} + (1-\beta_1)\boldsymbol{g}^{(e)}\right)$
    $\boldsymbol{r}^{(e)} \leftarrow \frac{1}{1-\beta_2^e}\left(\beta_2 \boldsymbol{r}^{(e-1)} + (1-\beta_2)\boldsymbol{g}^{(e)T}\boldsymbol{g}^{(e)}\right)$
    $\boldsymbol{\lambda}^{(e+1)} \leftarrow \boldsymbol{\lambda}^{(e)} - \eta \frac{\boldsymbol{s}^{(e)}}{\sqrt{\boldsymbol{r}^{(e)}}+\varepsilon}$
  end while
end while

RNNs are also trained using the same method by 'unfolding' their hidden layers, such that they become equivalent to a large DNN (except with shared weights), and then training through backpropagation (this is known as backpropagation through time) [285, 148].

Each iteration of gradient descent methods updates the parameters of all layers simultaneously based on their gradients and moments, under the assumption that they remain

constant relative to each other, ignoring higher order effects. This means that if the loss surface $C(\boldsymbol{\lambda}, \mathscr{S}_{train})$ is particularly rough, the optimisation process can become unstable and unable to converge on a minimum [97]. The loss surface when training neural networks has been shown to be very rough, non-convex and characterised by sharp twists ('kinks'), flat regions and sharp minima, all of which can cause contribute to instability of the optimisation process and make it particularly sensitive to its hyperparameters [163, 236]. The deeper and more complex the network, the greater all these effects become.

Approaches to resolving this issue by making the loss surface smoother through normalisation are discussed in §4.4.2. The hyperparameters of the optimisation process include the choice of initial parameter initialisation $\boldsymbol{\lambda}^{(0)}$ and the learning rate η. Methods for setting $\boldsymbol{\lambda}^{(0)}$ are discussed in §4.4.3, while the choice and adaptation of η are discussed in §4.4.4.

4.4.2 Normalisation

Ioffe et al. [127] showed that significant gains in the performance of deep neural networks could be obtained by adding a *batch normalisation* layer after every non-recurrent layer of the network, to ensure all activations follow the same zero mean, unit variance distribution on each batch.

Each element $h_i^{(l)}\left(\boldsymbol{x}^{(n)}\right)$ of the activation $\boldsymbol{h}^{(l)}\left(\boldsymbol{x}^{(n)}\right)$ of each non-recurrent layer l given original input $\boldsymbol{x}^{(n)}$ is normalised based on its mean value and variance in the current batch $\{\boldsymbol{x}^{(n_1:n_2)^{(b)}}, \boldsymbol{y}^{(n_1:n_2)^{(b)}}\}$, followed by scaling and shifting with learnable parameters β and γ:

$$\tilde{h_i}^{(l)}\left(\boldsymbol{x}^{(n)}\right) = \gamma \frac{h_i^{(l)}\left(\boldsymbol{x}^{(n)}\right) - \mathbb{E}\left[h_i^{(l)}\left(\boldsymbol{x}^{(n)}\right)\right]}{\sqrt{Var\left[h_i^{(l)}\left(\boldsymbol{x}^{(n)}\right)\right] + \varepsilon}} + \beta \tag{4.69}$$

where:

$$\mathbb{E}\left[h_i^{(l)}\left(\boldsymbol{x}^{(n)}\right)\right] = \frac{1}{N_b} \sum_{m=n_1^{(b)}}^{n_2^{(b)}} h_i^{(l)}\left(\boldsymbol{x}^{(m)}\right) \tag{4.70}$$

$$Var\left[h_i^{(l)}\left(\boldsymbol{x}^{(n)}\right)\right] = \frac{1}{N_b} \sum_{m=n_1^{(b)}}^{n_2^{(b)}} \left(h_i^{(l)}\left(\boldsymbol{x}^{(n)}\right) - \mathbb{E}\left[h_i^{(l)}\left(\boldsymbol{x}^{(n)}\right)\right]\right)^2 \tag{4.71}$$

$$N_b = n_2^{(b)} - n_1^{(b)} \tag{4.72}$$

and ε is a constant added for numerical stability.

The resultant full vector $\tilde{\boldsymbol{h}}^{(l)}\left(\boldsymbol{x}^{(n)}\right)$ is fed as before through the activation function into the next layer:

$$\boldsymbol{h}^{(l+1)}\left(\boldsymbol{x}^{(n)}\right) = f_{l+1}\left(\sigma_a\left(\tilde{\boldsymbol{h}}^{(l)}\left(\boldsymbol{x}^{(n)}\right)\right);\boldsymbol{\lambda}^{(l+1)}\right) \tag{4.73}$$

with the entire network, including Equations 4.69 to 4.71, being differentiated and backpropagated through in the calculation of $\nabla_{\boldsymbol{\lambda}} C(\boldsymbol{x}^{(n_1:n_2)^{(b)}}, \boldsymbol{y}^{(n_1:n_2)^{(b)}})$.

Santurkar et al. [236] demonstrated that batch normalisation significantly smooths the loss surface in neural network training tasks (as measured by Lipschitzness), thereby explaining the performance gains it has been widely seen to induce in practice. Batch normalisation also reduces the problem of saturation of non-linear activations by constraining their values [127], reduces sensitivity to parameter initialisation by making the parameters scale invariant [125] and improves generalisation through its regularising effect [191]. As expected, performance improvements increase with the depth of the network.

Santurkar et al. [236] also demonstrated similar smoothing and performance gains with normalisation of the form:

$$\tilde{\boldsymbol{h}}^{(l)}\left(\boldsymbol{x}^{(n)}\right) = \gamma \frac{\boldsymbol{h}^{(l)}\left(\boldsymbol{x}^{(n)}\right) - \frac{1}{N_b}\sum_{m=n_1^{(b)}}^{n_2^{(b)}} \boldsymbol{h}^{(l)}\left(\boldsymbol{x}^{(m)}\right)}{\frac{1}{N_b}\sum_{m=n_1^{(b)}}^{n_2^{(b)}} ||\boldsymbol{h}^{(l)}\left(\boldsymbol{x}^{(m)}\right)||_p} + \beta \tag{4.74}$$

where $||\boldsymbol{h}^{(l)}\left(\boldsymbol{x}^{(m)}\right)||_p$ is the l_p norm of $\boldsymbol{h}^{(l)}\left(\boldsymbol{x}^{(m)}\right)$. The l_1 norm in particular showed the best improvement.

A downside of batch normalisation is that it requires computation of batch means and variances (or l_p norms in the case of Eq. 4.74) during evaluation as well as training. This creates difficulties if the trained network is to be used to make predictions for single examples. Another downside is the difficulties in applying the technique to recurrent layers. To resolve these issues, Ba et al. [11] proposed *layer normalisation*, where it is the units of each layer rather than the unit values in each batch that are normalised:

$$\tilde{h_i}^{(l)} = \frac{h_i^{(l)} - \frac{1}{I}\sum_{j=1}^{I} h_j^{(l)}}{\sqrt{\frac{1}{I}\sum_{j=1}^{I}(h_j^{(l)} - \frac{1}{I}\sum_{j=1}^{I} h_j^{(l)})^2}} \tag{4.75}$$

where I is the size of $\boldsymbol{h}^{(l)}$.

This implementation can thus be trained and/or tested on minibatches or single examples as needed without creating difficulties, and can be applied on any layer, feed-forward or recurrent.

For visual tasks which typically use convolutional layers, variants of layer normalisation have been introduced including Group Normalisation [292], whereby the units of each layer are split into groups and each group is normalised instead of the entire layer, and Instance Normalisation [235], where each unit is normalised separately by dividing it by its own norm. These are not further discussed in this thesis as they are not generally applicable to the time sequence data dealt with herein.

4.4.3 Parameter Initialisation

The first step in any gradient descent method is to assign initial values to the parameters to be optimised. These values need to be asymmetric, so that different parts of the network learn different functions. They also need to avoid saturating the non-linear activation functions, which would impede learning.

The deeper a network and the rougher its cost surface, the more sensitive its stability and performance is to this choice of initial parameter values. The smaller the training data relative to the network's capacity, the more sensitive its generalisability is to the initialisation [97]. As seen in previous sections, batch normalisation and choice of activation functions such as Leaky ReLU can reduce this sensitivity, though not eliminate it.

To tackle the saturation issue, Glorot and Bengio [94] introduced an initialisation scheme based on initialising weights from a zero mean distribution with variance inversely proportional to the square root of the size of the layer. Specifically, for a fully-connected layer:

$$\boldsymbol{h}^{(l)} = \sigma_a(\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)}) \tag{4.76}$$

$\boldsymbol{b}^{(l)}$ is initialised to zero, and elements $w_i j^{(l)}$ of $\boldsymbol{W}^{(l)}$ are sampled from either:

$$w_{ij}^{(l)} \sim \mathscr{U}\left(-\sqrt{\frac{6}{I_l + I_{l-1}}}, \frac{6}{I_l + I_{l-1}}\right) \tag{4.77}$$

where I_l is the length of $\boldsymbol{h}^{(l)}$ (uniform Glorot), or:

$$w_{ij}^{(l)} \sim \mathscr{N}\left(0, -\sqrt{\frac{2}{I_l + I_{l-1}}}\right) \tag{4.78}$$

(normal Glorot).

Another common approach to initialisation is *fine-tuning*, where weights are initialised by setting them to the optimised values of the weights of another network with a similar architecture trained on a related task [97]. This approach can allow incorporation of additional

information from the data of the related task, and, if the related task is easier to optimise, help overcome difficulties in the optimisation of the new task by beginning its parameters in a state more likely to be closer to their optimum.

4.4.4 Learning Rate Schedules

The learning rate η determines the size of the step in each iteration of gradient descent. As was seen in §4.4.1, adaptive gradient descent algorithms set a different effective learning rate for each parameter in each iteration based on the moments of the gradient, however the baseline learning rate $\eta^{(i)}$ at each iteration i remains a hyperparameter.

When the learning rate is low, optimisation proceeds slowly and thoroughly and the effective capacity of the network is high. This makes it easier to learn complex patterns but also increases the risk of overfitting noisy data and falling into local minima. When the learning rate is high, optimisation proceeds quickly and effective capacity is limited, preventing overfitting but only learning simple patterns [97, 167, 294].

A common approach is thus to begin with a high learning rate at low iteration numbers to learn the simple patterns and escape local minima and then lower the learning rate at higher iteration numbers to learn more complex patterns [167, 294]. The most prevalent version of this method is exponential *learning rate decay* [157, 145]:

$$\eta^{(i)} = \eta^{(0)} e^{-\alpha i} \tag{4.79}$$

where α is a hyperparameter.

Smith [247–249] demonstrated further gains in performance by varying the learning rate *cyclically* between extremes $\eta^{(0)}$ and $\eta^{(max)}$ in a triangular fashion:

$$\eta^{(i)} = \eta^{(0)} + \frac{2(\eta^{(max)} - \eta^{(0)})}{\pi} \sin^{-1} \sin\left(\frac{2\pi}{p} i\right) \tag{4.80}$$

where p is a hyperparameter determining the period of the cycle.

If the normalisation techniques from §4.4.2 are used, the optimisation task becomes *scale invariant*, meaning $C(\boldsymbol{\lambda}, \mathscr{S}_{train})$ is unaffected by changes to the scale of $\boldsymbol{\lambda}$ [115]. Li et al. [168] showed that, under these circumstances, *exponentially increasing* the learning rate:

$$\eta^{(i)} = \eta^{(0)} e^{\alpha i} \tag{4.81}$$

is equivalent to keeping the learning rate constant and applying weight decay (see §4.3), while weight decay with learning rate decay is equivalent to exponentially increasing learning

rate with a decaying exponent:

$$\eta^{(i)} = \eta^{(0)} e^{(\alpha_0 e^{-\beta i})i} \tag{4.82}$$

where α and β are hyperparameters. Such schedules can thus be used instead of weight decay for the purposes of regularisation.

4.5 Chapter Summary

This chapter introduced the concept of neural networks and how they can be used to solve problems requiring the learning of complex representations from data.

Different types of neural network architecture were reviewed in §4.2, namely feed-forward networks for learning vector-to-vector relationships, recurrent networks and attention mechanisms for learning sequence-to-vector and sequence-to-sequence relationships, and siamese networks for learning pairwise distances. The criteria used to train networks for regression, classification and dimensionality reduction tasks for optimal accuracy and generalisation were explored in §4.3.

Finally, §4.4 reviewed the gradient descent methods used to train neural networks and the normalisation, learning rate adaptation and initialisation techniques that can be used to improve optimisation performance and ensure regularisation.

Chapter 5

Deep Learning for Spoken Language Proficiency Assessment

Approaches to assessing non-native speakers both holistically and with respect to individual views were reviewed in Chapter 2. Issues with availability and reliability of human-annotated grades for single-view systems were discussed. In the cases of pronunciation, rhythm and intonation, the weaknesses of existing methods were identified and it was hypothesised that performance could be improved by adopting approaches with a greater degree of representational capacity and tunability. Chapter 4 reviewed deep learning techniques to learn complex hierarchical representations mapping vectors to vectors, sequences to vectors and sequences to sequences. This chapter applies these techniques to propose a novel multi-view grading framework in a context where only holistic scores are available as training data.

The framework consists of end-to-end neural graders designed to limit the information available for grading to only that indicative of a single view, along with methods of combining them for holistic grading. Three hypotheses are developed to be experimentally evaluated in Chapter 6. The first is that graders designed in this way are able to predict single-view grades even when trained on holistic grades. The second is that designing such graders to be trainable end-to-end will make them outperform comparable two-stage graders (based on feature extraction followed by grading) both at predicting human-assigned grades and in terms of their generalisability to different tasks and datasets. Finally, the third is that systems combining single-view graders to predict holistic scores will outperform both two-stage multi-view holistic graders and end-to-end holistic graders at the task of holistic grading.

The general framework and the implications of training single-view graders on holistic scores are discussed in §5.1. Novel end-to-end single-view graders based on the framework are then proposed for each of pronunciation, rhythm and intonation in §5.2. Each is presented alongside two-stage grader counterparts, the second stage of which is always a DNN. In the

case of pronunciation, the feature extraction stage is adapted from previous work [151]. In the case of rhythm, it is based on aggregating handcrafted approaches from the literature. In the case of intonation, both a novel approach and approaches adapted from the literature are presented. Finally, §5.3 presents three methods of combining single-view graders to predict holistic scores.

5.1 Multi-view grading

This section motivates the three main ideas underlying the framework presented in this chapter, namely the introduction of end-to-end single-view graders to avoid the drawbacks of handcrafted features while maintaining their view specificity, the possibility of training view-specific graders on holistic grades without compromising their view specificity, and the combination of view-specific graders for holistic grading.

As per the discussion in §2.1, multi-view spoken language proficiency assessment involves using input data $\boldsymbol{x}_{1:T}^{(n)}$ from a speaker n to predict both a holistic grade $\bar{y}^{(n)}$ and grades $\hat{y}_j^{(n)}$ representing their proficiency with respect to each of a number of particular views j (pronunciation, rhythm etc.). The single-view graders reviewed in §2.3 fed hand-crafted features $\boldsymbol{v}_j^{(n)}$ for each view j through a grader $\mathscr{G}_j$ to predict single-view scores $y_j^{(n)}$:

$$\mathscr{F}_j(\boldsymbol{x}_{1:T}^{(n)}) \rightarrow \boldsymbol{v}_j^{(n)};\ \ \mathscr{G}_j(\boldsymbol{v}_j^{(n)}, \boldsymbol{\lambda}_j) \rightarrow \hat{y}_j^{(n)} \tag{5.1}$$

In §2.4, three approaches to holistic grading were reviewed. In the first approach, view-specific hand-crafted features for multiple views are concatenated to produce holistic feature sets $\boldsymbol{v}^{(n)}$, which are then passed through graders $\mathscr{G}$, to predict holistic grades:

$$\mathscr{G}(\boldsymbol{v}^{(n)}, \boldsymbol{\lambda}) \rightarrow \hat{\bar{y}}^{(n)} \tag{5.2}$$

The other two approaches use a neural holistic feature extractor $\mathscr{F}$, which, in one case, is trained together with the grader $\mathscr{G}$ in an end-to-end fashion, and, in the other case, in an unsupervised manner to compactly represent the information in $\boldsymbol{x}_{1:T}^{(n)}$:

$$\mathscr{F}(\boldsymbol{x}_{1:T}^{(n)}, \boldsymbol{\lambda}) \rightarrow \boldsymbol{v}^{(n)};\ \ \mathscr{G}(\boldsymbol{v}^{(n)}, \boldsymbol{\lambda}) \rightarrow \hat{\bar{y}}^{(n)} \tag{5.3}$$

The approach presented in this chapter relies on a compromise between the tunability and representational capacity offered by the neural holistic graders and the interpretability and view specificity made possible by the use of hand-crafted features. The idea is to use end-to-end single-view graders with trainable parametric feature extractors $\mathscr{F}_j$, which are

constrained, by their choice of input and structure, to only extract information regarding a particular view j:

$$\mathscr{F}_j(\boldsymbol{x}_{1:T}^{(n)}, \boldsymbol{\lambda}_j) \rightarrow \boldsymbol{v}_j^{(n)};\ \ \mathscr{G}(\boldsymbol{v}_j^{(n)}, \boldsymbol{\lambda}_j) \rightarrow \hat{y}_j^{(n)} \tag{5.4}$$

As these graders are predicting view-specific grades $\hat{y}_j^{(n)}$, they would ideally be trained on single-view annotations $y_j^{(n)}$:

$$\hat{\boldsymbol{\lambda}}_j = \arg\min_{\boldsymbol{\lambda}_j} \sum_{n=1}^{N} (\hat{y}_j^{(n)} - y_j^{(n)})^2 \tag{5.5}$$

Due to the problems with obtaining reliable human-annotated single-view grades, the use of holistic targets $\bar{y}^{(n)}$ is to be explored instead:

$$\hat{\boldsymbol{\lambda}}_j = \arg\min_{\boldsymbol{\lambda}_j} \sum_{n=1}^{N} (\hat{y}_j^{(n)} - \bar{y}^{(n)})^2 \tag{5.6}$$

Given the componential nature of holistic grades [67] and considering J views exhaustively covering the aspects of proficiency taken into account by holistic graders, the average view-specific grade should be equal to the holistic grade, if computed on the same scale:

$$\bar{y}^{(n)} = \frac{1}{J} \sum_{j=1}^{J} y_j^{(n)} \tag{5.7}$$

Equation 5.6 can thus be expressed as:

$$\hat{\boldsymbol{\lambda}}_j = \arg\min_{\boldsymbol{\lambda}_j} \sum_{n=1}^{N} (\hat{y}_j^{(n)} - \frac{1}{J} y_j^{(n)} - \frac{1}{J} \sum_{l \neq j}^{J} y_l^{(n)})^2 \tag{5.8}$$

If F_j indeed only extracts information relevant to view j, a given value of $\boldsymbol{v}_j^{(n)}$ should map to a wide range of inputs $\boldsymbol{x}_{1:T}^{(n)}$, by keeping view j unchanged but altering other views (e.g. identical text but different pronunciations). Any sufficiently large number N of speakers n should thus cluster into Q sets $S_q^{(j)}$, the L_q members of each of which map to the same $\boldsymbol{v}_j^{(n)}$:

$$n \in S_q^{(j)} \Leftrightarrow \boldsymbol{v}_j^{(n)} = \boldsymbol{v}_j^{(q)} \tag{5.9}$$

It follows that the members of each $S_q^{(j)}$ will also share the same true view-specific score $y_j^{(q)}$ and that the grader $\mathscr{G}_j$ will yield for all of them the same prediction $\hat{y}_j^{(q)}$:

$$y_j^{(n)} = y_j^{(q)},\ \ \hat{y}_j^{(n)} = \hat{y}_j^{(q)} \ \ \forall n \in S_q^{(j)} \tag{5.10}$$

Equation 5.8 can thus be re-written as:

$$\hat{\boldsymbol{\lambda}}_j = \arg\min_{\boldsymbol{\lambda}_j} \sum_{q=1}^{Q} (\hat{y}_j^{(q)} - \frac{1}{J} y_j^{(q)} - \frac{1}{JS_q^{(j)}} \sum_{n \in S_q^{(j)}} \sum_{l \neq j}^{J} y_l^{(n)})^2 \tag{5.11}$$

It follows that the effective target $\bar{y}_j^{(q)}$ that the system sees for the feature value $\boldsymbol{v}_j^{(q)}$ corresponding to each q is given by:

$$\bar{y}_j^{(q)} = \frac{1}{J} y_j^{(q)} + \frac{1}{JL_q} \sum_{n \in S_q^{(j)}} \sum_{l \neq j} y_l^{(n)} \tag{5.12}$$

Assuming each view-specific score $y_l^{(n)}$ consists of a general component $g^{(n)}$ which all views access (and thus accounts for their high correlation) and a unique component $u_l^{(n)}$:

$$y_l^{(n)} = g^{(n)} + u_l^{(n)} \tag{5.13}$$

Given knowledge of $y_j^{(n)}$, the expected value of $y_l^{(n)}$ for every $l \neq j$ is given by:

$$\mathbb{E}_{p(u_l^{(n)}|y_j^{(n)})}[y_l^{(n)}] = y_j^{(n)} - \mathbb{E}_{p(u_l^{(n)}|y_j^{(n)})}[u_j^{(n)}] + \mathbb{E}_{p(u_l^{(n)}|y_j^{(n)})}[u_l^{(n)}] \tag{5.14}$$

Assuming further that the unique component of every single-view grade is independent of all the others and, without loss of generality, has an expected value of zero:

$$\mathbb{E}_{p(u_l^{(n)}|y_j^{(n)})}[y_l^{(n)}] = y_j^{(n)} \tag{5.15}$$

Given that the scores of the speakers in each cluster can also be expected to be independent of each other, the expectation of Equation 5.12 thus also collapses to:

$$\mathbb{E}_{p(\boldsymbol{u}_j^{(q)}|y_j^{(q)})}[\bar{y}_j^{(q)}] = y_j^{(q)} \quad \text{where} \quad \boldsymbol{u}_j^{(q)} = \bigcup_{l \neq j} \bigcup_{n \in S_q^{(j)}} u_l^{(n)} \tag{5.16}$$

It follows that, insofar as the feature extractors only keep information about their respective views, the views are uncorrelated given their general component, and there is sufficient training data, a system trained on holistic grades according to Equation 5.6 will, on average, receive effective targets equal to the true view-specific score. With sufficient regularisation, a similar result should be approximated even with more moderate amounts of training data.

5.2 Single-view graders

5.2.1 Pronunciation

Overall pronunciation assessment in spontaneous non-native speech was reviewed in §2.3.2. It was seen how *phone distance* techniques, where a representation is learned for the speaker's pronunciation of each phone of English and then distances between each pair used to characterise them overall, can effectively grade speakers in a manner that should be robust to variation in irrelevant speaker attributes such as voice quality and sex. A method based on Kullback-Leibler (K-L) divergences between single-speaker monophone acoustic models, introduced in previous work [154], was presented, and its weaknesses discussed. This subsection explores how this work can be built on for more effective modelling of proficiency based on distances in phone pronunciation space.

In the original method, observations $\boldsymbol{o}_t$ (see Appendix A) for each frame t of speech are first passed through an ASR to recognise the most likely word sequence $\hat{w}_{1:I}$ (see §2.2):

$$\hat{w}_{1:I} = \arg\max_{w_{1:I}} \left\{ P(w_{1:I}) \sum_{\phi_{1:M} \in \mathscr{D}_{w_{1:I}}} P(\phi_{1:M}|w_{1:I}) \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \right\} \tag{5.17}$$

They are then force aligned to obtain the most probable phone sequence $\hat{\phi}_{1:M}$ and time alignment (defined by word and phone at each frame) $\hat{s}_{1:T}$ given $\hat{w}_{1:I}$:

$$\hat{\phi}_{1:M} = \arg\max_{\phi_{1:M} \in \mathscr{D}_{\hat{w}_{1:I}}} \left\{ \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \right\} \tag{5.18}$$

$$\hat{s}_{1:T} = \arg\max_{s_{1:T}|\hat{\phi}_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\hat{\phi}_{1:M}) \tag{5.19}$$

The time alignment $\hat{s}_{1:T}$ is used to obtain the sequence of observations $\boldsymbol{o}_{t_1,t_2}^{(\hat{\phi}_m)}$ corresponding to each phone instance ϕ_m. A set of models is then trained by maximum likelihood to represent the way the speaker pronounces each phone label ψ:

$$\hat{\mathscr{M}}_\psi = \arg\max_{\mathscr{M}_\psi} \left\{ \sum_{m|\hat{\phi}_m=\psi} p(\boldsymbol{o}_{t_1,t_2}^{(\hat{\phi}_m)}, \hat{s}_{t_1:t_2}^{(\hat{\phi}_m)}|\hat{\phi}_m, \mathscr{M}_\psi) \right\} \tag{5.20}$$

The parameters of these models should thus compactly represent the speaker's speech, while only retaining information about the way they tend to pronounce each phone. A

canonical dictionary $\mathscr{D}_{w_{1:I}}$ and an acoustic model trained on non-native speakers using the same dictionary (see Appendix C) are used for recognition and alignment. This maximises the chance that non-canonical realisations of words will still be recognised and will map to their closest canonical pronunciation during alignment. Non-canonically pronounced phones will thus be modelled as part of the distribution of the canonical phones they replaced, so the models capture how the speaker both pronounces and mispronounces each phone. To further compress the information and eliminate irrelevant attributes common across phones, pair-wise symmetric K-L divergence between models are computed:

$$\Delta_{\text{SKL}}(\psi_1||\psi_2) = \frac{1}{2}\Delta_{\text{KL}}(\psi_1||\psi_2) + \frac{1}{2}\Delta_{\text{KL}}(\psi_2||\psi_1) \tag{5.21}$$

where:

$$\Delta_{\text{KL}}(\psi_1||\psi_2) = \int p(\boldsymbol{o}_{1:\tau}|\mathscr{M}_{\psi_1}) \log\left(\frac{p(\boldsymbol{o}_{1:\tau}|\mathscr{M}_{\psi_1})}{p(\boldsymbol{o}_{1:\tau}|\mathscr{M}_{\psi_2})}\right) d\boldsymbol{o}_{1:\tau} \tag{5.22}$$

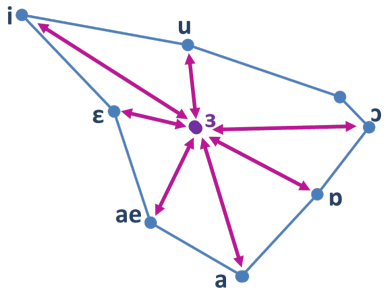


Fig. 5.1 Illustration of the phone distance concept

Following [154], each phone ψ is modeled by assuming all its frames $\boldsymbol{o}_t$ are drawn from a single multivariate Gaussian with mean, $\boldsymbol{\mu}_\psi$ and diagonal covariance matrix, $\boldsymbol{\Sigma}_\psi$:

$$p(\boldsymbol{o}_t|\mathscr{M}_\psi) = \mathscr{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_\psi, \boldsymbol{\Sigma}_\psi) \tag{5.23}$$

Equation 5.20 thus yields:

$$\boldsymbol{\mu}_\psi = \sum_{m|\hat{\phi}_m=\psi} \sum_{t=t_1^{(\hat{\phi}_m)}}^{t_2^{(\hat{\phi}_m)}} \boldsymbol{o}_t, \quad \boldsymbol{\Sigma}_\psi = \sum_{m|\hat{\phi}_m=\psi} \sum_{t=t_1^{(\hat{\phi}_m)}}^{t_2^{(\hat{\phi}_m)}} \left(\boldsymbol{o}_t \boldsymbol{o}_t^T\right) - \boldsymbol{\mu}_\psi \boldsymbol{\mu}_\psi^T \tag{5.24}$$

where $t_1^{(\hat{\phi}_m)}$ and $t_2^{(\hat{\phi}_m)}$ are the start and end times of each phone instance $\hat{\phi}_m$.

This simple model minimises the appearances of each phone necessary to train it and has the advantage of having a closed-form solution for its K-L divergence (a variant with HMMs,

model adaptation and variational approximations of relative entropy was also implemented in the original work but is not reproduced here as it did not yield significant benefits):

$$\Delta_{\mathrm{KL}}\left(\psi_1||\psi_2\right)=\frac{1}{2}\left(\mathrm{tr}(\boldsymbol{\Sigma}_{\psi_2}^{-1}\boldsymbol{\Sigma}_{\psi_1})+\Delta\boldsymbol{\mu}_{\psi_1,\psi_2}^{T}\boldsymbol{\Sigma}_{\psi_2}^{-1}\Delta\boldsymbol{\mu}_{\psi_1,\psi_2}-\mathtt{dim}(\boldsymbol{o})+\ln\frac{\det\boldsymbol{\Sigma}_{\psi_2}}{\det\boldsymbol{\Sigma}_{\psi_1}}\right) \tag{5.25}$$

where $\mathtt{dim}(\boldsymbol{o})$ is the dimension of $\boldsymbol{o}$, $\mathrm{tr}(\cdot)$ and $\det(\cdot)$ are the operators for the trace and determinant of the matrix, respectively. and:

$$\Delta\boldsymbol{\mu}_{\psi_1,\psi_2}=\boldsymbol{\mu}_{\psi_2}-\boldsymbol{\mu}_{\psi_1} \tag{5.26}$$

The resultant symmetric K-L divergences between each possible pair of phones are concatenated into a length $\frac{1}{2}P(P-1)$ vector $\boldsymbol{d}$:

$$\boldsymbol{d}=\left[\Delta_{\mathrm{SKL}}\left(\psi_1||\psi_2\right),\Delta_{\mathrm{SKL}}\left(\psi_1||\psi_3\right)...\Delta_{\mathrm{SKL}}\left(\psi_{P-1}||\psi_P\right)\right]^T \tag{5.27}$$

where P is the number of phones in the alphabet.

One issue that can arise with this type of feature, particularly for short utterances, is the case of phones that are missing from the candidate's speech and for which models, and thus pairwise distances, cannot be obtained. If the model is to be able to learn to deal with these cases automatically the missing phones must be marked in some way in the input. The most economical way of doing this is for all distances $\Delta_{\mathrm{SKL}}\left(\psi_p||\psi_q\right)$ corresponding to phone-pairs containing each missing phone ψ_p to be set to a pre-determined value which cannot be confused for a distance, such that the network can learn a separate behaviour for it. Since $log(\boldsymbol{d}+1)$ is always positive, a value of -1 is selected. The resultant vector thus also encodes information about which phones the speaker did not pronounce at all.

As the task is now one of vector-to-scalar regression, the vector $\boldsymbol{d}$ can be passed through a feed-forward fully-connected network (as described in §4.2.1) to predict grade. To compress the often wide dynamic range of K-L divergence, $log(\boldsymbol{d}+1)$ is used instead of $\boldsymbol{d}$ as the input (adding 1 serving to prevent a positive divergence from being matched to a negative input):

$$y=f_{DNN}(\log\left(\boldsymbol{d}+1\right);\boldsymbol{\lambda}_{DNN}) \tag{5.28}$$

where $\boldsymbol{\lambda}_{DNN}$ are the trainable parameters of the network.

The disadvantages of this approach include the large amounts of data still needed per speaker to obtain useful Gaussian models and the risk that the blunt techniques used to limit information not relevant to the speaker's pronunciation of phones may also discard useful information about pronunciation. The method treats all frame vectors equally, failing to take

into account the time-sequence nature of frames within a phone distance and the different role played by frames in different positions within the phone. Treating all frames equally also fails to account for the varying pronunciation of the same phone in different contexts, and the different salience of different phone instances to proficiency depending on their context. There is also a risk of the result being poisoned by badly aligned or incorrectly recognised phones, which cannot be distinguished from correct ones.

As can be seen in Figure 5.2, the phone distance calculation process is separate from the grading stage and the two cannot be trained end-to-end. Feature extraction is fixed and cannot be tuned using the training data.

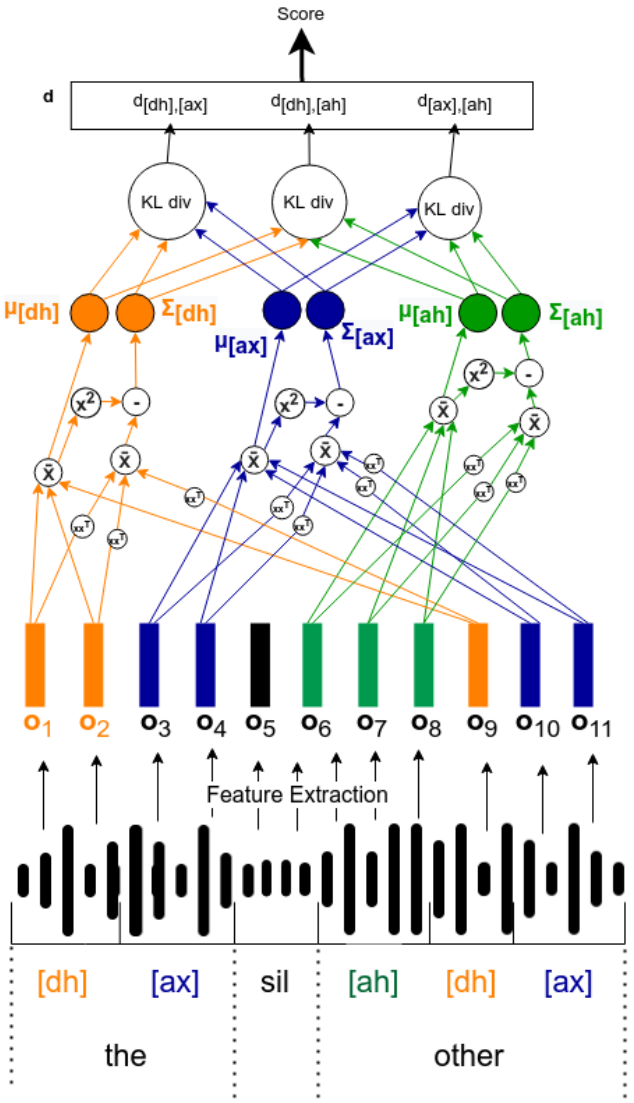


Fig. 5.2 Illustration of proficiency grading using phone distance features.

The next step is to explore whether deep systems can be used to resolve these issues while maintaining the specificity to the view of pronunciation achieved by the phone distance method. To ensure that the time-sequence nature of frames within each phone is taken into account and complexities and non-linearities between the values of $\boldsymbol{o}_t$ and distances in pronunciation space can be captured, a tunable distance metric is defined which can be learned directly from the frame sequences corresponding to individual phone instances.

The system implemented (Fig. 5.3) is based on the Siamese architecture described in §4.2.4.

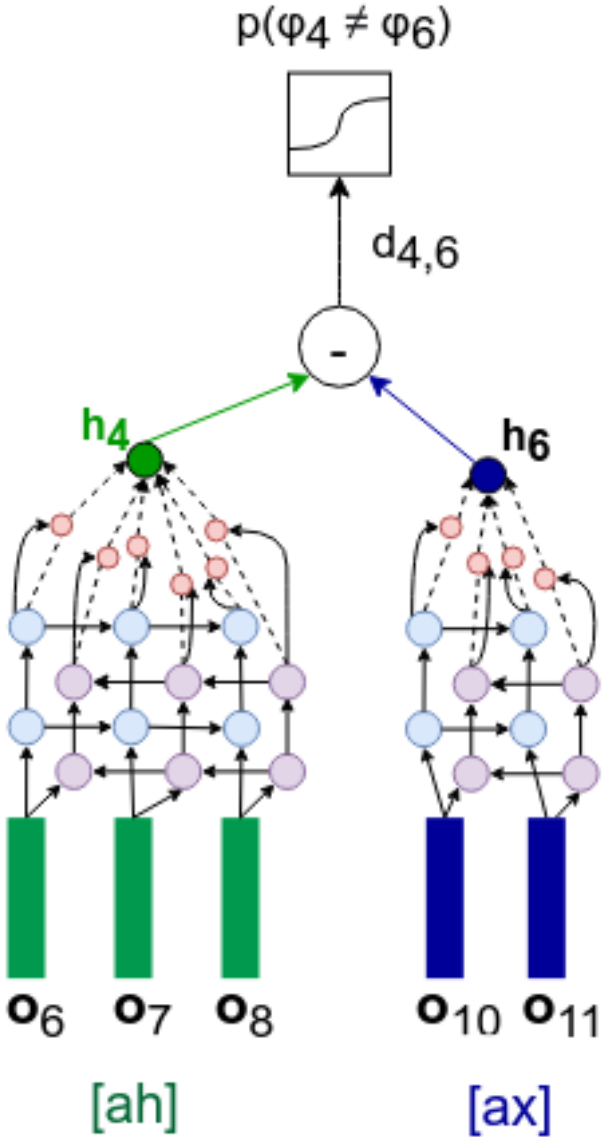


Fig. 5.3 Illustration of Siamese bi-directional RNN

The phone distance between any given pair of phone instances ϕ_1 and ϕ_2 is re-defined as the Euclidean distance between fixed-length embeddings of the corresponding sequences of frame vectors $\boldsymbol{o}_{t_1:t_2}^{(\phi_1)}$ and $\boldsymbol{o}_{t_1:t_2}^{(\phi_2)}$.

$$\Delta_{\phi_1,\phi_2} = |f_{seq2vec}(\boldsymbol{o}_{t_1:t_2}^{(\phi_1)}; \boldsymbol{\lambda}_{seq2vec}) - f_{seq2vec}(\boldsymbol{o}_{t_1:t_2}^{(\phi_2)}; \boldsymbol{\lambda}_{seq2vec})| \tag{5.29}$$

where $f_{seq2vec}$ takes the form of one of the sequence-to-vector transformations discussed in §4.2.2 and §4.2.3 (e.g. bi-directional LSTM with last outputs or attention over outputs or sequence-to-vector multi-head attention) and takes parameters $\boldsymbol{\lambda}_{seq2vec}$, the values of which are tied between the two halves of the Siamese network.

This distance is then passed through a sigmoid to predict the probability that ϕ_1 and ϕ_2 are instances of the same phones.

$$\hat{p}(\psi_1 \neq \psi_2) = \sigma_{softmax}(\Delta_{\psi_1,\psi_2}) \tag{5.30}$$

The network can now be trained on equal numbers of matched and non-matched phones, learning to embed to a space where Δ_{ϕ_1,ϕ_2} is small for similar and large for different phones.

The trained Siamese network can calculate a distance metric between two phone instances, however what is required is a distance metric for a given pair of phone labels reflecting all the instances of each of the two phones by the speaker in question, each instance being taken into account to the extent that it is predictive of the speaker's proficiency.

To this end, the frame vector sequence $\boldsymbol{o}_{t_1:t_2}^{(\phi_m)}$ corresponding to each phone instance ϕ_m is embedded to a phone instance vector representation as before:

$$\boldsymbol{h}_m = f_{seq2vec}(\boldsymbol{o}_{t_1:t_2}^{(\phi_m)}; \boldsymbol{\lambda}'_{seq2vec}) \tag{5.31}$$

but this time an attention mechanism is placed over all phone instance embeddings corresponding to the same phone label:

$$\tilde{\boldsymbol{h}}_\psi = \sum_{m|\hat{\phi}_m=\psi} \alpha_m \boldsymbol{h}_m \tag{5.32}$$

where:

$$\alpha_m = \frac{\exp f_{att}(\boldsymbol{h}_m; \boldsymbol{\lambda}_{att})}{\sum_{p=1}^{M} \exp f_{att}(\boldsymbol{h}_p; \boldsymbol{\lambda}_{att})} \tag{5.33}$$

The attention weights α_m are thus indicative of the relative importance of each phone instance to score.

Euclidean distances are then calculated between these resultant phone level embeddings rather than the original instance embeddings:

$$\Delta_{\psi_1,\psi_2} = |\tilde{\boldsymbol{h}}_{\psi_1} - \tilde{\boldsymbol{h}}_{\psi_2}| \tag{5.34}$$

As in the original phone distance case, pairwise distances are concatenated into a length $\frac{1}{2}P(P-1)$ vector $\boldsymbol{d}$:

$$\boldsymbol{d} = [\Delta_{\psi_1,\psi_2}, \Delta_{\psi_1,\psi_3} ... \Delta_{\psi_{P-1},\psi_P}]^T \tag{5.35}$$

which is passed through a fully-connected feed-forward layer to predict score:

$$y = f_{DNN}(\boldsymbol{d}; \boldsymbol{\lambda}'_{DNN}) \tag{5.36}$$

The entire network, from observation vectors to the final score output is trained through backpropagation, minimising a training criterion based on mean squared error (MSE) by optimising the parameters $\boldsymbol{\lambda}_{pron} = [\boldsymbol{\lambda}_{seq2vec}, \boldsymbol{\lambda}_{att}, \boldsymbol{\lambda}_{DNN}]$.

Given the hierarchical architecture of the network, its loss surface can be expected to be very rough and highly sensitive to the initialisation of parameters. If all the different parts of the network are initialised randomly, convergence to the globally optimal set of parameter values will likely be difficult.

Following the discussion in §4.4.3, the problem is tackled by initialising the parameters of each stage of the network using simpler tasks trained to perform a similar function to what it is intended to perform in the broader network. This also helps ensure each stage of the network performs its intended function.

Specifically, $\boldsymbol{\lambda}'_{seq2vec}$, which is intended to extract fixed-length phone instance representations in a space in which Euclidean distance is predictive of phone similarity, is initialised using the trained Siamese network, while the parameters $\boldsymbol{\lambda}'_{DNN}$, intended to predict score from a vector of phone-pair distances, are initialised using the feed-forward layer from the original network:

$$\boldsymbol{\lambda}'^{(0)}_{seq2vec} = \boldsymbol{\lambda}^{(E)}_{seq2vec}; \tag{5.37}$$

$$\boldsymbol{\lambda}'^{(0)}_{DNN} = \boldsymbol{\lambda}^{(E)}_{DNN} \tag{5.38}$$

The network is thus trained in multiple stages, learning simpler representations during the training of the DNN grader and the Siamese networks and more complex representations during end-to-end fine-tuning.

The full network is illustrated alongside the original phone distance approach in Figure 5.4.

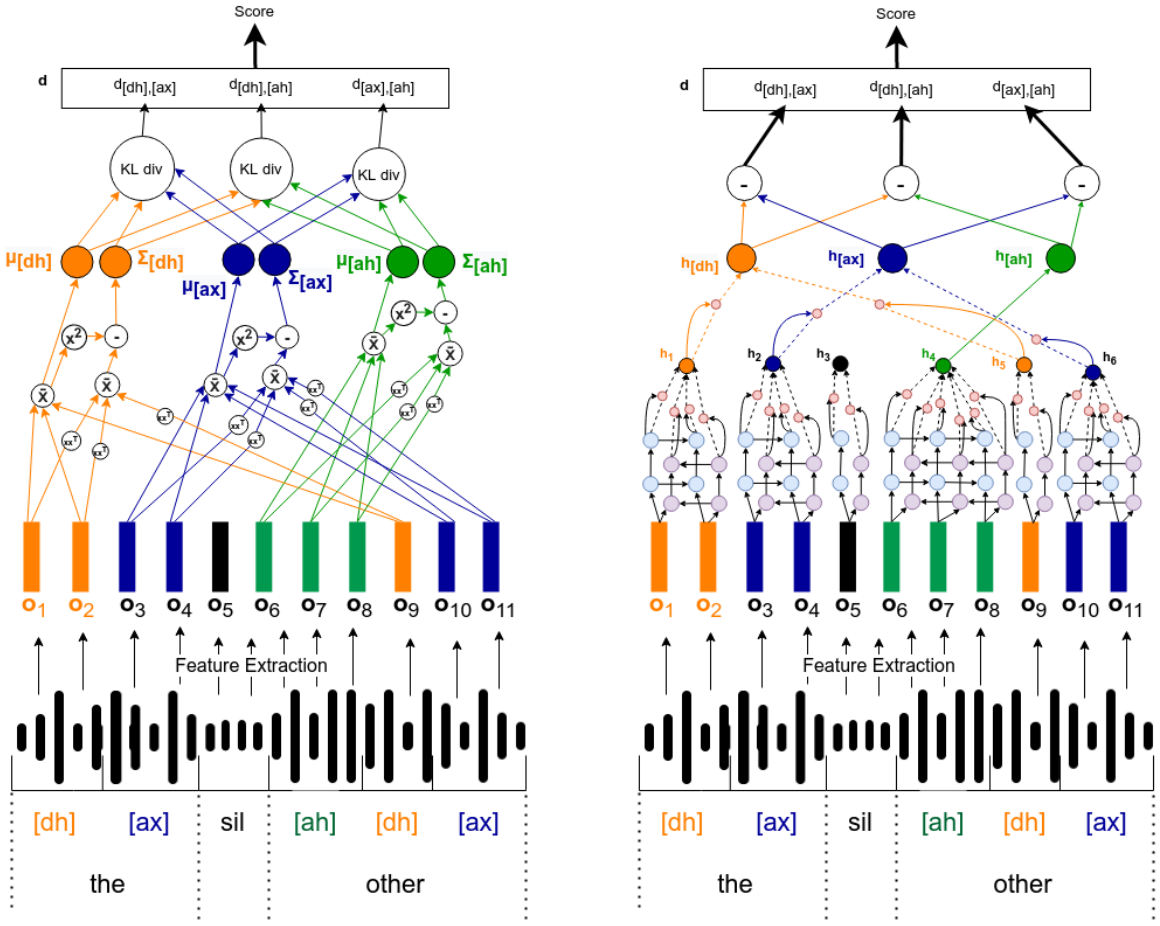


Fig. 5.4 Illustration of Deep Phone Distance Feature architecture (right) compared to Phone Distance Features (left - reproduced from Fig. 5.2)

Compared to the original phone distance feature extraction system, the embedding of the frame sequences corresponding to each phone and the assignment of weights to different instances of each phone are learned rather than hard-coded, with the entire system being trainable end-to-end and usable in one stage (excluding feature extraction).

The work in this sub-section and corresponding experiments (see Chapter 6) were published in [152].

5.2.2 Rhythm

The discussion in §2.3.5 introduced a bundle of state-of-the-art rhythm features to be used as a baseline for rhythm assessment, listed in Appendix E. The most promising were found to be PVI and CCI features:

- $\texttt{rPVI}_V = \frac{1}{K_V-1}\sum_{k=1}^{K_V-1}|d(\tau_k^{(V)}) - d(\tau_{k+1}^{(V)})|$ where $d(\tau_k)$ is the duration of the kth vocalic interval and K_V is the number of vocalic intervals.
- $\texttt{rPVI}_C = \frac{1}{K_C-1}\sum_{k=1}^{K_C-1}|d(\tau_k) - d(\tau_{k+1})|$ where $d(\tau_k)$ is the duration of the kth intervocalic segment and K_C is the number of intervocalic segments.
- $\texttt{CCI}_V = \frac{1}{K_V-1}\sum_{k=1}^{K_V-1}\left|\frac{d(\tau_k)}{l_k} - \frac{d(\tau_{k+1})}{l_{k+1}}\right|$ where $d(\tau_k)$ and l_k are the duration and number of vowel phones of the kth vocalic interval and K_V is the number of vocalic intervals.
- $\texttt{CCI}_C = \frac{1}{K_C-1}\sum_{k=1}^{K_C-1}\left|\frac{d(\tau_k)}{l_k} - \frac{d(\tau_{k+1})}{l_{k+1}}\right|$ where $d(\tau_k)$ and l_k are the duration and number of phones and silences in the kth intervocalic interval and K_C is the number of intervocalic measurements.

Their main weaknesses were seen to be their failure to capture effects beyond the interval-pair level as well as the effect of different levels of salience of each measurement and duration on the final proficiency score. Following the logic of §5.2.1, this subsection resolves these problems by introducing a deep, tunable, generalisation of PVI and CCI, in the form of a hierarchical deep neural network architecture.

Consider an utterance spoken by a given speaker and divided into K_V vocalic and K_C intervocalic intervals (e.g. the phrase 'on the mat' consists of vocalic intervals /oh/, /ax/ and /ae/ and intervocalic segments {/n/,/sil/,/dh/}, /m/ and /t/ as illustrated in Figure 5.5).

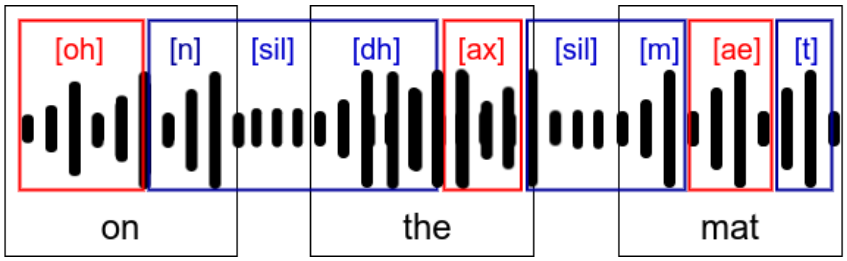


Fig. 5.5 Illustration of words (black), vocalic intervals (red), intervocalic intervals (blue) and sub-segments (square brackets) in the phrase 'on the mat'

For each of the two types of interval, we measure the duration $d(\upsilon_m^{(k)})$ of each sub-segment m of M_k in a given interval τ_k. Each interval is represented by the sum of its sub-segment durations in PVI, and by their mean in CCI. Since these metrics both capture

useful information, a representation is devised to include both. To allow the different salience of each sub-segment to be taken into account (and particularly to learn to ignore sub-segments the anomalous durations of which suggest alignment errors), the ordinary mean is replaced by a weighted mean in the form of an attention mechanism.

Thus, attention over the sub-segments is used to capture the relative salience of each, and the result concatenated with the total duration of the interval $d(\tau_k)$, to produce the vector $\boldsymbol{x}_k$ representing what we know about the segment k:

$$\boldsymbol{x}_k = \left[\sum_{m=1}^{M^{(k)}} \alpha_m d(\upsilon_m^{(k)}), d(\tau_k) \right] \tag{5.39}$$

where

$$\alpha_m = \frac{\exp f_{att}(d(\upsilon_m^{(k)}), \boldsymbol{\lambda} att)}{\sum_{p=1}^{M^{(k)}} \exp f_{att}(d(\tau_p^{(k)}), \boldsymbol{\lambda} att)} \tag{5.40}$$

To capture dependencies across the whole sequence of durations rather than just pairs of adjacent durations, as was the case with PVI and CCI, the summation of pairwise differences over k is replaced by passing the full sequence of vectors $\boldsymbol{x}_{1:K}$ for each of all vocalic and all inter-vocalic segments in each speaker's speech is passed through one of the sequence models f_{seq} from §4.2.1 or §4.2.3 (bi-directional LSTM or a transformer):

$$\boldsymbol{h}_{1:K_V}^{(V)} = f_{seq}(\boldsymbol{x}_{1:K_V}^{(V)}, \boldsymbol{\lambda}_V) \tag{5.41}$$

$$\boldsymbol{h}_{1:K_C}^{(C)} = f_{seq}(\boldsymbol{x}_{1:K_C}^{(C)}, \boldsymbol{\lambda}_C) \tag{5.42}$$

Further attention mechanisms project each of the resulting sequences to fixed length vocalic and intervocalic features, to capture the relative salience of each segment to the overall rhythm characterisation. For vocalic segments:

$$\tilde{\boldsymbol{h}}^{(V)} = \sum_{k=1}^{K_V} \alpha_k^{(V)} \boldsymbol{h}_k^{(V)} \quad \text{where} \quad \alpha_k^{(V)} = \frac{\exp f_{att}^{(V)}(\boldsymbol{h}_k^{(V)}, \boldsymbol{\lambda}_{att}^{(V)})}{\sum_{p=1}^{K_V} \exp f_{att}^{(V)}(\boldsymbol{h}_p^{(V)}, \boldsymbol{\lambda} att^{(V)})} \tag{5.43}$$

and, similarly, for intervocalic segments:

$$\tilde{\boldsymbol{h}}^{(C)} = \sum_{k=1}^{K_C} \alpha_k^{(C)} \boldsymbol{h}_k^{(C)} \quad \text{where} \quad \alpha_k^{(C)} = \frac{\exp s(\boldsymbol{h}_k^{(C)}, \boldsymbol{\lambda} att^{(C)})}{\sum_{p=1}^{K_C} \exp s(\boldsymbol{h}_p^{(C)}, \boldsymbol{\lambda} att^{(C)})} \tag{5.44}$$

This system is illustrated alongside the original PVI in Figure 5.6. The features $\tilde{\boldsymbol{h}} = [\tilde{\boldsymbol{h}}^{(V)}, \tilde{\boldsymbol{h}}^{(C)}]$ can now be used to represent the speaker's overall rhythm and can be projected through a simple feed forward layer to predict the speaker's grade or accent.

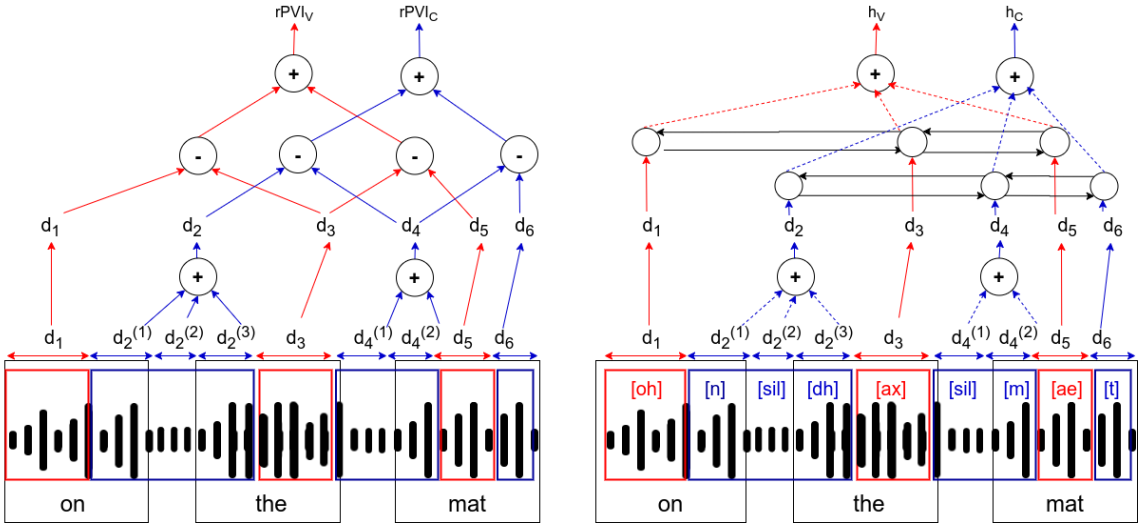


Fig. 5.6 Illustration of extraction of deep rhythm features from sample phrase 'on the mat' (right) compared to the original PVI (left)

As with pronunciation, hard-coded choices of how to aggregate durations have been replaced by more general parameterised operations, which can be trained end-to-end with with the grader. The work in this sub-section, along with its corresponding experiments (see Chapter 6), has been published in [153].

5.2.3 Intonation

In §2.3.6, the literature on intonation assessment was reviewed. It was seen how the probability of voicing $p_v^{(t)}$ and the fundamental frequency $f_0^{(t)}$ at every time position t in the speaker's speech can be extracted and the variation of $f_0^{(t)}$ over time used to characterise intonation.

Baseline features were presented in the form of the mean μ_{f_0}, median $Q_{f_0}^{(0.5)}$, maximum $\max_{1:T}(f_0^{(t)})$ and lower and upper quartiles $Q_{f_0}^{(0.25)}$ and $Q_{f_0}^{(0.75)}$ of f_0 across voiced regions of the speaker's audio. These features can be concatenated into a vector $\boldsymbol{i}$ for each speaker:

$$\boldsymbol{i} = \left[\mu_{f_0}, Q_{f_0}^{(0.5)}, \max_{1:T}(f_0^{(t)}), Q_{f_0}^{(0.25)}, Q_{f_0}^{(0.75)}\right] \tag{5.45}$$

and passed to a fully-connected feed-forward network (§4.2.1) to predict grade y:

$$y = f_{DNN}(\boldsymbol{i}; \boldsymbol{\lambda}_{DNN}) \tag{5.46}$$

As illustrated in Fig. 5.7, f_0 forms contours over the course of the speech based on the message being conveyed and the intonation rules of English. The f_0 contours contain holes, caused by unvoiced regions, which any intonation assessment framework must be able to deal with. While the contours can be annotated using a framework such as ToBI, the goal is to implement an assessment system that doesn't need such annotations to function and which can capture relationships beyond those codified by the annotation scheme.

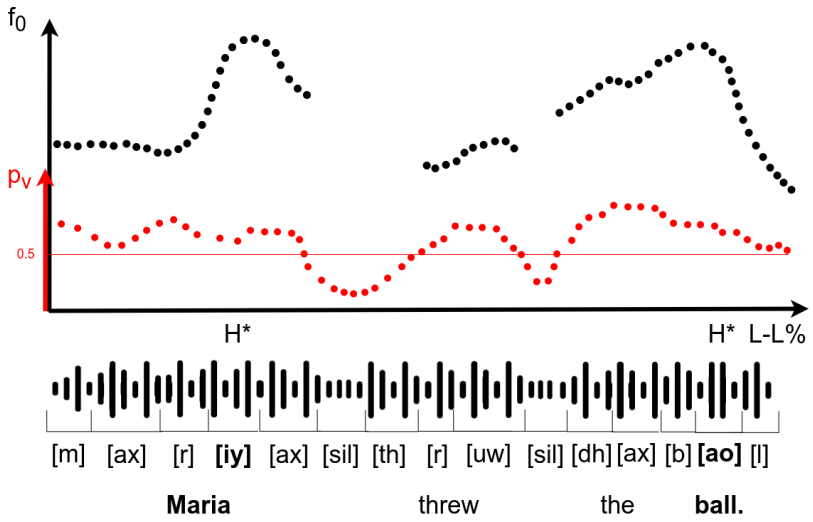


Fig. 5.7 Illustration of words, phones, emphases, ToBI annotations and per-frame f_0 and p_v (black and red dots respectively) for a simple realisation of the statement *Maria threw the ball* (Fig. F.1). Equal emphasis is placed on words *Maria* and *ball* by pronouncing the stressed vowels of each with a higher pitch (H* pitch stress). Pitch drops at the end of *ball* (L-L%) to indicate the end of the statement. f_0 is only extracted for voiced regions (i.e. $p_v \geq 0.5$).

While the baseline features deal with the issue of unvoiced audio, they represent a very coarse approach which doesn't capture the progression of f_0 from phone to phone. A simple way of resolving this issue is to calculate the f_0 statistics for each phone m, concatenating them into a vector $\boldsymbol{i}_m$:

$$\boldsymbol{i}_m = \left[\mu_{f_0}^{(\phi_m)}, Q_{f_0}^{(0.5)}(\phi_m), \max_{t_1^{(\phi_m)}:t_2^{(\phi_m)}} (f_0^{(t)}), Q_{f_0}^{(0.25)}(\phi_m), Q_{f_0}^{(0.75)}(\phi_m) \right] \tag{5.47}$$

To account for unvoiced phones and silences, the same statistics are also calculated for the probability of voicing p_v and concatenated to $\boldsymbol{i}_m$ to produce $\boldsymbol{j}_m$. Proficiency score is then predicted from the sequence $\boldsymbol{j}_{1:M}$ corresponding to each utterance using one of the sequence-to-vector approaches from §4.2.2 or §4.2.3:

$$y = f_{seq2vec}(\boldsymbol{j}_{1:M}; \boldsymbol{\lambda}_{seq2vec}) \tag{5.48}$$

The two systems are illustrated in Figure 5.8 below.

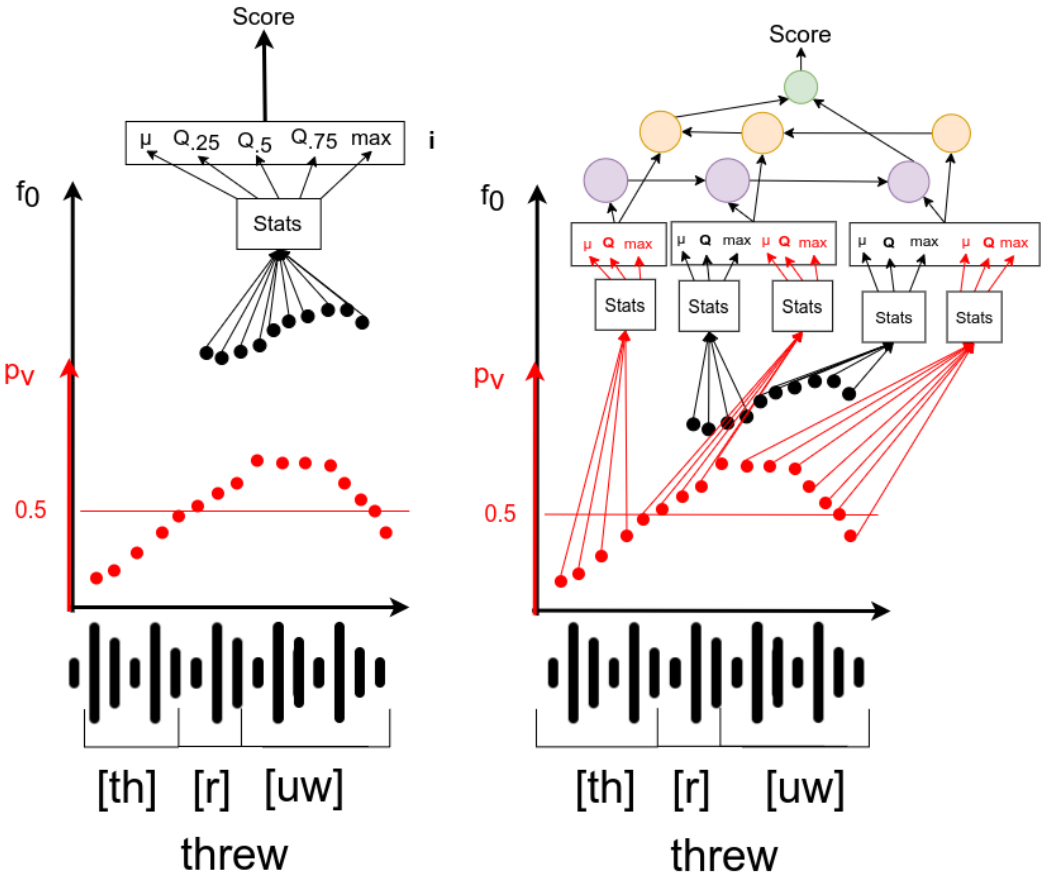


Fig. 5.8 Illustration of intonation assessment for an utterance consisting of the word *threw* using overall f_0 statistics through DNN (left) and phone-wise f_0 statistics through bi-directional RNN (right). Score is predicted starting from per-frame f_0 (black dots) and per-frame probability of voicing (red dots). Note $\boldsymbol{Q} = [Q_{.25}, Q_{.5}, Q_{.75}]$.

The second promising set of methods discussed in §2.3.6 were based on fitting the progression of f_0 over the utterance to smooth contours corresponding to ToBI annotations. As these require supervision with ToBI annotations, they need to be replaced them with a suitable unsupervised contour extraction method.

A standard methodology for extracting contours from time sequences is the Discrete Cosine Transform (DCT) [5], which is equivalent to fitting a sum of cosines by the least-squares algorithm and using the cosine weights to characterise the contours. The holes due to unvoiced regions prevent use of the DCT itself but allow least-squares cosine fitting to be performed directly, thereby implicitly interpolating the unvoiced regions and characterising the speaker's intonation based on its frequency domain properties.

As shown in Appendix G, voiced frames $f_0^{(t_1)}...f_0^{(t_N)}$ can be fit, by the method of least-squares, to a sum of K cosines:

$$\hat{f_0}^{(t)} = \sum_{k=0}^{K-1} x_k C_{k,t} \tag{5.49}$$

by the expression:

$$\hat{x}_k = \sum_{n=1}^{N} u_{n,k} f_0^{(t_n)} \tag{5.50}$$

where $u_{n,k}$ is the nth element of the kth row of the matrix $(\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T$, such that:

$$\hat{\boldsymbol{x}} = (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{f} \tag{5.51}$$

where $\boldsymbol{f}$ is a vector of the frames $f_0^{(t_1)}...f_0^{(t_N)}$, and:

$$c_{k,t} = \sqrt{\frac{1+\min(k,1)}{K}} \cos\left[\frac{\pi}{K}\left(t+\frac{1}{2}\right)k\right] \quad k = 0,1...K-1 \tag{5.52}$$

such that if the audio signal were uninterrupted and evenly sampled the result would be equivalent to a DCT:

$$\hat{\boldsymbol{x}} = \boldsymbol{C}^T\boldsymbol{f} \tag{5.53}$$

The length-K vector $\hat{\boldsymbol{x}}$ can now be used to predict proficiency score:

$$y = f_{DNN}(\hat{\boldsymbol{x}}; \boldsymbol{\lambda}_{DNN}) \tag{5.54}$$

Equation 5.50 demonstrates that cosine-fitting is equivalent to a series of weighted sums of $f_0^{(t)}$ by different weight schemes dependent on cosine encodings of position t. Each weight scheme extracts effects at different frequencies, accounting for holes in the contour.

This can be thought of as a special case of multi-head attention with positional encoding (see §4.2.3) followed by scaling of each head:

$$\hat{h_k} = u_k \sum_{n=1}^{N} a_{n,k} f_0^{(t_n)} \quad \text{where} \quad a_{n,k} = \frac{\exp f_{att}\left(f_0^{(t_n)}, p_v^{(t_n)}, C_{k,t_n}; \boldsymbol{\lambda}_{att}^{(k)}\right)}{\sum_{m=1}^{N} \exp f_{att}(f_0^{(t_m)}, p_v^{(t_m)}, C_{k,t_m}; \boldsymbol{\lambda}_{att}^{(k)})} \tag{5.55}$$

where u_k is a hyperparameter weight and f_{att} represents additive attention (as it is the magnitude of f_0 and p_v that determines salience rather than the difference between them).

This more general system can be trained end-to-end and allows the effect of the value of $f_0^{(t)}$ and the probability of voicing p_v to be taken into account in addition to the position t in determining the salience of f_0 at each frame t for each head k.

The two systems are contrasted in Figure 5.9 below.

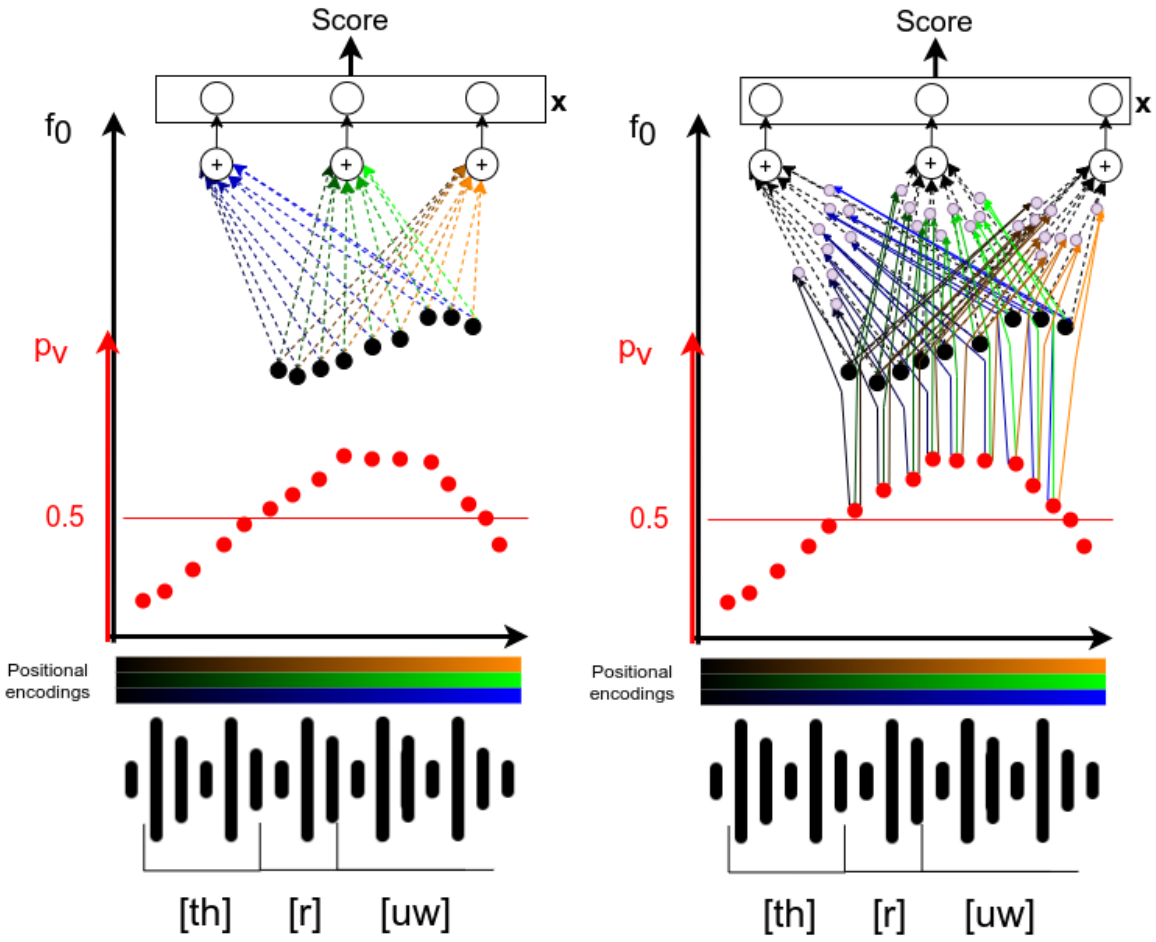


Fig. 5.9 Illustration of intonation assessment for an utterance consisting of the word *threw* using cosine fitting (left) and multi-head sequence-to-vector attention (right). Intonation grade is predicted from per-frame f_0 (black dots) and per-frame probability of voicing (red dots).

A disadvantage of both approaches is that, for typical utterance lengths, the number of frames fed into the attention mechanism will be very large, making training computationally difficult and slow and risking problems associated with very long sequences, such as vanishing gradients. Further, the locations of phone boundaries, which also carry salient information about pitch, are not taken into account.

To account for these issues, a third and final variant of the architecture is defined, consisting of two stages.

First a sequence-to-vector mapping is learnt for the sequence of frame-level f_0 and p_v in each phone:

$$\boldsymbol{h}_m = f_{seq2vec}([f_0^{(1:T_m)}, p_v^{((1:T_m)}, C_{k,t_n}]; \boldsymbol{\lambda}_{seq2vec}^{(phone)}) \tag{5.56}$$

This sequence is then passed through a second sequence-to-vector transformation to predict grade:

$$y = f_{seq2vec}(\boldsymbol{h}_{1:M}; \boldsymbol{\lambda}_{seq2vec}) \tag{5.57}$$

As in §5.2.1, the rough cost surface of this hierarchical network is likely to be highly sensitive to initialisation, suggesting advantages from initialising $\boldsymbol{\lambda}_{seq2vec}$ using one or more of the previous tasks. Thus, simpler longer-term f_0 patterns can be expected to be learned during the training of the simple graders, and more complex shorter-term patterns learned during the fine-tuning stage.

5.3 Grader Combination

In §5.2, end-to-end graders were presented for the views of pronunciation, rhythm and intonation, aiming for their inputs and structures to constrain them to only extract features representative of their respective views.

View specificity in the rhythm grader is achieved by limiting the input to consist exclusively of the durations of phones and silences, grouped into consonant and inter-consonant intervals, so that the grader doesn't have a choice but to grade on the basis of patterns of interval duration. Similarly view specificity in the intonation grader is established by limiting the input to frame-level f_0 and probability of voicing.

In the pronunciation grader, acoustic observations are grouped by phone label, attended over, and Euclidean distances between phone representations used to predict grade. The goal is for information from the observation vectors to only be preserved insofar as it characterises the way the speaker pronounced each phone relative to other phones. Phones that were wholly absent from the speech are marked by setting their distances to -1, which may cause the model to also indirectly model utterance and word length and richness of vocabulary, in turn causing unwanted overlap with the view of text.

In addition to these systems, a text grader is adapted from Raina et al. [223][1] which consists of an LSTM with attention over hidden representations, the inputs to which are word embeddings obtained by passing the words of each utterance through a trained BERT language model [69] (see (§4.2.3). The text grader only sees the identity of words and so, as with the rhythm and intonation systems, is limited to only grade on the basis of message construction.

The four systems are illustrated in Figure 5.10.

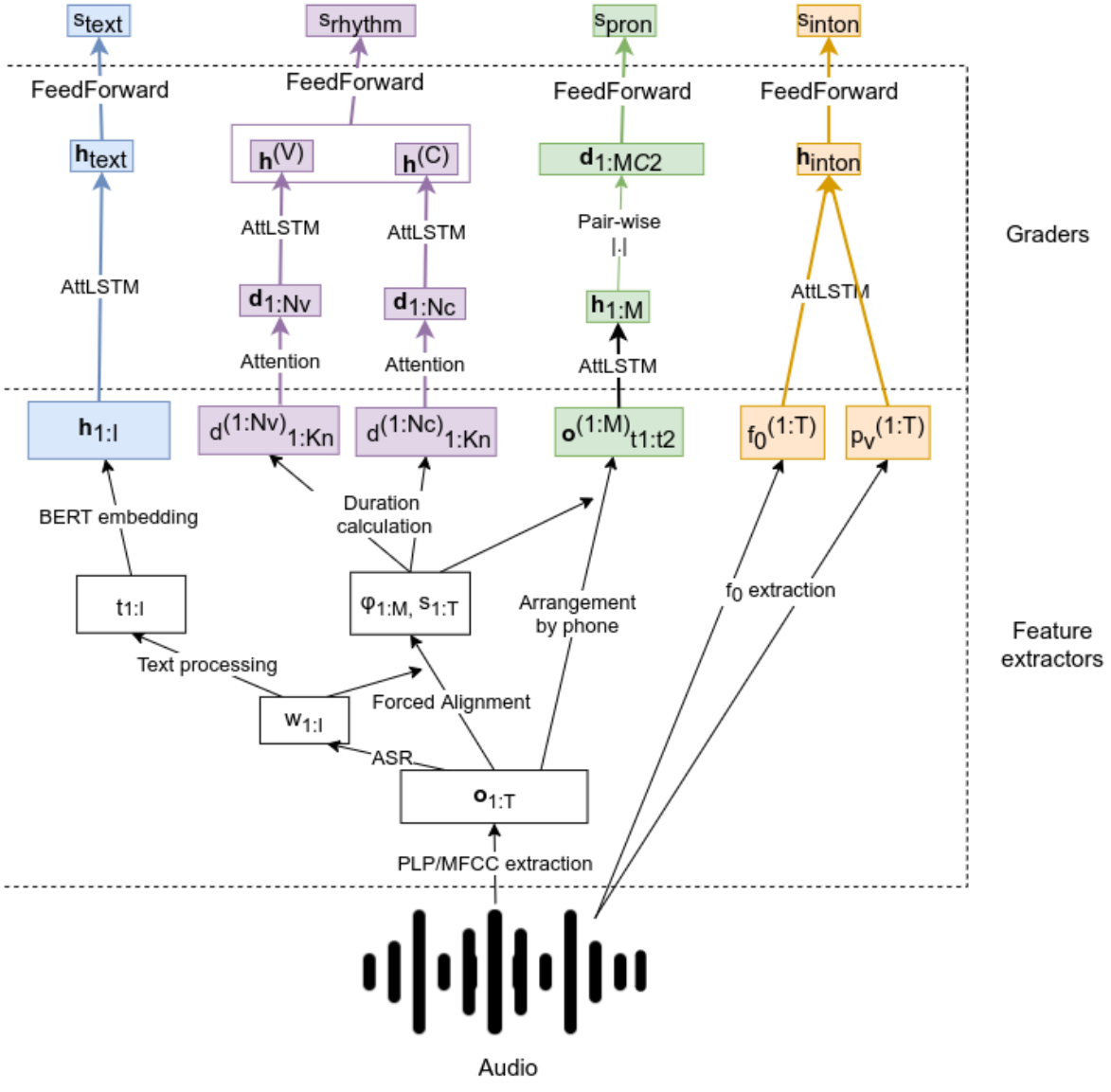


Fig. 5.10 Illustration of text, rhythm, pronunciation, and intonation grading.

[1]Note that in this work a single system is trained to assign a single grade to each speaker on all their speech while in the original paper separate graders were trained for each of five sections of a specific exam, which therefore yielded superior results to those seen here.

Three methods of combining the graders are investigated. The first and simplest is to compute the mean of the scores they predict for each speaker, mirroring Equation 5.7:

$$\hat{\bar{y}}^{(n)} = \frac{1}{J}\sum_{j=1}^{J}\hat{y}_j^{(n)} \tag{5.58}$$

The second mirrors the method used to combine hand-crafted features and consists of concatenating the intermediate representations of each view taken from its respective grader into a single representation:

$$\boldsymbol{v}^{(n)} = [\boldsymbol{v}_1^{(n)}...\boldsymbol{v}_J^{(n)}] \tag{5.59}$$

which is then passed through a single DNN grader:

$$\mathscr{G}(\boldsymbol{v}^{(n)}, \boldsymbol{\theta}) \rightarrow \hat{\bar{y}}^{(n)} \tag{5.60}$$

The third approach builds on the first, but allows different weights to be assigned to each view for each speaker, using an attention mechanism over view-specific scores:

$$\hat{\bar{y}}^{(n)} = \sum_{j=1}^{J}\alpha_j^{(n)}\hat{y}_j^{(n)} \tag{5.61}$$

where:

$$\alpha_j^{(n)} = \frac{\exp(s_j^{(n)})}{\sum_{n=1}^{I}\exp(s_j^{(n)})} \tag{5.62}$$

and $s_j^{(n)}$ is predicted from the intermediate representation by a feed-forward network:

$$\mathscr{A}(\boldsymbol{v}_j^{(n)}, \boldsymbol{\theta}) \rightarrow s_j^{(n)} \tag{5.63}$$

Each of the three combination systems can either be trained on its own using the outputs of the trained single-view graders or together with all the single-view graders as a single system in an end-to-end configuration. The latter configuration would allow each single-view grader to be fine-tuned to predict grades complementary to the others.

5.4 Chapter summary

This chapter investigated techniques to automatically grade speakers on the basis of their pronunciation, rhythm and intonation, using the deep learning methodologies reviewed in Chapter 4, and holistically, by combining single-view systems.

In §5.1, it was proposed that single-view graders that limit the information they extract to be indicative of a particular view can predict single-view grades, even when trained on only holistic grades. Single-view graders for each of pronunciation, rhythm and intonation were then proposed in §5.2. A set of features was first defined to characterise each of the three aspects for use in two-stage graders, where features are first extracted and then fed into a DNN to predict score. End-to-end systems were then crafted for each of the three aspects, generalising the feature extraction mechanism and allowing it to be tuned to the task and optimised jointly with the grader.

In §5.2.1, pronunciation was characterised by representing the way the speaker pronounces each of the phones of English relative to each of the others. This is first achieved by computing relative entropies between distributions of the acoustic observations of all the times the speaker used each phone. The log-one-plus symmetric K-L divergences between multi-variate Gaussian distributions of per-frame feature vectors then become features for a DNN. An end-to-end version of this framework is then presented, where representations of the way the speaker pronounces each phone are obtained by attending over representations of each instance of each phone, themselves projected by a sequence-to-vector transformation from frame-level acoustic observations. Instead of relative entropies, Euclidean distances between all possible pairs of phone representations are projected through a feed-forward layer to predict score. The weights of the phone instance embeddings are initialised by training a Siamese network to classify whether given pairs of phone instances belong to the same or different phones.

In §5.2.2, rhythm was characterised by the pattern of durations across syllables in the speaker's utterances. This is first represented by a series of ratios of adjacent vocalic and intervocalic interval durations, which are generalised into an end-to-end system that attends over phone and silence durations to yield an interval representation, then uses sequence-to-vector transformations to represent the progression of vocalic and intervocalic intervals across the utterance. In §5.2.3, intonation is characterised by patterns of fundamental frequency. Two types of DNN graders are defined, one using speaker-level statistics of f_0 and one using coefficients obtained by fitting cosines to the f_0 progression of each utterance. The first is generalised by a sequence-to-vector projection of phone-level f_0 statistics and the second by a sequence-to-vector projection if frame-level f_0 and probability of voicing.

Finally, in §5.3, three methods of combining the deep graders with each other and with a text grader were introduced. The first was a simple averaging of scores, the second a DNN trained on the inputs to the penultimate layer of each constituent grader, and the third a a way to join graders together and train them jointly in an end-to-end fashion for optimal combined score prediction.

Experiments to compare the performance of the two-stage and end-to-end single-view graders, investigate the extent to which the single-view graders are indeed capturing different views of proficiency, and compare the combined graders to each other and to baseline holistic graders, are described in Chapter 6.

Chapter 6

Experiments on Spoken Language Proficiency Assessment

Chapter 5 proposed a framework for multi-view spoken language proficiency assessment. For each of the views of pronunciation, rhythm, and intonation, end-to-end trainable neural models were presented alongside two-stage graders based on feeding hand-crafted features through DNNs. Both sets of graders were designed to limit the information available for grading to only that indicative of their respective views. It was thus hypothesised that the graders would be able to predict single-view grades even when trained on holistic grades. It was further expected that the tunability of the end-to-end graders would allow them to be better at predicting human-assigned grades and better generalisable to different tasks and datasets than their two-stage counterparts. Three methods of combining single-view graders with each other for holistic grading were then proposed, hypothesising that the combined systems would be able to outperform both two-stage multi-view holistic graders, as well as direct holistic graders that did not take individual views into account.

This chapter presents experiments conducted to test these hypotheses. The data used for the experiments is first outlined in §6.1. The baseline systems used to evaluate each novel system are then presented in §6.2. Experiments comparing the end-to-end graders to their two-stage counterparts are reported in §6.3. Performance of each grader was evaluated by the effectiveness with which it predicted human-assigned holistic grades in evaluation sets held-out from but comparable to the data on which it was trained. Generalisability of the models was evaluated by their ability to predict grades on held-out data sets not comparable to the training data. Transferability to different tasks was evaluated on its performance on data from different exams and, for the pronunciation grader, on the related task of predicting each speaker's native language (L1). The relationship between the two-stage and end-to-end pronunciation graders was further examined §6.4.

The experiments reported in §6.5 then attempted to investigate whether the single-view graders trained on holistic grades are indeed grading their intended aspects of proficiency. Finally, §6.6 reports experiments comparing the methods of combining the graders to each other and to the baselines.

6.1 Data

The speaker data used in the experiments of this chapter was taken from candidate responses to the spoken component of the Linguaskills (LS) [176] and Business Language Testing Service (BULATS or BLT) [43] English proficiency tests, provided by Cambridge English Language Assessment. Each test has five sections. Section A consists of short responses to prompted questions, Section B asks candidates to read 8 sentences aloud, while Sections C-E consist of spontaneous responses of several sentences in length to a series of spoken and visual prompts. Each response (of which there are usually 2-6 per section) corresponds to one utterance.

Each speaker has been assigned a score by operational human graders (i.e. those who provide the grades to candidates actually taking the examination), based on their holistic proficiency (original grades). These were averaged to produce the speaker's overall score. A small subset of the speakers were also assigned holistic scores by a different group of expert human graders, whose inter-annotator agreement is much higher than that of the original operational graders (expert grades). Candidates were scored on a 0-6 scale, mapping to CEFR levels [57] as shown in Table 6.1.

BLT score range	Level description	CEFR level
5.8 - 6.0	Upper advanced	C2
5.0 - 5.8	Advanced	C1
4.0 - 5.0	Upper intermediate	B2
3.0 - 4.0	Intermediate	B1
2.0 - 3.0	Elementary	A2
0.0 - 2.0	Beginner	A1

Table 6.1 Mapping from BLT/LS proficiency scores to CEFR levels (adapted from [38])

Metadata including each candidate's L1, gender and country of origin was also provided. The data was segmented into non-overlapping sets, each gender balanced and spread across CEFR levels, with some used to train and develop ASR systems (see Appendix C) and others to train and evaluate graders.

The sets used to train and evaluate graders in this chapter are outlined in Table 6.2. There are three pairs of matched datasets, each consisting of a larger training set graded by operational graders and a smaller evaluation set graded by experts. Given the scarcity and expense of obtaining expert annotations for an operational system, this setup ensures that models are trained on the data most likely to be available to train systems in practice but evaluated on the most reliable ground-truth available. The first pair of datasets represent speakers across a common L1 (Gujarati), while the other two are cross-L1 sets, with a similar mix of L1s in the training and testing set. By training and evaluating on unmatched sets, it will be possible to test the generalisation performance of each system.

Set	Source	#Speakers	Graders	L1s
BL_GRD_GJ	BLT	1013	original	Gujarati
BL_EVL_GJ	BLT	223	expert	Gujarati
BL_GRD_M1	BLT	994	original	Viet., Thai, Ar., Fr., Du., Pol.
BL_EVL_M	BLT	226	expert	Viet., Thai, Ar., Fr., Du., Pol.
LS_GRD	LS	4092	original	Viet., Thai, Ar., Span., Pr., Hi., Jp. (86%) 43 other L1s (rest)
LS_EVL	LS	248	expert	Viet., Thai, Ar., Span., Pr., Hi., Jp.

Table 6.2 BLT and Linguaskills (LS) datasets used for training and evaluating automatic proficiency graders in this thesis. Datasets annotated by the original operational graders are used for training while those annotated by experts are used for evaluation. L1 Key: Ar. = Arabic, Fr. = French, Du. = Dutch, Hi. = Hindi, Viet. = Vietnamese, Pr. = Portuguese, Jp. = Japanese., Span. = Spanish, Pol. = Polish

Two further cross-L1 datasets are used for experiments on L1 classification (Table 6.3).

Set	Source	#Speakers
BL_GRD_M2	BLT	11259
BL_EVL_M2	BLT	5082

Table 6.3 BLT sets used for training and evaluating L1 classifiers. L1s are Tamil, Telugu, Malayalam, Kannada, Gujarati, Hindi, Bengali, Marathi, Spanish, French, Portuguese, Italian

The utterances in each dataset were first recognised and force-aligned to obtain the time-aligned word and phone sequences needed to extract the input features of each grader. In each case one of the systems in Appendix C was used, specifically the GMM-HMM (GH), hybrid DNN-HMM (DH) or TDNN-F (TD) acoustic model, with either a phonetic (ph) or graphemic (gr) lexicon. The 47-phone broad transcription alphabet listed in Table D.1 of Appendix D is used throughout.

6.2 Baseline Systems

In §5.2, hand-crafted baseline feature sets for use with DNNs were introduced alongside hierarchical trainable feature extractors to grade proficiency on the basis of each of pronunciation, rhythm and intonation. The effectiveness of the latter approach can thus be investigated by comparing each hierarchical single-view model to its respective two-stage baseline. The Gaussian Process grader from Kyriakopoulos et al. [154] was also reproduced against which to compare the performance of both types of deep learning models while keeping features unchanged. Its mean prediction $\hat{y}_i$ for input $\boldsymbol{x}_i$ given training data $\boldsymbol{x}_{1:N}^{(tr)}, \boldsymbol{y}_{1:N}^{(tr)}$ is given by:

$$\hat{y}_i = \boldsymbol{k}(\boldsymbol{x}_i, \boldsymbol{x}_{1:N}^{(tr)})^T (\boldsymbol{K}(\boldsymbol{x}_{1:N}^{(tr)}, \boldsymbol{x}_{1:N}^{(tr)}) + \sigma_0^2 \boldsymbol{I})^{-1} \boldsymbol{y}_{1:N}^{(tr)} \tag{6.1}$$

where the covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ is a radial basis function (RBF):

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_y^2 \exp\left(-0.5||\boldsymbol{x} - \boldsymbol{x}'||/l^{-2}\right) \tag{6.2}$$

and σ_0, σ_y, and l are hyper-parameters tuned on the training set.

Three methods for combining the single-view graders were proposed in §5.3. To investigate whether these combined systems outperform more direct approaches to holistic grading, two baselines were implemented inspired by the literature (§5.7), one based on a hand-crafted feature extractor and one based on a trainable feature extractor. First, the ALTA baseline features (Appendix H), a concatenation of hand-crafted features for the views of tempo, rhythm, pronunciation and text, developed by other members of the ALTA project, were used to train a DNN grader. Performance is evaluated by the metrics shown in Table 6.4. It is seen that this baseline system is able to predict holistic score well, correlating strongly to actual grades and falling within 1 point (i.e. approximately one CEFR level) of the ground-truth score 91.5% of the time. The second-stage DNN is seen to be somewhat sensitive to random initialisation, but not so much as to risk serious performance reduction.

Model	PCC	MSE	MAE	%<0.5	%<1.0
base	0.862	0.359	0.454	62.1	91.5
	±0.02	±0.051	±0.023	±1.7	±1.5

Table 6.4 Performance of the DNN grader with ALTA baseline features (base) trained using exponential learning rate with five different random seeds on BL_GRD_M1 and evaluated on BL_EVL_M. Features are derived from MFCC-13 vectors from the output of TD-gr aligned with GH-ph. Accuracy of mean predictions is evaluated using Pearson correlation coefficient (PCC), mean squared error (MSE), mean absolute error (MAE) and by the percentages of predictions with an error below 0.5 (%<0.5) and 1.0 (%<1.0). Sensitivity to random initialisation is measured by the standard deviation of each metric across the five runs (±).

Next, building on the logic of the i-vector approaches of Takai et al. [259] and Cheng et al. [51], the x-vector extraction system developed by Wang et al. [276] (in turn based on Snyder et al. [250]) was employed to extract speaker-level features from per-frame filterbank features (see Appendix A).

In its original form, this feature extractor was trained on a speaker classification target and thus learns to extract features that characterise the way the particular speaker produces speech overall. In addition to the original extractor, Wang provided versions trained on the tasks of L1 classification and grading. The latter system thus directly learns to extract features representative of the speaker's holistic proficiency. All three were trained on BULATS data separate from the grader training and evaluation sets used in this chapter.

Features extracted using each of these systems were then used to train DNN graders with hidden layers 30×30. Results are compared to each other and against the ALTA baseline features in Table 6.5. X-vectors extracted using all three criteria performed reasonably well. This is consistent with the information extracted to represent the unique way each speaker speaks being also representative of their L1 and their proficiency level.

System	PCC	MSE	MAE	%<0.5	%<1.0
base	**0.862**	**0.359**	**0.454**	**62.1**	**91.5**
	±0.02	±0.051	±0.023	±1.7	±1.5
speaker	0.766	0.590	0.605	50.0	80.8
	±0.029	±0.065	±0.039	±4.5	±2.6
L1	0.797	0.512	0.552	56.3	82.1
	±0.049	±0.095	±0.05	±3.8	±4.4
grader	**0.806**	**0.489**	**0.556**	**52.2**	**83.9**
	±0.0099	**±0.022**	**±0.0091**	**±1.6**	**±1.5**

Table 6.5 Performance of DNN graders using x-vectors extracted based on speaker classification, L1 classification, and grade prediction criteria, compared against the baseline ALTA features (base), after recognition with the TD-gr ASR and alignment with the GH-ph acoustic model, trained on BL_GRD_M1 and evaluated on BL_EVL_M.

As expected, the x-vectors extracted with a grading criterion yielded higher performance and less sensitivity to random initialisation than those extracted with the other two criteria. In turn, the hand-crafted model significantly outperformed the x-vectors across all metrics. This is consistent with domain-aware view-specific feature extraction across multiple views resulting in more and better information about holistic grade than extracting features for holistic proficiency directly.

Figure 6.1 shows the performance of the three x-vector systems on the task of L1 classification, which was used to test the tunability of the end-to-end systems to different tasks.

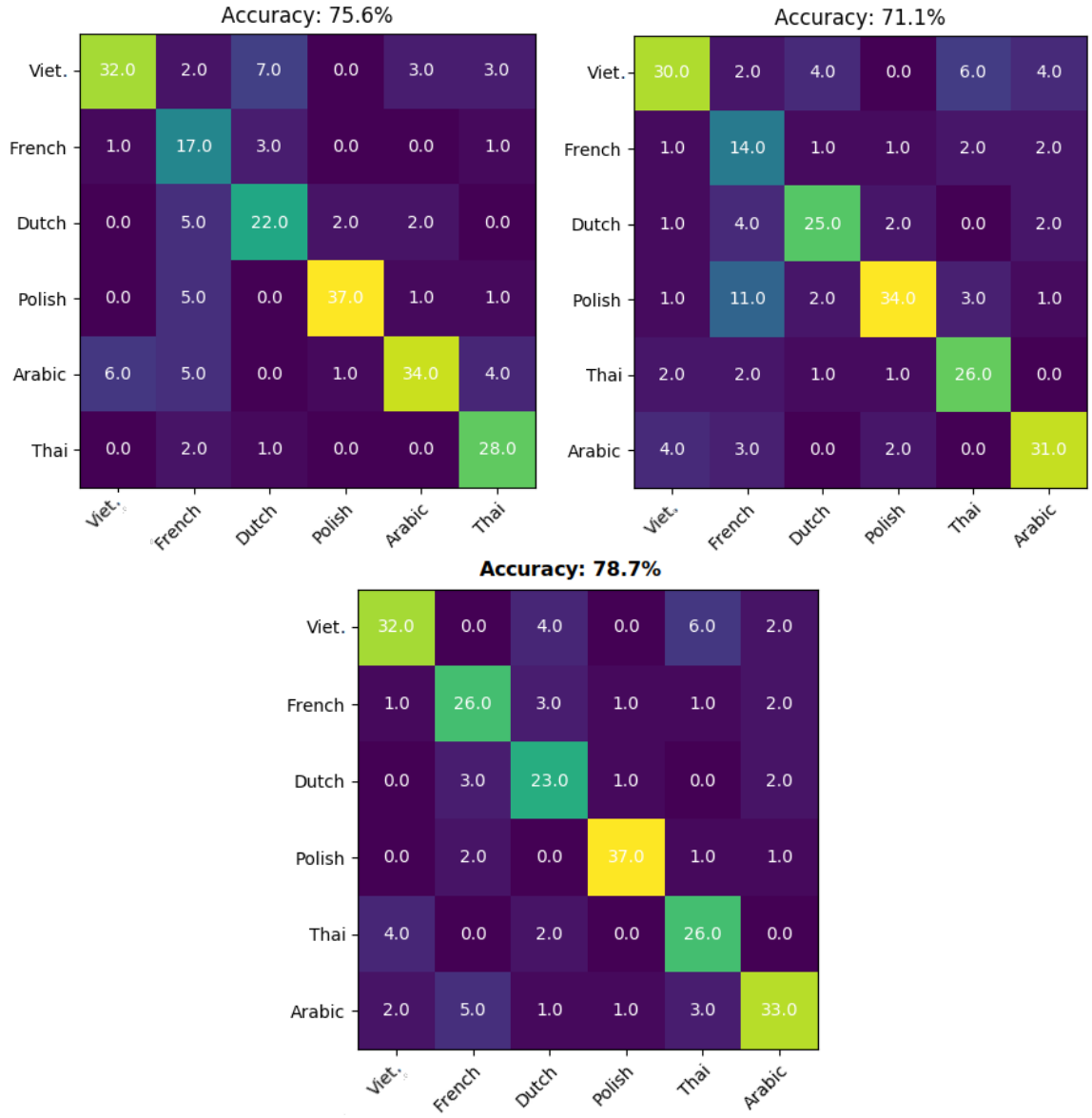


Fig. 6.1 Confusion matrices and overall % matches for DNN classifiers using x-vectors extracted based on speaker classification (top left), grade prediction (top right) and L1 classification (bottom) criteria, after recognition with the TD-gr ASR and alignment with the GH-ph acoustic model, trained with five different random seeds on BL_GRD_M1 and evaluated on BL_EVL_M.

As in the case of grading, x-vectors extracted using all three methods performed well, again consistent with the existence of a natural low-dimensional representation of the way a

speaker speaks, which is similarly indicative of their L1, proficiency and unique identifiable voice. As expected, the x-vectors extracted using the L1 criterion demonstrate the superior performance of the three.

The performance of the ALTA baseline features on the same L1 classification task is shown in Figure 6.2 below.

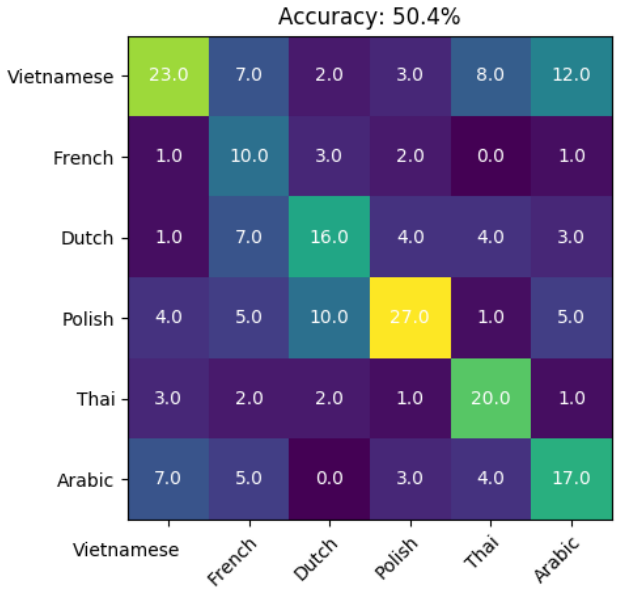


Fig. 6.2 Confusion matrix and overall % matches for DNN classifier using ALTA baseline features, after recognition with the TD-gr ASR and alignment with the GH-ph acoustic model, trained with five different random seeds on BL_GRD_M1 and evaluated based on average of predictions on BL_EVL_M.

In contrast to the grading experiments, where the ALTA baseline features were seen to outperform the best x-vector system by a large margin, the performance of the same features on this L1 classification task is much worse than all three types of x-vectors. This is consistent with the hand-crafted features being honed to exclusively represent proficiency to a much greater extent than the neural feature extractors and therefore carrying less information that is usable for L1 detection.

6.3 Performance of single-view graders

This section describes experiments that were conducted to investigate the performance of the graders introduced in §5.2. The aim is to evaluate the hypotheses that the deep end-to-end graders are more powerful, better at generalising and more tunable to different tasks than

their corresponding two-stage baselines. As human-annotated single view grades are not available, prediction of holistic grades will be used as a proxy for evaluation.

For the purposes of the experiments reported in the section, the HMM Toolkit (HTK) [296] was used for MFCC/PLP extraction and forced alignment, while graders and classifiers were implemented using TensorFlow [1] in Python. The training criteria used were mean squared error (MSE) in the case of graders and cross-entropy for classifiers (§4.3).The networks were trained in batches of 100 speakers at a time. Each was trained five times with different random seeds, 50% dropout at each layer, and a random 10% held out validation set for early stopping if loss on it increased thrice in a row (checked every 10 iterations).

Average (ensemble) predictions were evaluated against expert human scores using Pearson correlation coefficient (PCC), mean squared error (MSE), mean absolute error (MAE) and by the percentages of errors below 0.5 (%<0.5) and 1.0 (%<1.0). Sensitivity to random initialisation was measured by the standard deviation of each metric across the five runs (marked by $\pm$).

6.3.1 Pronunciation

The discussion in §5.2.1 proposed two systems for pronunciation grading. The first, two-stage, method is to train a DNN on phone distances (log of one plus the KL divergence between multivariate Gaussians of observations of each possible pair of phones). The second system is an end-to-end grader (based on Euclidean distances between attention over embeddings of all instances of each possible pair of phones), initialised using a Siamese network.

The second stage of the first system was implemented using a DNN with a hidden structure of 30×30, as well as using the baseline Gaussian Process [154]. The deep phone distance grader used the weights of the trained Siamese network to initialise the phone instance embedding stage and the trained phone distance grader to initialise the final feed-forward layer (which has the same 30×30 structure as the DNN).

Experiments reported in §J.1 of Appendix J established that the choice of speech features (MFCC vs PLP) makes little difference to performance while those in §J.2 confirmed that the phone distance configuration is superior to the alternative of grapheme distances. As further reported in Appendix J, the use of a bi-directional LSTM with additive attention over its final layer (§4.2.3) was found to be the optimal architecture for the sequence-to-vector transformations for the Siamese network and deep grader. Batch normalisation (§4.4.2) was also found to significantly boost performance and was therefore applied in all experiments with deep graders. Following §J.6, an exponential learning rate schedule (§4.4.4) was applied unless otherwise specified.

Applying a loss penalty term on the entropy of the attention weights was found to be necessary for the network optimisation to converge. The cost function thus becomes:

$$C(\boldsymbol{\lambda}, \mathscr{S}_{train}) = MSE_{train} - \beta \sum_{m=1}^{M} \sum_{n=1}^{N_m} \alpha_{nm} \log \alpha_{nm} \tag{6.3}$$

where α_{nm} is the weight of the nth item of the attention mechanism for the mth phone label in the network and β is a hyperparameter the optimal value of which is selected by grid search.

Having developed the DNN phone distance grader and the deep phone distance grader, their performance was then evaluated against each other and the baseline Gaussian Process. Systems were trained on BULATS and Linguaskills training sets and evaluated on corresponding evaluation sets (taken from the same task across similar L1s). To test each system's ability to generalise, the model trained on each training set was evaluated on the evaluation set corresponding to the other one. The results are displayed in Table 6.6.

Train set	Test set	Model	PCC	MSE	MAE	%<0.5	%<1.0
BL_GRD_M1	BL_EVL_M	DP	**0.82**	**0.531**	0.573	53.6	83.5
		DNN	0.785	0.556	**0.552**	**59.4**	**86.6**
		GP	0.811	**0.526**	0.559	54.5	83.9
LS_GRD	LS_EVL	DP	**0.777**	**0.499**	**0.567**	50.7	83.6
		DNN	0.719	0.524	0.584	**52.1**	**84.9**
		GP	**0.786**	0.524	0.582	46.6	83.6
BL_GRD_M1	LS_EVL	DP	**0.631**	**0.637**	**0.522**	**67.1**	**89.0**
		DNN	0.628	1.06	0.777	41.1	72.6
		GP	0.482	1.54	0.977	32.9	64.4
LS_GRD	BL_EVL_M	DP	**0.689**	**0.904**	**0.742**	**43.3**	**72.3**
		DNN	0.486	1.08	0.816	42.9	64.7
		GP	0.0465	1.46	0.984	30.8	57.6

Table 6.6 Performance of deep phone distance grader (DP) and each of DNN and Gaussian Process (GP) phone distance graders, trained on BULATS and Linguaskills datasets and evaluated on matched and non-matched evaluation sets, after recognition with TD-gr and alignment with GH-ph.

While the Gaussian Process outperformed the DNN on the first two experiments, it yielded poor performance on the unmatched datasets. This is consistent with the DNN's parametric representational learning being better than the Gaussian Process at capturing the underlying relationship between the variables in a way that can be generalised to out-of-domain data. On the matched task, the deep phone distance grader outperformed the DNN significantly

in terms of both PCC and MSE, supporting the hypothesis that the feature extractor can be tuned end-to-end to learn to extract features better representative of proficiency. On the unmatched datasets, the deep phone grader outperformed both other systems by a large margin, suggesting that its increased representational capacity and freedom to tune the feature extraction stage allows it to capture the underlying relationship even better and therefore overfit less to the data it is trained on.

Inspecting Figure 6.3, it can be seen that the points for the deep phone grader are, in both cases, less spread out that those for the shallow grader, but appear to have a larger systematic bias (i.e. the intercept of their line of best fit would be further from zero). This suggests that the deep phone grader predictions are less well calibrated (more systematically biased) for both tasks, but are more precise (i.e. have lower aleatoric uncertainty). This would explain why the PCC performance of the deep grader relative to the DNN is stronger than that measured by other metrics.

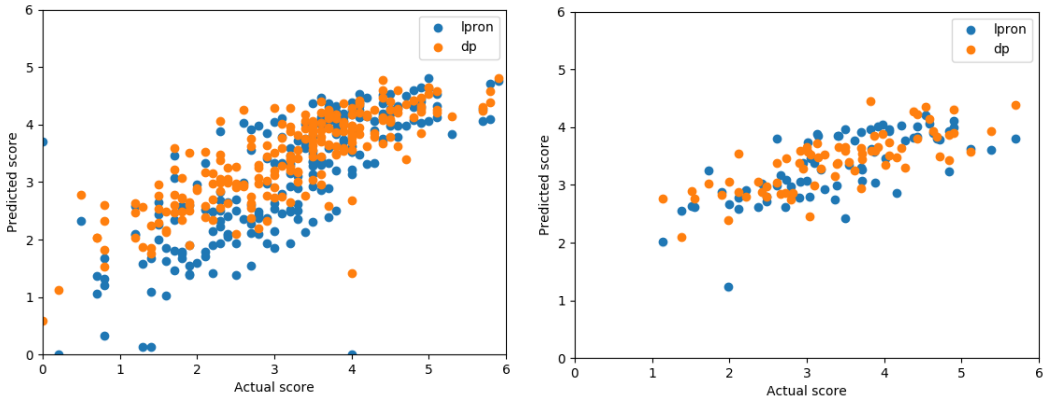


Fig. 6.3 Average (ensemble) predicted scores from deep phone distance grader (dp) and DNN phone distance feature (log K-L divergence) grader (lpron), trained with five different random seeds on BL_GRD_M1 (left) and LS_GRD_M (right) and evaluated on corresponding evaluation sets, plotted against expert human scores.

It follows that the end-to-end grader is both better performing and more generalisable than its end-to-end counterpart at the task of grading.

To evaluate the ability of the deep phone distance network to be tuned to tasks other than assessment, the output layers of both the deep grader and the phone distance DNN are modified to allow them to be used for 6-way L1 classification on the same data. The use of phone distances for L1 detection and country of origin classification was investigated on larger datasets in the experiments reported in Appendix I, where they were seen to be predictive of the speaker's native language, language family and country of origin. These results confirm the power of phone distances as representations of the speaker's pronunciation.

As illustrated in Figure 6.4, the deep phone system outperformed both the phone distance DNN and the x-vector system by a large margin at this L1 classification task. This result is consistent with the greater tunability expected of the deep phone distance extraction process compared to the phone distance DNN and its domain-aware information extraction compared with the x-vector DNN.

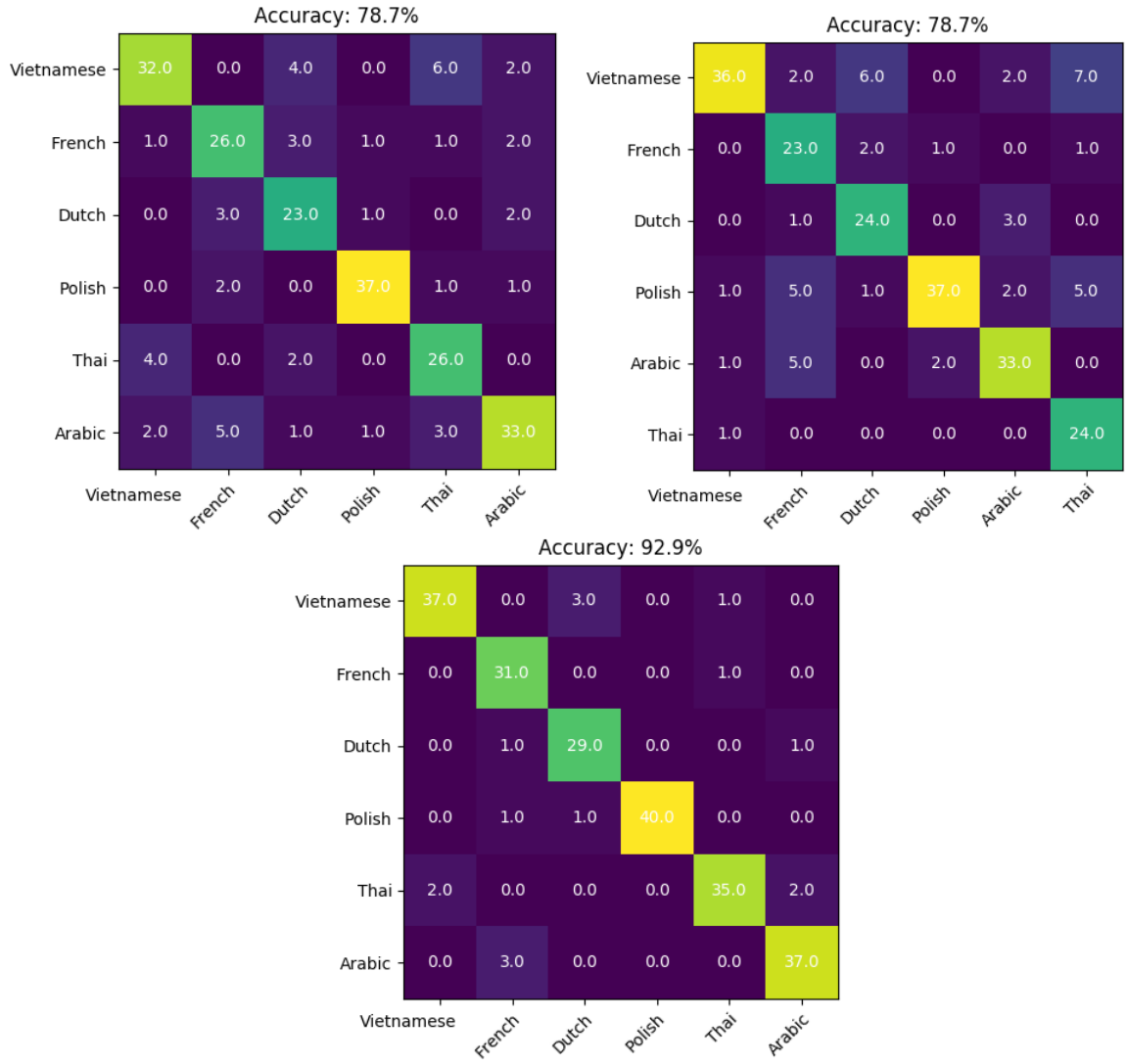


Fig. 6.4 Confusion matrices and overall % matches for DNN L1 classifier using x-vectors extracted based on speaker classification (top left), DNN L1 classifier using phone distance (log KL divergence) features (top right) and deep phone distance L1 classifier (bottom), after recognition with the TD-gr ASR and alignment with the GH-ph acoustic model, trained with five different random seeds on BL_GRD_M1 and evaluated on BL_EVL_M.

The phone distance DNN in turn has comparable performance to the x-vector DNN, which is can be explained as a trade-off between the benefits of the x-vector's increased

tunability and the phone distances' domain-aware information extraction. The larger gap in performance between the deep phone classifier and the other two compared to the case of the grader is also consistent with performance of the grader being undermined by the calibration issue which does not occur with the L1 task.

6.3.2 Rhythm

In §5.2.2, a DNN grader using the baseline features listed in Appendix §E and an end-to-end system generalising them were proposed for rhythm grading. The systems were implemented using feed-forward stages consisting of two 30-unit hidden layers. Following the discussion in Appendix J, the deep system was implemented using bi-directional LSTMs with additive attention over their final layer for the sequence-to-vector stages, batch normalisation and an exponential learning rate schedule.

The performance of each system was evaluated on a matched dataset task, where the train and test set were taken from the same examination and have similar L1s, as well as on an unmatched dataset task, to investigate the system's ability to generalise.

As seen in Table 6.7, the deep grader considerably outperformed the DNN on the unmatched task, indicating superior generalisability.

Test set	Model	PCC	MSE	MAE	%<0.5	%<1.0
BL_EVL_M	DR	**0.819** ±0.068	0.541 ±0.13	0.578 ±0.05	49.6 ±3.0	82.6 ±4.6
	DNN	0.785 ±0.01	**0.533** ±0.019	**0.555** ±0.013	**55.4** ±2.0	**88.4** ±1.8
LS_EVL	DR	**0.801** ±0.077	**0.474** ±0.11	**0.511** ±0.056	**54.8** ±8.4	**91.8** ±5.5
	DNN	0.651 ±0.013	0.719 ±0.048	0.522 ±0.018	61.6 ±2.9	89.0 ±2.5

Table 6.7 Performance of deep rhythm grader (DR) and DNN rhythm grader (DNN), trained on BL_GRD_M1 with five random seeds and evaluated on BL_EVL_M and LS_EVL (matched and unmatched respectively), after recognition with TD-gr and alignment with GH-ph.

This is comparable to what was observed in the case of the pronunciation grader and is consistent with the end-to-end systems, having the freedom to tune both the feature extraction and feed-forward stages, overfitting less to the specific dataset they are trained on and better capturing the underlying relationship between the inputs and outputs. On the matched task, the end-to-end grader outperformed the DNN in terms of PCC only, but not other metrics.

Plotting actual against predicted score (Figure 6.5), it is apparent that there is a strong systematic bias in the predictions of the deep grader.

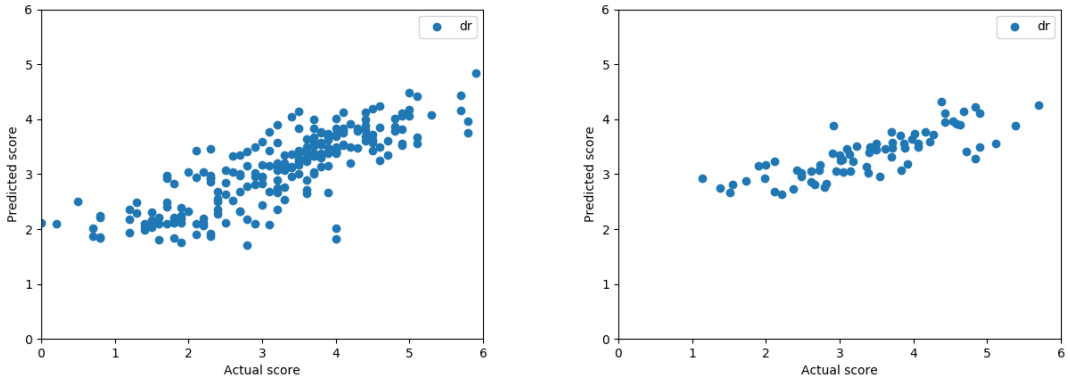


Fig. 6.5 Expert human scores plotted against ensemble predictions from deep rhythm grader, trained on BL_GRD_M1 (left) and LS_GRD_M (right) and evaluated on corresponding test sets.

It is thus likely that, as with the pronunciation grader, the predictions of the end-to-end system are more precise but less well-calibrated, explaining the superior PCC performance but inferior MSE performance. The same issue likely also affects the unmatched task, but is outweighed when considered relative to the DNN system by the end-to-end system's superior generalisation.

In conclusion, as in the pronunciation case, the end-to-end rhythm grader outperforms its two-stage counterpart at predicting grades which correlate with scores and demonstrates superior generalisability. However, it introduces a bias towards predicting systematically less extreme scores than the ground-truth.

6.3.3 Intonation

Following the discussion in §2.3.6, five intonation graders were implemented. The first two are two-stage baselines, specifically a feed-forward neural network trained on f_0 statistics computed across all data from each speaker (DNN) and a feed-forward neural network trained on the cosine fitting coefficients introduced in §2.3.6 (CF). As an intermediate system, bridging the gap between two-stage and end-to-end systems, a bi-directional attention LSTM is trained on f_0 statistics computed for each phone (RNN). The final two systems are end-to-end systems, namely a multi-head attention transformer over per-frame f_0 and probability of voicing (8HA) and an attention LSTM over the same per-frame f_0 and probability of voicing

(AttLSTM). The in-domain and generalisation performance of each of these systems was evaluated experimentally as with the previous two views.

Table 6.8 compares the performance of each system on a matched dataset task and an unmatched dataset task.

Test set	Model	PCC	MSE	MAE	%<0.5	%<1.0
BL_EVL_M	AttLSTM	**0.826** ±0.090	**0.437** ±0.061	**0.493** ±0.03	**60.7** ±4.0	**88.8** ±2.0
	8HA	0.708 ±0.038	0.729 ±0.049	0.675 ±0.023	46.9 ±2.9	75.9 ±2.9
	CF	0.711 ±0.21	0.944 ±0.32	0.779 ±0.15	38.8 ±11.0	70.1 ±10.0
	RNN	0.795 ±0.23	0.625 ±0.32	0.608 ±0.16	53.6 ±10.0	78.1 ±10.0
	DNN	0.734 ±0.069	0.813 ±0.081	0.725 ±0.04	40.2 ±3.9	73.7 ±3.4
LS_EVL	AttLSTM	**0.729** ±0.0064	0.572 ±0.041	**0.513** ±0.014	60.3 ±2.7	**90.4** ±0.67
	CF	0.728 ±0.16	0.68 ±0.16	0.64 ±0.11	47.9 ±9.1	76.7 ±10.0
	RNN	0.728 ±0.23	**0.562** ±0.23	0.551 ±0.16	**63.0** ±13.0	84.9 ±10.0
	DNN	0.582 ±0.15	0.799 ±0.15	0.654 ±0.057	58.9 ±5.8	79.5 ±2.0

Table 6.8 Performance of attention LSTM (AttLSTM) and 8-head attention (8HA) deep graders, f_0 statistics RNN and DNN graders and cosine fitting DNN grader (CF), trained on BL_GRD_M1 and evaluated on BL_EVL_M and LS_EVL (TD-gr ASR and GH-ph to align)

As expected, the 8-head attention mechanism, which was previously shown to be a generalisation of the CF method, outperformed it at both the matched and unmatched tasks, though by a relatively small margin. Replacing multi-head attention with an attention LSTM further improved performance, consistent with what was seen with the pronunciation and rhythm graders. Similarly, the f_0 statistics RNN outperformed the corresponding DNN. Both f_0 statistics systems outperformed the 8-head attention and CF systems but were outperformed by the attention LSTM. Overall, the two attention LSTM systems, over per-frame f_0 and p_v (DI) and over per-phone f_0 statistics (RNN), have the highest performance on both sets, with the attention LSTM being the clear winner.

Also consistent with previous results, the DNN system is the worst at generalising, having the lowest performance (both in absolute terms and relative to its matched task performance)

on the unmatched task. When considering the performance on the unmatched task, relative to that on the matched task, the RNN and cosine fitting system appear to be the least affected by the distributional shift, with the latter actually performing better on the unmatched task than the matched task.

As seen in Figure 6.6, this time it was the deep intonation grader not the DNNs that was the best calibrated, likely explaining why it clearly outperformed the two-stage systems across all metrics, rather than outperforming them on PCC but not on MSE, as was observed in some cases with its pronunciation and rhythm counterparts.

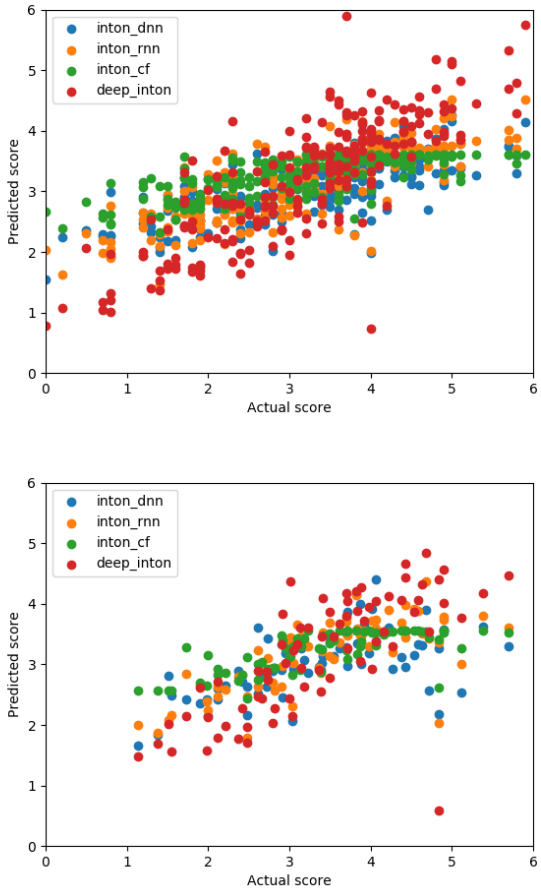


Fig. 6.6 Expert human scores vs predictions by AttLSTM deep intonation grader, trained on BL_GRD_M1 and evaluated on BL_EVL_M (top) and LS_EVL (bottom).

6.4 Relationship between two-stage and end-to-end graders

This section discusses experiments that were performed to investigate the relationship between the two-stage and end-to-end pronunciation graders. The pronunciation graders are focused on for further exploration as the mechanism by which they limit the information they represent for exclusive view-specificity (see §5.1) is the least transparent (as it is based on the topology of the network rather than simply limiting input to only duration or f_0 as is the case in the other networks).

Specifically, the experiments in this section aim to investigate three hypotheses. The first is that the phone distance graders are indeed assigning scores on basis of the speaker's pronunciation and not spurious effects. The second is that there exists a concept of pronunciation distance between phones, which the shallow phone distances (K-L divergences), pairwise Siamese networks and the intermediate representations of the deep pronunciation grader are all capturing. Finally, the third is that individual phone instances can be more or less representative of the speaker's overall pronunciation, either due to how well they've been recognised by the ASR, or how clearly or correctly they've been realised by the speaker, and that this representativeness is captured by the deep grader's attention mechanisms.

The two-stage system was first investigated. A histogram of the correlation between phone distance and scores (excluding undefined phones) is displayed in Figure 6.7. As was established in Kyriakopoulos et al. [154], most distances are moderately negatively correlated with score. This is consistent with unproficient speakers mispronouncing certain phones in ways that diverge from all the others.

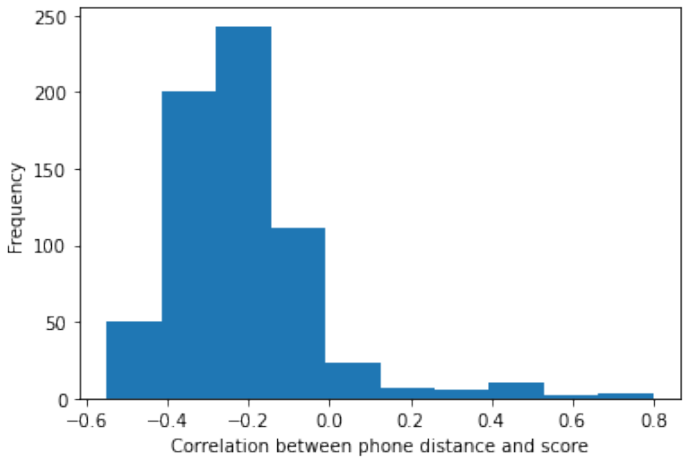


Fig. 6.7 Histogram of correlations between score and phone distance for each phone-pair extracted from MFCC13 features after recognition with TD-gr and alignment with GH-ph

Select phone distances plotted against score are displayed in Figure 6.8. It is confirmed that the relation to score is considerably stronger at low grades, with intermediate to high proficiency speakers mostly equally unlikely to register divergent pronunciations.

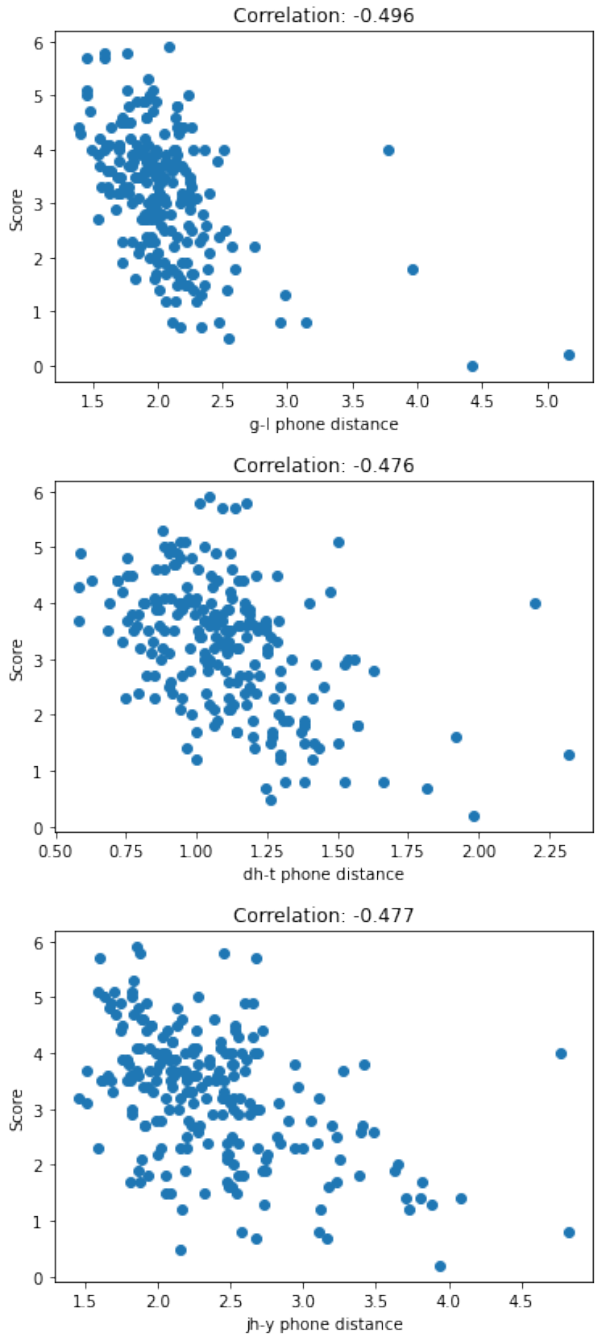


Fig. 6.8 Plot of expert human assigned score for speakers in BL_EVL_M against select log-one-plus K-L divergences extracted from MFCC13 features after recognition with TD-gr and alignment with GH-ph acoustic model

A particular phone distance is undefined for a speaker if their speech contains no examples of one of the phones. Such cases are assigned a value of -1. Figure 6.9 shows the relationship between the number of such undefined phone distances and score.

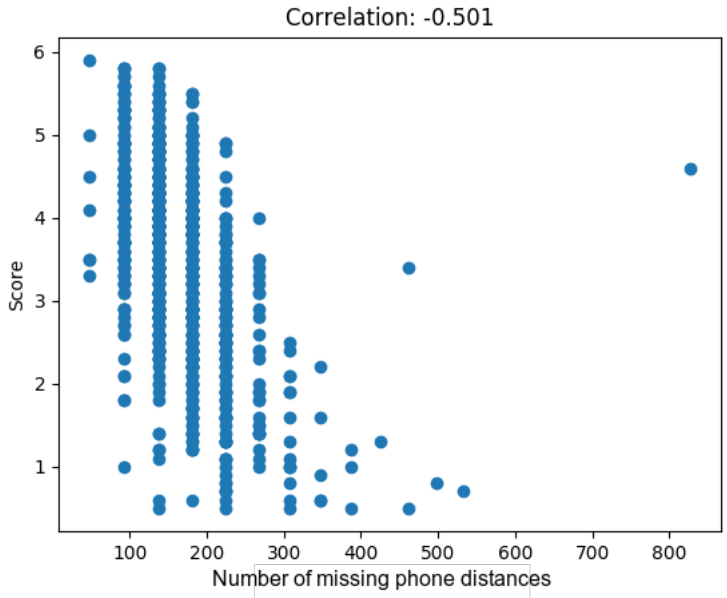


Fig. 6.9 Plot of expert human assigned score for speakers in BL_EVL_M against the number of phone distances that have no instances in each speaker's speech

Omitting more phones appears predictive of lower proficiency, likely a product of number of words spoken and vocabulary use. The features thus indirectly encode an aspect of text proficiency, likely improving performance but diluting view-specificity and possibly introducing bias in favour of candidates that spoke longer.

The above two effects appear insufficient to explain the strong predictive power of phone distance features for grade (PCC > 0.75 as seen in Table 6.6). This performance, as well their ability to predict the speaker's L1 and country of origin (Appendix I), suggest that the phone distances are indeed able to encode complex properties of the speakers' pronunciation space which non-linear models can learn to extract. This is consistent with the first hypothesis at the beginning of this section.

Next, the Siamese phone instance pair classifier used to initialise the end-to-end pronunciation grader was investigated. In addition to performance at classifying pairs of instances as being of the same or different phones, the correlation between the value of the Euclidean distance and the original phone distance feature between the two phone labels was calculated over all pairs that belong to different phones.

A second version of the Siamese network was also trained to minimise MSE between the distance metric and the K-L divergence and evaluated using the same two criteria. These results are displayed in Table 6.9.

Model	Accuracy (%)	Correlation with KL
siam_bin	75.0 ±1.2	0.399 ±0.085
siam_kl	68.0 ±2.2	0.789 ±0.10

Table 6.9 Performance, measured by mean and standard deviation of % accuracy and correlation of distance metric to phone distance of Siamese LSTM phone instance pair classifier trained using binary classification (siam_bin) and phone distance prediction (siam_kl) criteria, on 100,000 pairs of matched phone instances and 100,000 pairs of unmatched phone instances randomly sampled from BL_GRD_M1 and evaluated on 10,000 matched and 10,000 unmatched pairs sampled from BL_EVL_M (input is MFCC-13 from TD-gr aligned with GH-ph).

As expected, the networks performed better on the task they are trained for than on the other task. Both systems performed well, suggesting that the Siamese networks are capable of extracting interpretable distance metrics indicative of both the clustering together of instances of the same phone as well as the distances between distributions of different phones. Further, the fact that the system trained for each task also performed reasonably on the other task suggests that these two concepts of distance are closely related. This lends support to the second hypothesis.

The deep grader attends over the embeddings of all instances of each phone before calculating pair-wise Euclidean distance at the phone label level rather than at the instance level as in the Siamese network. The attention mechanism is expected to allow the network to learn to ignore outlying (e.g. badly aligned) phone instances and focus on instances that are more predictive of proficiency score (e.g. errors).

To investigate the operation of the attention mechanism in the trained grader, the entropy of the attention weights over instances of each phone for each speaker was calculated as a ratio of its maximum possible value:

$$ER = \frac{\sum_{n=1}^{N_m} -\alpha_{nm} \log \alpha_{nm}}{\log N_m} \tag{6.4}$$

where $a_n m$ is the attention weight corresponding to the nth of N_m instances of the mth phone of a given speaker.

To illustrate the interpretation of this metric, Figure 6.10 displays three plots of the distribution of attention weights for select phones and speakers at different values of entropy.

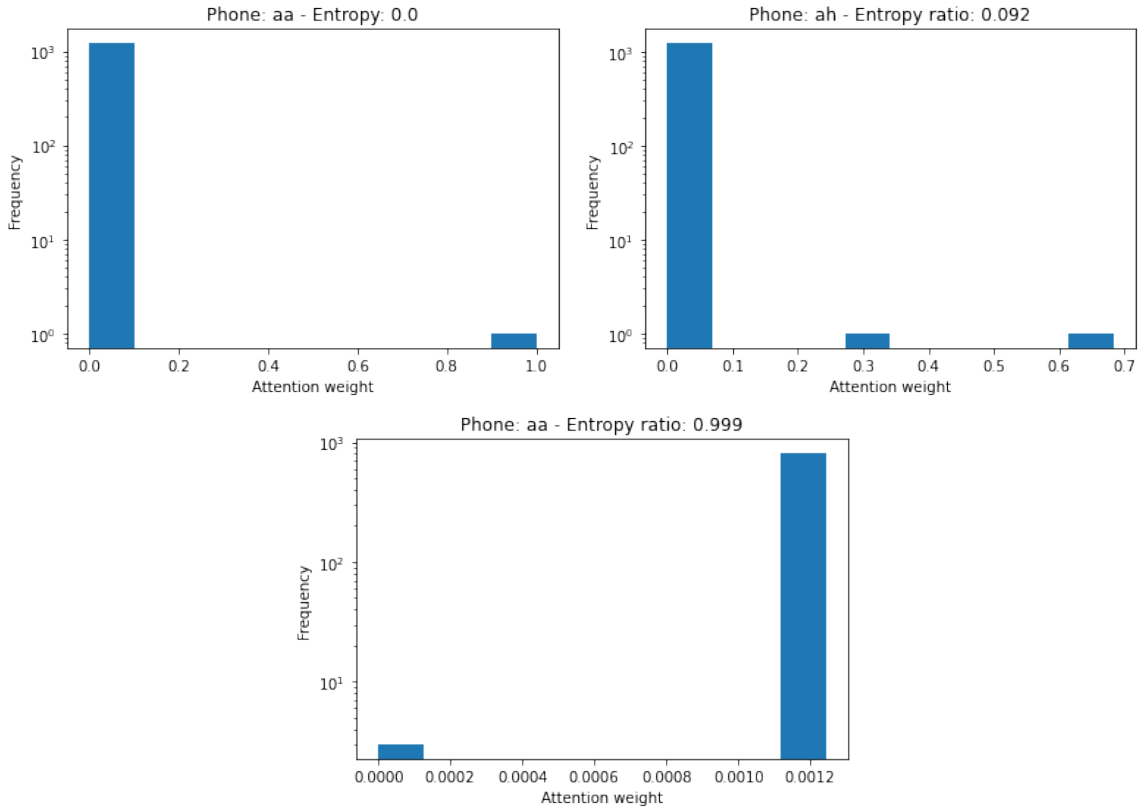


Fig. 6.10 Histograms of attention weight values over instances for 3 phones for select speakers in BL_EVL_M, from deep phone distance grader (with AttLSTM instance embedding) trained on BL_GRD_M1, with batch norm and exponential learning rate

A value of zero (e.g. Figure 6.10, left) indicates that only one phone instance is determining the phone representation for the speaker (as all but one weight are zero). This means that the layer extracting the significance of each instance has extracted such a large value for the instance in question compared to all the others that the weights corresponding to the later all drop to within a rounding error of zero. This could suggest that the instance is a pronunciation error or a particularly clear realisation, but could also be a result of random spikes in outputs of the model. Figure 6.10, middle, illustrates a near-zero case, where a single phone instance determines over 60% of the phone representation, a second instance determines over three quarters of what remains and the rest is determined by the over a thousand remaining instances. As before, this could be explained by a small number of pronunciation errors or clear realisations or by random artifacts. On the other extreme, a value of one indicates all phone instances are counted equally. In the near-one case of Figure 6.10,

right, all instances are given equal weight except a single outlier which is ignored. Ignoring of a small number of outliers and equal treatment of remaining phones could indicate the attention mechanism learning to ignore badly aligned or otherwise invalid phone instances.

To investigate which of the above hypothesised effects may be taking place in the attention mechanism of the deep pronunciation grader, the overall distribution of the entropy ratios across all phones across all speakers is illustrated in Figure 6.11.

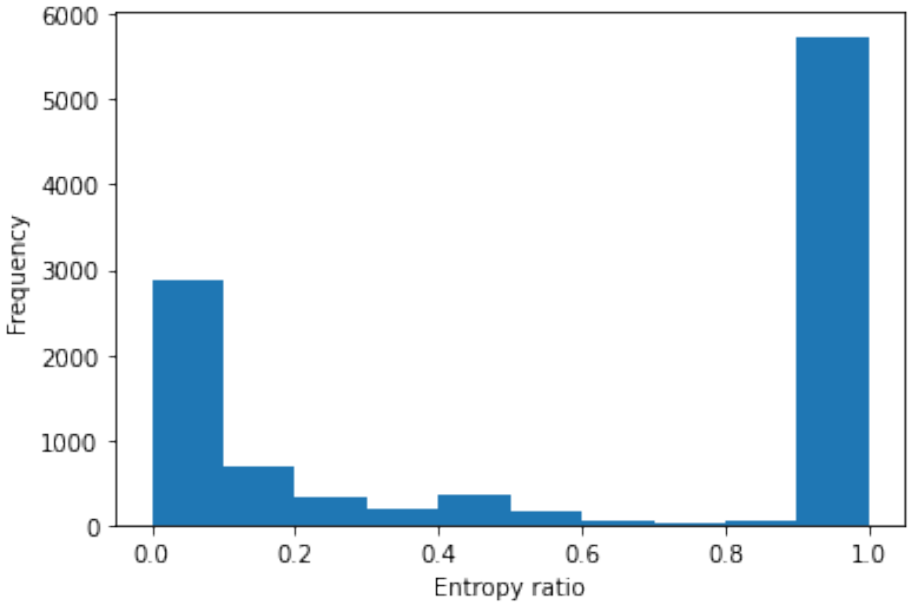


Fig. 6.11 Overall distribution overall of the entropy ratio of attention weights over instances of each phone of each speaker in BL_EVL_M, when evaluating a deep phone distance grader (with attention LSTM phone instance embedding) trained on BL_GRD_M1, with batch normalisation and exponential learning rate schedule

It is seen that the majority of entropy values are slightly below one. This suggests that the dominant effect explaining the advantages yielded by the addition of the attention mechanism is that the mechanism is learning to detect and exclude small numbers of instances that are outliers or otherwise invalid. This is consistent with the third hypothesis from the beginning of this section.

A significant minority of entropy values are at the low end of entropy ratio, with the attention having learned to focus on only one or a small number of representative phone instances. Intermediate entropy ratios between the two extremes are much rarer. As discussed above, this minority effect could either be a disruptive random phenomenon or could be indicative of a strategy of focusing on particularly errorful or particularly well rendered phone instances.

The above insights were confirmed by plotting the entropy ratio for each phone for each speaker in Figure 6.12.

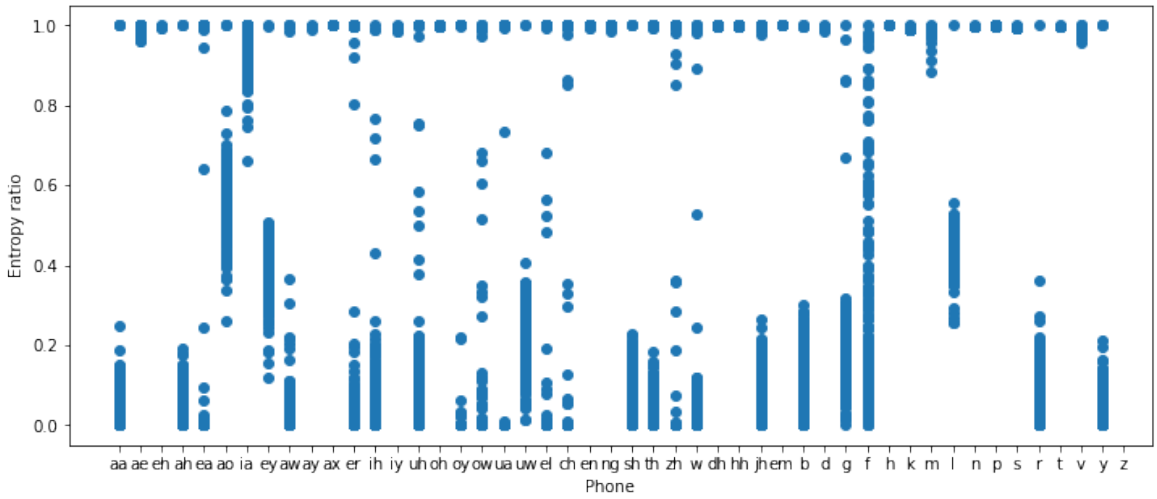


Fig. 6.12 Distribution by phone of the entropy ratio of attention weights over instances of each phone of each speaker in BL_EVL_M, when evaluating an attLSTM deep phone distance grader trained on BL_GRD_M1, with batch normalisation and exponential learning rate schedule

While it is apparent different speakers can have radically different attention weight distributions for the same phone, it can also be seen some phones are considerably more prone overall to the low entropy (i.e. sharp focus) attention outcome than others.

The extent to which the pairwise Euclidean distances between phone-level representations calculated by the network match those extracted using the K-L divergence method was then investigated. Given the fairly strong correlation between the phone distances and instance-pair distances observed in the Siamese network experiments, a decent correlation with the new phone-level distances was also expected. In fact, the correlation was found to vary considerably. Figure 6.13 on the following page illustrates three indicative pairs.

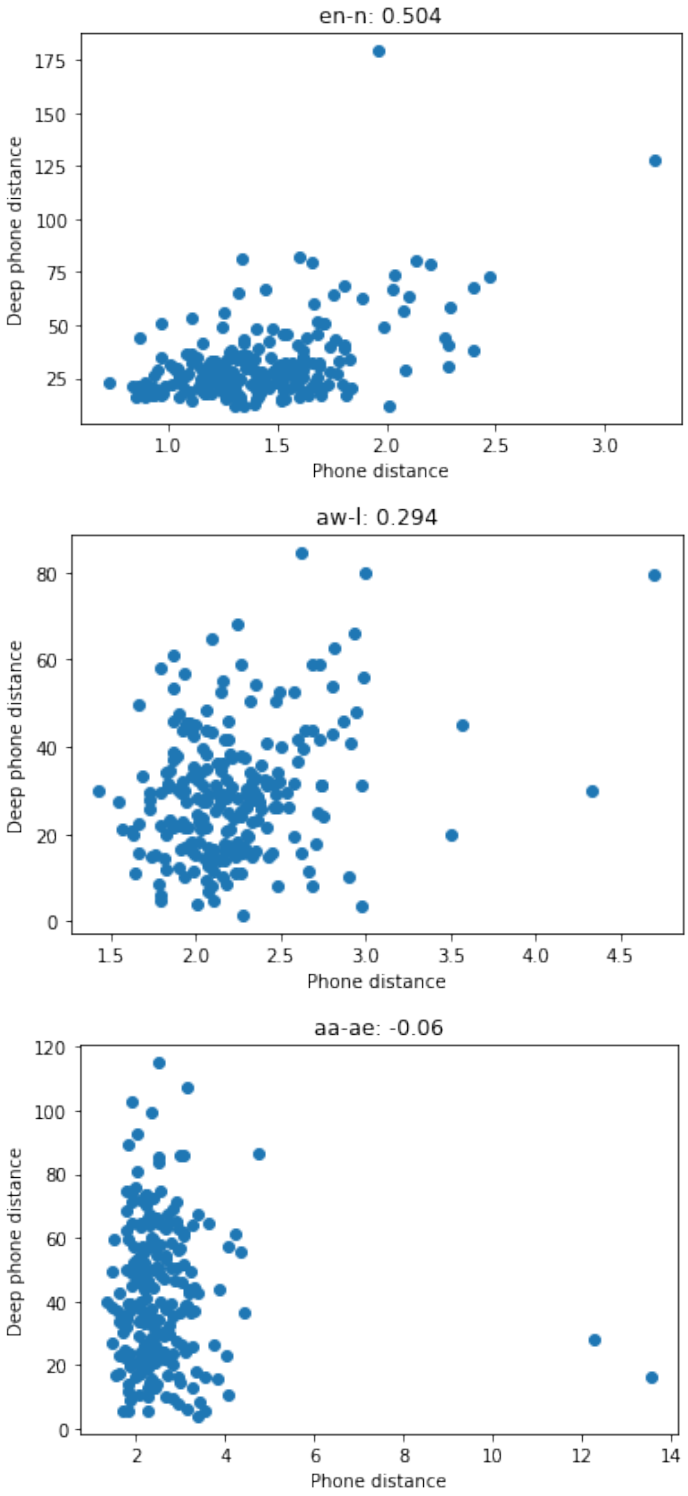


Fig. 6.13 Correlation between phone distance and deep phone distance across all speakers in BL_EVL_M for selection of phone pairs.

To investigate this effect, the correlation between each phone distance and its corresponding deep phone distance was plotted against the correlation between the phone distance and score for all phone-pairs (Figure 6.14).

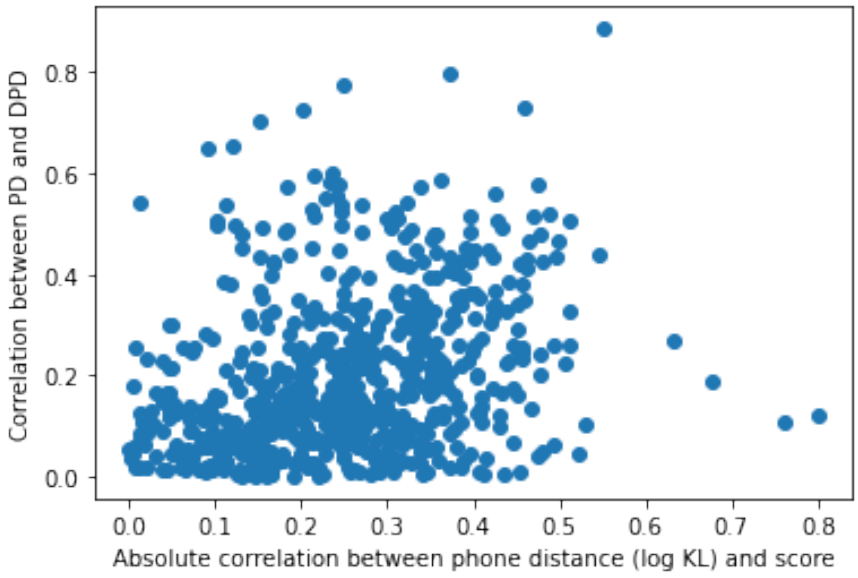


Fig. 6.14 Correlation between phone-pair phone distance (PD) and deep phone distance (DP) across speakers in BL_EVL_M plotted against absolute correlation of phone distance (PD) with score.

The deep phone distances are more likely to match the original distances the more strongly they predict score. This could be interpreted as the deep phone grader system using its increased flexibility to improve on poorly performing distances.

6.5 Validity of single-view graders

As the experiments reported in §6.3 relied on holistic grades to evaluate single-view systems, it was not possible to directly evaluate whether each system was indeed grading with respect to its intended view. This section presents experiments conducted to indirectly investigate this question.

First, an investigation is carried out into whether the single-view graders are complementary to each other. If the different graders are indeed grading on the basis of different views, they should each be capturing complementary information about the proficiency of the speaker.

Combining single-view graders should therefore lead to a greater increase in performance than combining graders of the same view in an ensemble. As in previous experiments,

each single-view grader is evaluated on the basis of the performance of an ensemble of its randomly initialised instantiations. These performance figures are compared to those of each possible pairwise combination of the graders based on grade averaging. The results are displayed in Table 6.10.

Model	PCC	MSE	MAE	%<0.5	%<1.0
text	0.820 ±0.031	0.459 ±0.039	0.505 ±0.001	60.7 ±0.85	88.8 ±2.9
pron	0.820 ±0.021	0.531 ±0.051	0.573 ±.017	53.6 ±2.1	83.5 ±1.8
rhythm	0.819 ±0.038	0.541 ±0.13	0.578 ±0.05	49.6 ±3.0	82.6 ±4.6
intonation	0.826 ±0.039	0.437 ±0.041	0.493 ±0.015	60.7 ±0.9	88.8 ±3.4
text+pron	**0.852**	**0.418**	**0.488**	**61.6**	*87.9*
text+rhythm	**0.861**	0.428	**0.494**	*59.8*	89.7
text+inton	**0.866**	**0.356**	**0.443**	**67.0**	90.2
pron+rhythm	0.823	0.445	**0.499**	**57.6**	**87.9**
pron+inton	**0.865**	**0.393**	**0.475**	**62.2**	*87.1*
inton+rhythm	0.854	0.421	*0.499*	60.9	*88.4*

Table 6.10 Performance of single-view graders and their pair-wise averages on BL_EVL_M. Single-view grader performances are performances of ensemble averages of predictions and are reported together with the standard deviations of their underlying ensembles (note the means of the underlying ensembles are smaller than the ensemble performances and are not reported). Pair-wise average performance average figures are marked in italics if they are worse than the ensemble average performance of either of their constituent graders and in bold if they are more than one ensemble standard deviation better than the ensemble average performances of both of their constituent graders.

It is seen that every pairwise combination of single-view graders yields a superior result on almost all indicators to both of its constituent single-view graders combined by averaging in an ensemble. Further, in the majority of cases, the difference in performance between the combination and each of the constituents is greater than the standard deviation of the underlying ensemble of the latter. These results suggest that the single-view graders are indeed extracting complementary information to each other, consistent with them representing different views.

To further examine the relationships between the predictions of the single-view graders so as to determine whether they appear to be measuring their respective grades, the Kendall

rank coefficient between the sets of grades predicted by each pair of graders was computed, with results displayed in Table 6.11.

	text	pron	inton
pron	0.638	-	
inton	0.588	0.653	-
rhythm	0.613	0.699	0.690

Table 6.11 Kendall's τ between single-view grader predictions

Rhythm and intonation grades are closer to each other and to pronunciation than to text, consistent with pronunciation, rhythm and intonation being more closely related (as aspects of realisation per §2.1) than each is to text. Text is closer to pronunciation than to the other two, which is consistent with the hypothesised text-pronunciation overlap due to the latter's explicit encoding of missing phones.

Next, the prediction errors for each of the four graders were plotted against expert-assigned score (Figure 6.15).

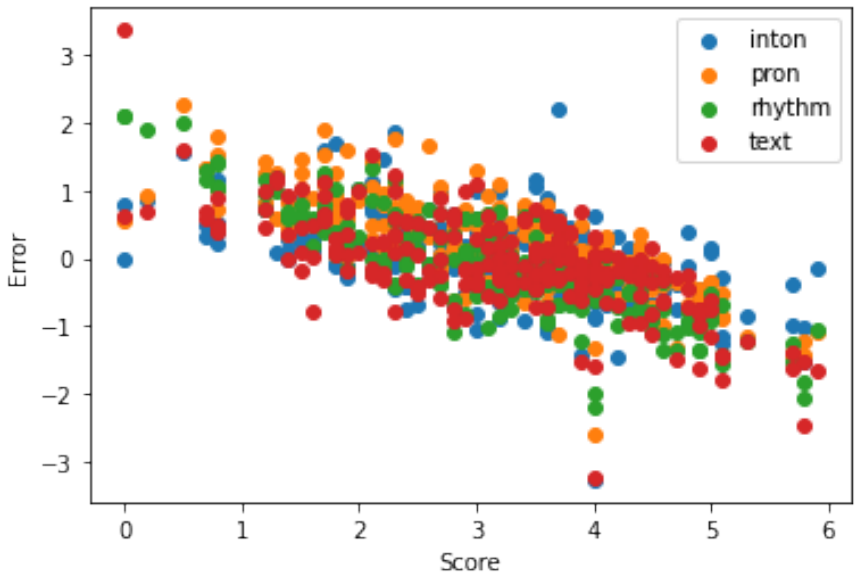


Fig. 6.15 True holistic score plotted against error between holistic and each single-view predicted score for all speakers in BL_EVL_M

It is seen that all graders suffer from a calibration problem, whereby low holistic scores are systematically overestimated and high scores systematically underestimated. This effect was modelled by fitting a linear relationship between score and error, which was then used to adjust the errors.

These adjusted errors were plotted against score to find other systematic biases (Figure 6.16).

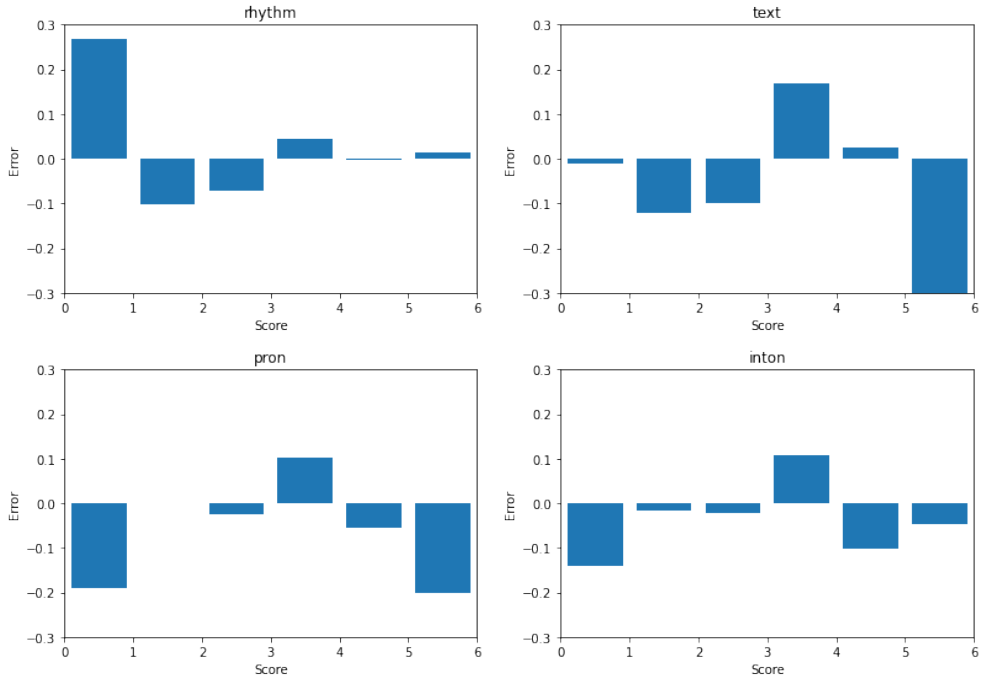


Fig. 6.16 Error between holistic scores and single-view predictions with each of the end-to-end rhythm, text, pronunciation (pron), and intonation graders adjusted for mis-calibration, plotted across score ranges for speakers in BL_EVL_M

The rhythm grader systematically overestimates very low grades (0-1), even after adjustment, while underestimating lower intermediate grades (1-3). This is consistent with rhythm being limiting at higher grades, only becoming relevant after the speaker has mastered other aspects of proficiency. The graders would thus have no way of telling the difference between the rhythm of poor and intermediate speakers. By contrast, the pronunciation and text graders underestimate high grades (5-6) while overestimating upper intermediate speakers (3-4). This is consistent with text and pronunciation being limiting at lower grades, such that human annotators consider all speakers above a certain level to be good enough. Though these conclusions support each grader indeed measuring proficiency with respect to its intended view, they constitute a deviation from the assumption that holistic grade is a simple average of view-specific grades (Equation 5.7) and illustrates a limitation of the use of holistic grades to train view-specific systems for grade ranges where the view in question is not limiting for holistic grading. In the case of the pronunciation grader, low grades are also under-estimated. It was hypothesised that this is due to the higher incidence of ASR errors for poorer speakers, as more proficient speakers speak more intelligibly, making the pronunciation appear more erroneous than it actually is.

To confirm this, word error rate (WER) is plotted against score (Figure 6.17).

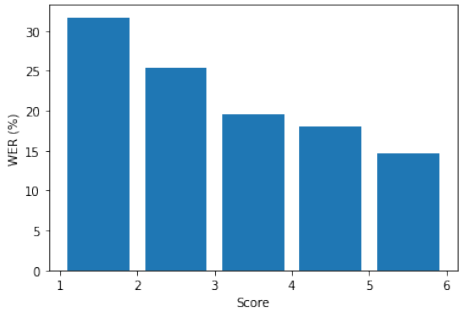


Fig. 6.17 Performance of the TD-gr ASR on speakers in BL_EVL_M of each CEFR level (shown on the score axis) evaluated by word error rate (WER).

The experiment was repeated, plotting the absolute values of the adjusted errors against score, to measure the unbiased precision of the grader at different grade levels (Figure 6.18).

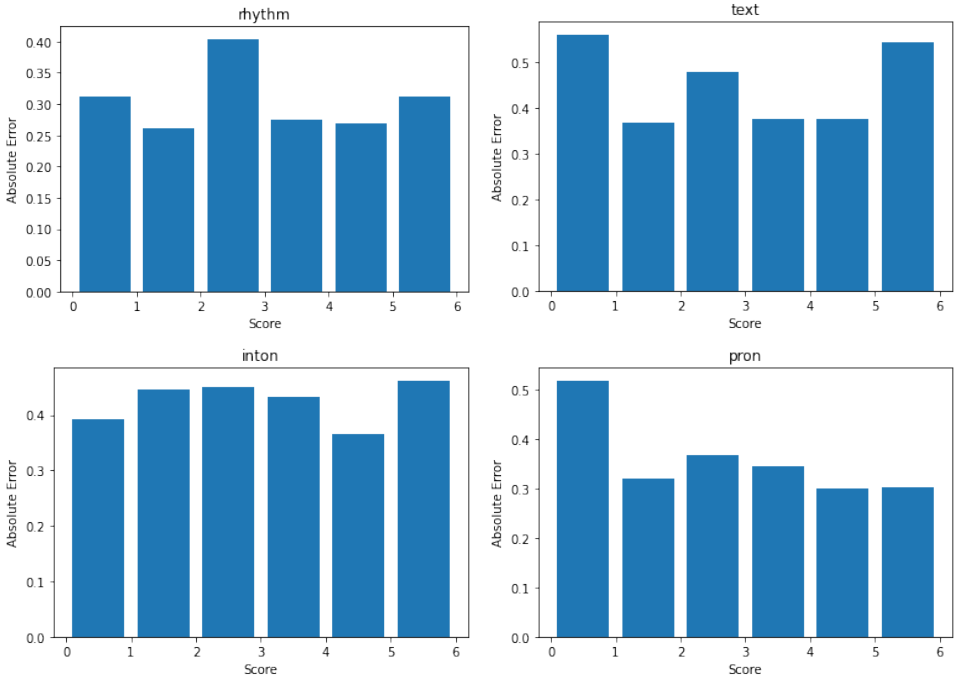


Fig. 6.18 Absolute error between holistic scores and single-view predictions with each of the end-to-end rhythm, text, pronunciation (pron), and intonation graders adjusted for mis-calibration, plotted across score ranges for speakers in BL_EVL_M

The pronunciation and text graders have larger absolute errors at the lowest scores, consistent with performance being lower for poorer speakers due to a higher incidence of ASR errors. The rhythm and intonation grader do not display the same effect, which is to be expected as the text and pronunciation graders are much more dependent on the ASR.

Performance of phone distances at different scores was also investigated for the L1 and country classification tasks (Appendix I), using the two-stage grader, with results displayed in Table 6.12. As with the grader, performance is in both cases best at intermediate grades. This is consistent with ASR errors distorting the system for low proficiency speakers and the L1 and country of origin of high proficiency speakers being objectively harder to distinguish based on their pronunciation due to their speech being more native-like.

A1	A2	B1	B2	C1	C2	Overall
60.0	60.1	70.0	70.5	71.8	57.5	69.0

A1	A2	B1	B2	C1	C2	Overall
34.4	75.7	88.7	89.8	86.3	86.7	85.5

Table 6.12 L1 (top) and country of origin of Spanish speakers (bottom) detection rate on BLT_EVL_M2, broken down by CEFR level, of 3-way phone distance DNN country of origin classifier using phone distance features, trained on BLT_GRD_M2

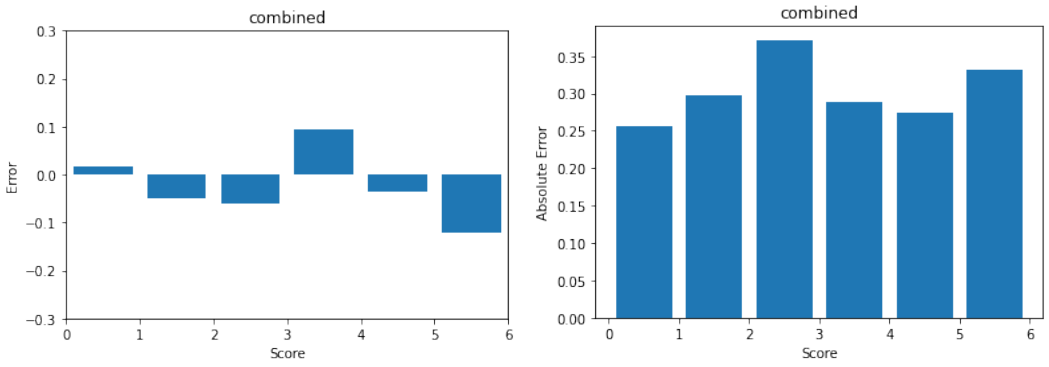
6.6 Grader combination

The four end-to-end single-view graders (pronunciation, rhythm, intonation, and text) are now combined using each of the three methods discussed in §5.3, compared to the two baselines from §6.2 (Table 6.13). Combination with the best system outperforms both baselines, suggesting the multi-view approach is also a superior approach to holistic grading.

Model	PCC	MSE	MAE	%<0.5	%<1.0
hand	0.862	0.359	**0.454**	62.1	**91.5**
xvec	0.806	0.489	0.556	52.2	83.9
mean	0.879	0.383	0.461	64.4	89.8
concat	0.855	0.414	0.482	63.1	88.9
att	**0.881**	**0.358**	0.465	**64.2**	90.0

Table 6.13 Performance of end-to-end pronunciation, rhythm, intonation, and text graders combined using each of score averaging (mean), concatenating intermediate representations (concat) and attention mechanism over scores (att), compared to performance of DNNs trained on ALTA baseline features (hand) and x-vectors trained with grading criterion (xvec), trained on BL_GRD_M1 and evaluated on BL_EVL_M

As illustrated in Figure 6.19 below, the combined grader's adjusted scores perform evenly across all score levels, suggesting the limiting factor and ASR effects seen in the single-view graders have been smoothed out.



f

Fig. 6.19 Mean adjusted error (left) and mean absolute error (right) between expert score and that predicted by combined grader (attention method) for different score ranges for all speakers in BL_EVL_M

6.7 Chapter Summary

This chapter reported experiments that were conducted to evaluate the framework for multi-view grading introduced in Chapter 5. Three main hypotheses were evaluated: that the end-to-end graders introduced in §5.2 would outperform and display superior generalisation and transferability to their corresponding two-stage graders; that each end-to-end single-view grader indeed assesses its respective intended view of proficiency, even when trained on holistic grades; and that combining different end-to-end single-view graders would result in superior holistic grading to baseline approaches.

The data used in the experiments, consisting of holistically graded recordings of mostly spontaneous candidate responses to two spoken English exams, was presented in §6.1. The baseline systems used to test the hypotheses were discussed in §6.2. Specifically, end-to-end single-view graders were compared to their respective two-stage graders, while combined graders were compared to multi-view handcrafted features (Appendix H) and x-vectors extracted to predict holistic grade.

In §6.3, it was shown that all three novel view-specific end-to-end graders outperform their two-stage baselines on the task of holistic proficiency grading, and are better at generalising to other datasets. The end-to-end pronunciation grader was also tested on the related task of L1 classification, again found to considerably outperform its two-stage counterpart. The investigation in §6.4 indicated that the structures in the end-to-end pronunciation grader

intended to generalise equivalents in the hand-crafted feature extraction stage of the two-stage grader were indeed performing similar functions, and suggested that the attention mechanism over instances was improving their representational capacity by learning to exclude and focus on instances particularly indicative of proficiency as needed.

The extent to which the four end-to-end graders (the three novel systems plus the text grader) were indeed assessing their intended views was evaluated, insofar as was possible given the lack of human-annotated single-view grades, in §6.5. It was first demonstrated that each possible pair of graders yield a considerably superior performance when combined with each other than when different versions of each are combined in an ensemble. This indicated that the different graders were extracting complementary information. Rank correlation analysis showed rhythm and intonation grades predicted for a given speaker to be more consistently closer to each other than to text, which was again consistent with what would be theoretically expected. Analysis of performance at different grades also yielded the results that would be expected if each grader was assessing its own view, with the text and pronunciation graders being more limiting at lower grades, while rhythm was more limiting at higher grades. The text and pronunciation graders were also confirmed to be the most sensitive to ASR errors, as expected.

Combined with the fact that the text, rhythm and intonation graders were strongly limited by their inputs to only assess their respective views, only having access to words, duration and pitch inputs respectively, it was concluded to be highly likely that the end-to-end graders were indeed effectively assessing their respective views, even though they were trained on holistic grades. Two important caveats to this were discovered however. The first was that holistic grades encode less information about views for speakers at proficiency levels where they are not limiting (e.g. rhythm for low proficiency speakers and pronunciation for high proficiency speakers), meaning single-view graders trained on holistic grades will be less reliable in these cases. The second was that the pronunciation grader also captured aspects of text, owing to the way it encoded missing phones, as had been predicted when it was introduced.

Finally, in §6.6, a holistic grader developed by combining the single-view end-to-end graders with an attention mechanism over grades was shown to outperform x-vector and multi-view hand-crafted feature-set baselines, as well as two other forms of grader combination, and to no longer display the asymmetric performance at different proficiency levels and ASR error sensitivity of the single-view graders.

Chapter 7

Experiments on Pronunciation Error Detection

Pronunciation assessment was introduced in Chapter 3. It was defined as characterising a non-native speaker's proficiency based on the way they speak words as series of phones [61]. The discussion in §3.7 established that effective systems should, in addition to scoring a candidate on the overall way they pronounce the phones of English across their speech, determine the types of pronunciation errors they make at a word and utterance level. The use of deep learning for the overall pronunciation assessment was explored in Chapter 5 while Chapter 6 investigated how it can be combined with other views of speakers' overall proficiency. This chapter will investigate pronunciation error detection at the word and utterance level to further enhance the feedback that can be provided about a speaker by supplementing the overall pronunciation score with information on individual utterances and words that were pronounced incorrectly.

Approaches to pronunciation error detection in the literature were reviewed in Chapter 3. Native speaker comparison methods (§3.1) were rejected as they can't distinguish types of error and risk of bias towards irrelevant attributes of the native speakers. ASR confidence methods (§3.2) similarly cannot distinguish error types, risk bias due to factors other than proficiency affecting confidence, and are highly sensitive to the recognised start and end times of words and phones, the latter two issues being particularly problematic with spontaneous speech. Supervised methods were rejected as they required large amounts of error-annotated training data which, as was seen in §3.6, is both difficult to find and suffers problems of inconsistency. Approaches based on forced alignment followed by comparison with canonical pronunciations, namely extended recognition networks (ERNs) (§3.3) and phone recognition methods (§3.4), emerged as the most suited to the task, as they allow word and utterance level error detection on spontaneous speech with the ability to separately detect and get

feedback on different types of errors without the need for human labelled training data or native reference data.

In ERNs, the utterance is aligned in one go with an extended dictionary allowing canonical and a finite number of candidate errorful pronunciations, with the resultant 1-best phone sequence determining which words were pronounced errorfully and which canonically. ERNs require a method for generating errorful pronunciations and are limited as to the number of pronunciations that can be considered per word without the network becoming unmanageably large, which would result in the need for extensive pruning, distorting the results. In phone recognition methods, the utterance is first aligned with a canonical dictionary and each word re-aligned, fixing its start and end frames, but allowing any possible sequence of phones. The main issue with this approach is its reliance on the word boundaries from the initial canonical alignment, which may distort the result. To resolve these issues, this chapter presents a modification of the ERN approach to avoid the problem of network size growth without fixing word boundaries. This is achieved by repeating alignment of the utterance, allowing only canonical pronunciations for all but one word each time.

In §7.1, a framework is introduced for generating candidate errorful pronunciations, decomposed into accent and lexical errors, generalising the generation methods for ERNs reviewed in §3.3. Next, §7.2 presents the modified ERN approach for detecting errors at the word and utterance levels. One of the main challenges in error detection is obtaining consistent human annotated datasets. Even with unsupervised techniques such as ERNs, annotated data is still necessary for system development, evaluation and comparison across sites. In practice, word-level annotations tend to be inconsistent within and among annotators and within and across datasets (§3.6). Since the approaches in the literature were almost exclusively used with read speech, the additional issue of uncertainty in the word being spoken, which if recognised incorrectly will lead to incorrect error detection, was also not addressed. In §7.3, three corpora of human annotated speech are presented, on which to investigate annotator consistency as well as detection of pronunciation errors using this method. Finally, in §7.4, experimental results are presented to evaluate the quality of human annotations, establish the extent to which accent and lexical errors are distinct and evaluate the performance of the framework for detecting them.

7.1 Accent and Lexical Errors

Consider a speaker uttering a word w_i (e.g. the word *the*). The way the word is pronounced can be expressed as a sequence of phone instances $\boldsymbol{\phi}_{1:M}^{(w_i)} = \phi_1^{(w_i)}, \phi_2^{(w_i)} \ldots \phi_M^{(w_i)}$ representing its *phonetic pronunciation*. Given a chosen phonetic alphabet, the word w_i has a set of canonical

phonetic pronunciations which a listener would recognise as correct, represented by the pronunciation dictionary entry $\mathscr{D}_{w_i}^{(can)}$, e.g.

$$\mathscr{D}_{the}^{(can)} = \{[\text{dh ax}], [\text{dh iy}]\}) \tag{7.1}$$

As discussed in §3.3, using narrow rather than broad transcriptions makes canonical pronunciation dictionaries more difficult to obtain and is expected to make pronunciation error modelling more complex, with more candidate pronunciations per word, resulting in distortions due to excessive pruning during alignment. Consistent with this argument, this work uses broad phonetic transcriptions (specifically the two alphabets described in Appendix D). The publicly available CMU [269] and the proprietary COMBILEX [81] pronunciation dictionaries are used, depending on the dataset, with the same dictionary used in ASR, alignment and error detection in each case.

Given each word's canonical pronunciation dictionary entry, it is now desired to generate a set of candidate non-canonical pronunciations:

$$\boldsymbol{\phi}_{1:M}^{(w_i)} \notin \mathscr{D}_{w_i}^{(can)} \tag{7.2}$$

The process of error detection (§7.2) will then involve determining the probability, given the acoustic observations found by the ASR to correspond to the particular word instance, that the word was pronounced as one of the non-canonical pronunciations rather than one of the canonical pronunciations.

Following the discussion in §3.3, two types of deviations from the canonical pronunciation of a word are modelled, accent errors and lexical errors (illustrated in Fig. 7.1).

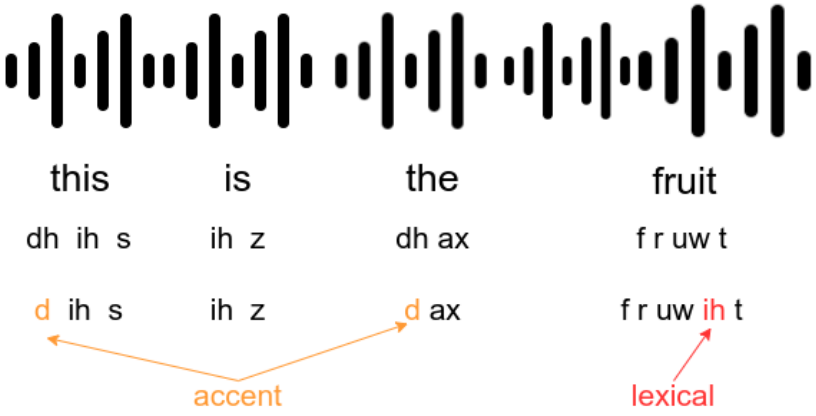


Fig. 7.1 Illustration of accent and lexical errors. A speaker that pronounces *the* as [d ax] is likely to also pronounce *this* as [d ih s]. A speaker that pronounces *fruit* as [f r uw ih t] is unlikely to also pronounce *boot* as [b uw ih t].

Accent errors are repeated patters of insertions, deletions and substitutions caused by effects of the speaker's L1 on their speech e.g. a speaker pronouncing *the* as [d ax] as part of a general tendency to pronounce [dh] as [d]. *Lexical errors* are errors particular to a specific word, caused by not knowing its canonical pronunciation and depend on the word's spelling e.g. a speaker pronouncing *subtle* as [s ax b t l] because they don't know that the b is silent. Someone pronouncing *the* as [d ax] would be more likely to also pronounce *that* as [d ah t]), but someone who pronounces the silent b in subtle is not likely to also pronounce *scuttle* as [s k ax b t l].

Possible accent errors for any given word should be dependent on the word's canonical pronunciation, while for lexical errors on its spelling. It is hypothesised that the two types of errors are distinct and can be separately detected. Separate models are thus defined to respectively generate potential candidate accent and lexical errors for a given word.

7.1.1 Generating candidate accent errors

Let an accent error type ε be defined as a phone insertion, deletion or substitution that can be applied to an eligible pronunciation. For instance, applying the substitution error type $dh \rightarrow d$ to the canonical pronunciation of the word *the* [dh ax] turns it into the errorful pronunciation [**d** ax]. A canonical pronunciation can be eligible for multiple instances of the same error type. For instance, $s \rightarrow z$ can be applied to [s ih s t ax r] (*sister*) in two different locations, to yield [**z** ih s t ax r] or [s ih **z** t ax r], or to both locations, yielding [**z** ih **z** t ax r].

Accent error types identified in the literature are generally caused by differences between the phonological rules of English and the speaker's L1 [107, 53, 63, 212]. For example, words in Vietnamese never end in certain consonants making Vietnamese learners of English prone to final consonant deletion errors [204, 203], while French speakers are prone to omitting [h] sounds which don't exist in French phonology [244]. Some accent error types, such as confusion of the short o [oh] and the long o [ow], are found across L1s and are generally associated with particularities of English that it shares with few other languages.

To populate $\mathscr{E}$, common accent error types among non-native speakers are compiled from the literature, focusing on 10 first languages (L1s) and on cross-L1 errors (Appendix K - Tables K.1, K.2, K.3, K.4 and K.5). These tables are not intended to be exhaustive or definitive, but to provide a practical tool to propose common and interpretable errors for detection. The use of broad transcriptions means that substitutions of narrow English phones with non-English narrow phones that map to the same broad phone (e.g. a French speaker pronouncing r as [ɣ]) can't be modelled.

Given a canonical dictionary entry $\mathscr{D}_{w_i}^{(can)}$, a group of accent error types $\mathscr{E}$, and a maximum number of errors R, the dictionary entry can be expanded to the combined entry

$\mathscr{D}_{w_i}^{(can)} \oplus \mathscr{E}_R$, containing pronunciations formed by applying up to R instances of the error types in $\mathscr{E}$ to the initial canonical pronunciations. e.g.

$$\mathscr{D}_{the}^{(can)} = \{[\text{dh ax}], [\text{dh iy}]\}) \tag{7.3}$$

$$\mathscr{D}_{the}^{(can)} \oplus \{dh \rightarrow d\}_1 = \{[\text{dh ax}], [\text{dh iy}], [\textbf{d} \text{ ax}], [\textbf{d} \text{ iy}]\}) \tag{7.4}$$

$$\mathscr{D}_{the}^{(can)} \oplus \{dh \rightarrow d, iy \rightarrow eh\}_1 = \{[\text{dh ax}], [\text{dh iy}], [\textbf{d} \text{ ax}], [\textbf{d} \text{ iy}], [\text{dh } \textbf{eh}]\} \tag{7.5}$$

$$\mathscr{D}_{the}^{(can)} \oplus \{dh \rightarrow d, iy \rightarrow eh\}_2 = \{[\text{dh ax}], [\text{dh iy}], [\textbf{d} \text{ ax}], [\textbf{d} \text{ iy}], [\text{dh } \textbf{eh}][\textbf{d eh}]\} \tag{7.6}$$

Setting R thus allows multiple accent errors on the same word (though strictly on different phones), while constraining the final dictionary size. The process is illustrated in Figure 7.2.

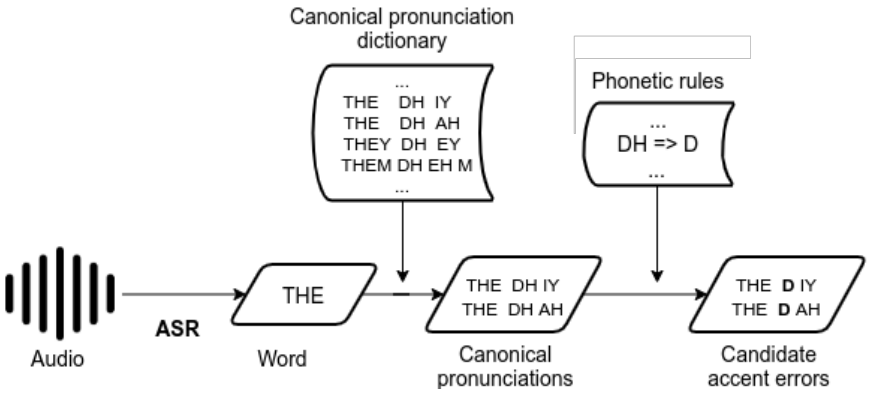


Fig. 7.2 Illustration of process for generating candidate accent errors from recognised word, canonical pronunciation dictionary and phonetic rules.

Accent error candidate generation can be run in an L1-dependent manner, generating candidates based on error types corresponding to the speaker's L1 ($\mathscr{E}_R^{L1}$) e.g.

$$\mathscr{D}_{hot}^{(can)} = \{[\text{h oh t}]\}) \tag{7.7}$$

$$\mathscr{D}_{hot}^{(can)} \oplus \mathscr{E}_1^{French} = \{[\text{h oh t}], [\textbf{oh} \text{ t}], [\text{h } \textbf{ow} \text{ t}], [\text{h oh } \textbf{d}]\}) \tag{7.8}$$

$$\mathscr{D}_{hot}^{(can)} \oplus \mathscr{E}_1^{Vietnamese} = \{[\text{h oh t}], [\text{h } \textbf{oh}], [\text{h } \textbf{ow} \text{ t}], [\text{h oh } \textbf{d}]\}) \tag{7.9}$$

or in an L1-independent manner, allowing all error types ($\mathscr{E}_R^A$), irrespective of L1 e.g.

$$\mathscr{D}_{hot}^{(can)} \oplus \mathscr{E}_1^A = \{[\text{h oh t}], [\textbf{oh} \text{ t}], [\text{h } \textbf{oh}], [\text{h } \textbf{ow} \text{ t}], [\text{h oh } \textbf{d}]\}) \tag{7.10}$$

The result of this process is a dictionary entry containing, for each word, a combination of canonical pronunciations and errorful pronunciations representing up to R accent errors of, as needed, a particular type $\mathscr{D}^{(can)} \oplus \varepsilon_R$, of types corresponding to a particular L1 $\mathscr{D}^{(can)} \oplus \mathscr{E}_1^{L1}$, or of any type $\mathscr{D}^{(can)} \oplus \mathscr{E}_1^A$.

7.1.2 Generating lexical errors

The discussion in §3.3 defined lexical errors as those that arise from speakers failing to correctly convert the spelling of a word to its phonetic pronunciation (also known as sound-to-letter conversion errors [220]). The English language lacks a one-to-one correspondence between spelling and pronunciation, such that knowing the general phonetic rules of the language is not necessarily sufficient to know how to pronounce each word. It is also possible for a learner to misunderstand or misinterpret the phonetic rules of English, particularly where those contain exceptions and special cases. A model is therefore required to predict possible misinterpretations of the pronunciation of a word given its spelling.

It was seen that approaches in the literature include hand-crafted rules [238], which require extensive manual analysis and only cover one-to-one substitutions, generative statistical models [165], which require accurate human error annotations to train, and grapheme-to-phoneme (G2P) based approaches [220], which were determined to be the most suitable for the purposes of this thesis as they operate in an unsupervised manner and can learn more complex mappings.

Following these approaches, a grapheme-to-phoneme (G2P) system is trained on a standard canonical pronunciation dictionary to predict the canonical pronunciations of English words given their spelling. The hypothesis is that non-canonical pronunciations predicted by such a G2P are also likely to be made by a non-native speaker who hasn't encountered the word and guesses its pronunciation from its spelling. In contrast to accent errors, which are predicted based only on the word's canonical pronunciation and not spelling, lexical errors are predicted based on only spelling.

Note that this model doesn't capture L1-influenced letter-to-sound conversion failures, wherein a speaker whose native language also uses the Latin alphabet assigns phonetic values to letters based on the rules of their L1 rather than English (e.g. an Italian speaker pronouncing *city* as [ts iy t iy]). Such errors are expected to be captured, at least in part, by the accent error framework.

Following the G2P architecture from Bisani and Ney [27], parameters are learned to predict the probabilities of phone sub-sequences given grapheme sub-sequences (eg. tion => [sh ax n]) using expectation-maximisation (E-M) on all possible partitions and alignments of each words and corresponding phone sequence in the training dictionary. The trained model can then be used to initialise a similar model which predicts each phone sub-sequence given a grapheme sub-sequence and the immediately previous grapheme-phone sub-sequence pair (a 2-gram model), which is in turn trained by E-M and its final parameters used to initialise a 3-gram model and so on, up to a maximum context window size L. The value of L thus determines the number of neighbouring grapheme sub-segments that each mapping in the final model's predictions will take into account and is a key hyperparameter of the G2P.

The greater the value of L, the better the G2P will learn the rules of English, including memorising the pronunciation of words and morphemes, and thus be better able to predict canonical pronunciations. Setting the value too high will therefore lead to fewer candidate errors being proposed, as the system will only output canonical pronunciations, while setting it too low will lead to an output that is too noisy and deviates too much from the canonical pronunciation.

The canonical pronunciations and the top ranking non-canonical pronunciations predicted by the G2P constitute the combined pronunciation dictionary $\mathscr{D}_w^{(can)} \oplus \{lex\}_L$. The process of generating candidate lexical errors for each word as described above is illustrated in Figure 7.3 below.

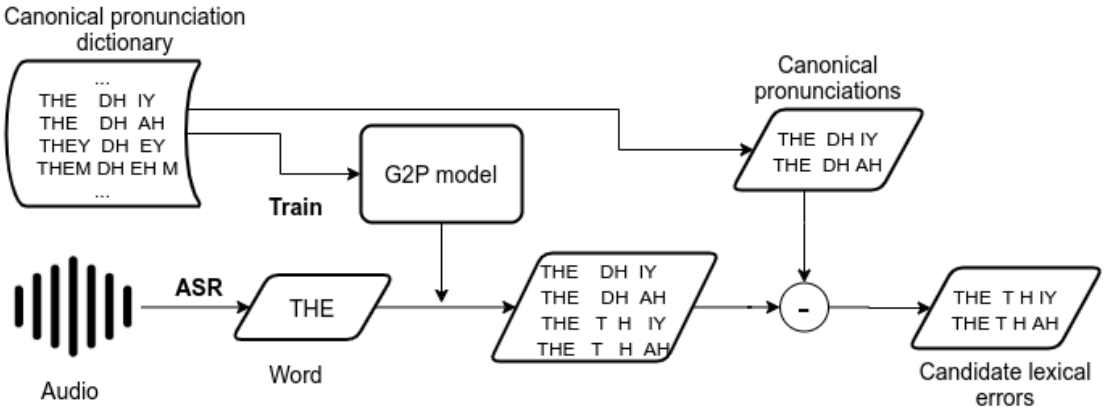


Fig. 7.3 Illustration of process for generating candidate lexical errors

Following from above discussion and that in the previous subsection, the tuning of the context window L as well as the maximum number of allowable accent errors per word R will need to be considered when running experiments on the effectiveness of the framework described in this section.

7.2 Accent and Lexical Error Detection

Given the framework for generating candidate accent and lexical errors introduced in the previous section, a methodology must now be developed to estimate the probability that any given word in a recorded utterance was pronounced with one of these candidate errorful pronunciations rather than one of the canonical pronunciations in the original dictionary. This probability estimate is then to be thresholded to detect errors at the word and utterance level. The described methodology must work with spontaneous speech, where the words spoken are not known in advance but recognised by an ASR with an accompanied degree of uncertainty.

Consider a spontaneous utterance with audio frames $\boldsymbol{o}_{1:T}$ in which errors of types contained in $\mathscr{E}$ are to be detected. The phones $\boldsymbol{\phi}_{m_1:m_2}^{(w_i)}$ corresponding to each word w_i are taken to belong to the dictionary $\mathscr{D}^{(can)} \oplus \mathscr{E}$ containing canonical pronunciations and errorful pronunciations of the type being detected for each word:

$$\boldsymbol{\phi}_{m_1:m_2}^{(w_i)} \in \mathscr{D}_{w_i}^{(can)} \oplus \mathscr{E} \tag{7.11}$$

such that:

$$\mathscr{D}_{w_i}^{(can)} \oplus \mathscr{E} = \begin{cases} \mathscr{D}^{(can)} \oplus \{lex\}_L & \mathscr{E} \text{ represents lexical errors} \\ \mathscr{D}^{(can)} \oplus \mathscr{E}_R & \mathscr{E} \text{ is all or a type of accent errors} \end{cases} \tag{7.12}$$

where L and R are hyperparameters respectively representing the context window size and maximum accent error per word as per the discussion in §7.1.

An error is present in w_i if its phones belong to one of the errorful rather than canonical pronunciations:

$$e(w_i) = \boldsymbol{\phi}_{m_1:m_2}^{(w_i)} \notin \mathscr{D}_{w_i}^{(can)} \tag{7.13}$$

The spontaneous nature of the speech means that the utterance word sequence $w_{1:I}$ and phone sequence $\phi_{1:M}$ and the words and phones each frame t corresponds to $s_{1:T}$ are not known a priori. Likely possibilities can be thought of as being arranged into a lattice $\mathscr{P}_{\mathscr{E}}$ such that each path $\pi \in \mathscr{P}_{\mathscr{E}}$ through the lattice represents a possible value of $\{w_{1:I}, \phi_{1:M}, s_{1:T}\}$. A simple example is illustrated in Fig. 7.4. It can be seen that the uncertainty in what the speaker said due to the spontaneous nature of speech adds a degree of complexity above the standard ERNs discussed in §3.3. To detect a pronunciation error it is thus not only necessary

to establish that the speaker pronounced a given recognised word non-canonically, but also that the recognised word was indeed the word that the speaker pronounced.

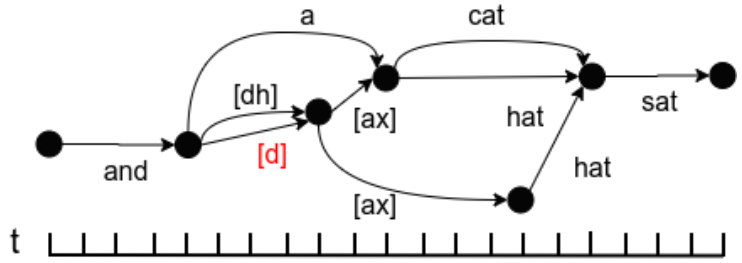


Fig. 7.4 Illustration of an example lattice for the problem of error detection in spontaneous speech. The speaker is saying *and a cat sat*, *and a hat sat*, *and the cat sat* or *and the hat sat*, the word *the* is pronounced as either the canonical pronunciation [dh ax] or the accent error [d ax] (corresponding to error type $\mathscr{E} = [dh] \rightarrow [d]$), and, if the third word is *hat*, there are two different time stamps that could be the boundary between the second and third words.

Let $P(\pi|\boldsymbol{o}_{1:T})$ be the posterior probability of the path π given the observed $\boldsymbol{o}_{1:T}$ such that:

$$\sum_{\pi \in \mathscr{P}_{\mathscr{E}}} P(\pi|\boldsymbol{o}_{1:T}) = 1 \tag{7.14}$$

Suppose an ASR has recognised the 1-best word sequence $\hat{w}_{1:I}$ given $\boldsymbol{o}_{1:T}$ (see §2.2):

$$\hat{w}_{1:I} = \arg\max_{w_{1:I}} \left\{ P(w_{1:I}) \sum_{\phi_{1:M} \in \mathscr{D}^{(can)}_{w_{1:I}}} \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \right\} \tag{7.15}$$

where $P(w_{1:I})$ is the language model, $p(\boldsymbol{o}_{1:T} s_{1:T}|\phi_{1:M})$ is the acoustic model and $\mathscr{D}^{(can)}$ is the canonical pronunciation dictionary. A pronunciation error is said to be detected in a recognised word $\hat{w}_i$ if it has been recognised correctly and pronounced non-canonically:

$$e(\hat{w}_i) = (w_i = \hat{w}_i) \cap (\phi^{(w_i)}_{m_1:m_2} \notin \mathscr{D}^{(can)}_{w_i}) \tag{7.16}$$

$$e\,(\hat{w}_i) = (w_i = \hat{w}_i) \cap e(w_i) \tag{7.17}$$

The posterior probability of an error in $\hat{w}_i$ is thus the sum of the posterior probabilities of paths $\pi \in \mathscr{P}_{\mathscr{E}} \mid e(\hat{w}_i)$ for which Equation 7.16 holds:

$$P(e(\hat{w}_i)|\boldsymbol{o}_{1:T}) = \sum_{\pi \in \mathscr{P}_{\mathscr{E}}|e(\hat{w}_i)} P(\pi|\boldsymbol{o}_{1:T}) \tag{7.18}$$

The application of Equation 7.18 is illustrated, for the joint ASR-alignment example from

Figure 7.4, in Figure 7.5 below.

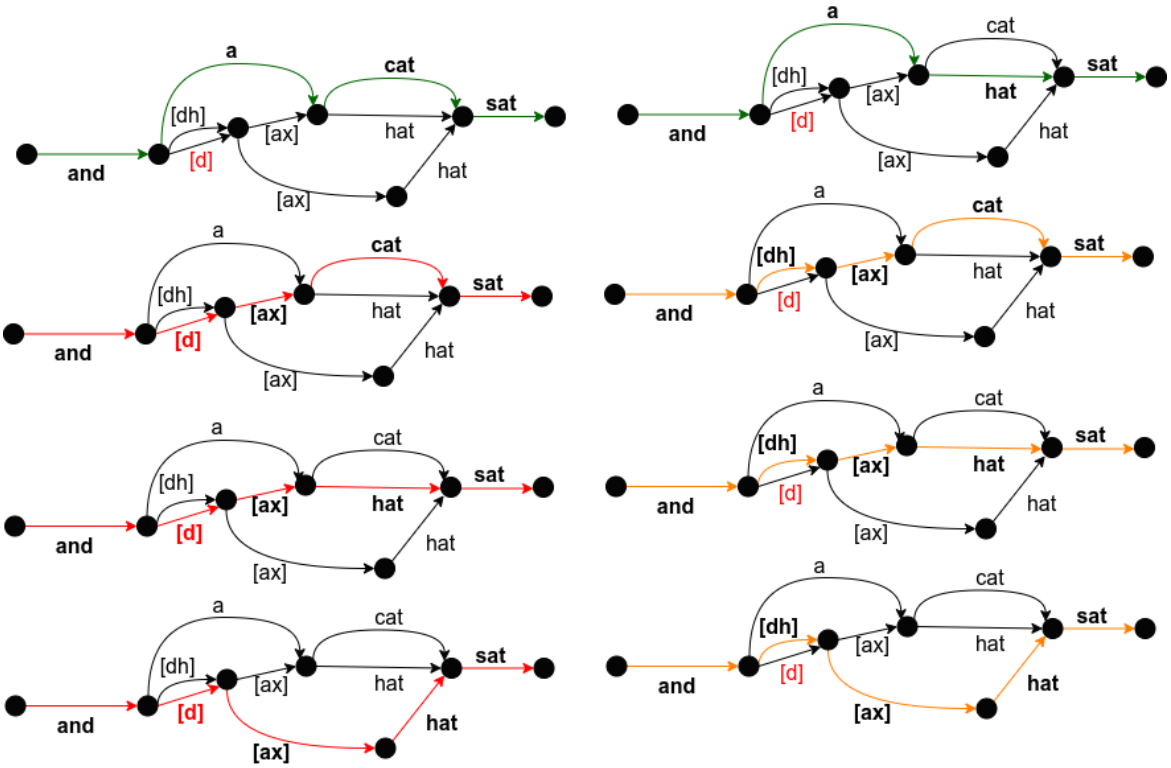


Fig. 7.5 Illustration of all possible paths through the lattice in Figure 7.4. The red and orange paths satisfy $w_1 = \textit{the}$, of which the red paths satisfy $\phi_{1:M}^{(w_1)} \notin \mathscr{D}_{w_1}^{(can)}$. The posterior probability of an error per Eq. 7.18 is given by the sum of the likelihoods of the red paths normalised by the sum of all paths. An estimate of confidence in the word *the* per Eq. 7.20 is given by the sum of the red and orange paths normalised by the sum of all paths.

As in practice it is not feasible to align with such a large lattice directly, the error posterior of Equation 7.18 is decomposed into:

$$P(e(\hat{w}_i)|\boldsymbol{o}_{1:T}) = P(e(w_i), \hat{w}_i|\boldsymbol{o}_{1:T}) = P(\hat{w}_i|\boldsymbol{o}_{1:T})P(e(w_i)|\hat{w}_i, \boldsymbol{o}_{1:T}) \tag{7.19}$$

where the probability $P(\hat{w}_i|\boldsymbol{o}_{1:T})$ of the ith word being recognised correctly can be estimated from the output lattice of the ASR as:

$$P(\hat{w}_i|\boldsymbol{o}_{1:T}) = \sum_{\pi|w_i^{(\pi)}=\hat{w}_i} P(\pi|\boldsymbol{o}_{1:T}) \tag{7.20}$$

or approximated by one of the methods for obtaining the confidence of the word given its aligned location [34, 286, 130, 150]:

$$P(\hat{w}_i|\boldsymbol{o}_{1:T}) \approx P(\hat{w}_i|\boldsymbol{o}_{t_1:t_2}^{(\hat{w}_i)}) \tag{7.21}$$

To compute the word-conditioned posterior $P(e(w_i)|\hat{w}_i, \boldsymbol{o}_{1:T})$, the utterance is force aligned for each word $\hat{w}_i$ with a dictionary $\mathscr{D}^{(can)}_{\hat{w}_{1:I}} \oplus \mathscr{E}(w_i)$ containing canonical pronunciations for all words except $\hat{w}_i$ and both canonical and errorful pronunciations for $\hat{w}_i$:

$$\mathscr{D}^{(can)}_{\hat{w}_j} \oplus \mathscr{E}(w_i) = \begin{cases} \mathscr{D}^{(can)}_{\hat{w}_i} \oplus \mathscr{E} & j = i \\ \mathscr{D}^{(can)}_{\hat{w}_j} & j \neq i \end{cases} \tag{7.22}$$

Force aligning the entire utterance each time allows different pronunciations of w_i to have different start and end frames, thus preventing the distortions associated with phone recognition methods. Using canonical pronunciations for all words except $\hat{w}_i$ mitigates the network size explosion effect associated with ERNs and increases the number of possible candidate errorful pronunciations that can be checked for per word. A simplified illustration of the resultant lattice is displayed in Fig. 7.6.

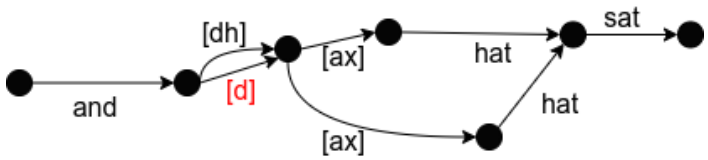


Fig. 7.6 Lattice for force alignment after ASR on Fig. 7.4 has yielded the 1-best word sequence *and the hat sat*. There are now only four paths. The posterior probability of an error is obtained by summing the likelihoods of the two paths through [d], normalised by the sum of the likelihoods of all paths, multiplied by an estimate of ASR word confidence (Eq. 7.25)

Following the lattice-based posterior estimation technique from Evermann and Woodland [79], the posterior $P(e(w_i)|\boldsymbol{o}_{1:T}, \hat{w}_i)$ is estimated using Bayes' Rule:

$$P(e(w_i)|\boldsymbol{o}_{1:T}, \hat{w}_i) = \frac{P(\boldsymbol{o}_{1:T}, e(w_i)|\hat{w}_i)}{P(\boldsymbol{o}_{1:T}|\hat{w}_i)} \tag{7.23}$$

by summing path likelihoods:

$$P(e(w_i)|\boldsymbol{o}_{1:T}, \hat{w}_i) = \frac{\sum_{\pi|e(w_i)} p(\boldsymbol{o}_{1:T}, \pi)^{\frac{1}{\gamma}}}{\sum_{\pi|w_i^{(\pi)}=\hat{w}_i} p(\boldsymbol{o}_{1:T}, \pi)^{\frac{1}{\gamma}}} \tag{7.24}$$

where γ is an acoustic model scaling factor used to flatten the posterior distribution, compensating for overestimation of the probability of the 1-best pronunciation caused by invalid independence assumptions during alignment [79].

Equation 7.24 is now substituted back into Equation 7.19 to yield an estimate for the posterior probability $P(e(\hat{w}_i)|\boldsymbol{o}_{1:T})$ that, given the observed frames of audio, the ith word

in the ASR output $\hat{w}_i$ was both correctly recognised and pronounced by the speaker in an errorful manner:

$$P(e(\hat{w}_i)|\boldsymbol{o}_{1:T}) = P(\hat{w}_i|\boldsymbol{o}_{1:T}) \frac{\sum_{\pi|e(w_i)} p(\boldsymbol{o}_{1:T},\pi)^{\frac{1}{\gamma}}}{\sum_{\pi|w_i^{(\pi)}=\hat{w}_i} p(\boldsymbol{o}_{1:T},\pi)^{\frac{1}{\gamma}}} \tag{7.25}$$

It is also possible to separately obtain the log likelihood of the errorful pronunciations:

$$\mathscr{L}_{err}(w_i) = \log P(\boldsymbol{o}_{1:T}, e(w_i)|\hat{w}_i) = \log \left(\sum_{\pi|e(w_i)} p(\boldsymbol{o}_{1:T},\pi)^{\frac{1}{\gamma}} \right) \tag{7.26}$$

and the negative log likelihood that the word was pronounced canonically:

$$\mathscr{L}_{\text{neg\_can}}(w_i) = -\log P(\boldsymbol{o}_{1:T}, \bar{e}(w_i)|\hat{w}_i) = -\log \left(\sum_{\pi|\bar{e}(w_i)} p(\boldsymbol{o}_{1:T},\pi)^{\frac{1}{\gamma}} \right) \tag{7.27}$$

where $\bar{e}(w_i)$ indicates that w_i is pronounced with a canonical pronunciation $\phi_{m_1:m_2}^{(w_i)} \in \mathscr{D}_{w_i}^{(can)}$.

Obtaining and thresholding $\mathscr{L}_{err}(w_i)$ and $\mathscr{L}_{neg\_can}(w_i)$ separately could be useful in the event there are systematic differences between the dynamic ranges of the path likelihoods obtained for errorful and canonical pronunciations. Experiments will also investigate the ratio of the two logs, a larger value of which indicates that the canonical likelihood is more negative than the error likelihood (i.e. an error is more likely than a canonical pronunciation):

$$\mathscr{L}_{\text{ratio}}(w_i) = -\frac{\log P(\boldsymbol{o}_{1:T}, e(w_i)|\hat{w}_i)}{\log P(\boldsymbol{o}_{1:T}, \bar{e}(w_i)|\hat{w}_i)} \tag{7.28}$$

Finally, it is possible to only consider the 1-best errorful and canonical pronunciations:

$$\mathscr{L}_{\text{err}}^{*}(w_i) = \log \left(\max_{\pi|e(w_i)} p(\boldsymbol{o}_{1:T},\pi) \right) \tag{7.29}$$

$$\mathscr{L}_{\text{neg\_can}}^{*}(w_i) = -\log \left(\max_{\pi|\bar{e}(w_i)} p(\boldsymbol{o}_{1:T},\pi) \right) \tag{7.30}$$

and the ratio between them (γ being no longer necessary due to the logs):

$$\mathscr{L}_{\text{ratio}} = -\frac{\log\left(\max_{\pi|e(w_i)} p(\boldsymbol{o}_{1:T},\pi)\right)}{\log\left(\max_{\pi|\bar{e}(w_i)} p(\boldsymbol{o}_{1:T},\pi)\right)} \tag{7.31}$$

Word-level errors can be detected by thresholding each of the metrics in Equations 7.25 - 7.31 or by training a classifier using all the metrics as inputs.

An utterance error is said to be present in utterance $e(\hat{w}_{1:I})$ if any of the words in the utterance contains an error:

$$e(\hat{w}_{1:I}) = \bigcup_{i=1}^{I} (e(\hat{w}_i)) \tag{7.32}$$

such that the posterior probability of an utterance error given the observed frames is:

$$P(e(\hat{w}_{1:I})|\boldsymbol{o}_{1:T}) = P(e(\hat{w}_1) \vee e(\hat{w}_2) ... \vee e(\hat{w}_I)|\boldsymbol{o}_{1:T}) \tag{7.33}$$

Given this definition, one option to detect utterance error detection is to re-align the whole utterance using a dictionary with both canonical and errorful pronunciations for all words, compute the posterior from path likelihoods and multiply with the ASR word confidence:

$$P(e(\hat{w}_{1:I})|\boldsymbol{o}_{1:T}) = P(\hat{w}_{1:I}|\boldsymbol{o}_{1:T}) \frac{\sum_{\pi|e(w_{1:I})} p(\boldsymbol{o}_{1:T}, \pi)^{\frac{1}{\gamma}}}{\sum_{\pi|w_{1:I}^{(\pi)}=\hat{w}_{1:I}} p(\boldsymbol{o}_{1:T}, \pi)^{\frac{1}{\gamma}}} \tag{7.34}$$

Such an approach would lead to an overly large lattice, requiring considerable pruning to evaluate, which in turn would cause distortions, such as many pronunciations not appearing the output lattice at all, and therefore prevent the meaningful estimation of posterior probabilities. Instead, it is desired to estimate $P(e(\hat{w}_{1:I})|\boldsymbol{o}_{1:T})$ from the individual word error posteriors $P(e(\hat{w}_i)|\boldsymbol{o}_{1:T})$ calculated as per Equation 7.25. These posteriors cannot be assumed to independent, especially in the case of accent errors, so the disjunction probability can only be evaluated in terms of its upper and lower bounds, given by the Frechet inequalities [87]:

$$P(e(\hat{w}_{1:I})|\boldsymbol{o}_{1:T}) \geq \max_i P(e(\hat{w}_i)|\boldsymbol{o}_{1:T}) \tag{7.35}$$

$$P(e(\hat{w}_{1:I})|\boldsymbol{o}_{1:T}) \leq \min\left(\sum_{i=1}^{I} P(e(\hat{w}_i)|\boldsymbol{o}_{1:T}), 1\right) \tag{7.36}$$

Utterance-level errors can thus be detected by either thresholding the lower bound from Equation 7.35, equivalent to checking whether any word errors were detected by thresholding the posterior of Equation 7.25, or thresholding the upper bound from Equation 7.36.

7.3 Corpora and Annotations

One of the advantages of ERN-based techniques such as those described in this chapter is their ability to detect errors in an unsupervised fashion without requiring a ground-truth

training set of manually annotated pronunciation errors. However, such annotated sets are still required to evaluate the performance of the system during development. This data should have high inter-annotator agreement and represent a consistent standard for marking words as errors which an automatic error detector tuned to the right threshold could match.

Two types of annotation are possible: exhaustive phonetic annotation (from which errors can be identified by looking up the annotated pronunciations in a canonical pronunciation dictionary) or annotation of errors directly. The discussion in §3.6 established a number of issues with annotated corpora of pronunciation errors in the literature, including poor inter-annotator agreement and the preponderance of read over spontaneous speech.

In the work in this chapter, three datasets are investigated: one of phonetically annotated spontaneous speech (LPINT), one of error-annotated spontaneous speech (BLT), and one of error-annotated read speech (SELL), on which experiments will be performed in §7.4. This section describes these datasets and investigates inter-annotator agreement in BLT and LeaP.

The LPINT dataset consists of the spontaneous part of the publicly available Learning Prosody in a Foreign Language (LeaP) corpus [102] and contains recordings of 35 non-native speakers of English, with 11 L1s (German, Thai, Korean, Arabic, French, Persian, Chinese, Hungarian, Polish, Russian and Spanish), being interviewed before and/or after a prosody training course. It is phonetically annotated by humans using the X-SAMPA alphabet [284], unlike BULATS and SELL where annotators marked pronunciation errors.

While each utterance in the LeaP corpus proper is only annotated once, the authors conducted a study [103] in which a single utterance (of spontaneous speech from a story retelling - not in LPINT) was annotated twice by each of five different annotators, two years apart. Intra-annotator agreement for each annotator was calculated along with inter-annotator agreement, measured by Cohen's κ [56], defined, for a pair of annotations, as:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{7.37}$$

where p_o is the fraction of words on which the two annotators agree and p_e is the fraction that would be expected by chance, given by:

$$p_e = p_1 p_2 + (1 - p_1)(1 - p_2) \tag{7.38}$$

where p_1 and p_2 are the proportion of words marked as errors by each annotator.

The results from the study [103] are replicated in Table 7.1. They indicate only fair agreement, as noted in the study in accordance with the guidelines for evaluating κ values set out in Landis et al. [155]. Intra-annotator agreement is higher than inter-annotator agreement, as would be expected, but is still only fair to moderate. These numbers are consistent with

those seen for similar tasks in the rest of the literature, as discussed in §3.6. They meanwhile contrast sharply with the substantial to almost perfect agreement between the same annotators on the tasks of transcribing words and prosodic segments and phrases. These observations suggest that the annotations encode some meaningful information that an unsupervised error detector should be able to learn to extract. However, there also appears to be an inherent ambiguity in the task of phonetic annotation that will limit the degree of agreement any effective system can attain with the single-annotator annotations on LPINT.

Annotator	1	2	3	4	5
1	0.34	0.28	0.31	0.37	0.23
2	0.28	0.32	0.25	0.30	0.15
3	0.31	0.25	0.38	0.30	0.22
4	0.37	0.38	0.30	0.57	0.23
5	0.23	0.15	0.57	0.23	0.36

Table 7.1 Cohen's κ for intra-annotator agreement between successive phonetic annotations of an utterance from LeaP by each of five annotators (leading column) and inter-annotator agreement between each pair thereof, from Tables 2 and 3 in [103]

The LPINT data was further pre-processed by performing speaker diarisation to remove the sections of interviewer speech, converting the X-SAMPA annotations to ARPABET, identifying annotated errors by looking up annotated pronunciations in the pronunciation dictionary and selecting corresponding corrections by minimising Levenshtein distance to the annotated errorful pronunciation. A COMBILEX pronunciation dictionary [229] of RP English was used.

The BLT dataset consists of candidate recordings from the Business Language Testing Service (BULATS) spoken English test for foreign learners [43]. It was kindly provided by Cambridge Assessment. It comprises spontaneous speech, manually transcribed by colleagues in the Cambridge University Computer Laboratory [39], with pronunciation errors and corrections annotated using ARPABET [144]. The dataset used in this work contains 226 speakers of varying proficiency, balanced for gender and between 6 L1s (Arabic, Dutch, French, Polish, Vietnamese and Thai). The same pronunciation dictionary is used with it as with LeaP.

As in the case of LPINT, most of the dataset is annotated by a single annotator (Annotator A). However, 315 of the 1453 utterances (i.e. 21.7% of the corpus) were also re-annotated by one of three additional annotators (Annotators B, C, D), such that each utterance was annotated by exactly two annotators, one of which was always A. Inter-annotator agreement

between each pair of annotations in this subset of the data are computed, measured by Cohen's κ and cross-correlation. The results are displayed in Table 7.2 below.

	Utterances #	Utterances %	Cohen's κ	Cross-correlation
Annotator A only	1138	78.3		-
Annotators A and B	103	7.1	0.220	0.396
Annotators A and D	129	8.9	0.159	0.240
Annotators A and C	83	5.7	0.186	0.366
Total	1453	100		-

Table 7.2 Statistics of human annotations of pronunciation errors on BLT corpus

The results indicate only slight to fair agreement between annotators, meaning the data contains even more noise and is likely to place even harsher limits on the performance of an automatic system than LPINT. Since the additional annotators annotated non-overlapping utterances, it is not possible to evaluate the agreement between annotators B, C and D and therefore determine whether the problem is unique to annotator A.

Aggregate annotation statistics of BLT are compared to those of LeaP in Table 7.3. The LeaP annotations appear more consistent, even though the task of phonetic transcription is more complicated than that of binary error annotation. It is also clear from the proportion of the words in each dataset that are marked as errors (in the LeaP case implicitly, by transcribing non-canonical pronunciations), that the BLT annotators are annotating a much smaller proportion of words as errorful than the LeaP annotators.

Corpus	Task	# Ann.	# Pairs	# Utt.	% Errors	Mean Cohen's κ
LeaP	Phonetic transcription	5	10	1	76.2†	0.26
BLT	Pron. error annotation	4	3	315	9.7	0.19

Table 7.3 Annotation statistics for the BLT and LeaP corpora. † This value is calculated on the data in LPINT as the held out utterance used in [103] was not available

A possible explanation for these results is that annotators asked to point out pronunciation errors are under-annotating the errors actually present, selecting those they believe are most important according to their differing subjective judgements, whereas annotators asked to exhaustively phonetically transcribe the data are forced to consider every word and therefore transcribe most errors. This would suggest that the BLT annotations lack the consistent standard of errorfullness necessary in a data set used for error detector development and evaluation, and would therefore be less suited to the task than LPINT.

The SELL-CORPUS [50] consists of recordings of 389 volunteer Chinese speakers of English of varying proficiency, gender balanced and spread across 8 L1s/dialects (Northern Mandarin, Southwest Mandarin, Wu, Cantonese, Xiang, Minnan, Hakka and Gan), reading phonetically balanced utterances sampled from Project Guttenberg, with pronunciation errors and corrections human annotated using ARPABET. SELL data comes with only one annotator and so inter-annotator agreement cannot be evaluated. The annotator assumed US pronunciations, so the CMU [281] pronunciation dictionary is used when processing it.

Detecting errors in spontaneous speech requires acoustic models to recognise the text spoken and align observations to sequences of phones. The accuracy of ASR systems using standard “off-the-shelf" acoustic models is, however, too low on non-native learner English, so, systems trained on non-native learners of English are used, as per Appendix C, specifically TD-gr for ASR and GH-ph for alignment.

To isolate the impact of the ASR on error detection performance, the manual word transcriptions corresponding to the annotations are also used alongside ASR outputs, such that the results using forced alignment based on each can be compared. In the case of BLT, a third word sequence via crowd-sourced transcription is also added, which is expected to be more accurate than the ASR output but less accurate than the expert manual transcription. Table 7.4 shows the size of each of the three corpora, the number of words and word error rates (WER) of the ASR and crowd-souced transcriptions evaluated against the manual transcription, and the proportions of those words that are marked as errors.

Data Set	Transcription	#Utterances	#Words	WER (%)	#Errors
BLT	MAN	1438	61722	-	5968 (9.7%)
	CS	1438	53668	15.5	4546 (8.5%)
	ASR	1438	51535	19.5	4464 (8.7%)
SELL	MAN	149	3003	-	363 (12.1%)
	ASR	149	2701	10.1	296 (11.0%)
LPINT	MAN	45	6536	-	4982 (76.2%)
	ASR	45	6732	19.0	4383 (65.1%)

Table 7.4 Number of detectable annotated errors in each dataset, using original manual (MAN), ASR and crowd-sourced (CS) [271] transcriptions.

As expected, the crowd-source transcription differs from the manual transcription but matches it more closely than the ASR. The two error-annotated datasets (BLT and SELL) have similar proportions of words marked as errors, while the phonetically transcribed LPINT has radically more, in line with what was previously noted.

7.4 Experiments

In §7.1, a framework was introduced for predicting accent and lexical candidate pronunciation errors. It was hypothesised that accent and lexical errors are distinct types of error that can be separately detected. In §7.2, a method based on forced alignment was proposed to use this framework to detect accent and lexical errors at the word level and diagnose the speaker's tendency to make lexical and specific types of accent error at the utterance level. §7.3 introduced three datasets: one containing phonetically transcribed spontaneous speech (LPINT), one containing error annotated spontaneous speech (BLT), and one containing error annotated read speech (SELL). Analysis of the annotations led to the hypothesis that error annotators in BLT and LPINT systematically under-annotate pronunciation errors.

This section describes experiments that were performed to investigate the above hypotheses and the performance of the error detection system, and discusses their results. In §7.4.1, the under-annotation hypothesis is investigated by having human annotators complete a re-formulated error annotation of BLT and comparing annotation statistics to those of the original annotators. The distinctness of accent and lexical errors and the effect of candidate generation hyper-parameters is investigated in §7.4.2 by comparing predicted candidate errors of each type to each other and to the annotated errors. The performance of the described system on error detection, its relationship to the patterns of human error annotations, and its sensitivity to the ASR are then evaluated in §7.4.3. Finally, §7.4.4 examines the relationship between numbers of pronunciation errors and proficiency grade.

7.4.1 Error annotation

It was hypothesised in §7.3 that annotators detecting errors in an utterance (such as in BLT and SELL) systematically under-annotate errors compared to those transcribing every phone spoken (such as in LeaP). This would in turn explain the higher number of errors annotated in LPINT compared to BLT and SELL and the higher degree of annotator agreement in LeaP compared to BLT. As an initial test of this hypothesis, a re-formulated error annotation task is developed, designed to limit the under-annotation effect, to be completed by a small number of human annotators on a subset of BLT.

Rather than being given an entire utterance and asked to mark incorrectly pronounced words, annotators are given individual words, with their surrounding context, one at a time, and asked to choose between a specified canonical and errorful pronunciation. If the under-annotation hypothesis holds, annotators in this new task are expected to label a greater proportion of words as errors and have a higher degree of agreement between them than in the original annotation.

For a subset of the annotated utterances in the BLT test set, words with high ASR confidence are identified and the system described in §7.2 used to propose the most likely canonical pronunciation and the most likely errorful pronunciation. For comparison a similar process is also undertaking for stress errors. The interfaces used are illustrated in Appendix L. They come in batches of 20 rows, each asking the annotator to choose between the canonical pronunciation, the 1-best errorful pronunciation or another errorful pronunciation. No word is evaluated more than once by the same annotators. A total of 18 batches i.e. 360 total annotations, of which 342 were valid, are completed by 8 annotators, across the two categories. Most words are annotated by 2-3 different annotators. The results are presented in Table 7.5, contrasted with the annotation figures for LeaP and BLT from §7.3.

Dataset	Task	$\bar{\kappa}$	% Error
LeaP	Phonetic transcription	0.26	76.2†
BLT	Pron. error annotation	0.19	9.7
BLT	Pron. error classification	0.24	41.1
BLT	Stress error classification	0.36	34.1

Table 7.5 Mean pairwise Cohen's κ and proportion of words annotated as errors on different annotation tasks † Error % is calculated on LPINT but κ on a held-out utterance

As expected, the proportion of words annotated as errors considerably increases with the new formulation of the task. Inter-annotator agreement is also higher and more closely matches that observed on LeaP. These results would therefore seem to support the under-annotation hypothesis. Inter-annotator agreement is higher for stress errors than pronunciation errors, consistent with the view that phonetic pronunciation is more subjective and harder to evaluate than stress. It is noted, however, the ability to confidently draw conclusions is limited by the small numbers of annotators involved, the fact that the phonetic transcription method was run on a different dataset to the other two, and the fact that different annotators were used in each of the three tasks. These issues should be addressed in future work.

7.4.2 Candidate error generation

In §7.1, it was hypothesised that non-native speakers make two distinct types of pronunciation error, accent and lexical, with methods presented for predicting each given an input word. Experiments are now conducted to investigate whether accent and lexical errors are indeed distinct types of error that can be separately predicted using the methods proposed.

The canonical pronunciations of each recognised word in each dataset are looked up in the corresponding dictionary and candidate accent errors generated as described in §7.1, with a

maximum number of error type applications per word of $R = 2$ to satisfy computational complexity constraints. Annotated errorful pronunciations which match one of these candidates are identified. The words in each dataset annotated by the humans as incorrectly pronounced have thus been separated into accent errors and unidentified errors. If the hypothesis of the distinctness of accent and lexical errors holds, the unidentified errors should be much more likely to be detected as lexical errors than the accent errors.

To investigate this, a Sequitur [202] G2P, with a context window size of $L = 3$, is trained on the full canonical dictionary and evaluated on each word, to produce a ranking of the 50 most likely pronunciations given each word's spelling. Each annotated error is looked up in this ranking. Based on the framework of §7.1, pronunciations included in the top 50 are more likely to be lexical errors than those that are not. If accent and lexical errors are indeed distinct, more unidentified errors should thus occur than accent errors. The cumulative frequencies of rankings of annotated errors in the output are plotted in Figure 7.7.

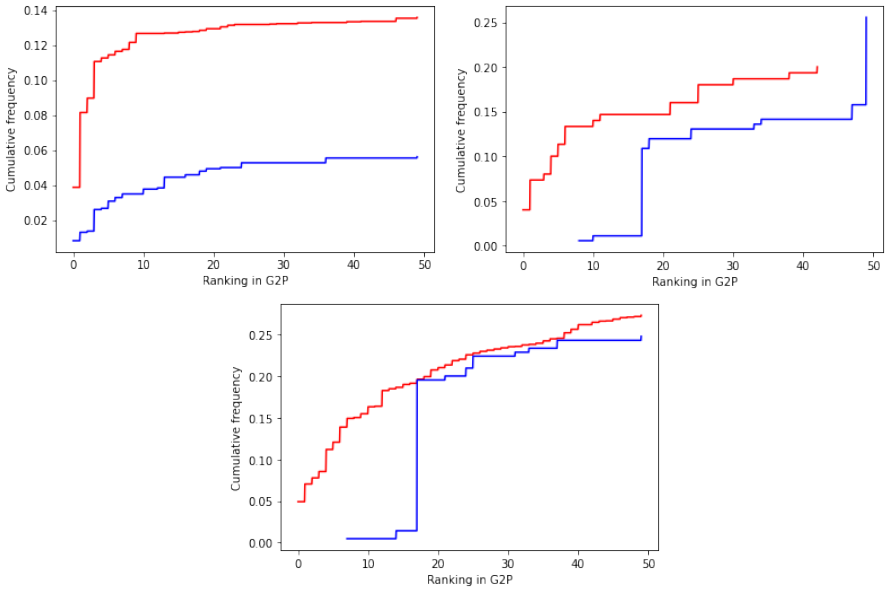


Fig. 7.7 Cumulative frequency of the ranking of identified accent errors (blue) and remaining errors (red) in BLT (top left), SELL (top right) and LPINT (bottom) among the 50-best outputs of a G2P system trained on the corresponding canonical dictionary

Across all datasets, most accent errors are absent from the G2P output and more unidentified errors are present than accent errors. This suggests that, while the G2P is not good at predicting accent errors, there exists a body of other annotated errors which which the G2P is considerably better at predicting. This is consistent with a significant fraction of the unidentified errors being lexical errors.

The experiment is repeated, splitting accent errors by type (Figure 7.8). It is seen that all accent error types rank below unidentified errors, further supporting the hypothesis that the unidentified errors contain lexical errors, which are distinct from accent errors. It is further seen that final deletions are the most likely to be predicted by G2P systems and voicing errors the least likely. This could be explained by final deletions overlapping with lexical errors the most (due to cases of believing a final letter to be silent) and voicing errors the least (as the voicing of English letters is mostly unambiguous [219]).

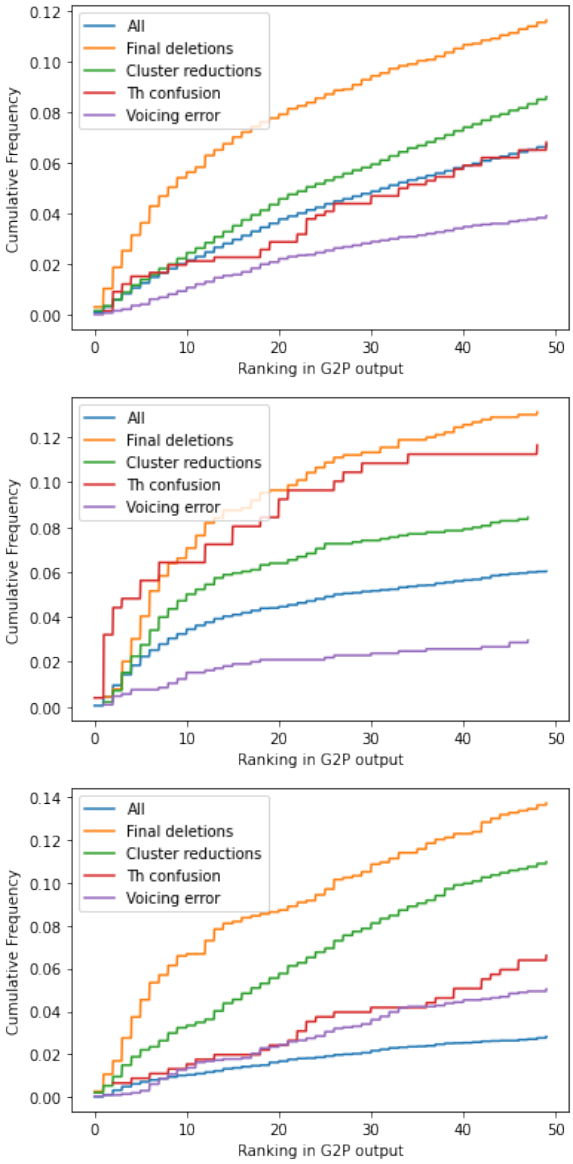


Fig. 7.8 Cumulative frequency of the ranking among 50-best G2P outputs of a sample of types of identified accent errors in BLT (top), SELL (middle) and LPINT (bottom)

A lexical error dictionary is now generated, keeping the first 10 pronunciations in each G2P output. The pronunciations in the accent and lexical error dictionaries are compared (Figure 7.9) and it is confirmed that the overlap is minimal.

Fig. 7.9 Overlap of accent and lexical error candidate pronunciations in the dictionaries generated for the words in BULATS (top left), SELL (top right) and LeaP (bottom)

The dictionaries are then used to identify the annotated errors. As seen in Figure 7.10, large numbers of annotated errors do not match any candidate errors, especially in LPINT. This is to be expected given the non-exhaustive nature of the generation algorithms.

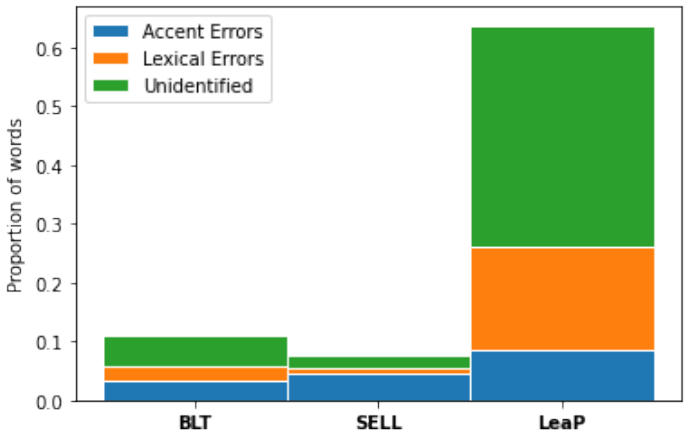


Fig. 7.10 Proportions of words annotated as errors and identified as accent and lexical.

As discussed in §7.1, one of the settings of the G2P system used to generate candidate lexical errors is the context window size L. The greater the value of L, the better the G2P

should be able to learn the letter-to-sound conversion rules of English. If L is too low, the G2P should suggest spurious pronunciations that neither a proficient nor non-proficient speaker would make. If L is set too high, its ability to predict lexical errors should be impeded. This hypothesis is tested by repeating the G2P ranking experiment with different values of L and plotting the median rankings of accent and remaining errors (Fig. 7.11).

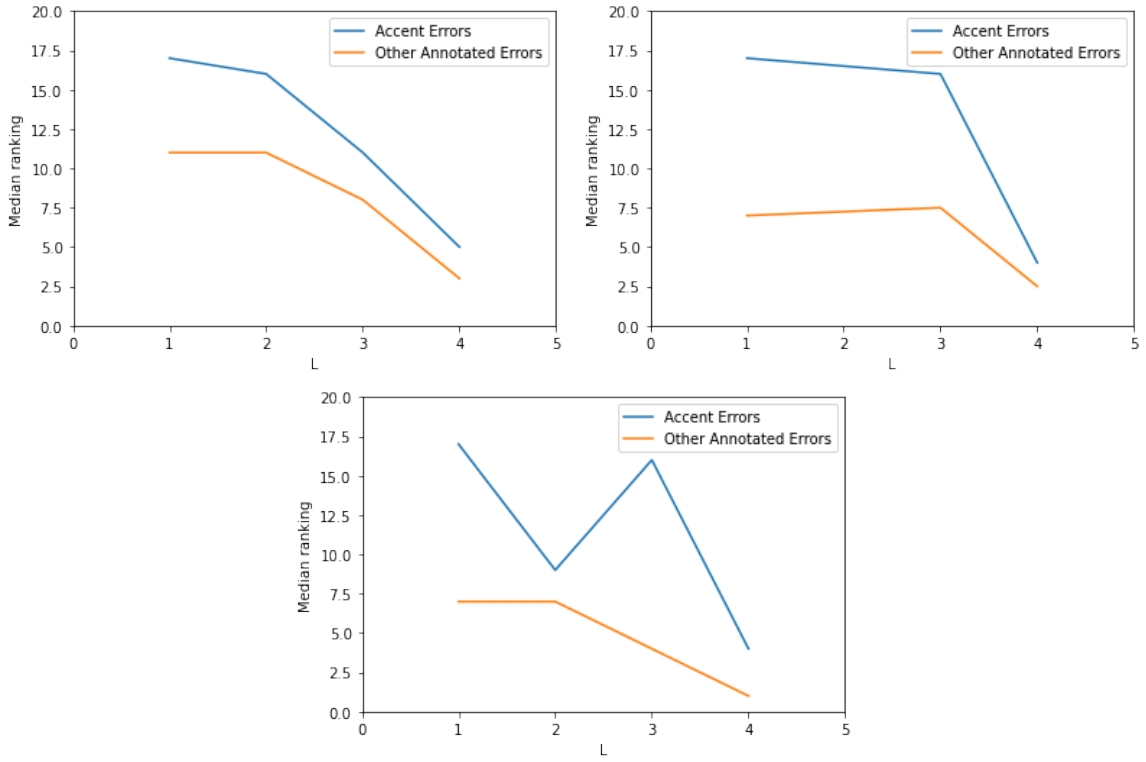


Fig. 7.11 Median ranking of accent and other errors among the 50-best G2P predictions for different context window sizes L on BLT (top left), SELL (top right) and LPINT (bottom)

Across all three datasets, the median rankings of both accent and unidentified errors are seen to improve as L is increased. This is consistent with the system making fewer spurious predictions and therefore improving the rankings of all other pronunciations. However, increasing L also shrinks the gap between the rankings of accent and lexical errors. This is consistent with wider G2P systems learning the letter-to-sound conversion rules of English too well and therefore no longer confusing lexical errors for canonical pronunciations. These results suggest that, as hypothesised, an intermediate value of L would be optimal for lexical error prediction. In the rest of this chapter, $L = 3$ is used across all three datasets.

7.4.3 Detection Performance

Having generated candidate accent and lexical errors for each word $\hat{w}_i$ recognised by the ASR in each utterance of each dataset, errors are now detected as described in §7.2. First, each utterance is force aligned, for each $\hat{w}_i$:

- with a dictionary $\mathscr{D}^{(can)} \oplus \{lex\}_3(\hat{w}_i)$ containing canonical pronunciations for all words except $\hat{w}_i$ and both canonical pronunciations and lexical errors for $\hat{w}_i$
- with a dictionary $\mathscr{D}^{(can)} \oplus \mathscr{E}_2^A(\hat{w}_i)$ containing canonical pronunciations for all words except $\hat{w}_i$ and both canonical pronunciations and accent errors for $\hat{w}_i$
- with dictionaries $\mathscr{D}^{(can)} \oplus \mathscr{E}_2^{final\_del}(\hat{w}_i)$, $\mathscr{D}^{(can)} \oplus \mathscr{E}_2^{dh\to d}(\hat{w}_i)$ etc. for each type of accent error, containing canonical pronunciations for all words except $\hat{w}_i$ and both canonical pronunciations and accent errors of the respective type for $\hat{w}_i$

The metrics detailed in Equations 7.25 - 7.31 are then extracted for each word and used to detect word-level errors. Results are reported based on the three prevailing methods. The first is thresholding the word-level posterior (Equation 7.25 - reproduced below):

$$P(e(\hat{w}_i)|\boldsymbol{o}_{1:T}) = P(\hat{w}_i|\boldsymbol{o}_{1:T}) \frac{\sum_{\pi|e(w_i)} p(\boldsymbol{o}_{1:T},\pi)^{\frac{1}{\gamma}}}{\sum_{\pi|w_i^{(\pi)}=\hat{w}_i} p(\boldsymbol{o}_{1:T},\pi)^{\frac{1}{\gamma}}} \tag{7.39}$$

The second is thresholding the 1-best likelihood ratio (Equation 7.31 - reproduced below):

$$\mathscr{L}_{\text{ratio}} = -\frac{\log\left(\max_{\pi|e(w_i)} p(\boldsymbol{o}_{1:T},\pi)\right)}{\log\left(\max_{\pi|\phi_{m_1:m_2}^{(w_i)} \in \mathscr{D}_{w_i}^{(can)}} p(\boldsymbol{o}_{1:T},\pi)\right)} \tag{7.40}$$

Finally, the third method is to feed all the features of Equations 7.25 to 7.31 as inputs to a binary classifier to detect errors, trained on held-out error annotations in a cross-validation fashion.

Utterance-level errors are detected by thresholding the upper and lower bounds on the probability of the utterance containing at least one error as per Equations 7.36 and 7.35, respectively reproduced below:

$$P(e(\hat{w}_{1:I})|\boldsymbol{o}_{1:T}) \leq \min\left(\sum_{i=1}^{I} P(e(\hat{w}_i)|\boldsymbol{o}_{1:T}), 1\right) \tag{7.41}$$

$$P(e(\hat{w}_{1:I})|\boldsymbol{o}_{1:T}) \geq \max_i P(e(\hat{w}_i)|\boldsymbol{o}_{1:T}) \tag{7.42}$$

A number of metrics are defined with which to evaluate the performance of each configuration of the system on each dataset. The first is *precision*, which measures the proportion of the errors detected by the system that were also annotated as such by the humans:

$$\text{precision} = \frac{|\{\text{annotated}\} \cap \{\text{detected}\}|}{|\{\text{detected}\}|} \tag{7.43}$$

where $\{\text{annotated}\}$ is, for word error detection, the number of words annotated as containing errors and, for utterance error detection, the number of utterances containing at least one example of the error type being detected, while $\{\text{detected}\}$ is the number of those words or utterances detected as errorful by the system.

The second is *recall*, also known as True Positive Rate (TPR), which measures the proportion of the errors annotated by the humans that the system is also able to detect:

$$\text{recall} = \text{TPR} = \frac{|\{\text{annotated}\} \cap \{\text{detected}\}|}{|\{\text{annotated}\}|} \tag{7.44}$$

The trade-off between precision and recall can be captured by plotting them against each other for different values of the relevant threshold or by characterising the performance of the system by their harmonic mean, known as F1 score, computed as:

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{7.45}$$

Next, False Positive rate (FPR) measures the proportion of those words or utterances that are not annotated as errors by the humans which the system does mark as errors:

$$\text{FPR} = \frac{|\{\text{detected}\}| - |\{\text{annotated}\} \cap \{\text{detected}\}|}{|\{\text{all}\}| - |\{\text{annotated}\}|} \tag{7.46}$$

where $|\{\text{all}\}|$ is the total number of annotated words or utterances.

The trade-off between TPR and FPR can be captured by plotting them against each other for different values of the relevant threshold, known as the the Receiver Operator Characteristic (ROC).

Figure 7.12 shows precision plotted against recall on the three datasets for accent error detection using the posterior thresholding method. To evaluate the sensitivity of the system to the ASR, the experiment is also repeated using expert and crowd-sourced manual transcriptions of each utterance instead of ASR recognised word sequences. It is seen that precision is very poor on BLT and SELL, though better on the latter, and considerably better on LPINT, consistent with the hypothesis that BLT and SELL were under-annotated. Using the manual transcription yields an improvement over ASR, but the difference is mostly minor. This is consistent with the system being robust to ASR error. As expected, performance using the

crowd-sourced transcription is worse than those using the manual transcription but better than those using the ASR output, consistent with the relative word error rates of the ASR and crowd-sourced transcriptions.

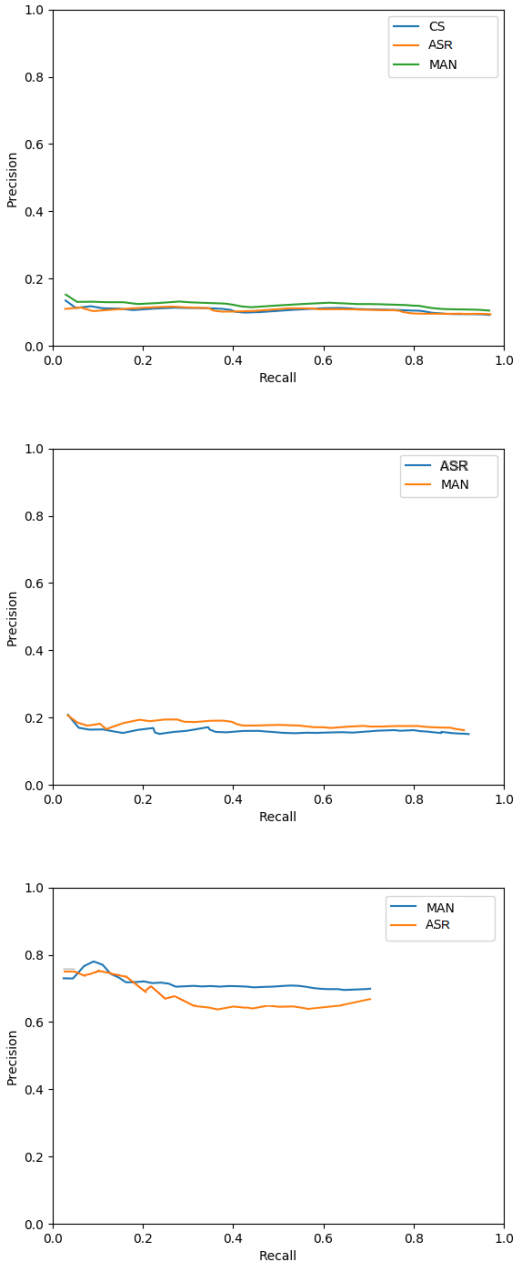


Fig. 7.12 Precision-recall curves for accent error detection on BLT (top), SELL (middle) and LPINT (bottom) using posterior thresholding, repeated using ASR output, manual transcription (MAN) and, the case of BLT, crowd-sourced transcriptions (CS)

The experiments are repeated with 1-best log ratio thresholding (Figure 7.13), which outperforms posterior thresholding across all three datasets. As before, performance on LPINT exceeds that on BLT and SELL, consistent with under-annotation of the latter, and that on manually transcribed data surpasses that on data from crowd-sourcing and ASR.

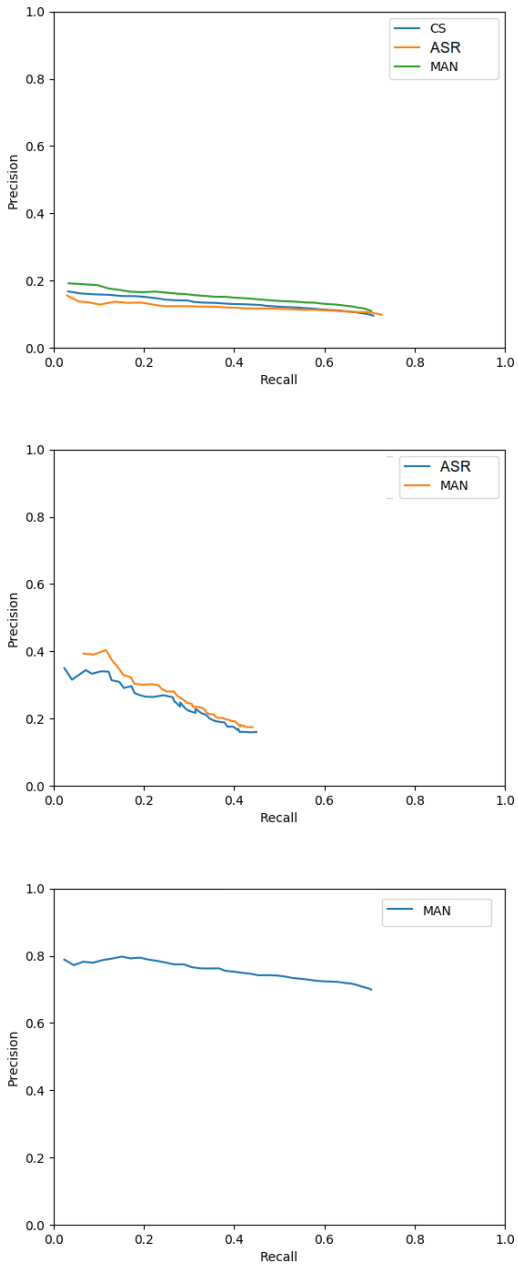


Fig. 7.13 Precision-recall curves for accent error detection on BLT (top), SELL (middle) and LPINT (bottom) using 1-best log ratio thresholding repeated with manual transcription (MAN), ASR output - for BLT and SELL -, and crowd-sourced transcriptions (CS) - for BLT

Figure 7.14 shows F1 scores for the tasks of detecting accent errors, detecting specific types of accent errors and detecting lexical errors at the word and utterance level, using posterior thresholding for the word error detection and each of the upper and lower bound thresholding methods for the utterance error detection. It is seen that the system can accurately predict the presence of accent and lexical errors at the word-level for LeaP, but not for SELL and BULATS. At the utterance level, on the other hand, the system can diagnose tendency for accent errors, lexical errors and specific types of accent errors, across all three corpora, using both methods, with the upper bound method outperforming the lower bound method.

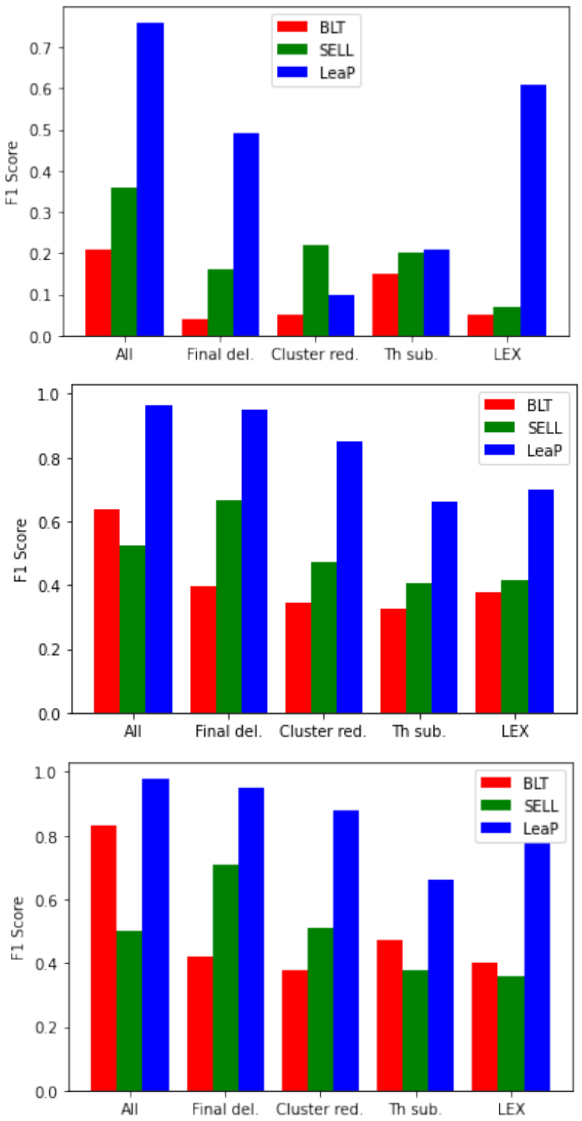


Fig. 7.14 F1 scores on each dataset for detecting accent errors ('All'), specific types of accent errors and lexical errors (top) and the presence of one or more of the above in a particular utterance, using the lower bound (middle) and upper bound (bottom) methods.

Investigating these results further, Figure 7.15 shows the expected number of accent errors plotted against the number of actual annotated errors for each dataset. It is seen that the two correlate strongly, especially for SELL and LPINT and, in the cases of BLT and SELL, more strongly than would be expected given the F1 scores of the prediction tasks. It is also noted that, for BLT and SELL, the expected number of accent errors is almost an order of magnitude greater than the actual annotated number of errors, while for LPINT this is not the case. These results, combined with those from Figure 7.14, are consistent with the annotators in BLT and SELL, who were instructed to specifically identify pronunciation errors, having annotated only a fraction of the accent errors actually present in the dataset. This would explain why both utterance-level performance and correlation between aggregated word-level posteriors and number of annotated errors are high, while word-level performance is low. It would also explain why for LPINT, where annotators were instructed to label every single phone and the expected and annotated numbers of errors match, all three results are instead consistent and high.

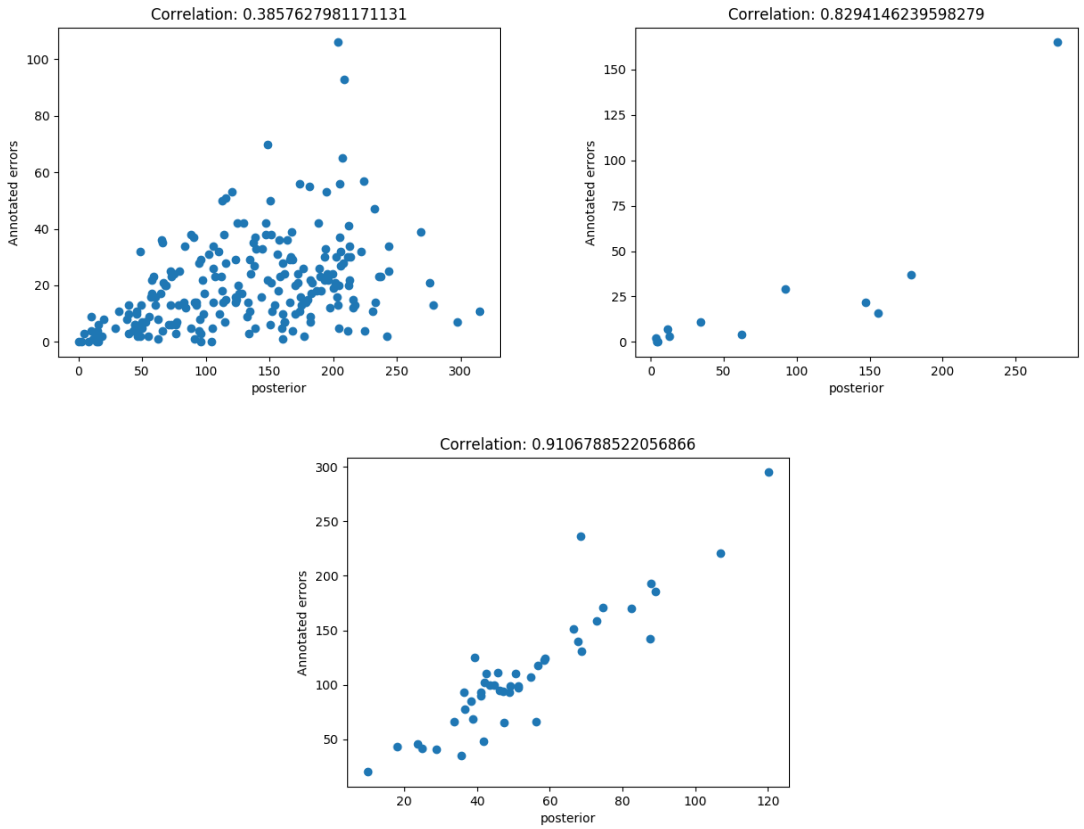


Fig. 7.15 Detected expected number of errors (sum of word-level posteriors) against annotated errors for each speaker in BLT (top left), SELL (top right) and LPINT (bottom).

7.4.4 Relationship to proficiency grade

Each speaker in the BLT and LPINT datasets has proficiency grades assigned to them by expert graders on a scale of 0-6, such as those used in Chapter 5. Figures 7.16 and 7.17 show the relationship between the number of annotated and detected errors and proficiency grade in each dataset. As expected, the less proficient a speaker, the more errors were annotated and detected. Consistent with the discussion in the §7.4.3, the system detects more errors than were annotated, but the number of errors detected has a similarly strong correlation with grade as the number annotated. The number of errors appears to have a stronger relationship to grade at lower proficiency scores, consistent with a limiting factor effect, whereby above a certain proficiency pronunciation errors are no longer an important component of assessment.

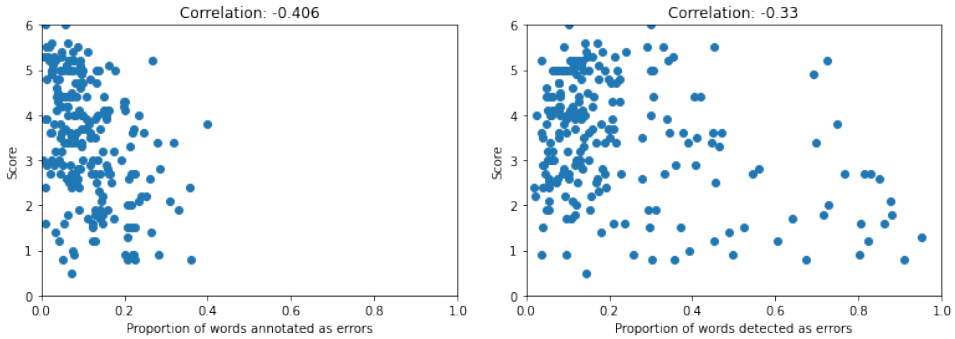


Fig. 7.16 Relationship between proficiency score and the numbers of annotated (left) and detected, at the F1-maximising threshold, (right) errors in BLT

On LPINT, the numbers of errors annotated and detected are more closely matched consistent with the previous discussion. Correlation between number of errors and grade is weaker, which, given the limited factor effect discussed above, could be a consequence of the proficiency of LPINT speakers being higher.

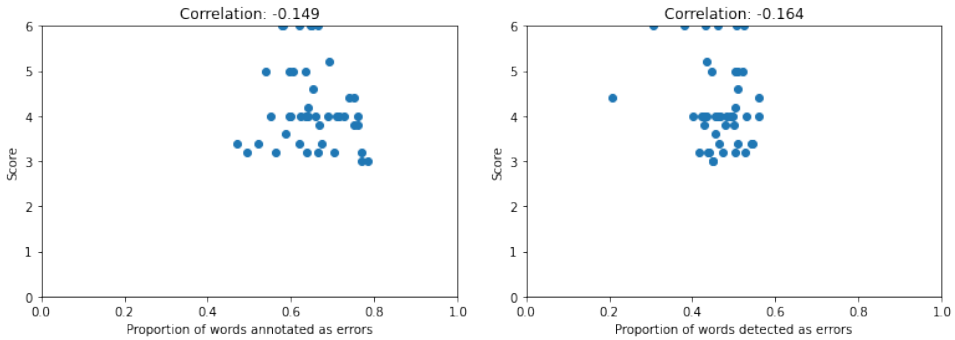


Fig. 7.17 Relationship between proficiency grade and the numbers of annotated (left) and detected, at the F1-maximising threshold, (right) errors in LPINT

The analysis is repeated for lexical errors (Figure 7.18). The relationship between number of lexical errors and proficiency is weaker than that of total errors, but nonetheless continues to hold equally strongly for both annotated and detected number of errors.

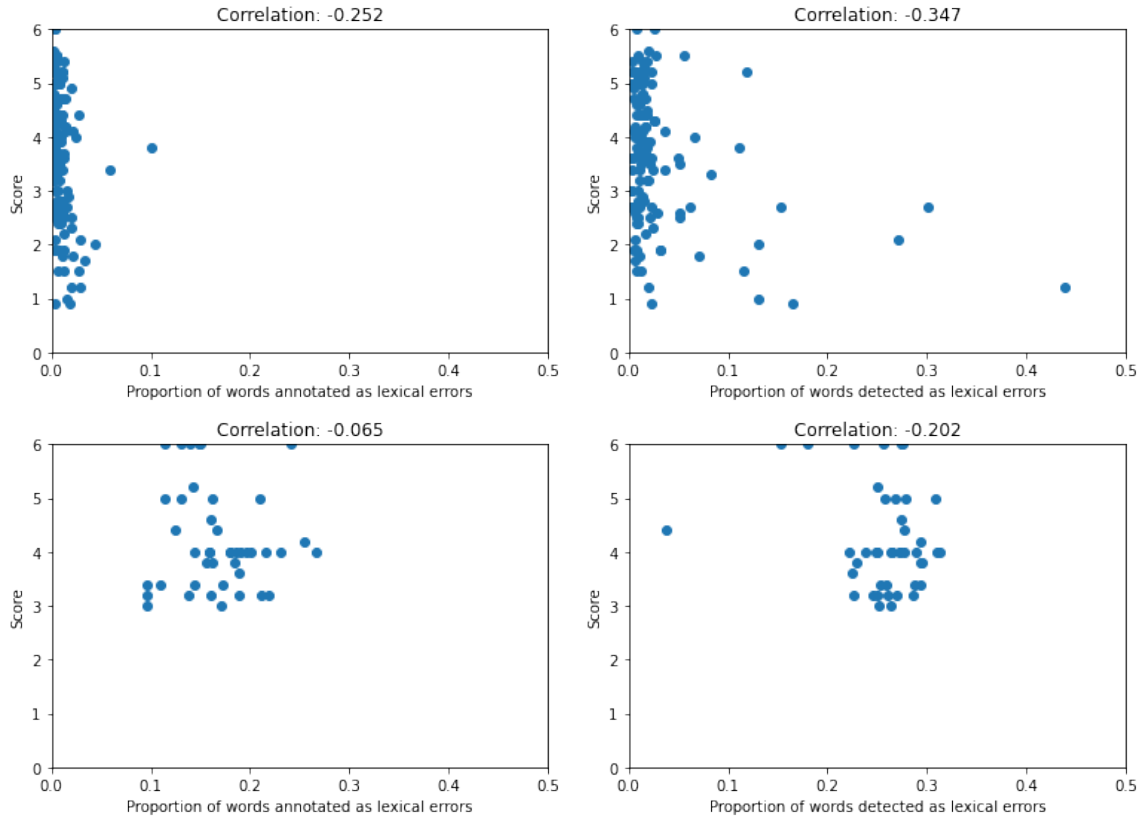


Fig. 7.18 Relationship between number of annotated (left) and detected (right) lexical errors in BLT (top) and LPINT (bottom)

7.5 Chapter Summary

This chapter investigated techniques to provide feedback on individual pronunciation errors made by a speaker, as well as on the speaker's tendency to make particular types of pronunciation errors at the utterance level. A framework was devised for generating candidate errorful pronunciations, divided into accent errors, patterns of phone-based substitutions, insertions and deletions which a speaker makes consistently, and lexical errors, word-specific errors caused by not knowing how to convert spelling to sounds (§7.1).

It was then seen how forced alignment can be used to assess the relative likelihood of these candidate errors compared to each word's canonical pronunciation given the audio of each word spoken by a non-native speaker, and how these confidence measures can be

used to predict which words were pronounced errorfully and whether an utterance contains a particular type of error (§7.2).

Three datasets were investigated, one consisting of phonetically transcribed spontaneous speech, one of error annotated spontaneous speech and one of error annotated read speech. Patterns in inter-annotator agreement and proportion of words annotated suggested that the error-annotated datasets might be under-annotated (§7.3).

A sample of error annotations were repeated with a re-formulated annotation task designed to reduce the under-annotation and both the proportion of words marked as errors and the inter-annotator agreement substantially increased (§7.4.1). Analysis of the G2P-based lexical error candidate generation framework was undertaken to determine the optimal context window size and confirm that the candidates produced are different to those produced by the accent error candidate generation framework and that both sets of candidates match substantial proportions of the errors marked by annotators in the data (§7.4.2).

It was seen that individual accent and lexical errors could be detected with precision in the phonetically transcribed dataset, while the tendency to make particular types of accent errors and lexical errors generally could be detected at the utterance level across all three datasets. Individual error detection in the error annotated datasets, however, was found to perform very poorly. Further analysis showed that the F1-maximising number of detected errors was considerably larger but correlated strongly with the number of annotated errors. This was found to be consistent with the hypothesis that the datasets were under-annotated. It was concluded that further work is needed, particularly in the area of annotation, but that the evidence so far seems to suggest the method described for error detection is promising (§7.4.3).

Finally, the numbers of errors annotated and detected were found to be negatively correlated with proficiency grade, with the relationship being stronger for lower grades. This suggests the number of pronunciation errors a speaker makes, whether annotated by humans or detected by the system described could be used as a partial indicator of overall proficiency for lower proficiency speakers, but that as speakers improve their English, other factors become more important in determining their proficiency (§7.4.4).

Chapter 8

Discussion

The main contributions of this thesis consist of the novel approaches to grading introduced in Chapter 5, the experiments into them reported in Chapter 6 and the investigation into pronunciation error detection undertaken in Chapter 7. This chapter discusses the implications and limitations of the results of the previous two chapters, as well as a number of future avenues of research suggested by the work done so far.

First, the work on grading is considered. The experiments reported in §6.3 compared two-stage single-view graders based on hand-crafted features to end-to-end networks generalising the principles behind them. The latter were seen to generally outperform the former, especially at generalisation tasks. In the case of pronunciation, it was also seen that the handcrafted features performed better at generalisation with a DNN than with a Gaussian Process grader. The suggested explanation for both these outcomes was that the systems with greater parametric representational learning capacity are better at capturing the underlying relationship between the input and output variables and thereby allow better generalisation to out-of-domain data. It could be useful to conduct more research into these phenomena. This could include replicating both sets of results on more datasets, tasks and graders and conducting experiments to investigate whether the suggested hypothesis is indeed the best explanation.

The experiments of this section also detected a calibration issue the end-to-end graders, which led to their PCC performance being better than MSE performance. While the existence and implications of the issue were clearly demonstrated, further investigation into the causes and and possible remedies of the problem is warranted.

One of the advantages of the attention mechanisms used in the end-to-end networks is the interpretability provided by their weights. This was exploited in the investigation into the attention weights of the pronunciation grader in §6.4. This work could be continued to test some of the hypotheses drawn based on the initial results. It would be interesting

to investigate whether the selection of instances by the attention weights indeed correlates to badly aligned phones and pronunciations errors as hypothesised. The latter could also provide an interesting point of linkage with the work on pronunciation error detection in Chapter 7. This analysis should also be extended to the attention mechanisms in the rhythm, intonation, and text graders as well as to that in the grader combination stage, which could help provide richer insight on the relative weights given to different views of proficiency in different contexts.

§6.5 investigated the question of whether the single-view end-to-end graders are indeed predict true single-view grades even though they are trained on holistic grades. As set out in §5.1, the graders were designed by leveraging the topology and inputs of hierarchical neural networks so that they only retain information salient to a single view. It was argued that such networks should be able to accurately predict true single-view grades even when trained on holistic grades, as long as the feature extractors only keep information about their respective views, the views are uncorrelated given their general component, and there is sufficient training data.

To ensure each feature extractor only retained information about its respective view, the inputs of the rhythm, intonation and text graders were respectively limited to durations of phones and intervals, measures of f_0 and probability of voicing, and the sequence of words spoken. In the case of pronunciation, the network was limited to considering only the manner of pronunciation of each phone relative to each of the others, by computing pair-wise scalar distances between aggregated representations of the instances of each phone. A limitation in the pronunciation grader was identified in its use of values of -1 to encode distances for which one of the phones is missing, thereby indirectly encoding aspects of message construction (i.e. phone occurrence frequency). Evidence of the resultant predicted overlap with the text grader was also observed experimentally (Table 6.11 in §6.5 and discussion in §6.4). Finding alternatives for dealing with missing phones in the pronunciation grader is thus an area of potential improvement.

By analysing the complementarity, correlation and behaviour of the scores predicted by different single-view graders, the results reported in §6.5 supported the conclusion that the graders indeed predict true single-view grades. However, the ability to demonstrate this conclusively is limited by the lack of human-annotated single-view grades to test against. Obtaining such data would make possible a number of experiments to shed light on the systems tested. It would be very useful, for example, to evaluate the performance of the single-view graders trained on holistic grades on human-annotated single-view grades, compared to the same systems trained on the single-view grades. Access to human-annotated single-view grades would allow hypotheses such as the assumptions regarding composition

and conditional independence of single-view scores (e.g. Equations 5.7 and 5.13) and the conclusions about different views being limiting at different score ranges (from §6.5) to be independently verified and further investigated.

The argument in §5.1 also suggested that a more powerful holistic grader could be obtained by aggregating the predictions of single-view graders for views exhaustively covering the aspects of proficiency taken into account by holistic grades. In addition to the novel pronunciation, rhythm and intonation graders, a text grader was reproduced to better approximate this range of aspects. In future work, the framework could also be applied to stress, the only one of the views considered in Chapter 2 not used in any of the systems in this thesis. A hierarchical attention-based end-to-end grader aggregating energy, duration and intonation features at the syllable level and then attending over syllables could be examined.

The results of §6.6 showed that combining single-view graders indeed outperformed baseline systems at the task of holistic grading. However, there is potential for additional improvements to further boost this grading performance. The three methods of grader combination proposed in §5.3 could be trained in both a two-stage configuration, fixing the weights of the trained component graders, or an end-to-end configuration, where the single-view graders are trained simultaneously with each other and the combination layer to optimally predict score. If single-view grades are available, more complex loss functions exploiting the multiple annotated grades per speaker could also be investigated. This could include composite loss functions for holistic grading and adversarial loss functions (penalising performance on views other than the one intended) to enforce view-specificity of single-view graders. In the end-to-end holistic grading case, the effect of the joint training on each single-view grader also suggests itself as a subject of investigation. The experiments of §6.3 and §6.5 could be repeated before and after this fine-tuning, to determine what effect being trained to be complementary to graders for other views will have on each grader's single-view performance and behaviour.

Regarding error detection, the work in Chapter 7 revealed a number of issues regarding the quality of data available to test error detection systems. The small scale re-annotation experiment of Appendix L should be extended to a full-scale re-annotation of all the datasets by multiple annotators. More research is needed into the guidelines given to annotators in relation to the usability of the annotations in error detection systems. It would be useful to run more tests of inter-annotator agreement and consistency across publicly available datasets, and to develop metrics and experiments that allow better comparison of different datasets to each other.

If better quality data could be obtained, more informative investigations could be conducted into the performance of the system proposed as well as other systems replicated from

the literature. With enough data, supervised error detection methods could be devised, using one of the sequence-to-sequence models of Chapter 4. Another avenue of work that could prove promising is the generation of artificial or modified data to train such supervised error detection systems.

Further, while the work on error detection in this thesis focused on pronunciation, intonation and stress also operate and are largely evaluated locally, as discussed in §2.3.6 and §2.3.4 respectively. Extension of the work of Chapter 3 and 7 to these two other areas of error detection would thus also be warranted.

Finally, while feedback and adaptive learning were an important motivator for the approaches taken in this thesis (e.g. in §2.1 and §3.7), it was not possible to conduct experiments to evaluate the hypotheses made regarding the types of feedback it was believed would be more useful for learners. Being able to conduct experiments on data from actual learners is essential for progress in this area. This could include collecting and analysing data from language learning apps in current use and evaluating the effectiveness of proposed feedback and adaptive learning techniques by deploying them on users.

Chapter 9

Conclusions

This thesis investigated the use of deep learning and other statistical techniques for automatic single-view grading, holistic grading and pronunciation error detection in spontaneous non-native English speech. The work was motivated by the need for useful feedback in the contexts of Computer Assisted Language Learning and auto-marking, within the framework of the ALTA project.

The scope of the work and research questions were introduced in Chapter 1. The literatures on grading and pronunciation error detection were reviewed in Chapters 2 and 3 respectively, while the deep learning methodologies used in the rest of the thesis were reviewed in Chapter 4. A novel approach to single and multi-view grading was presented in Chapter 5 and evaluated experimentally in Chapter 6 while a novel framework for error detection as well as experiments regarding the implications of error annotation was reported in Chapter 7. Finally, the implications and limitations of the experimental results and avenues of potential future work were discussed in Chapter 8. This Chapter summarises the conclusions of this thesis, with respect to the nine research questions set up in the introduction:

1. *Do deep learning approaches offer superior accuracy and generalisability to alternative machine learning approaches (specifically Gaussian Processes) on the task of grading the proficiency of non-native speakers?*

   In §6.3, deep learning models including DNNs fed with handcrafted features and end-to-end neural graders (see below) were evaluated against a Gaussian Process baseline. The neural systems were seen to outperform the baseline, particularly in terms of generalisation, while the end-to-end neural systems, which leveraged deep learning techniques to process highly structured inputs and limit the information retained by the network, outperformed the DNNs. It was concluded the neural models' capability for parametric representational learning is better at capturing the underlying relationship

between the input and output variables for these tasks, allowing improved performance and better generalisation to out-of-domain data.

2. *Can single-view end-to-end neural graders (i.e. end-to-end neural systems constrained by their input and structure to grade on the basis of specific views) offer superior accuracy and generalisability at the task of single-view proficiency grading to methods based on hand-crafted features?*

   The problem of single-view grading was framed in Chapter 5 as limiting the information that a parametric feature extractor retains from its speech input to be exclusively representative of a particular view of proficiency, while otherwise allowing it the freedom to learn the representation that can best predict human-annotated grades. A general approach to this was followed, identifying the raw information and structural relationships necessary to assess each view and then crafting a parametric, end-to-end neural grader following the structure. End-to-end graders were presented alongside expert features exploiting the same structures and were shown to act as their generalisations.

   For the view of rhythm, the input was limited to durations of vocalic and inter-vocalic intervals and their constituent phones and silences, grouped by interval and interval type. The inputs were then fed through a hierarchical series of attention mechanisms and LSTMs, framed as a generalisation of the expert feature extractors in the literature. For intonation, view-specificity was approached by limiting the input to fundamental frequency and probability of voicing per frame. A two-stage grader was proposed based on features extracted by a modified DCT. These were then generalised into an end-to-end grader in the form of a multi-head attention mechanism. The pronunciation grader, generalising a two-stage grader developed in previous work, approached the problem of imitating information transmitted to that indicative of the speaker's manner of realisation of phones by grouping phone instances by phone label, attending over embeddings of each instance and computing Euclidean distances between pairs of phones, thus characterising each relative to the others. To get the embeddings into a space in which Euclidean distance is representative of distance in pronunciation space, the parameters of the embedding stage were initialised using a Siamese network trained to classify pairs of phone instances as belonging to the same or different phones. A text grader, based on feeding BERT embeddings of recognised word sequences into a sequence-to-vector grader was replicated from other work at ALTA.

   In §6.3, the novel end-to-end systems proposed in Chapter 5 were shown to outperform novel and replicated two-stage counterparts on the task of holistic proficiency grading,

being particularly better at generalising to other datasets and tasks. The end-to-end pronunciation grader was also tested on the related task of L1 classification, again found to considerably outperform its two-stage counterpart. It was concluded that giving the network the freedom to tune its feature extraction stage in an end-to-end fashion enabled it to better capture the underlying relationship between inputs and score and therefore both more accurately predict human-assigned grades and overfit less to the data it is trained on.

3. *Can single-view end-to-end neural graders be interpretable as to their reasons for assigning grades?*

   In addition to the increased interpretability already afforded by validly predicting single-view grades compared to merely holistic grading (see below), §6.4 investigated whether intermediate representations of the pronunciation grader could be used to give further feedback on the reasons the system assigned a particular grade. It was seen that the attention weights aggregating phone instance representations to phone representations can be used to determine which phone instances were taken into account the most during grading. The identity of phone instances that were discarded as well as instances that provided the bulk of the information used to characterise a particular phone could thus be fed back. The phone distance representations could also be used to determine which phones were missing and which phones had high distance values to other phones (both of which were shown to correlate negatively with grade).

   As discussed in Chapter 8, there is considerable room to continue this work to test some of the hypotheses drawn based on the initial results, including testing whether instances with very low attention weights indeed correlate to badly aligned phones and whether instances with high attention weights and/or phones with high pair-wise distances correlate to the pronunciations errors considered in Chapter 7. The analysis should also be extended beyond pronunciation to the rhythm, intonation, and text graders.

4. *Can single-view end-to-end neural graders still validly grade on the basis of those views when trained on holistic grades?*

   The lack of ground-truth single-view grades for speakers in the corpora made it impossible to directly investigate the hypothesis that each single-view grader validly grades its respective view of proficiency. Instead, experiments were conducted in §6.5 to indirectly examine this question, the results of which, when considered together, provided considerable evidence in support of the hypothesis, with the caveat of the

overlap between the pronunciation and text grader due to the method of encoding missing phones.

In particular, pair-wise combination experiments first demonstrated that each of the four graders was complementary to each of the others, producing a greater gain when combining graders for different views than when combining graders of the same view in an ensemble. Next, analysis on rank correlations between the grader predictions yielded results consistent with what would be expected if each grader was indeed specific to its respective view.

Finally, the relationship between performance and ground-truth holistic grade was analysed. The rhythm grader was seen to be more accurate on high proficiency graders, which analysis suggested was due to rhythm not being a limiting factor in human annotators assigning holistic grades to poorer speakers. The converse was established for pronunciation and text, which results suggested were limiting at lower grades, such that human annotators consider all speakers above a certain level to be good enough and stop using these views for discrimination. These results further demonstrate the view-specificity of the single-view graders. However, they also illustrate an important limitation of the use of single-view graders trained on holistic grades to assess speakers at proficiency levels where the view in question is not limiting for holistic grading.

5. *Do systems based on combining multiple single-view end-to-end neural graders offer superior accuracy at the task of holistic grading to systems based on concatenating single-view handcrafted features and neural systems trained end-to-end on holistic grades?*

   In §5.3, holistic grading was approached based on three methods of combining the single-view end-to-end graders to yield a single score. In §6.6, the best approach was evaluated and found to outperform both a neural holistic grader without any view-specificity and a multi-view approach based on concatenated hand-crafted features. It was also seen that the baseline handcrafted grader outperformed the non-view-specific holistic grader. It was concluded that separately characterising individual views of proficiency and then aggregating the results is a superior approach for proficiency assessment to direct end-to-end holistic grading, but that end-to-end approaches to increase the tunability of each single-view can significantly boost performance over handcrafted features. This was attributed to the end-to-end single-view grader approach acting as a synthesis of the tunability of the end-to-end holistic grader and the domain knowledge-incorporation of the handcrafted approach.

6. *Does the approach of phonetic transcription (asking annotators to exhaustively transcribe the way a speaker pronounced each word) capture the pronunciation errors made by non-native speakers in their spontaneous speech than the approach of pronunciation error annotation (asking annotators to mark which words in recorded speech contain pronunciation errors)?*

   The experiments on error detection focused largely on issues with the data, which was identified as the main challenge for this task. §7.3 introduced three datasets, two annotated using error annotation (one in spontaneous speech and one in read speech) and one using phonetic transcription. Analysis of annotation statistics and inter-annotator agreement showed that the phonetic transcribers both annotated more errors in total and agreed more with each other on their annotations. In §7.4.1, it was shown that by re-annotating part of the error annotated dataset, this time asking annotators to explicitly select between an errorful and correct pronunciation for each word, the number of errors annotated in total and the inter-annotator agreement both increased.

   When testing error detection frameworks on the three datasets in §7.4.3, the automatic system predicted similar numbers of errors to those annotated by humans in the phonetically transcribed dataset but a much larger number of errors in the two error annotated datasets. While this resulted in word error detection performance being very low on the latter two datasets, both utterance error detection performance and the correlation between number of errors detected and proficiency grade on these datasets was much stronger than this low performance would suggest.

   Given all this evidence, it was concluded that the error annotated datasets are systematically under-annotated by the human annotators and that the phonetic transcriptions provide a superior ground-truth. As the same effect was observed on two datasets obtained from completely different sources, there is reason to believe this is general result, with phonetic annotation concluded to be a overall preferable paradigm for data collection.

7. *Are accent errors (errors caused by the speaker systematically inserting, deleting or substituting phones across their speech) distinct and capable of being separately detected to lexical errors (errors caused by a speaker not knowing the correct pronunciation of specific words based on their spelling)?*

   §7.1 introduced a framework for generating candidate pronunciation errors for any word, divided into accent and lexical errors. Accent errors were modelled by applying L1-dependent rules obtained from the literature across multiple L1s (Appendix K),

while lexical errors were modelled by training a G2P system to predict pronunciations from a word's spelling.

The experiments reported in §7.4.2 investigated whether these two types of errors can indeed be considered distinctly. It was seen that the accent error model only explained a fraction of the human-annotated errors in the datasets, that the lexical error model mostly generated errors that were not also generated by the accent error model, that human-annotated errors not explained by the accent error models ranked higher in the output of the lexical error model (i.e. were deemed more likely to be lexical errors) than those explained as accent errors, and that the sets of human-annotated errors explained by each of the accent and lexical models had a very small overlap with each other. Taken together, this evidence was interpreted to demonstrate that the two types of errors are indeed distinct and can be detected separately.

8. *Can a system based on calculating word-level probabilities from lattice path likelihoods obtained using force alignment of spontaneous non-native utterances with multiple candidate pronunciations be used to accurately detect individual lexical errors made by the speaker as well as the overall tendency of the speaker to make different types of accent errors?*

   A methodology for pronunciation error detection was introduced in §7.2 based on generating candidate errorful pronunciations for each of accent and lexical errors and repeatedly force aligning utterances with all but one word limited to canonical pronunciations. The resultant lattice was used to assess the probability that the one word for which errorful pronunciations were allowed contains an error of one of the types modelled. This probability was then thresholded for detection. Estimates of the probability of each utterance containing errors were also thresholded to perform utterance error detection. The accent error generation process can be narrowed to model individual types of error, allowing arbitrarily focused detection. It was argued that identifying the types of errors made, rather than their locations, is more important for feedback and a method for doing so was presented.

   The proposed pronunciation error detection framework was evaluated in §7.4.3. It was found to be able to precisely detect individual accent and lexical errors in the phonetically annotated data set and utterance-level errors across all three datasets. Utterance-level error detection could be broken down by pronunciation error type to give richer feedback. Given the under-annotation phenomenon noted above, the word-level errors detected by the system in the error annotated datasets (which generally had high recall but low precision) may actually provide a better estimate of the errors

actually present in the dataset than the human annotated ground-truth. Higher-quality data and more research into data collection were seen to be essential to be able to better evaluate such systems.

9. *Is the number of pronunciation errors detected by an error detection system predictive of their holistic proficiency grade?*

   The results reported in §7.4.4 demonstrated the numbers of errors annotated and detected in all three datasets to be negatively correlated with proficiency grade, with the relationship being stronger for lower grades. This suggests that the number of pronunciation errors a speaker makes, whether annotated by humans or detected by an automated system, could be used as an indicator of overall proficiency for lower proficiency speakers, but that as speakers improve their English, other factors become limiting. As discussed in Chapter 8 there are a number of potential avenues of future work to further explore the relationship between error detection and proficiency assessment.

References

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283.

[2] Abercrombie, D. (1967). *Elements of General Phonetics*. Edinburgh: University Press. p.97.

[3] Abrahamsson, N. (1999). Vowel epenthesis of/sc (c)/onsets in spanish/swedish interphonology: A longitudinal case study. *Language learning*, 49(3):473–508.

[4] Adda-Decker, M. and Lamel, L. (1999). Pronunciation variants across system configuration, language and speaking style. *Speech Communication*, 29(2-4):83–98.

[5] Ahmed, N., Natarajan, T., and Rao, K. R. (1974). Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93.

[6] Akinnaso, F. N. (1982). On the differences between spoken and written language. *Language and speech*, 25(2):97–125.

[7] Alikaniotis, D., Yannakoudakis, H., and Rei, M. (2016). Automatic text scoring using neural networks. *arXiv preprint arXiv:1606.04289*.

[8] Anh, N. T. N. (2019). *The Pronunciation Difficulties of Vietnamese Young Adult English Learners and Their Perceptions of These Difficulties*. PhD thesis, UNIVERSITY OF PEDAGOGY, HCMC.

[9] Arvaniti, A. (2009). Rhythm, timing and the timing of rhythm. *Phonetica*, 66(1-2):46–63.

[10] Asakawa, S., Minematsu, N., Isei-Jaakkola, T., and Hirose, K. (2005). Structural representation of the non-native pronunciations. In *INTERSPEECH*, pages 165–168.

[11] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

[12] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[13] Barros, A. M. d. V. (2003). Pronunciation difficulties in the consonant system experienced by arabic speakers when learning english after the age of puberty. *West Virginia University*.

[14] Batliner, A., Buckow, J., Niemann, H., Nöth, E., and Warnke, V. (2000). The prosody module. In *Verbmobil: foundations of speech-to-speech translation*, pages 106–121. Springer.

[15] Beckman, M. E. and Ayers, G. (1997). Guidelines for tobi labelling. *The OSU Research Foundation*, 3:30.

[16] Beckman, M. E. and Hirschberg, J. (1994). The tobi annotation conventions. *Ohio State University*.

[17] Beckman, M. E., Hirschberg, J., and Shattuck-Hufnagel, S. (2006). The original tobi system and the. *Prosodic typology: The phonology of intonation and phrasing*, 1:9.

[18] Behrman, A. (2017). A clear speech approach to accent management. *American Journal of Speech-Language Pathology*, 26(4):1178–1192.

[19] Bell, S., Yannakoudakis, H., and Rei, M. (2019). Context is key: Grammatical error detection with contextual word representations. *arXiv preprint arXiv:1906.06593*.

[20] Bengio, Y. (2009). *Learning deep architectures for AI*. Now Publishers Inc.

[21] Bernstein, J., Cohen, M., Murveit, H., Rtischev, D., and Weintraub, M. (1990). Automatic evaluation and training in english pronunciation. In *ICSLP*, volume 90, pages 1185–1188.

[22] Bertinetto, P. M. and Bertini, C. (2008). On modeling the rhythm of natural languages. In *Proceedings of the Fourth International Conference on Speech Prosody*.

[23] Bett, S. (2002). The number of phonemes in english. *In Memory of Ken Ives (1917–2002)*, 30:1.

[24] Bhat, S. and Yoon, S.-Y. (2015). Automatic assessment of syntactic complexity for spontaneous speech scoring. *Speech Communication*, 67:42–57.

[25] Bigi, B. and Hirst, D. (2012). Speech phonetization alignment and syllabification (sppas): a tool for the automatic analysis of speech prosody. In *Speech Prosody*, pages 19–22.

[26] Bingham, G., Macke, W., and Miikkulainen, R. (2020). Evolutionary optimization of deep learning activation functions. *arXiv preprint arXiv:2002.07224*.

[27] Bisani, M. and Ney, H. (2008). Joint-Sequence Models for Grapheme-to-Phoneme Conversion. *SPEECHCOM*, 50:434–451.

[28] Bishop, C. M. (2006a). *Pattern recognition and machine learning*. springer.

[29] Bishop, C. M. (2006b). *Pattern Recognition and Machine Learning*. Springer.

[30] Bohn, O.-S. and Flege, J. E. (1992). The production of new and similar vowels by adult german learners of english. *Studies in Second Language Acquisition*, pages 131–158.

[31] Bonaventura, P., Howarth, P., and Menzel, W. (2000). Phonetic annotation of a non-native speech corpus. In *Proceedings International Workshop on Integrating Speech Technology in the (Language) Learning and Assistive Interface, InStil*, pages 10–17.

[32] Bondarenko, O. (2014). Does russian english exist. *American Journal of Educational Research*, 2(9):832–839.

[33] Bourlard, H. A. and Morgan, N. (2012). *Connectionist speech recognition: a hybrid approach*, volume 247. Springer Science & Business Media.

[34] Brems, D. J. and Schoeffler, M. S. (1996). Automatic speech recognition (asr) processing using confidence measures. US Patent 5,566,272.

[35] Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the workshop on Speech and Natural Language*, pages 112–116. Association for Computational Linguistics.

[36] Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E., and Shah, R. (1993). Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688.

[37] Bugdol, M., Segiet, Z., and Kręcichwost, M. (2014). Pronunciation error detection using dynamic time warping algorithm. In *Information Technologies in Biomedicine, Volume 4*, pages 345–354. Springer.

[38] BULATS (BULATS). BULATS. Business Language Testing Service. Available: http://www.bulats.org/computer-based-tests/online-tests.

[39] Caines, A., Nicholls, D., and Buttery, P. (2017). Annotating errors and disfluencies in transcriptions of speech. Technical Report UCAM-CL-TR-915, University of Cambridge Computer Laboratory.

[40] Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D., and Wellner, P. (2006). The AMI meeting corpus: A pre-announcement. In *Machine learning for multimodal interaction*, pages 28–39. Springer.

[41] Carlisle, R. S. (1991). The influence of environment on vowel epenthesis in spanish/english interphonology. *Applied linguistics*, 12(1):76–95.

[42] Caruana, R., Lawrence, S., and Giles, C. L. (2001). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, pages 402–408.

[43] Chambers, L. and Ingham, K. (2011). The BULATS online speaking test. *Research Notes*, 43:21–25.

[44] Chan, A. Y. (2006). Cantonese esl learners' pronunciation of english final consonants. *Language, Culture and Curriculum*, 19(3):296–313.

[45] Chan, A. Y. and Li, D. C. (2000). English and cantonese phonology in contrast: Explaining cantonese esl learners' english pronunciation problems. *Language Culture and Curriculum*, 13(1):67–85.

[46] Chen, J.-C., Jang, J.-S. R., Li, J.-Y., and Wu, M.-C. (2004). Automatic pronunciation assessment for Mandarin Chinese. In *Proc. of the 2004 IEEE Int. Conference on Multimedia and Expo (ICME)*, volume 3, pages 1979–1982.

[47] Chen, L. and Evanini, K. (2010). Assessment of non-native speech using vowel space characteristics. *Speech Prosody*.

[48] Chen, L., Tao, J., Ghaffarzadegan, S., and Qian, Y. (2018). End-to-end neural network based automated speech scoring. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6234–6238. IEEE.

[49] Chen, L.-Y. and Jang, J.-S. R. (2012). Stress detection of english words for a capt system using word-length dependent gmm-based bayesian classifiers. *Interdisciplinary information sciences*, 18(2):65–70.

[50] Chen, Y., Hu, J., and Zhang, X. (2019). Sell-corpus: an open source multiple accented Chinese-English speech corpus for L2 English learning assessment. In *ICASSP*, pages 7425–7429.

[51] Cheng, S., Liu, Z., Li, L., Tang, Z., Wang, D., and Zheng, T. F. (2020). Asr-free pronunciation assessment. *arXiv preprint arXiv:2005.11902*.

[52] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[53] Chotimongkol, A., Thatphithakkul, S., Chootrakool, P., Hansakunbuntheung, C., and Wutiwiwatchai, C. (2011). The design and development of pelecan: Pronunciation errors from learners of english corpus and annotation. In *2011 International Conference on Speech Database and Assessments (Oriental COCOSDA)*, pages 36–41. IEEE.

[54] Cincarek, T., Gruhn, R., Hacker, C., Nthöb, E., and Nakamura, S. (2009). Automatic pronunciation scoring of words and sentences independent from the non-native's first language. *Computer Speech and Language*.

[55] Cleland, J., Lloyd, S., Campbell, L., Crampin, L., Palo, J.-P., Sugden, E., Wrench, A., and Zharkova, N. (2020). The impact of real-time articulatory information on phonetic transcription: ultrasound-aided transcription in cleft lip and palate speech. *Folia Phoniatrica et Logopaedica*, 72(2):120–130.

[56] Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

[57] Council of Europe (2001). *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Cambridge University Press.

[58] Coutinho, E., Hönig, F., Zhang, Y., Hantke, S., Batliner, A., Nöth, E., and Schuller, B. (2016). Assessing the prosody of non-native speakers of english: Measures and feature sets. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*.

[59] Cox, S. and Dasmahapatra, S. (2002). High-level approaches to confidence estimation in speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 10(7):460–471.

[60] Crossley, S. and McNamara, D. (2013). Applications of text analysis tools for spoken response grading. *Language Learning & Technology*, 17(2):171–192.

[61] Crystal, D. (2011). *A dictionary of linguistics and phonetics*, volume 30. John Wiley & Sons.

[62] Cucchiarini, C., Doremalen, J. v., and Strik, H. (2010). Fluency in non-native read and spontaneous speech. In *DiSS-LPSS Joint Workshop 2010*.

[63] Cucchiarini, C., Nejjari, W., and Strik, H. (2012). My pronunciation coach: Improving english pronunciation with an automatic coach that listens. *Language Learning in Higher Education*, 1(2):365–376.

[64] Cucchiarini, C., Strik, H., and Boves, L. (1997). Automatic evaluation of dutch pronunciation by using speech recognition technology. In *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 622–629. IEEE.

[65] Dauer, R. M. (1983). Stress-timing and syllable-timing reanalyzed. *Journal of phonetics*.

[66] Davis, L. (2016). The influence of training and experience on rater performance in scoring spoken language. *Language Testing*, 33(1):117–135.

[67] De Jong, N. H., Steinel, M. P., Florijn, A. F., Schoonen, R., and Hulstijn, J. H. (2012). Facets of speaking proficiency. *Studies in Second Language Acquisition*, 34(1):5–34.

[68] Deterding, D. (2001). The measurement of rhythm: A comparison of singapore and british english.

[69] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[70] Dikli, S. (2006). An overview of automated scoring of essays. *The Journal of Technology, Learning and Assessment*, 5(1).

[71] Diment, A., Fagerlund, E., Benfield, A., and Virtanen, T. (2019). Detection of typical pronunciation errors in non-native English speech using convolutional recurrent neural networks. In *Proc. Int. Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

[72] Douglas, S. R. (2015). The relationship between lexical frequency profiling measures and rater judgements of spoken and written general english language proficiency on the celpip-general test. *TESL Canada Journal*, 32(9):43–64.

[73] Duan, R., Kawahara, T., Dantsuji, M., and Nanjo, H. (2019). Cross-lingual transfer learning of non-native acoustic modeling for pronunciation error detection and diagnosis. *ATASLP*, 28:391–401.

[74] Duan, R., Kawahara, T., Dantsuji, M., and Zhang, J. (2017). Effective articulatory modeling for pronunciation error detection of l2 learner without non-native training data. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 5815–5819. IEEE.

[75] El Zarka, A. M. E. S. (2013). *The Pronunciation errors of L1 Arabic learners of L2 English: The role of modern standard Arabic and vernacular dialects transfer*. PhD thesis, The British University in Dubai (BUiD).

[76] Emflazie (2020). *Source-filter model diagram*. Wikimedia. https://creativecommons.org/licenses/by-sa/4.0/legalcode.

[77] Engwall, O. (2012). Pronunciation analysis by acoustic-to-articulatory feature inversion. In *ADEPT*, page 79.

[78] Engwall, O. and Bälter, O. (2007). Pronunciation feedback from real and virtual language teachers. *Computer Assisted Language Learning*, 20(3):235–262.

[79] Evermann, G. and Woodland, P. (2000). Posterior probability decoding, confidence estimation and system combination. In *Proc. Speech Transcription Workshop*, volume 27, pages 78–81. Baltimore.

[80] Feng, Y., Fu, G., Chen, Q., and Chen, K. (2020). SED-MDD: Towards sentence dependent end-to-end mispronunciation detection and diagnosis. In *ICASSP*, pages 3492–3496.

[81] Fitt, S., Richmond, K., and Clark, R. (2009). The combilex lexicon.

[82] Flege, J. E. (1996). English vowel productions by dutch talkers: More evidence for the "similar" vs "new" distinction. *Second-language speech: Structure and process*, 13:11–52.

[83] Flege, J. E., Munro, M. J., and Skelton, L. (1992). Production of the word-final english/t/–/d/contrast by native speakers of english, mandarin, and spanish. *The Journal of the Acoustical Society of America*, 92(1):128–143.

[84] Franco, H., Bratt, H., Rossier, R., Rao Gadde, V., Shriberg, E., Abrash, V., and Precoda, K. (2010). Eduspeak®: A speech recognition and pronunciation scoring toolkit for computer-aided language learning applications. *Language Testing*, 27(3):401–418.

[85] Franco, H., Neumeyer, L., Digalakis, V., and Ronen, O. (2000). Combination of machine scores for automatic grading of pronunciation quality. *Speech Communication*.

[86] Franco, H., Neumeyer, L., Kim, Y., and Ronen, O. (1997). Automatic pronunciation scoring for language instruction. SRI International.

[87] Fréchet, M. (1935). Généralisation du théoreme des probabilités totales. *Fundamenta mathematicae*, 1(25):379–387.

[88] Gales, M. and Young, S. (2008). *The application of hidden Markov models in speech recognition*. Now Publishers Inc.

[89] Gales, M. J. F. (1999). Semi-tied covariance matrices for hidden Markov models. *TSAP*, 7(3):272–281.

[90] Gao, Y., Xie, Y., Cao, W., and Zhang, J. (2015). A study on robust detection of pronunciation erroneous tendency based on deep neural network. In *Sixteenth annual conference of the international speech communication association*.

[91] Ghahremani, P., Manohar, V., Povey, D., and Khudanpur, S. (2016). Acoustic modelling from the signal domain using cnns. In *Interspeech*, pages 3434–3438.

[92] Gharsellaoui, S., Selouani, S. A., Cichocki, W., Alotaibi, Y., and Dahmane, A. O. (2018). Application of the pairwise variability index of speech rhythm with particle swarm optimization to the classification of native and non-native accents. *Computer Speech & Language*, 48:67–79.

[93] Gleason, J. (2011). Beaches and peaches: common pronunciation errors among 11 spanish speakers of english. *SOCIAL FACTORS IN PRONUNCIATION ACQUISITION*, page 205.

[94] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

[95] Goldman-Eisler, F. (1956). The determinants of the rate of speech output and their mutual relations. *Journal of Psychosomatic Research*, 1(2):137–143.

[96] Gonet, W. and Pietron, G. (2006). English interdental fricatives in the speech of polish learners of english. *Dydaktyka fonetyki języka obcego. Neofilologia*, 8:73–86.

[97] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

[98] Grabe, E. and Low, E. L. (2002). Durational variability in speech and the rhythm class hypothesis. *Papers in laboratory phonology*, 7(515-546).

[99] Graddol, D. (2006). *English Next*. British Council.

[100] Graham, C., Nolan, F., Caines, A., and Buttery, P. (2015). Automated assessment of non-native speech using vowel formant features. ALTA Institute, Phonetics Lab, DTAL - University of Cambridge.

[101] Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052 vol. 4.

[102] Gut, U. (2012). The LeaP corpus: A multilingual corpus of spoken. *Multilingual corpora and multilingual corpus analysis*, 14:3–23.

[103] Gut, U. and Bayerl, P. S. (2004). Measuring the reliability of manual annotations of speech corpora. In *Speech Prosody 2004, International Conference*.

[104] Hahn, L. D. (2004). Primary stress and intelligibility: Research to motivate the teaching of suprasegmentals. *TESOL quarterly*, 38(2):201–223.

[105] Hallé, P. A., Best, C. T., and Levitt, A. (1999). Phonetic vs. phonological influences on french listeners' perception of american english approximants. *Journal of phonetics*, 27(3):281–306.

[106] Hardison, D. M. (2004). Generalization of computer assisted prosody training: Quantitative and qualitative findings. *Language Learning & Technology*, 8(1):34–52.

[107] Harrison, A. M., Lo, W.-K., Qian, X.-j., and Meng, H. (2009). Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training. In *SLATE*.

[108] Hecht-Nielsen, R. et al. (1988). Theory of the backpropagation neural network. *Neural Networks*, 1(Supplement-1):445–448.

[109] Heckel, R. and Yilmaz, F. F. (2020). Early stopping in deep networks: Double descent and how to eliminate it. *arXiv preprint arXiv:2007.10099*.

[110] Hermansky, H. (1990). Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752.

[111] Hermansky, H., Ellis, D. P., and Sharma, S. (2000). Tandem connectionist feature extraction for conventional hmm systems. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 3, pages 1635–1638. IEEE.

[112] Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8).

[113] Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

[114] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[115] Hoffer, E., Banner, R., Golan, I., and Soudry, D. (2018). Norm matters: efficient and accurate normalization schemes in deep networks. In *Advances in Neural Information Processing Systems*, pages 2160–2170.

[116] Hönig, F., Batliner, A., and Nöth, E. (2012). Automatic assessment of non-native prosody annotation, modelling and evaluation. In *International Symposium on Automatic Detection of Errors in Pronunciation Training (IS ADEPT)*, pages 21–30.

[117] Hönig, F., Batliner, A., Weilhammer, K., and Nöth, E. (2010). Automatic assessment of non-native prosody for english as l2. In *Speech Prosody 2010-Fifth International Conference*.

[118] Hu, W., Qian, Y., Soong, F. K., and Wang, Y. (2015). Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers. *SPEECHCOM*, 67:154–166.

[119] Huang, L.-f., Kubelec, S., Keng, N., and Hsu, L.-h. (2018). Evaluating cefr rater performance through the analysis of spoken learner corpora. *Language Testing in Asia*, 8(1):1–17.

[120] Huckvale, M. (2007). Hierarchical clustering of speakers into accents with the accdist metric. ICPhS.

[121] Hudson, J. (2019). Pronunciation studio. *www. pronunciationstudio. com*, 5(4.15):3–15.

[122] Hulstijn, J. H. (2011). Language proficiency in native and nonnative speakers: An agenda for research and suggestions for second-language assessment. *Language Assessment Quarterly*, 8(3):229–249.

[123] Hussein, M. A., Hassan, H., and Nassef, M. (2019). Automated language essay scoring systems: a literature review. *PeerJ Computer Science*, 5:e208.

[124] Iadkert, K. and Hashim, A. (2020). The production of english codas by thai speakers. *Linguistics Journal*, 14(1).

[125] Im, D. J., Tao, M., and Branson, K. (2016). An empirical analysis of the optimization of deep network loss surfaces. *arXiv preprint arXiv:1612.04010*.

[126] Imoto, K., Tsubota, Y., Raux, A., Kawahara, T., and Dantsuji, M. (2002). Modeling and automatic detection of english sentence stress for computer-assisted english prosody learning system. In *Seventh International Conference on Spoken Language Processing*.

[127] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[128] Isaacs, T. (2008). Towards defining a valid assessment criterion of pronunciation proficiency in non-native english-speaking graduate students. *Canadian Modern Language Review*, 64(4):555–580.

[129] Ito, A., Lim, Y.-L., Suzuki, M., and Makino, S. (2005). Pronunciation error detection method based on error rule clustering using a decision tree. *Interspeech*.

[130] Jiang, H. (2005). Confidence measures for speech recognition: A survey. *Speech communication*, 45(4):455–470.

[131] Kalischová, M. I. H. and Volkova, Y. (2014). Analysis of errors in english pronunciation typical of russian speakers. *Pronunciation Studio*.

[132] Kamimura, K. and Takano, K. (2019). Pronunciation error detection in voice input for correct word suggestion. In *Proc. of 2019 Int. Electronics Symposium (IES)*, pages 490–493.

[133] Kane, M., Ahmed, Z., and Carson-Berndsen, J. (2012). Underspecification in pronunciation variation. In *ADEPT*, page 101.

[134] Kane, M., Cabral, J. P., Zahra, A., and Carson-Berndsen, J. (2011). Introducing difficulty-levels in pronunciation learning. In *SLATE*.

[135] Kang, S., Lee, G. G., Lee, H.-Y., and Kim, B. (2012). An automatic pitch accent feedback system for english learners with adaptation of an english corpus spoken by koreans. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 432–437. IEEE.

[136] Karhila, R., Smolander, A.-R., Ylinen, S., and Kurimo, M. (2019). Transparent pronunciation scoring using articulatorily weighted phoneme edit distance. *arXiv preprint arXiv:1905.02639*.

[137] Kato, T., Truong, Q.-T., Kitamura, K., and Yamamoto, S. (2019). Referential vowel duration ratio as a feature for automatic assessment of l2 word prosody. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6595–6599. IEEE.

[138] Kawai, G. and Hirose, K. (1998). A method for measuring the intelligibility and nonnativeness of phone quality in foreign language pronunciation training. In *ICSLP*.

[139] Kim, C. and Sung, W. (2002). Implementation of an intonational quality assessment system. In *INTERSPEECH*.

[140] Kim, Y., Franco, H., and Neumeyer, L. (1997). Automatic pronunciation scoring of specific phone segments for language instruction. In *EUROSPEECH*.

[141] Kingma, D. P. and Ba, J. (2015). Adam (2014), a method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR), arXiv preprint arXiv*, volume 1412.

[142] Kitamura, K., Kato, T., and Yamamoto, S. (2020). Tree-based clustering of vowel duration ratio toward dictionary-based automatic assessment of prosody in l2 english word utterances. In *Proc. 10th International Conference on Speech Prosody 2020*, pages 980–984.

[143] Kitikanan, P. (2016). *L2 English fricative production by Thai learners*. PhD thesis, Newcastle University.

[144] Klautau, A. (2001). ARPABET and the TIMIT alphabet.

[145] Kleinberg, R., Li, Y., and Yuan, Y. (2018). An alternative view: When does sgd escape local minima? *arXiv preprint arXiv:1802.06175*.

[146] Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *1995 international conference on acoustics, speech, and signal processing*, volume 1, pages 181–184. IEEE.

[147] Knill, K., Gales, M., Kyriakopoulos, K., Ragni, A., and Wang, Y. (2017). Use of graphemic lexicons for spoken language assessment. In *Proceedings of INTERSPEECH*, pages 2774–2778.

[148] Kombrink, S., Mikolov, T., Karafiát, M., and Burget, L. (2011). Recurrent neural network based language modeling in meeting recognition. In *Twelfth annual conference of the international speech communication association*.

[149] Koniaris, C. and Engwall, O. (2011). Phoneme level non-native pronunciation analysis by an auditory model-based native assessment scheme. In *INTERSPEECH*, pages 1157–1160.

[150] Kumar, A., Singh, S., Gowda, D., Garg, A., Singh, S., and Kim, C. (2020). Utterance confidence measure for end-to-end speech recognition with applications to distributed speech recognition scenarios. In *Proc. Interspeech*.

[151] Kyriakopoulos, K. (2016). Automatic assessment of english as a second language. Master's thesis, University of Cambridge.

[152] Kyriakopoulos, K., Knill, K. M., and Gales, M. J. (2018). A deep learning approach to assessing non-native pronunciation of english using phone distances. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2018, pages 1626–1630.

[153] Kyriakopoulos, K., Knill, K. M., and Gales, M. J. (2019). A deep learning approach to automatic characterisation of rhythm in non-native english speech. In *INTERSPEECH*, pages 1836–1840.

[154] Kyriakopoulos, K., Knill, K. M., and Gales, M. J. (2020). Automatic detection of accent and lexical pronunciation errors in spontaneous non-native english speech. *Interspeech*.

[155] Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

[156] LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

[157] LeCun, Y., Kanter, I., and Solla, S. (1990). Second order properties of error surfaces: Learning time and generalization. *Advances in neural information processing systems*, 3:918–924.

[158] Lee, A., Chen, N. F., and Glass, J. (2016). Personalized mispronunciation detection and diagnosis based on unsupervised error pattern discovery. In *ICASSP*, pages 6145–6149.

[159] Lee, A. and Glass, J. R. (2013). Pronunciation assessment via a comparison-based system. In *SLaTE*, pages 122–126.

[160] Lee, G. G., Lee, H.-Y., Song, J., Kim, B., Kang, S., Lee, J., and Hwang, H. (2017). Automatic sentence stress feedback for non-native english learners. *Computer Speech & Language*, 41:29–42.

[161] Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867.

[162] Levis, J. and Barriuso, T. A. (2011). Nonnative speakers'pronunciation errors in spoken and read english. *Social factors in pronunciation acquisition*, page 187.

[163] Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018). Visualizing the loss landscape of neural nets. In *Advances in neural information processing systems*, pages 6389–6399.

[164] Li, K., Qian, X., Kang, S., Liu, P., and Meng, H. (2015). Integrating acoustic and state-transition models for free phone recognition in l2 english speech using multi-distribution deep neural networks. In *SLaTE*, pages 119–124.

[165] Li, K., Qian, X., and Meng, H. (2017). Mispronunciation detection and diagnosis in L2 English speech using multidistribution deep neural networks. *ATASLP*, 25(1):193–207.

[166] Li, K., Wu, X., and Meng, H. (2017). Intonation classification for l2 english speech using multi-distribution deep neural networks. *Computer Speech & Language*, 43:18–33.

[167] Li, Y., Wei, C., and Ma, T. (2019). Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information Processing Systems*, pages 11674–11685.

[168] Li, Z. and Arora, S. (2019). An exponential learning rate schedule for deep learning. *arXiv preprint arXiv:1910.07454*.

[169] Lin, J., Gao, Y., Zhang, W., Wei, L., Xie, Y., and Zhang, J. (2020). Improving pronunciation erroneous tendency detection with multi-model soft targets. *Journal of Signal Processing Systems*, pages 1–11.

[170] Liu, Z., Xu, G., Liu, T., Fu, W., Qi, Y., Ding, W., Song, Y., Guo, C., Kong, C., Yang, S., et al. (2020). Dolphin: a spoken language proficiency assessment system for elementary education. In *Proceedings of The Web Conference 2020*, pages 2641–2647.

[171] Loos, E. (1997). *LinguaLinks: Glossary of linguistic terms*. SIL International.

[172] Loukina, A., Lopez, M., Evanini, K., Suendermann-Oeft, D., and Zechner, K. (2015). Expert and crowdsourced annotation of pronunciation errors for automatic scoring systems. In *Sixteenth Annual Conference of the International Speech Communication Association*.

[173] Low, E. L. and Grabe, E. (1995). Prosodic patterns in singapore english. In *Proceedings of the International Congress of Phonetic Sciences, Stockholm*, volume 3, pages 636–639.

[174] Lu, Y., Gales, M. J., Knill, K. M., Manakul, P., Wang, L., and Wang, Y. (2019a). Impact of asr performance on spoken grammatical error detection. In *INTERSPEECH*, pages 1876–1880.

[175] Lu, Y., Gales, M. J. F., Knill, K. M., Manakul, P. P., Wang, L., and Wang, Y. (2019b). Impact of ASR performance on spoken grammatical error detection. In *INTERSPEECH*, pages 1876–1880.

[176] Ludlow, K. (2020). *Official Quick Guide to Linguaskill*. Cambridge University Press.

[177] Luong, M.-T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

[178] Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3.

[179] Malinin, A., Van Dalen, R., Knill, K., Wang, Y., and Gales, M. (2016). Off-topic response detection for spontaneous spoken english assessment. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1075–1084.

[180] Maqsood, M., Habib, H. A., Nawaz, T., and Haider, K. Z. (2016). A complete mispronunciation detection system for arabic phonemes using svm. *IJCSNS*, 16(3):30.

[181] Martin, J. H. and Jurafsky, D. (2000). Speech and language processing. *International Edition*.

[182] Mehta, G. and Cutler, A. (1988). Detection of target phonemes in spontaneous and read speech. *Language and Speech*, 31(2):135–156.

[183] Metallinou, A. and Cheng, J. (2014). Using deep neural networks to improve proficiency assessment for children english language learners. In *INTERSPEECH*, pages 1468–1472.

[184] Miles, S. and Kwon, C.-J. (2008). Benefits of using call vocabulary programs to provide systematic word recycling. *ENGLISH TEACHING*, 63(1):199–216.

[185] Minematsu, N., Asakawa, S., and Hirose, K. (2006). Structural representation of the pronunciation and its use for call. In *Spoken Language Technology Workshop, 2006. IEEE*, pages 126–129. IEEE.

[186] Minematsu, N., Kobashikawa, S., Hirose, K., and Erickson, D. (2002). Acoustic modeling of sentence stress using differential features between syllables for english rhythm learning system development. In *Seventh International Conference on Spoken Language Processing*.

[187] Miodonska, Z., Bugdol, M. D., and Krecichwost, M. (2016). Dynamic time warping in phoneme modeling for fast pronunciation error detection. *Computers in Biology and Medicine*, 69:277–285.

[188] Mohamed, A.-r., Dahl, G., and Hinton, G. (2009). Deep belief networks for phone recognition. In *Nips workshop on deep learning for speech recognition and related applications*, number 9 in 0, page 39. Vancouver, Canada.

[189] Mok, P. and Dellwo, V. (2008). Comparing native and non-native speech rhythm using acoustic rhythmic measures: Cantonese, beijing mandarin and english. *University of Zurich*.

[190] Moore, R. K. and Skidmore, L. (2019). On the use/misuse of the term'phoneme'. *arXiv preprint arXiv:1907.11640*.

[191] Morcos, A. S., Barrett, D. G., Rabinowitz, N. C., and Botvinick, M. (2018). On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*.

[192] Morrill, T., Baese-Berk, M., and Bradlow, A. (2016). Speaking rate consistency and variability in spontaneous speech by native and non-native speakers of english. In *Proceedings of the International Conference on Speech Prosody*, volume 2016, pages 1119–1123.

[193] Moustroufas, N. and Digalakis, V. (2007). Automatic pronunciation evaluation of foreign speakers using unknown text. *Computer Speech and Language*, 21:219–230.

[194] Mueller, J. and Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. In *AAAI*, pages 2786–2792.

[195] Müller, P., De Wet, F., Van Der Walt, C., and Niesler, T. (2009). Automatically assessing the oral proficiency of proficient l2 speakers. In *SLaTE*, pages 29–32.

[196] Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

[197] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *ICML*.

[198] Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. (2019). Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*.

[199] Neri, A., Cucchiarini, C., and Strik, H. (2001). Effective feedback on l2 pronunciation in asr-based call. *University of Nijmegen, The Netherlands*.

[200] Neumeyer, L., Franco, H., Digalakis, V., and Weintaub, M. (2000). Automatic scoring of pronunciation quality. *Speech Communication*, 30.

[201] Neumeyer, L., Franco, H., Weintraub, M., and Price, P. (1996). Automatic text-independent pronunciation scoring of foreign language student speech. In *ICSLP*, volume 3, pages 1457–1460.

[202] Nevill-Manning, C. G. and Witten, I. H. (1997). Identifying hierarchical structure in sequences: A linear-time algorithm. *Journal of Artificial Intelligence Research*, 7:67–82.

[203] Nguyen, N. (2008). Interlanguage phonology and the pronunciation of english final consonant clusters by native speakers of vietnamese. *Unpublished master's thesis, Ohio University*.

[204] Nguyen, T. T. T. (2007). Difficulties for vietnamese when pronouncing english: Final consonants.

[205] Nicolao, M., Beeston, A. V., and Hain, T. (2015). Automatic assessment of English learner pronunciation using discriminative classifiers. In *ICASSP*, pages 5351–5355.

[206] Nitta, T., Manosavan, S., Iribe, Y., Katsurada, K., Hayashi, R., and Zhu, C. (2012). Pronunciation training by extracting articulatory movement from speech. In *Int. Symposium on Automatic Detection of Errors in Pronunciation Training*, page 75.

[207] of Europe. Council for Cultural Co-operation. Education Committee. Modern Languages Division, C. (2001). *Common European Framework of Reference for Languages: learning, teaching, assessment*. Cambridge University Press.

[208] Palaz, D., Collobert, R., and Doss, M. M. (2013). End-to-end phoneme sequence recognition using convolutional neural networks. *arXiv preprint arXiv:1312.2137*.

[209] Palaz, D., Doss, M. M., and Collobert, R. (2015). Convolutional neural networks-based continuous speech recognition using raw speech signal. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4295–4299. IEEE.

[210] Park, J., Diehl, F., Gales, M., Tomalin, M., and Woodland, P. C. (2011). The efficient incorporation of mlp features into automatic speech recognition systems. *Computer Speech & Language*, 25(3):519–534.

[211] Pike, K. L. (1945). The intonation of american english. *ERIC*.

[212] Porzuczek, A. (2015). Handling global and local english pronunciation errors. In *Teaching and researching the pronunciation of English*, pages 169–187. Springer.

[213] Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M., and Khudanpur, S. (2018). Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Interspeech*, pages 3743–3747.

[214] Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., and Khudanpur, S. (2016). Purely sequence-trained neural networks for asr based on lattice-free mmi. In *Interspeech*, pages 2751–2755.

[215] Povey, D. and Woodland, P. C. (2002). Minimum phone error and i-smoothing for improved discriminative training. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–105. IEEE.

[216] Power, T. (2011). Pronunciation by nationality. *www. tedpower. co. uk*.

[217] Prafianto, H., Nose, T., Chiba, Y., and Ito, A. (2019). Improving human scoring of prosody using parametric speech synthesis. *Speech Communication*, 111:14–21.

[218] Prechelt, L. (1998). Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer.

[219] Press, O. U. (2015). *Oxford Advanced Learner's Dictionary*. Oxford University Press.

[220] Qian, X., Meng, H., and Soong, F. (2010). Capturing l2 segmental mispronunciations with joint-sequence models in computer-aided pronunciation training (capt). In *2010 7th International Symposium on Chinese Spoken Language Processing*, pages 84–88. IEEE.

[221] Rabiner, L. (1993). Fundamentals of speech recognition. *Fundamentals of speech recognition*.

[222] Radwanska-Williams, J. and Yam, J. P. (2001). The acquisition of english plosives by chinese learners. *PTLC2000*.

[223] Raina, V., Gales, M., and Knill, K. (2020). Universal adversarial attacks on spoken language assessment systems. *Interspeech 2020*.

[224] Ramanarayanan, V., Lange, P. L., Evanini, K., Molloy, H. R., and Suendermann-Oeft, D. (2017). Human and automated scoring of fluency, pronunciation and intonation during human-machine spoken dialog interactions. In *INTERSPEECH*, pages 1711–1715.

[225] Ramus, F. (2002). Acoustic correlates of linguistic rhythm: Perspectives. *Laboratoire de Sciences Cognitives et Psycholinguistique (EHESS/CNRS)*.

[226] Ramus, F., Nespor, M., and Mehler, J. (1999). Correlates of linguistic rhythm in the speech signal. *Cognition*, 73(3):265–292.

[227] Rashid, M., van Dalen, R. C., Malinin, A., Knill, K., and Gales, M. (2015). Spontaneous spoken language assessment using a statistical parser. ALTA Institute/Department of Engineering, University of Cambridge.

[228] Rehman, I., Silpachai, A., Levis, J., Zhao, G., and Gutierrez-Osuna, R. (2020). The english pronunciation of arabic speakers: A data-driven approach to segmental error identification. *Language Teaching Research*, page 1362168820931888.

[229] Richmond, K., Clark, R., and Fitt, S. (2010). On generating Combilex pronunciations via morphological analysis. In *INTERSPEECH*, pages 1974–1977.

[230] Robertson, S., Munteanu, C., and Penn, G. (2016). Pronunciation error detection for new language learners. In *INTERSPEECH*, pages 2691–2695.

[231] Rogers, C. L. and Dalby, J. (2005). Forced-choice analysis of segmental production by chinese-accented english speakers. *Journal of Speech, Language, and Hearing Research*.

[232] Ronanki, S., Henter, G. E., Wu, Z., and King, S. (2016). A template-based approach for speech synthesis intonation generation using lstms. In *INTERSPEECH*, pages 2463–2467.

[233] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

[234] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

[235] Salimans, T. and Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 901–909.

[236] Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization? *Advances in neural information processing systems*, 31:2483–2493.

[237] Sao Bui, T. (2016). Pronunciation of consonants/ð/and/θ/by adult vietnamese efl learners. *Indonesian Journal of Applied Linguistics*, 6(1):125–134.

[238] Schaden, S. (2004). Generating non-native pronunciation lexicons by phonological rule. In *Proc. ICSLP*, number 4 in 1.

[239] Schmidhuber, J. (2015). Deep learning. *Scholarpedia*, 10(11):32832.

[240] Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

[241] Selkirk, E. (1995). Sentence prosody: Intonation, stress, and phrasing. *The handbook of phonological theory*, 1:550–569.

[242] Semina, A. (2014). Comparison of english pronunciation errors made by czech and russian speakers on the segmental level. Master's thesis, MASARYK UNIVERSITY.

[243] Shahin, M. A., Epps, J., and Ahmed, B. (2016). Automatic classification of lexical stress in english and arabic languages using deep learning. In *INTERSPEECH*, pages 175–179.

[244] Sheriff, A. (2015). Phonology in english 1 phonological problem areas in english for native french speakers. In *French L1 Phonology in English*.

[245] Shermis, M. D. and Hamner, B. (2012). Contrasting state-of-the-art automated scoring of essays: Analysis. In *Annual national council on measurement in education meeting*, pages 14–16.

[246] Shriberg, L. D. and Lof, G. L. (1991). Reliability studies in broad and narrow phonetic transcription. *Clinical Linguistics & Phonetics*, 5(3):225–279.

[247] Smith, L. N. (2015). No more pesky learning rate guessing games. *CoRR, abs/1506.01186*, 5.

[248] Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*.

[249] Smith, L. N. and Topin, N. (2019). Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612. International Society for Optics and Photonics.

[250] Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018). X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. IEEE.

[251] Sobkowiak, W. (2001). *English Phonetics for Poles: A resource book for learners and teachers*. Wydaw. Poznañskie.

[252] SONG, H.-p. and ZHOU, W.-j. (2015). A review of research on english pronunciation errors made by chinese efl learners. *Journal of Zhenjiang College*, 2(2):8.

[253] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

[254] Stevens, S. S., Volkmann, J., and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190.

[255] Strik, H. and Cucchiarini, C. (1999). Automatic assessment of second language learners' fluency. *A2RT, Dept. of Language and Speech, University of Nijmegen, the Netherlands*.

[256] Strik, H., Cucchiarini, C., and Binnenpoorte, D. (2000). L2 pronunciation quality in read and spontaneous speech. *China Military Friendship Publish*.

[257] Strik, H., Truong, K., De Wet, F., and Cucchiarini, C. (2009). Comparing different approaches for automatic pronunciation error detection. *Speech Communication*, 51(10):845–852.

[258] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27:3104–3112.

[259] Takai, K., Heracleous, P., Yasuda, K., and Yoneyama, A. (2020). Deep learning-based automatic pronunciation assessment for second language learners. In *International Conference on Human-Computer Interaction*, pages 338–342. Springer.

[260] Talkin, D. (2015). Reaper: Robust epoch and pitch estimator. *Github: https://github.com/google/REAPER*.

[261] Tam, H. C. (2005). Common pronunciation problems of vietnamese learners of english. *VNU Journal of Foreign Studies*, 21(1).

[262] Tamburini, F. and Caini, C. (2005). An automatic system for detecting prosodic prominence in american english continuous speech. *International Journal of speech technology*, 8(1):33–44.

[263] TechNavio (2015). Global digital english language learning market 2015-2019. Technical report, TechNavio.

[264] Tepperman, J. and Narayanan, S. (2005). Automatic syllable stress detection using prosodic features for pronunciation evaluation of language learners. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 1, pages I–937. IEEE.

[265] Tepperman, J. and Narayanan, S. (2007). Using articulatory representations to detect segmental errors in nonnative pronunciation. *TASLP*, 16(1):8–22.

[266] Thomson, R. I. (2011). Computer assisted pronunciation training: Targeting second language vowel perception improves pronunciation. *Calico Journal*, 28(3):744–765.

[267] Timmons, N. G. and Rice, A. (2020). Approximating activation functions. *arXiv preprint arXiv:2001.06370*.

[268] Tóth, L. (2015). Phone recognition with hierarchical convolutional deep maxout networks. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):1–13.

[269] University, C. M. (1998). CMU Pronunciation Dictionary.

[270] van Dalen, R., Knill, K., and Gales, M. (2015a). Automatically Grading Learners' English Using a Gaussian Process. In *Proceedings of Workshop on Speech and Language Technology for Education (SLaTE)*.

[271] van Dalen, R. C., Knill, K. M., Tsiakoulis, P., and Gales, M. J. F. (2015b). Improving multiple-crowd-sourced transcriptions using a speech recogniser. In *ICASSP*, pages 4709–4713.

[272] van Dalen, R. C., Knill, K. M., Tsiakoulis, P., and Gales, M. J. F. (2015c). Improving multiple-crowd-sourced transcriptions using a speech recogniser. In *Proc. of ICASSSP*.

[273] Van Doremalen, J., Cucchiarini, C., and Strik, H. (2009). Automatic detection of vowel pronunciation errors using multiple information sources. In *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*, pages 580–585. IEEE.

[274] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.

[275] Walsh, P. (2018). Analysis of pronunciation errors and oral reading fluency in a read corpus of spanish learners of english as a foreign language. *Universidad CEU San Pablo*.

[276] Wang, L., Wang, Y., and Gales, M. J. (2019). Non-native speaker verification for spoken language assessment. *arXiv preprint arXiv:1909.13695*.

[277] Wang, Y., Gales, M., Knill, K. M., Kyriakopoulos, K., Malinin, A., van Dalen, R. C., and Rashid, M. (2018a). Towards automatic assessment of spontaneous spoken english. *Speech Communication*, 104:47–56.

[278] Wang, Y., Wong, J. H. M., Gales, M., Knill, K. M., and Ragni, A. (2018b). Sequence teacher-student training of acoustic models for automatic free speaking language assessment. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 994–1000. IEEE.

[279] Wei, L., Dong, W., Lin, B., and Zhang, J. (2019). Multi-task based mispronunciation detection of children speech using multi-lingual information. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1791–1794.

[280] Wei, S., Hu, G., Hu, Y., and Wang, R.-H. (2009). A new method for mispronunciation detection using support vector machine based on pronunciation space models. *Speech Communication*, 51(10):896–905.

[281] Weide, R. (1998). CMU pronunciation dictionary, rel. 0.6.

[282] Weigelt, L. F., Sadoff, S. J., and Miller, J. D. (1990). Plosive/fricative distinction: The voiceless case. *The Journal of the Acoustical Society of America*, 87(6):2729–2737.

[283] Wells, J. C. (1982). *Accents of English*, volume 1. Cambridge University Press.

[284] Wells, J. C. (1995). Computer-coding the IPA: a proposed extension of SAMPA. *Revised draft*, 4(28):1995.

[285] Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

[286] Wessel, F., Schluter, R., Macherey, K., and Ney, H. (2001). Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on speech and audio processing*, 9(3):288–298.

[287] Wester, F., Gilbers, D., and Lowie, W. (2007). Substitution of dental fricatives in english by dutch l2 speakers. *Language Sciences*, 29(2-3):477–491.

[288] Williamson, D. M. (2009). A framework for implementing automated scoring. In *Annual Meeting of the American Educational Research Association and the National Council on Measurement in Education, San Diego, CA*. Citeseer.

[289] Witt, S. M. (2012). Automatic error detection in pronunciation training: Where we are and where we need to go. *Proc. IS ADEPT*, 6.

[290] Witt, S. M. and Young, S. J. (2000). Phone-level pronunciation scoring and assessment for interactive language learning. *SPEECHCOM*, 30(2-3):95–108.

[291] Wong, J. H. and Gales, M. (2016). Sequence student-teacher training of deep neural networks. *Speech Communication*.

[292] Wu, Y. and He, K. (2018). Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19.

[293] Yan, B.-C., Wu, M.-C., Hung, H.-T., and Chen, B. (2020). An end-to-end mispronunciation detection system for l2 english speech leveraging novel anti-phone modeling. *arXiv preprint arXiv:2005.11950*.

[294] You, K., Long, M., Wang, J., and Jordan, M. I. (2019). How does learning rate decay help modern neural networks? *arXiv*.

[295] Young, S. (2008). Hmms and related speech recognition technologies. In *Springer Handbook of Speech Processing*, pages 539–558. Springer.

[296] Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2015). *The* HTK *book (for* HTK *version 3.5)*. University of Cambridge.

[297] Zhang, L., Zhao, Z., Ma, C., Shan, L., Sun, H., Jiang, L., Deng, S., and Gao, C. (2020). End-to-end automatic pronunciation error detection based on improved hybrid CTC/Attention architecture. *Sensors*, 20(7):1809.

[298] Zhao, G., Sonsaat, S., Silpachai, A. O., Lucic, I., Chukharev-Khudilaynen, E., Levis, J., and Gutierrez-Osuna, R. (2018). L2-arctic: A non-native english speech corpus. *Perception Sensing Instrumentation Lab*.

Appendix A

Speech Feature Extraction

Before speech can be processed for the purposes of acoustic modelling or proficiency assessment (see §2.2), the raw recorded audio must be converted into a compact and informative format which preserves the same salient information that the human auditory system uses to convert the sound of speech into meaning.

The most common way of achieving this is to divide the audio into 10-25ms segments called *frames*, over the length of each of which speech is be assumed to be stationary. A vector of features (or observation) $\boldsymbol{o}_t$ is extracted to represent each frame t. The frame-series $\boldsymbol{o}_{1:T}$ over all frames of the utterance then becomes the input for further stages of processing. The sections below discuss the four feature extraction methods referenced in this thesis, Filterbank features (§A.1), Mel Frequency Cepstral Coefficients (MFCCs) (§A.2), Perceptual Linear Prediction (PLP) features (§A.3), and bottleneck (BN) features (§A.4).

An alternative method used in recent approaches involves extracting features from the spectrum using convolutional or other neural layers [208, 209, 91] which can be trained in an end-to-end fashion with the rest of the system. These methods are not used in this thesis due to the computational load they would add to the training and evaluation of systems, which was deemed not to be matched by expected gains in performance. However, these methods are discussed as part of future work in Chapter 8.

A.1 Filterbank Features

Under the source-filter model [181], the time domain audio signal representing recorded speech $y(t)$ is treated as the convolution of an *excitation signal* $x(t)$, resulting from vocal cord vibrations in voiced speech and/or turbulence caused by forcing air through constrictions during fricative production, and a *vocal tract signal* $h(t)$, representing the transfer function of

the human vocal tract, itself a product of the position of the tongue and lips when the sound was made (Fig. A.1).

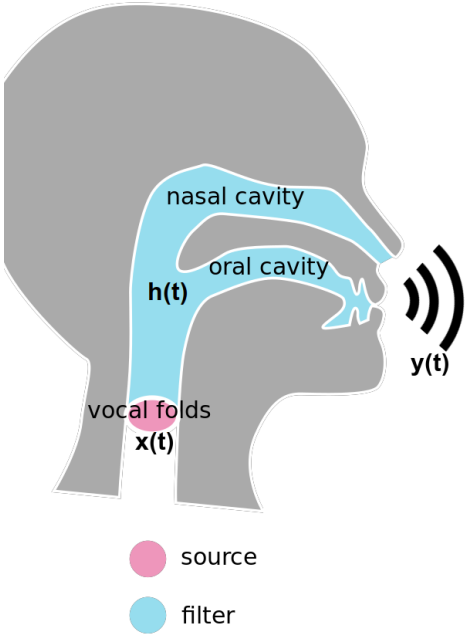


Fig. A.1 Illustration of the source filter model. Adapted from [76]

The relationship can be expressed as:

$$y(t) = x(t) * h(t) \tag{A.1}$$

The two signals clearly carry different and equally important information about the nature of the sound produced and allowing them to be represented separately is key to fully capturing the manner in which the phones constituting speech are rendered.

To separate the convolved signals, we take Fourier transforms to convert convolution to multiplication:

$$Y(\omega) = X(\omega)H(\omega) \tag{A.2}$$

take squares of both sides to obtain the power spectra:

$$|Y(\omega)|^2 = |X(\omega)|^2|H(\omega)|^2 \tag{A.3}$$

followed by logs to convert multiplication to addition:

$$2\log(|Y(\omega)|) = 2\log(|X(\omega)|) + 2\log(|H(\omega)|) \tag{A.4}$$

Inverse Fourier transforms are then taken to obtain the *cepstrum* $c(t)$:

$$c(t) = \mathscr{F}^{-1}\{2\log(|Y(\omega)|)\} = \mathscr{F}^{-1}\{2\log(|X(\omega)|) + 2\log(|H(\omega)|)\} \tag{A.5}$$

Unlike $y(t)$, $c(t)$ now has the property that the effects of $x(t)$ and $h(t)$ are linearly separable in the frequency domain. As the excitation and vocal tract signals are generally in different frequency ranges, this means they can be separated by filtering. In practice $c(t)$ is obtained from a regularly sampled $y(t)$ by applying discrete Fourier and Inverse Fourier transforms.

Filtering is usually done on the *mel-scale* (Eq. A.6), which is a scale of pitches designed to be perceived by listeners as equal in distance to each other. It is representative of the tonotopic properties of the human auditory system, making it ideal for extracting features to characterise properties of speech likely to be salient to humans. [254]

$$Mel(f) = 2595\log_{10}(1 + \frac{f}{700}) \tag{A.6}$$

Filtering is performed using N triangular overlapping windows spaced according to the mel-scale along the frequency axis. The filters are applied to coefficients of the FFT of $c(t)$ such that each filter yields a single coefficient m_n representing the amount of the energy of $c(t)$ at that part of the mel-scale. The logs of these parameters:

$$l_n = \log m_n \tag{A.7}$$

are taken, such that the vector $l_{1:N}$ serves as the filterbank feature representation of the frame in question. [296]

A.2 Mel Frequency Cepstral Coefficients

Having obtained N log filterbank amplitudes l_n as described in the previous section, it is further useful to decorrelate the coefficients, so that diagonal Gaussian models can be better used to represent them (which greatly saves on computational and storage cost). This can be achieved by taking the discrete cosine transform (DCT):

$$c_i = \sqrt{\frac{2}{N-1}} \sum_{j=0}^{N} m_j cos\left(\frac{\pi i}{N}(j - \frac{1}{2})\right) \quad i = 0...N-1 \tag{A.8}$$

with the resultant values known as mel-frequency cepstral coefficients (MFCCs).

In the case of 16kHz speech dealt with throughout this thesis, it is typical to calculate $N = 12$ such coefficients (MFCC12) and accompany them with a metric of energy (MFCC13). A common variation is to further append the first and second order frame-to-frame differences of each of these 13 features to capture the continuity of speech, resulting in 39 coefficients (MFCC39) in total for each frame. [296]

A.3 Perceptual Linear Prediction Coefficients

Perceptual Linear Prediction (PLP) features were introduced by Hermansky [110] and are based on modeling the way the human auditory system processes sound during perception, rather than the way sound is produced by the vocal system. After windowing, the Fourier Transform of the signal is taken the power spectrum obtained as above:

$$P(\omega) = |Y(\omega)|^2 = |\mathscr{F}_{\omega}(y(t))|^2 \tag{A.9}$$

This spectrum is then warped $\omega \rightarrow \Omega$ onto the Bark scale, a scale of perceptually equidistant pitches similar to the mel scale:

$$\Omega(\omega) = 6\ln(\omega/1200\pi + [(\omega/1200\pi)^2 + 1]^{0.5}) \tag{A.10}$$

It is then convolved with a masking curve $\Psi(\Omega)$ simulating the critical-band bandwidth of the human auditory system:

$$\Theta(\Omega(\omega)) = P(\Omega(\omega)) * \Psi(\Omega(\omega)) \tag{A.11}$$

The result is an array representing the amount of energy stimulating each section along the basilar membrane. This is then multiplied with an equal loudness curve $E(\Omega)$, representing the auditory system's differing sensitivity to energy at different frequencies:

$$\Xi(\Omega(\omega)) = E(\Omega(\omega))\Theta(\Omega(\omega)) = E(\Omega(\omega))(P(\Omega(\omega)) * \Psi(\Omega(\omega))) \tag{A.12}$$

and finally cube rooted, to represent the compression performed by the 'power-law of hearing' (the perceived loudness of sound is cubically compressed relative to its actual intensity):

$$\Phi(\Omega) = \sqrt[3]{\Xi(\Omega(\omega))} \tag{A.13}$$

$$\Phi(\Omega) = \sqrt[3]{E(\Omega(\omega))(|\mathscr{F}_{\Omega(\omega)}y(t))|^2 * \Psi(\Omega(\omega)))} \tag{A.14}$$

The inverse Fourier transform of this spectrum is then obtained, deriving the parameters of an all-pole filter that could have produced it:

$$\phi(t) = \mathscr{F}^{-1}\sqrt[3]{E(\Omega(\omega))(|\mathscr{F}_{\Omega(\omega)}y(t))|^2 * \Psi(\Omega(\omega)))} \tag{A.15}$$

These parameters are the PLP features and represent the most salient information about the audio, as it would have been extracted by the human auditory system. Where implemented in this thesis, 13 parameters are extracted (PLP13) to match the number of MFCC features. To capture continuity of speech, first and second deltas of each feature can also be appended to yield a length-39 feature vector (PLP39).

A.4 Bottleneck features

Bottleneck features are a way of learning to compress a high dimensionality feature representation (e.g. filterbank features) to a compact feature representation tuned to be optimally representative of the information necessary for a certain task.

A neural network (§4.2.1) with a much small number of units in one of its layers (bottleneck layer), generally its final or penultimate hidden layer, is trained on a relevant task e.g. to predict the state that each frame belongs to in a force aligned manually transcribed dataset. During training, the layers up to and including the bottleneck layer learn to map the input features to a representation that will best help the final layers complete the task. Once training is complete, the output layers are discarded and the trained network, with the bottleneck layer as its new output layer, used as a feature extractor to transform input features for any unseen frame into bottleneck features. [296]

Appendix B

HMM Acoustic Modelling

An acoustic model $p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}, \mathscr{M})$ models the production of acoustic observations $\boldsymbol{o}_{1:T}$ given the phones $\phi_{1:M}$ being spoken by a speaker (see §2.2). The most common form of acoustic model is the Hidden Markov Model (HMM).

An HMM $\mathscr{H} = \{N, \{a_{ij}\}, \{b_j(.)\}\}$ (Fig. B.1) is an N-state finite state machine which, at each frame t, occupies a state $s_t = i$ and, unless $i \in \{1, N\}$ (non-emitting states), generates an acoustic feature vector $\boldsymbol{o}_t$ (see Appendix A) with probability:

$$p(\boldsymbol{o}_t|s_t = i, \mathscr{H}) = b_i(\boldsymbol{o}_t) \tag{B.1}$$

At the next frame $t+1$, the system then moves to a state j with probability given i:

$$P(s_{t+1} = j|s_t = i, \mathscr{H}) = a_{ij} \tag{B.2}$$

There's usually a non-zero a_{ii}, meaning each state can emit multiple observations.

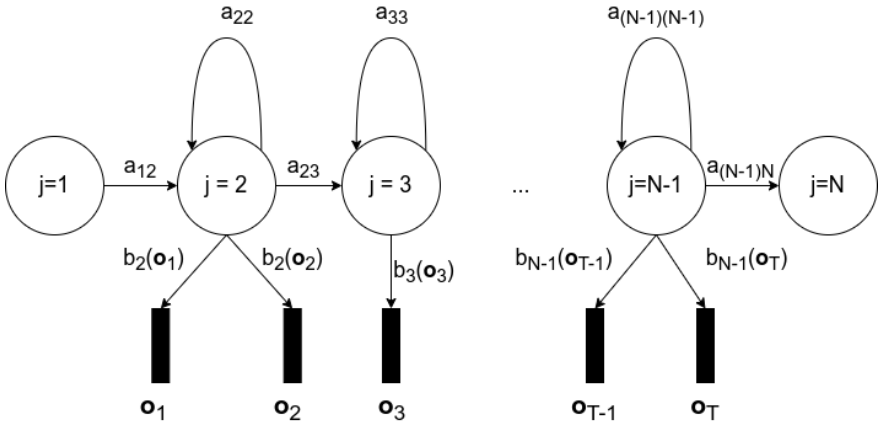


Fig. B.1 Illustration of a Hidden Markov Model (HMM)

In a GMM-HMM ASR system [221, 296], the output observation probabilities $b_j(\boldsymbol{o}_t)$ are generated by Gaussian Mixture Models (GMMs):

$$b_i(\boldsymbol{o}_t) = \sum_{m=1}^{M} c_{im}\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}) \tag{B.3}$$

where the mean $\boldsymbol{\mu}_{im}$, covariance $\boldsymbol{\Sigma}_{im}$ and weight c_{im} for each of M components m are the parameters to be learnt for each state i.

To allow more complex non-linear relationships to be modelled, stacked-hybrid DNN-HMMs [33, 188] instead generate $b_j(\boldsymbol{o}_t)$ using a DNN. Specifically, a DNN classifier is trained to predict the occupied state $s_t = i$ given observations $\boldsymbol{o}_t$:

$$\boldsymbol{y}_t = f_{DNN}(\boldsymbol{o}_t; \boldsymbol{\lambda}) \tag{B.4}$$

where $\boldsymbol{\lambda}$ represents the parameters of the DNN and $\boldsymbol{y}_t$ is a vector such that:

$$y_{ti} = P(s_t = i|\boldsymbol{o}_t) \tag{B.5}$$

where y_{it} is the ith element of $\boldsymbol{y}_t$. Applying Bayes' rule:

$$b_i(\boldsymbol{o}_t) = p(\boldsymbol{o}_t|s_t = i) = \frac{P(s_t = i|\boldsymbol{o}_t)p(\boldsymbol{o}_t)}{P(s_t = i)} \tag{B.6}$$

In practice, $b_i(\boldsymbol{o}_t)$ can be estimated by directly substituting $P(s_t = i|\boldsymbol{o}_t)$ or by dividing $P(s_t = i|\boldsymbol{o}_t)$ by a prior on the state $P(s_t = i)$ [33].

An alternative approach to combining DNNs with HMMs is the tandem methodology [111, 210], in which the values of the last layer of the DNN state classifier are orthogonalised and fed as features to the GMM model from above.

Given the parameters of b_j and a_{ij}, the joint likelihood of any T-frame long sequence of states $s_{1:T}$ and observations $\boldsymbol{o}_{1:T}$ is given by:

$$p(\boldsymbol{o}_{1:T}, s_{1:T}|\mathscr{H}) = P(s_{1:T}|\mathscr{H})p(\boldsymbol{o}_{1:T}|s_{1:T}, \mathscr{H}) = P(s_{1:T}|\mathscr{H})\left(\prod_{t=1}^{T} p(\boldsymbol{o}_t|s_{1:T}, \mathscr{H})\right) \tag{B.7}$$

$$p(\boldsymbol{o}_{1:T}, s_{1:T}|\mathscr{H}) = \left(P(s_1|\mathscr{H})\prod_{t=1}^{T} P(s_{t+1}|s_t, \mathscr{H})\right)\left(\prod_{t=1}^{T} p(\boldsymbol{o}_t|s_t, \mathscr{H})\right) \tag{B.8}$$

$$p(\boldsymbol{o}_{1:T}, s_{1:T}|\mathscr{H}) = a_{0,1}\prod_{t=1}^{T} a_{s_t,s_{t+1}} b_{s_t}(\boldsymbol{o}_t) \tag{B.9}$$

Individual HMMs are usually constructed for tri-phones $\phi_{t-1:t+1}$, such that an HMM $\mathscr{H}^{(\phi_{1:M})}$ can be compiled for any phone sequence $\phi_{1:M}$ by linking the states of the constituent tri-phone models. The likelihood of the observations given a known $\phi_{1:M}$ is thus obtained by summing Eq. B.9 over all possible state sequences $s_{1:T}|\phi_{1:M}$ consistent with $\phi_{1:M}$:

$$p(\boldsymbol{o}_{1:T}|\phi_{1:M},\mathscr{H}^{(\phi_{1:M})}) = \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T},s_{1:T}|\mathscr{H}^{(\phi_{1:M})}) \tag{B.10}$$

If it is instead the word sequence $w_{1:I}$ that is known, the likelihood ob the observations is obtained by summing Eq. B.10 over all possible (assumed equiprobable) phone sequences $\phi_{1:M}$ corresponding to $w_{1:I}$ given a pronunciation dictionary $\mathscr{D}$ (see §2.2):

$$p(\boldsymbol{o}_{1:T}|w_{1:I},\mathscr{M}) = \sum_{\phi_{1:M}\in\mathscr{D}_{w_{1:I}}} \sum_{s_{1:T}|\phi_{1:M}} a_{0,1}^{(\phi_{1:M})} p(\boldsymbol{o}_{1:T},s_{1:T}|\mathscr{H}^{(\phi_{1:M})}) b_{s_t}^{(\phi_{1:M})}(\boldsymbol{o}_t) \tag{B.11}$$

where $\mathscr{M}$ consists of HMMs $\mathscr{H}^{(\phi_{1:3})}$ for each possible tri-phone $\phi_{1:3}$ from which $\mathscr{H}^{(\phi_{1:M})})$ is constructed in each case.

Acoustic model training consists of optimising the parameters of the tri-phone HMMs to maximise the likelihood of observations $\boldsymbol{o}_{1:T'}^{(train)}$ in the acoustic model training set given their labelled word sequences $w_{1:I}^{(train)}$ [181, 296]:

$$\hat{\mathscr{M}} = \arg\max_{\mathscr{M}} \sum_{\phi_{1:M}\in\mathscr{D}_{w_{1:I}^{(train)}}} p(\boldsymbol{o}_{1:T}^{(train)}|w_{1:I}^{(train)},\mathscr{M}) \tag{B.12}$$

Given a trained acoustic model $\hat{\mathscr{M}}$, forced alignment consists of finding the state sequence $\hat{s}_{1:T}$ compatible with a known word sequence $w_{1:I}$ that maximises the likelihood of the observations $\boldsymbol{o}_{1:T}$ being aligned:

$$\hat{s}_{1:T} = \arg\max_{s_{1:T}|\phi_{1:M}\ \forall\ \phi_{1:M}\in\mathscr{D}_{w_{1:I}}} p(\boldsymbol{o}_{1:T},s_{1:T}|\mathscr{H}^{(\phi_{1:M})}) \tag{B.13}$$

with $\mathscr{H}^{(\phi_{1:M})}$ derived by linking the states of the tri-phone models in $\hat{\mathscr{M}}$ as above.

In the experiments reported in this thesis, both these optimisations are conducted using the Viterbi algorithm with the HTK framework [296]. During forced alignment, rather than just returning the 1-best state sequence $\hat{s}_{1:T}$, there is also the option of generating a lattice, each path through which represents one possible $s_{1:T}$.

Appendix C

Automatic Speech Recognition

As discussed in §2.2, Automatic Speech Recognition (ASR) aims to determine the most likely word sequence $\hat{w}_{1:I}$ given frame-wise acoustic observations $\boldsymbol{o}_{1:T}$:

$$\hat{w}_{1:I} = \arg\max_{w_{1:I}} \left\{ P(w_{1:I}) \sum_{\phi_{1:M} \in \mathscr{D}_{w_{1:I}}} P(\phi_{1:M}|w_{1:I}) \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \right\} \tag{C.1}$$

or, assuming all pronunciations of each word are equiprobable:

$$\hat{w}_{1:I} = \arg\max_{w_{1:I}} \left\{ P(w_{1:I}) \sum_{\phi_{1:M} \in \mathscr{D}_{w_{1:I}}} \sum_{s_{1:T}|\phi_{1:M}} p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M}) \right\} \tag{C.2}$$

where $P(w_{1:I})$ is the language model, $\mathscr{D}_{w_{1:I}}$ is the pronunciation dictionary and $p(\boldsymbol{o}_{1:T}, s_{1:T}|\phi_{1:M})$ is the acoustic model.

Different methods of extracting $\boldsymbol{o}_{1:T}$ from an audio signal were reviewed in Appendix A, while HMM-based acoustic models were discussed in Appendix B. This section describes the three ASR systems used in this thesis, specifically their acoustic models, their language models, their pronunciation dictionaries and the method used for extracting the observations they are trained on. The systems were developed by other members of the CUED speech group as part of the ALTA project, as described in Knill et al. [147] and Wang et al. [277] and summarised below.

All three systems are trained on datasets taken from the BULATS corpus (§6.1), non-overlapping with the corpora used to train and evaluate systems, with merged crowd-sourced transcriptions as described in van Dalen et al. [272]. Data from the AMI meeting corpus [40] is also used where specified. Phonetic systems use the COMBILEX British English

pronunciation dictionary which records pronunciations using the 47-phone alphabet from Appendix D. Graphemic systems instead use a lexicon representing each word by its constituent letters, such that the alphabet consists of the 26 letters of the alphabet plus two additional labels to deal with hesitations, as set out in Knill et al. [147].

Language modelling is performed using a Kneser-Ney (K-N) [146] trigram language model, trained on a 186,000-word transcribed dataset from BULATS and interpolated with a general English language model trained on a large native corpus. For the SELL corpus (§7.3), the language model is fine-tuned on a 17,000-word held-out dataset from SELL.

The first acoustic model (GH) is a GMM-HMM system, using PLP features as inputs (§A.3). It is trained using a transcribed non-native speaker dataset consisting of 8485 speakers, mostly from India (69.8%), with 28 different L1s, speaking for 334 hours in total. Two versions are implemented, one phonetic (*GH-ph*) and one graphemic (*GH-gr*). This system is used in most forced alignment tasks in this thesis.

The second system (DH) is based on a hybrid DNN-HMM approach, described in Knill et al. [147], trained on 75 hours of non-native English speech from 1075 Gujarati speakers. A GMM-HMM system is first trained on the data and used to force align data from AMI, which is used to train a $720 \times 1000^4 \times 39 \times 1000 \times 6000$ bottleneck DNN to predict HMM states from an input consisting of 9 consecutive frames of 40-dimensional filterbank features (§A.1) with their deltas appended. The 39-dimensional bottleneck features obtained from this extractor are transformed using a global semi-tied covariance matrix as described in Gales [89] and appended to PLP features projected using the HLDA method described in Gales and Young [88], as well as their first, second, and third order deltas. Cepstral mean normalisation (CMN) and cepstral variance normalisation (CVN) are applied at the speaker level to yield a 78-dimension frame-level feature vector $\boldsymbol{o}_t$. Groups of 9 such consecutive feature vectors are fed as inputs to a $702 \times 1000^5 \times 6000$ DNN state classifier, which is pre-trained on context-dependent targets from GH and fine-tuned using the frame-level cross-entropy criterion. This DNN is then used as part of the final hybrid DNN-HMM system, which is trained using the minimum phone error (MPE) criterion [215]. Like GH, DH also comes in a phonetic (*DH-ph*) and a graphemic (*DH-gr*) version.

The final acoustic model (*TD-gr*), described in Lu et al. [174], is based on factorised time-delay neural networks (TDNN-Fs) [213]. It uses a graphemic lexicon and is trained on a 505 hour dataset of 12375 non-native speakers across 75 L1s, automatically transcribed using DH-gr [278]. The system is trained in a teacher-student fashion as set out in Wong et al. [291]. 40-dimensional filterbank features, concatenated with 100-dimensional i-vectors are used as the input to three teacher TDNN-F models trained using lattice-free maximum mutual information (LF-MMI) [214], as an ensemble, using different random initialisations. A single

student TDNN-F model with the same input is then trained to minimise the KL divergence between its sequence-level posteriors and those from the combined teacher ensemble.

The details of the systems described above as well as their performance on the BLT_EVL_M dataset (from §6.1) used to evaluate most of the graders and error detector in this thesis, are summarised in Table C.1.

Model	Alphabet	Architecture	Speakers	L1s	WER (%)
GH-ph	phonetic (47-ph.)	GMM-HMM	8485 (334 hrs)	28 L1s	49.0
GH-gr	graphemic	GMM-HMM	8485 (334 hrs)	28 L1s	47.0
DH-ph	phonetic (47-ph.)	hybrid DNN-HMM	1075 (108 hrs)	Gujarati	47.5
DH-gr	graphemic	hybrid DNN-HMM	1075 (108 hrs)	Gujarati	46.1
TD-gr	graphemic	TDNN-F	12375 (505 hs)	75 L1s	21.3

Table C.1 Description of acoustic models used in this thesis, the data they are trained on and their word error rates (WER), when used together with a K-N trigram language model, evaluated on BLT_EVL_M (from §6.1). The COMBILEX pronunciation dictionary is used throughout.

Appendix D

Phonetic alphabets

When analysing the pronunciation of words as series of phones (e.g. when compiling a pronunciation dictionary) it is necessary to define a *phonetic alphabet* of all possible phone labels. A large phonetic alphabet corresponds to a narrow phonetic transcription, which can capture more detail of the way each sound is pronounced by the speaker but is harder to reliably annotate and detect. A small phonetic alphabet corresponds to a broad phonetic transcription, which is coarser and less descriptive but easier to obtain (see §2.2). Phonetic alphabets also vary depending on the variety of English they are based on, as some varieties contain sounds, or distinctions between sounds that others do not.

All experiments and results in this thesis employ one of two phonetic alphabets. The first is the one adopted by the proprietary COMBILEX-RP canonical pronunciation dictionary [81] and is used, along with the dictionary, for non-native speech where British English was used as the annotation standard. The second, used when American English was used as the annotation standard, is the one adopted by the openly available CMU pronunciation dictionary [281]. The former alphabet has has 47 distinct phones (20 vowels and 27 consonants) whlie the latter (CMU) has 39 distinct phones (15 vowels and 24 consonants). Both alphabets are displayed in Table D.1 on the following page. Each phone is represented by its International Phonetic Alphabet (IPA) symbol and by its corresponding ARPABET two-digit ASCII code [144]. The latter is mostly used in the text of this thesis.

Consonants					
Plosives			Fricatives		
Arpabet Transcription	IPA Symbol	Example	Arpabet	IPA	Example
b	b	**b**ad	v	v	**v**an
p	p	**p**en	f	f	**f**all
d	d	**d**id	dh	ð	**th**is
k	k	**c**at	th	*θ*	**th**in
t	t	**t**ea	sh	ʃ	**sh**oe
g	g	**g**et	s	s	**s**ee
Affricates			z	z	**z**oo
ch	tʃ	**ch**ain	zh	ʒ	vi**s**ion
jh	dʒ	**j**am	h	h	**h**at
Approximants			Nasals		
w	w	**w**et	m	m	**m**an
y	j	**y**es	em*	m̩	rhyth**m**
r	r	**r**ed	n	n	**n**ow
l	l	**l**eg	en*	n̩	butt**on**
el*	l̩	bott**le**	ng	ŋ	si**ng**
Vowels					
Monophthongs			ao	ʉ	s**aw**
aa	ɑ	f**a**ther	uh	ʊ	p**u**t
oh*	ɒ	g**o**t (RP)	Diphthongs		
ah	ʌ	c**u**p	aw	aʊ	n**ow**
ax*	ə	**a**bout	ay	aɪ	m**y**
uw	u	t**oo**	ua †	ʊə	f**ewer** (RP)
eh	*ɛ*	t**e**n	ea †	eə	h**air** (RP)
iy	i	s**ee**	ia †	ɪə	n**ear** (RP)
er	ɜ:	f**ur**	ow	oʊ	g**o**
ih	ɪ	s**i**t	oy	ʉɪ	b**oy**
ae	æ	c**a**t	ey	eɪ	s**ay**

Table D.1 Phones used in this project in Arpabet and IPA [219, 281, 283]. In CMU, phones with * are merged with phone above and those with † are merged with that to their left.

Appendix E

Baseline Rhythm Features

- mean and standard deviation of length of time between consecutive stressed syllables
- mean and standard deviation of length of time between consecutive unstressed syllables
- ratios of above two means and above two standard deviations
- %V: The proportion of time devoted to vocalic intervals in the sentence, disregarding word boundaries
- ΔV: The standard deviation of the duration of vocalic intervals
- ΔC: The standard deviation of the duration of consonantal intervals
- $\texttt{rPVI}_V = \frac{1}{m-1}\sum_{k=1}^{m-1}|d_k - d_{k+1}|$ where d_k is the duration of the kth vowel segment and m is the number of vowel segments.
- $\texttt{rPVI}_C = \frac{1}{m-1}\sum_{k=1}^{m-1}|d_k - d_{k+1}|$ where d_k is the duration of the kth intervocalic segment and m is the number of intervocalic segments.
- $\texttt{nPVI}_V = \frac{1}{m-1}\sum_{k=1}^{m-1}\frac{|d_k - d_{k+1}|}{(d_k + d_{k+1})/2}$ where d_k is the duration of the kth vowel segment and m is the number of vowel segments.
- $\texttt{nPVI}_C = \frac{1}{m-1}\sum_{k=1}^{m-1}\frac{|d_k - d_{k+1}|}{(d_k + d_{k+1})/2}$ where d_k is the duration of the kth intervocalic segment and m is the number of intervocalic segments.
- $\texttt{CCI}_V = \frac{1}{m-1}\sum_{k=1}^{m-1}\left|\frac{d_k}{n_k} - \frac{d_{k+1}}{n_{k+1}}\right|$ where d_k and n_k are the duration and number of vowel segments of the kth measurement and m is the number of vocalic measurements.
- $\texttt{CCI}_C = \frac{1}{m-1}\sum_{k=1}^{m-1}\left|\frac{d_k}{n_k} - \frac{d_{k+1}}{n_{k+1}}\right|$ where d_k and n_k are the duration and number of intervocalic segments of the kth measurement and m is the number of intervocalic measurements.

Appendix F

Intonation Annotation

The tones and break indices (ToBI) convention [16, 15, 17] provides a framework for annotating salient features of pitch variation that mark meaning in speech. The convention is used for human annotation based on the perceived intonation to the annotator, though they also generally correlate with distinct f_0 contours. The main labelled contours are described below then illustrated with examples in Figures F.1 to F.8.

- **Pitch stresses**: Mark words that are emphasized within phrases

  H* Peak in f_0 on stressed syllable of word - standard pitch stress - used to mark important words in phrase (see Fig. F.1)

  L* Trough in f_0 - Alternative pitch stress - used instead of H* particularly in yes-no questions to contrast with final H% (see Fig. F.2)

  L+H* H* stress on stressed syllable preceded by trough - Stronger than H* - used to stress earlier word more than other H* stressed words (see Fig. F.3)

  L*+H L* stress followed by peak - stronger version of L* (see Fig. F.4) - also used in calling and for contrastive emphasis

  !H Downstep - Peak that is lower than previous H* peak - marks all but first pitch stress in sentence with falling intonation (see Fig. F.7)

- **Intonational phrase boundaries:** Mark the end of intonational phrases

  L-L% Falling tone - Marks end of ordinary statement (see Fig. F.1)

  H-H% Sharp rise - Marks end of yes/no question (see Fig. F.2) or statement rendered in question-like uncertain or emphatic manner

  L-H% Weak rise - known as continuation rise - used to indicate that there are more phrases in the sentence (see Fig. F.6)

H-L% Level tone - used to mark transition between items in a list (see Fig F.8)

- **Intermediate phrase boundaries:** Mark the end of intermediate phrases within intonational phrases

  L- Trough before short pause - standard intermediate phrase ending (see Fig. F.7)

  H- Peak before short pause - alternative intermediate phrase ending - often for intermediate phrase of intonation phrase also ending in H-H% (see Fig. F.5).

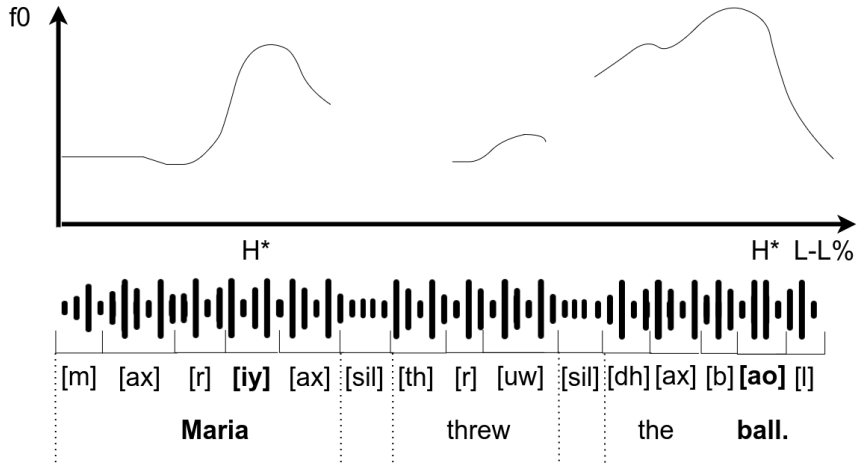


Fig. F.1 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for a simple rendering of the statement *Maria threw the ball*. Equal emphasis is placed on the words *Maria* and *ball* by pronouncing the stressed vowels of each with a higher pitch (H* pitch stress). Pitch drops at the end of *ball* (L-L%) to indicate the end of the statement.

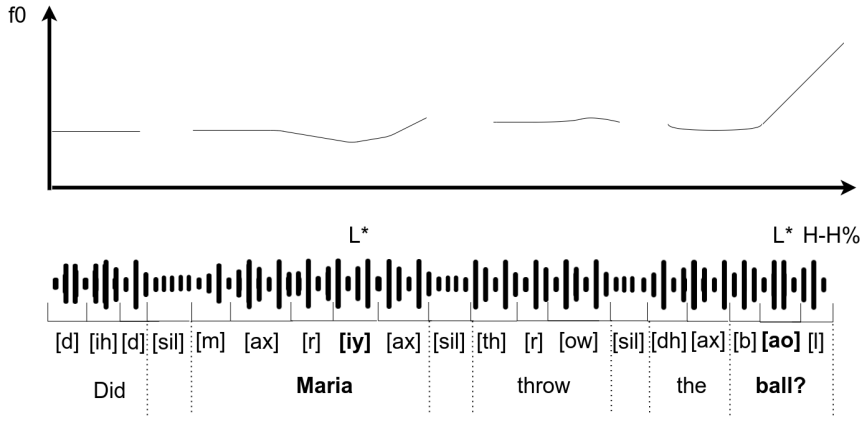


Fig. F.2 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for question *Did Maria throw the ball?*. Equal emphasis is placed on *Maria* and *ball* using lower pitch (L* pitch stress). These play the same role as H* in Fig. F.3 but are now low to contrast with the high pitch at the end. Pitch rises at the end of *ball* (H-H%) to indicate a yes/no question.

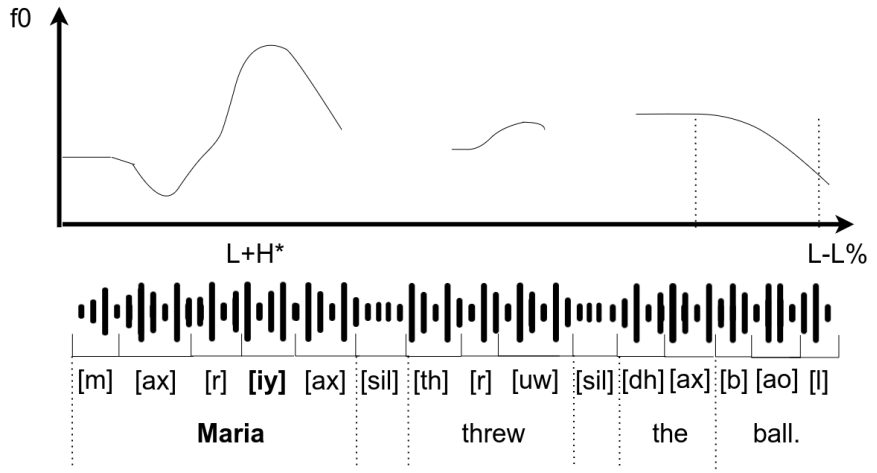


Fig. F.3 Illustration of words, phones, emphases, ToBI annotations and f_0 for a rendering of the statement *Maria threw the ball* with an emphasis on *Maria* (i.e. Maria is the one that threw the ball as opposed to someone else). Particular emphasis is placed on the word Maria by lowering the pitch before increasing it to form the pitch stress H*, forming a 'scoop' pitch stress L+H*. Pitch drops (L-L%) to indicate the end of the statement.

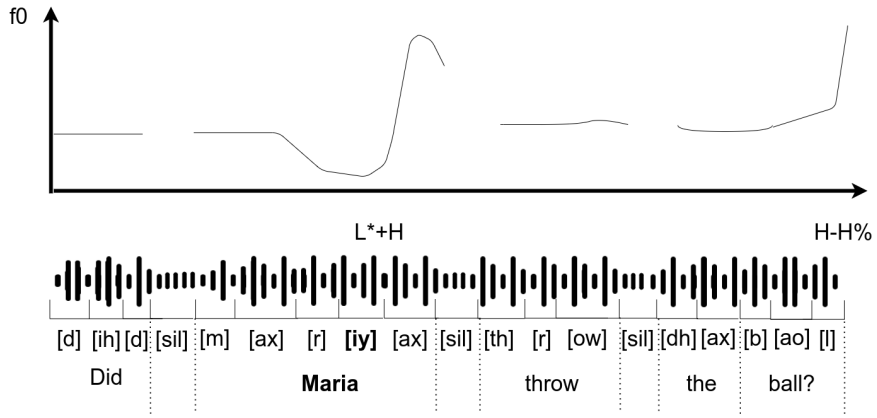


Fig. F.4 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for a rendering of the question *Did Maria throw the ball?* with an emphasis on the word *Maria* (i.e. asking whether Maria was the one that threw the ball as opposed to someone else). Particular emphasis is placed on the word Maria by sharply raising pitch immediately after having reduced it to form the pitch stress L* on the stress syllable [iy], thus the combined scoop stress L*+H. Pitch rises at the end of *ball* (H-H%) to indicate a yes/no question.

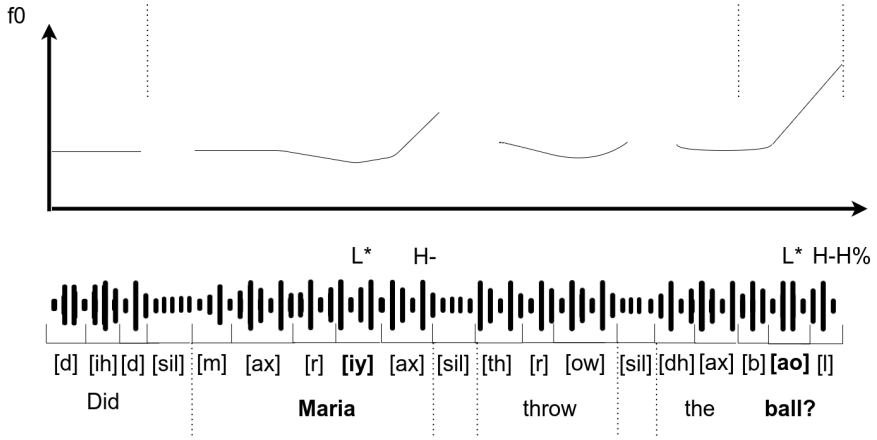


Fig. F.5 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for a rendering of the question *Did Maria throw the ball?*, such that *Did Maria* forms a separate intermediate intonational phrase to *throw the ball*? A small pitch rise after *Maria* separates the two intermediate phrases and signals that the first intermediate phrase is part of a question, such that the whole utterance sounds like *Did Maria*$^{(?)}$*throw the ball?*. Equal emphasis is still placed on the words *Maria* and *ball* by using lower pitch (L* pitch stress). The meaning is the same as in Fig. F.2 but the rendering is more spaced out. Pitch rises again at the end of *ball* (H-H%) to signal the end of the yes/no question.

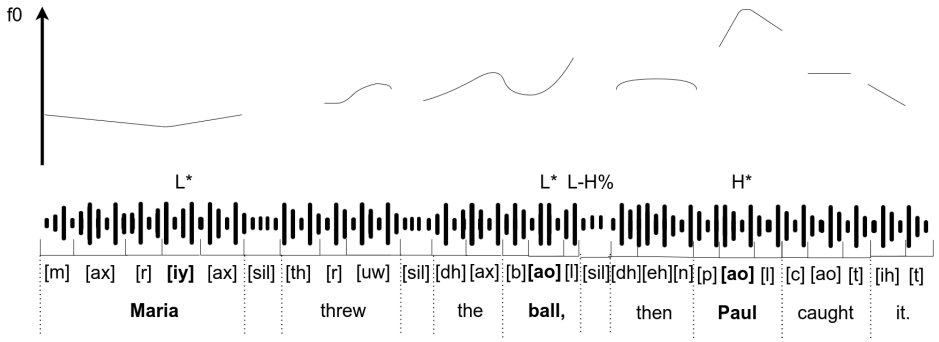


Fig. F.6 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for statement *Maria threw the ball, then Paul caught it* with emphasis on *Maria*, *ball*, and *Paul*. The clauses before and after the comma are separate intonational phrases, both in the form of statements. The end of the first is signalled by a drop L to indicate the end of a statement followed by a continuation rise H to signal that another related intonation phrase is coming, together making the boundary tone L-H%. The final phrase ends with a standard pitch drop L-L%. Emphases on *Maria* and *ball* are marked by low pitch stresses, contrasting the continuation rise, while the emphasis on *Paul* is marked by a high pitch stress as normal.

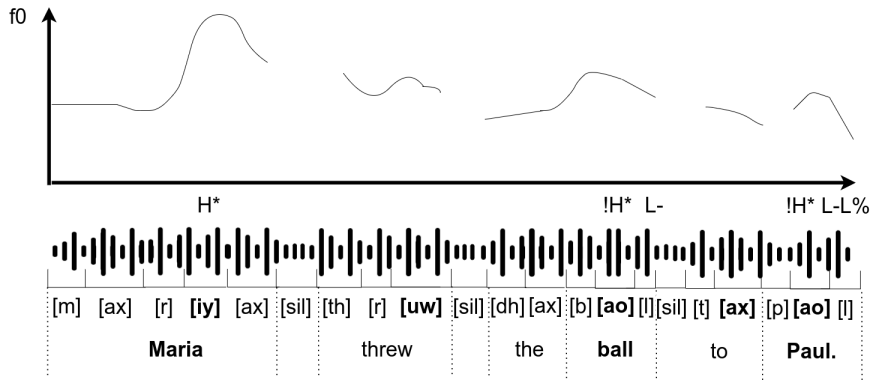


Fig. F.7 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for a rendering of the statement *Maria threw the ball to Paul*, with emphasis on *Maria*, *ball* and *Paul*, marked by high pitch stresses (H*), an intermediate phrase break (L-) before *to Paul*, marked by a small pitch drop after *ball*, and overall falling pitch (downstep) across the entire intonational phrase, signalled by each pitch stress !H* being lower than its predecessor. A final pitch drop at the end of *Paul* (L-L%) marks the end of the statement.

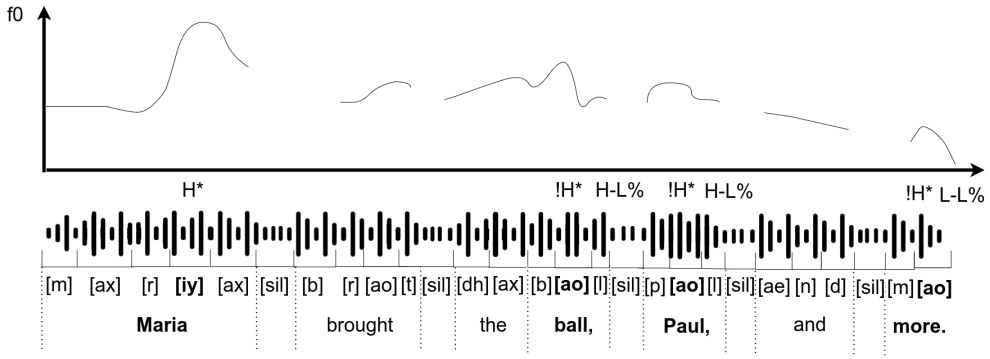


Fig. F.8 Illustration of words, phones, emphases, ToBI annotations and f_0 profile for the statement *Maria brought the ball, Paul, and more* as three intonational phrases. The ends of the first two intonational phrases are marked by level boundary tones L-H% after *ball* and *Paul*, signalling the progression to the next element of the list, while the end of the statement is marked with a standard pitch drop L-L%. Emphasis on *Maria*, *ball*, *Paul*, and *more* is marked by high pitch stresses (H*). Overall falling pitch (downstep) across the entire utterance is signalled by each pitch stress !H* being lower than its predecessor.

Appendix G

Equivalence of DCT-II to least-squares approximation

Consider a discrete time function y_t which we wish to approximate as a sum of K discrete time cosines:

$$\hat{y}_t = \sum_{k=0}^{K-1} x_k c_{k,t} \tag{G.1}$$

where:

$$c_{0,t} = \sqrt{\frac{1}{K}} \cos\left[\frac{\pi}{K}\left(t+\frac{1}{2}\right)k\right] \quad \forall\, t \in \mathbb{Z}^{\geq 0} \tag{G.2}$$

$$c_{k,t} = \sqrt{\frac{2}{K}} \cos\left[\frac{\pi}{K}\left(t+\frac{1}{2}\right)k\right] \quad \forall\, t \in \mathbb{Z}^{\geq 0},\ k \in \mathbb{Z}^{+} \tag{G.3}$$

such that:

$$\sum_{t=0}^{K-1} c_{k,t} c_{j,t} = \begin{cases} 1, & \text{for } k = j \\ 0, & \text{for } k \neq j \end{cases} \tag{G.4}$$

If y_t is sampled at times $t_1 \ldots t_N$ and the values placed in vector $\boldsymbol{y}$, we can express Equation G.1 as:

$$\hat{\boldsymbol{y}} = \boldsymbol{C}\boldsymbol{x} \tag{G.5}$$

where $\boldsymbol{x}$ is a length-K vector of cosine coefficients and $\boldsymbol{C}$ is an $N \times K$ matrix such that the element at the nth row and kth column is c_{k,t_n}.

The least-squares approximation $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ can be obtained by minimising the square distance between $\boldsymbol{y}$ and its approximation $\hat{\boldsymbol{y}}$ given $\boldsymbol{x}$:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} (\boldsymbol{y} - \hat{\boldsymbol{y}})^T (\boldsymbol{y} - \hat{\boldsymbol{y}}) = \arg\min_{\boldsymbol{x}} (\boldsymbol{y} - \boldsymbol{C}\boldsymbol{x})^T (\boldsymbol{y} - C\boldsymbol{x}) \tag{G.6}$$

$$\frac{\partial (\boldsymbol{y} - \boldsymbol{C}\boldsymbol{x})^T (\boldsymbol{y} - \boldsymbol{C}\boldsymbol{x})}{\partial \boldsymbol{x}} |_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = 0 \tag{G.7}$$

$$\frac{\partial}{\partial \boldsymbol{x}} \left(\boldsymbol{y}^T \boldsymbol{y} - 2\boldsymbol{x}^T \boldsymbol{C}^T \boldsymbol{y} + \boldsymbol{c}^T \boldsymbol{C}^T \boldsymbol{C}\boldsymbol{x} \right) |_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = 0 \tag{G.8}$$

$$-2\boldsymbol{C}^T \boldsymbol{y} + 2\boldsymbol{C}^T \boldsymbol{C}\hat{\boldsymbol{x}} = 0 \tag{G.9}$$

$$\hat{\boldsymbol{x}} = (\boldsymbol{C}^T \boldsymbol{C})^{-1} \boldsymbol{C}^T \boldsymbol{y} \tag{G.10}$$

If $t_1 \ldots t_N$ correspond to evenly spaced samples $0 \ldots K-1$ then, by Equation G.4, $C^T C = I$, so Equation G.10 becomes:

$$\hat{\boldsymbol{x}} = \boldsymbol{C}^T \boldsymbol{y} \tag{G.11}$$

or, equivalently:

$$\hat{c}_k = \sum_{t=0}^{K-1} x_{t_n} c_{k,t} \tag{G.12}$$

which is equivalent to the DCT-II.

Appendix H

ALTA Baseline Features

A set of features for various views, specifically tempo (rate of speech, silence and disfluency frequency and duration statistics), rhythm (word, syllable, phone and grapheme duration statistics), pronunciation (acoustic model confidence measures) and text (language model confidence scores and number of unique words), was developed for use with Gaussian Process and neural network graders by other members of the CUED group as part of the ALTA project, as described in Wang et al. [277] and van Dalen et al. [270].

The version used as a baseline for the experiments in Chapter 6 of this thesis was provided by Kate Knill and consists of the fluency and ASR confidence features listed in Table H.1 on the following page. The terms used in Table H.1 are further defined in Table H.2.

Item	Features
Long Silence Duration	mean standard deviation median mean absolute deviation
disfluencies	number all fraction all
Long Silences	number
Phone/Grapheme Duration	mean standard deviation median mean absolute deviation
Silence Duration	mean standard deviation median mean absolute deviation
Words	frequency mean duration
Time Used	fraction
Syllables	articulation rate
Disfluencies	number partial words
Hesitations	number fraction
Words	number number unique
Confidence score	per-section score mean score
Grader-dependent LM score	relative across CEFR grades normalised by grader-independent relative normalised by grader-independent

Table H.1 ALTA baseline grader features extracted from audio and ASR output

Item	Definition
Silence	all segments recognised as silence
Long Silence	within word silence $> 0.25s$ in length does not include start/end silence
Phone/Grapheme	all consonant/vowel segments not recognised as silence
Word	all word segments not recognised as silence
Disfluencies	partial words, hesitations, within word silences $> 0.25s$, non-silence words repeated twice, words repeated with short pause between
Word Frequency	speaking rate in terms of word frequency (number of words / speech duration) across all utterances - including speech silence in duration
Syllables Articulation Rate	syllable rate (number of vowels / speech duration) across all utterances not including speech silence in duration
Time Used	ratio of mean time from start of the 1st non-silence segment to the end of the last, over total duration of the utterance
Relative across CEFR grades	relative grader-dependent language model scores across 5 grades (e.g. GDLM A1 - GDLM A2)
Normalised by grader-independent	actual grader-dependent language model score normalised by the grader-independent language model (e.g. GDLM A1/GILM)
Relative normalised	relative as above normalised by grader-independent as above

Table H.2 Definition of terms in Table H.1

Appendix I

Phone Distance Features and L1

The relationship between phone distance features and the speaker's L1 is investigated by training and evaluating an 11-way L1 DNN classifier, with two 30-unit hidden layers. Results, broken down by the language family of each L1 are displayed in Table I.1. The overall performance is significantly better than chance and misclassified speakers are more likely to be misclassified into an L1 of the same language family than a different one.

	Spanish	French	Portuguese	Italian	Non-romance
Spanish	97.7	0.5	0.0	0.0	1.8
French	16.5	43.5	0.0	0.0	40
Portuguese	21.8	24.4	29.1	0.0	24.7
Italian	36.6	26.8	0.0	2.4	34.2

	Gujarati	Hindi	Bengali	Non-Indo-Aryan
Gujarati	74.3	10.9	1.7	13.1
Hindi	11.6	62.9	0.0	25.5
Bengali	10.5	55.9	20.3	13.3
Marathi	3.0	74.6	0.0	22.4

	Tamil	Telugu	Malayalam	Kannada	Non-Dravidian
Tamil	76.7	8.5	0.0	0.0	14.8
Telugu	37.6	27.5	0.0	0.0	34.9
Malayalam	49.5	23.4	12.4	0.0	14.7
Kannada	24.4	19.1	0.0	0.0	56.5

Table I.1 Percentage of speakers of Romance (top), Indo-Aryan (middle) and Dravidian (bottom) L1s in BLT_EVL_M2 classified as other languages in same group by phone distance DNN trained on BLT_GRD_M2 from MFCC-13 after decode and alignment with GH-ph

A similar methodology is now employed to attempt to classify the speaker's country of origin. The results in Table I.2 demonstrate that the features can indeed be used to detect Spanish speakers' countries of origin.

	Spain	Colombia	Mexico	Overall
% Correct	71.5	45.5	97.5	85.5

Table I.2 Detection rate, by country of origin, on Spanish speakers in BLT_EVL_M2, of phone distance DNN 3-way country classifier, trained on Spanish speakers in BLT_GRD_M2

This suggests that phone distances are able to capture phonological differences in the way speakers of the same L1 with different regional dialects pronounce the phones of English. Taken together with previous results, phone distance features have been shown to be predictive of grade, L1, L1 language family and country of origin, consistent with them being an informative characterisation of the speaker's accent.

Appendix J

Experiments on grader architecture

This appendix presents the results of experiments exploring the effect of various architectural choices on the performance of the graders from §5.2. Varying the features extracted in the speech processing stage is investigated in §J.1, while the effect of the alphabet used for ASR and alignment is explored in §J.2. Different sequence modelling layers are compared in §J.3, with types of attention examined in §J.4. Finally the effects of batch normalisation and learning rate schedule are investigated in §J.5 and §J.6 respectively.

J.1 Speech features

The two-stage phone distance grader from §5.2.1 is replicated with three different types of observation vector $\boldsymbol{o}_t$ (see Appendix A), namely MFCCs without deltas (MFCC13), PLPs without deltas (PLP13) and PLPs with first and second deltas (PLP39). Results are shown in Table J.1. The grader performs very similarly in all three cases, suggesting robustness to the choice of feature type. Similar results were also observed with all three end-to-end graders.

Features	PCC	MSE	MAE	%<0.5	%<1.0
MFCC13	0.785	0.556	0.552	**59.4**	**86.6**
	±.03	±.073	±.026	±2.6	±1.2
PLP13	**0.786**	**0.552**	0.56	57.4	85.2
	±.041	±.11	±.029	±2.4	±.75
PLP39	0.781	0.566	**0.536**	58.3	86.5
	±.081	±.2	±.042	±3.2	±1.1

Table J.1 Performance of DNN graders with phone distances from MFCC-13, PLP-13 and PLP-39 observations after ASR by TD-gr and alignment with the GH-ph acoustic model. Each is trained on BL_GRD_M1 and evaluated on BL_EVL_M.

J.2 Phonetic alphabet

The effect of the phonetic alphabet on the two-stage phone distance grader from §5.2.1 is investigated, by separately varying the alphabet used for ASR and for alignment, with the alphabet used for alignment also used for feature extraction. An alternative to segmenting speech into its constituent phones is to instead treat graphemes (i.e. the letters of the text) as the most basic speech units. The idea is that non-native speakers are likely to make large numbers of lexical errors, pronouncing words in ways that more closely resemble their spelling that their canonical pronunciation. A graphemic alphabet is thus hypothesised to improve speech recognition performance on such speakers. On the other hand, a graphemic representation would be expected to be less useful for characterising pronunciation proficiency, as correct pronunciation is a product of the phones rather than the graphemes of a word. Results using the baseline Gaussian Process grader and comparing phone distance features to baseline ALTA features (Appendix H) are displayed in Table J.2 below.

Train Set	Test Set	ASR	Align/Grd	Grader PCC	
				Base	Pron
BL_GRD_GJ	BL_EVL_GJ	DH-ph	GH-ph	0.843	0.838
		DH-gr	GH-ph	0.841	0.804
		DH-gr	GH-gr	0.832	0.771
BL_GRD_M1	BL_EVL_M	DH-ph	GH-ph	0.852	0.806
		DH-gr	GH-ph	0.863	0.804
		DH-gr	GH-gr	0.859	0.734

Table J.2 Performance, measured by Pearson correlation (PCC), of Gaussian Process graders using ALTA baseline features (Base) and phone/grapheme distance features (Pron) obtained from PLP-39 observation vectors, after recognition (ASR) and alignment (Align/Grd) with phonetic (DH-ph and GH-ph) and graphemic (DH-gr and GH-gr) acoustic models. Feature extraction for each of Base and Pron follows the same phonetic alphabet used for alignment.

While switching from a phonetic to a graphemic ASR improves ASR performance [147] and performance using the ALTA baseline features, it deteriorates performance using phone distance features across both pairs of datasets. As expected, switching from phonetic to graphemic alignment (and thus from phone distance features to grapheme distance features) deteriorates performance even further. The grader has significantly worse performance on the mixed L1 datasets compared to the Gujarati-only datasets, despite the reverse being the case with the baseline features. This is consistent with the types of deviations from proficient pronunciation made by different low proficiency speakers varying dependent on their L1, such that determining a speaker's proficiency is easier if the L1 is fixed.

J.3 Sequence modelling

Experiments are now performed to investigate the effect of different sequence-to-vector transformations on the performance of the deep pronunciation and rhythm graders (the equivalent experiments for the deep intonation grader are reported in §6.3.3).

The architectures explored include two types of bi-directional LSTM (Figure 4.6, left and Figure 4.6, right) and a transformer (Equation 4.45) with each of 1 and 8 attention heads (relevant diagrams reproduced below - note the transformers used in this case have an extra attention mechanism over the output layer of the multi-head attention mechanism to project to a fixed-length representation).

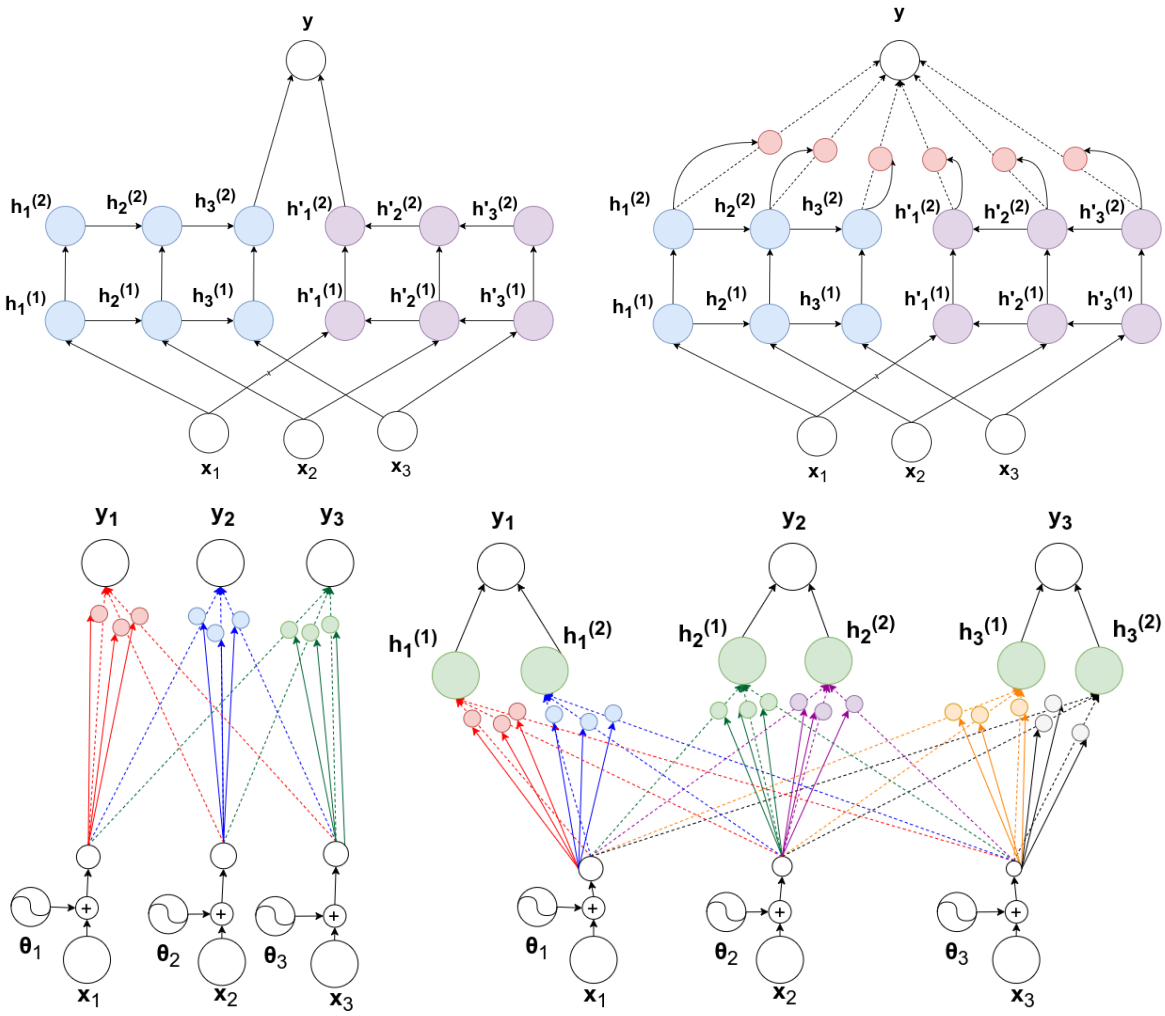


Fig. J.1 Illustrations of standard (top, left) and attention (top, right) bi-directional RNNs, and single-head (bottom, left) and multi-head (bottom, right) attention, reproduced from Figures 4.6 and 4.7

Results for both graders are displayed in Table J.3. In both cases, the LSTM models clearly outperform the transformer models in terms of both prediction accuracy and robustness to random initialisation. For the pronunciation grader, the attention LSTM performs better than the regular LSTM in terms of Pearson correlation coefficient, while the regular LSTM is ahead in terms of other accuracy metrics. The reverse is the case for the rhythm grader.

Grader	Model	PCC	MSE	MAE	%<0.5	%<1.0
pron	8-head Transformer	0.782 ±0.11	0.579 ±0.28	0.581 ±0.056	54.9 ±2.2	83.9 ±3.1
	1-head Transformer	0.699 ±0.022	0.753 ±0.05	0.687 ±0.022	43.3 ±3.0	76.3 ±1.5
	AttLSTM	**0.82** ±.021	0.531 ±.051	0.573 ±.017	53.6 ±2.1	83.5 ±1.8
	LSTM	0.804 ±0.031	**0.492** ±0.089	**0.54** ±0.036	**56.3** ±1.5	**86.2** ±2.0
rhythm	8-head Transformer	0.775 ±0.036	0.623 ±0.065	0.61 ±0.024	**50.9** ±1.5	79.0 ±1.0
	AttLSTM	0.819 ±0.068	**0.541** ±0.13	**0.578** ±0.05	49.6 ±3.0	**82.6** ±4.6
	LSTM	**0.821** ±0.062	0.598 ±0.11	0.612 ±0.056	49.6 ±4.3	81.3 ±4.6

Table J.3 Performance of end-to-end graders trained on BL_GRD_M1 and evaluated on BL_EVL_M with MFCC-13 from TD-gr aligned with GH-ph.

The source of this apparent inconsistency in the pronunciation case is illustrated in Figure J.2. While the predictions of the attention LSTM model are more precise, high and low scores are suppressed towards the centre (a calibration issue).

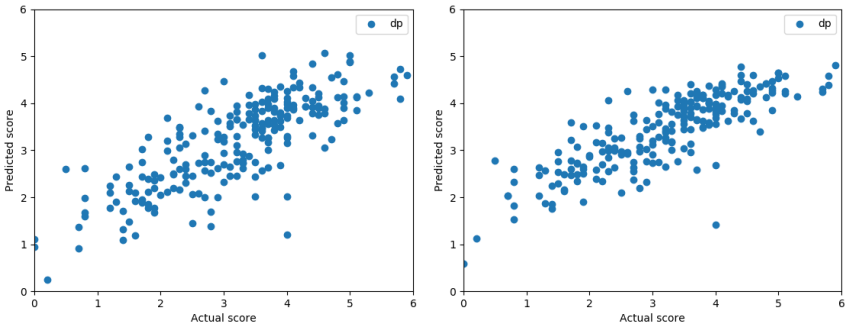


Fig. J.2 Expert human scores against predictions of deep phone distance grader with regular (left) and attention (right) LSTM trained on BL_GRD_M1 and evaluated on BL_EVL_M1.

J.4 Attention

Two forms of attention are tested, additive and scaled dot product (§4.2.3) in each of the pronunciation and rhythm end-to-end systems. As seen in Table J.4, additive attention outperforms scaled dot product attention in both cases. This makes sense given the greater representational power of additive attention and the fact that the similarity paradigm underlying scaled dot product attention isn't fully applicable to these systems.

Grader	Attention	PCC	MSE	MAE	%<0.5	%<1.0
pron	add	**0.815** ±0.025	**0.501** ±0.052	**0.544** ±0.027	**58.5** ±2.4	**85.3** ±2.2
	sdp	0.786 ±0.024	0.556 ±0.036	0.572 ±0.028	52.7 ±2.8	83.9 ±2.6
rhythm	add	**0.807** ±0.055	**0.569** ±0.11	**0.592** ±0.057	**54.0** ±6.1	**80.8** ±2.6
	sdp	0.785 ±0.08	0.624 ±0.14	0.631 ±0.063	50.4 ±4.5	79.5 ±3.0

Table J.4 Performance of deep pronunciation and rhythm graders with additive (add) and scaled dot-product (sdp) attention, in att-LSTM configuration, trained with CLR on BL_GRD_M1 and evaluated on BL_EVL_M, starting from MFCC-13 observation vectors.

J.5 Batch normalisation

Next, the impact of batch normalisation (§4.4.2) on the end-to-end pronunciation grader is evaluated. Given the highly non-linear and hierarchical nature of the deep phone grader, it is expected to have a rough cost surface and therefore benefit from the smoothing provided by batch normalisation. The results are shown in Table J.5.

Normalisation	PCC	MSE	MAE	%<0.5	%<1.0
batch norm	**0.815** ±0.025	**0.501** ±0.052	**0.544** ±0.027	**58.5** ±2.4	**85.3** ±2.2
none	0.795 ±0.15	0.606 ±0.24	0.62 ±0.12	46.4 ±7.7	79.5 ±7.4

Table J.5 Performance of deep phone distance grader with and without batch normalisation trained using clr with five different random seeds on BL_GRD_M1 and evaluated on BL_EVL_M, with MFCC-13 from TD-gr aligned with GH-ph.

They confirm that this is indeed the case with the batch normalised network being significantly more accurate as measured by all metrics and considerably less sensitive to random initialisation compared to its unnormalised counterpart. Similar results were seen with the end-to-end rhythm and intonation graders.

J.6 Learning rate schedule

Finally, the effect of the learning rate schedule on the end-to-end pronunciation and rhythm graders is investigated, with results reported in Table J.6. For both graders, the exponential learning rate schedule emerges as best performing in terms of Pearson correlation, but due to a calibration issue is outperformed by the constant schedule (for the pronunciation grader) and the decaying schedule (for the rhythm grader) in terms of other metrics.

Grader	Schedule	PCC	MSE	MAE	%<0.5	%<1.0
pron	exponential	**0.82** ±.021	0.531 ±.051	0.573 ±.017	53.6 ±2.1	83.5 ±1.8
	constant	0.815 ±0.025	**0.501** ±0.052	**0.544** ±0.027	**58.5** ±2.4	**85.3** ±2.2
	decaying	0.799 ±0.15	0.519 ±0.48	0.556 ±0.14	55.8 ±5.6	83.5 ±5.1
rhythm	exponential	**0.819** ±0.068	0.541 ±0.13	0.578 ±0.05	49.6 ±3.0	82.6 ±4.6
	constant	0.807 ±0.055	0.569 ±0.11	0.592 ±0.057	54.0 ±6.1	80.8 ±2.6
	decaying	0.812 ±0.021	**0.53** ±0.043	**0.558** ±0.023	**56.7** ±1.2	**83.5** ±1.7

Table J.6 Performance of deep pronunciation and rhythm graders trained using constant, decaying, and exponential learning rate schedules with five seeds on BL_GRD_M1 and evaluated on BL_EVL_M, starting from MFCC-13 observation vectors derived from TD-gr ASR aligned with the GH-ph acoustic model.

Appendix K

Accent Error Types

Common patterns of phone insertions, deletions and substitutions (accent errors) are collected from the literature for 10 L1s present in the BULATS, SELL and LeaP data sets used for error detection in this work (see §7.3). Error pattern are obtained from two aggregators [216, 121] as well as from work specific to each L1, namely: French [105, 244], Chinese [83, 45, 222, 231, 44, 252], Vietnamese [261, 203, 237, 8], Thai [53, 143, 124], Spanish [41, 3, 93, 18, 275], Russian [131, 242, 32], Dutch [82, 287, 63], Arabic [13, 75, 228], Polish [251, 212, 96] and German [30].

The results are displayed below in the categories of deletions (Table K.1), insertions (Table K.2), vowel substitutions (Table K.3) and consonant substitutions (Table K.5).

Error type	Description	Example	L1s
H silencing	Deleting h sound. Common in L1s where the letter h is silent.	hotel => otel	French
Final consonant deletion	Deleting final consonant of a word. In L1s with monosyllabic words and/or that always end in a vowel.	road => row	Chinese Vietnamese
Cluster reduction	Deleting one of the consonants in a cluster. In L1s where clusters do not occur.	stable => sable	Chinese Vietnamese Thai Spanish

Table K.1 Deletion accent error types

Error type	Description	Example	L1s
Anaptyxis	Insertion of a vowel between two consonants. In L1s where consonant clusters do not occur.	stable => s-uh-table	Chinese Vietnamese Thai Spanish
Final vowel insertion	Insertion of vowel at the end of a word ending in a consonant. In L1s where words usually end in vowels.	start => start-ah	Chinese
Initial vowel insertion	Insertion of vowel at the start of a word. In L1s where words never start with certain consonants.	sport => eh-sport	Spanish

Table K.2 Insertion accent error types

Error type	Description	Phones	Example	L1s
Reduction failure	Failure to reduce unstressed vowels to schwas. Found in L1s without similar schwa reduction rules.	AX => [AE, EH, IH, OH, AH]	about => ae-bout	Chinese Spanish Russian
Vowel length confusion	Replacing long vowels with their shortened forms and vice versa. Common in L1s with fewer vowels than English.	IH <=> IY UH <=> UW EH <=> ER AH/AA <=> AE	fish => f-ee-sh	All
Diphthong confusion	Confusing some dipthongs with monophthongs and each other. Found in L1s that lack those diphthongs.	EY<=>EH/AE AY<=>EH OW/AW<=>OH AY<=>EY UA<=>UW	let => l-ey-t	All
Back vowel confusion	Confusing back vowels [aw] and [aa] with each other and with [ow]. Found in L1s with fewer vowels than English.	AW<=>AA AW<=>OW AA<=>OW	phone => ph-aw-n	Thai Vietn. Spanish Dutch
UW - AH confusion	Pronouncing [uw] as [ah] and vice versa. Common in L1s with many u sounds	UW <=> AH	food => f-uh-d	French Arabic
IH - EH confusion	Pronouncing short i as short e and vice versa. Common in L1s with [eh] but no [ih].	IH <=> EH	rid => r-e-d	French Arabic

Table K.3 Vowel substitution accent error types (Vietn. = Vietnamese, All = French, Chinese, Vietnamese, Thai, Spanish, Russian, Dutch, Arabic, Polish and German)

Error type	Description	Phones	Example	L1s
L-N confusion	Pronouncing l as n and vice versa. In L1s without a distinct [l].	L <=> N	ladder => n-adder	Chinese Vietn.
L-W confusion	Pronouncing l as w and vice versa. In L1s without a distinct [l].	L <=> W	ladder => w-adder	Thai
Y-J confusion	Spanish has no [y] and uses y to represent [jh].	Y <=> JH	you => j-ou	Spanish
V-B confusion	Confusing [v] and [b]. In L1s where they map to the same phoneme.	V <=> B	very => b-ery	Spanish Chinese
V-W confusion	Pronouncing v as w and vice versa. Found in L1s where the two phones are represented by the same phoneme.	V <=> W	very => w-ery	Chinese Thai Vietn. Spanish Dutch German

Table K.4 Consonant pairwise confusion substitution error types (Vietn. = Vietnamese, All = French, Chinese, Vietnamese, Thai, Spanish, Russian, Dutch, Arabic, Polish and German)

Error type	Description	Phones	Example	L1s
Rhotic failure	Failure to pronounce the letter r, replacing it with [l] or [w]. In L1s without a distinct r sound.	R => L R => W	red => l-ed	Chinese Thai Vietn. Polish
Dental fricative fortition	Pronouncing [dh] and [th] as each other or their respective stops. In L1s without dental fricatives.	TH <=> DH TH => T DH => D	this => d-is	All
Other Fortition	Strengthening fricative/affricate to stop. In L1s with fewer fricatives.	F => P S/SH/CH => T	effort => e-p-ort	Thai Vietn.
Incorrect voicing	Pronouncing unvoiced consonants as their voiced counterparts.	F => V S => Z T => D P => B	sea => z-ea	All
Affricate failure	Pronouncing an affricate as its corresponding fricative. In L1s with fewer affricates than English.	SH => S ZH => Z	shoe => s-oo	Spanish Arabic Dutch
Affricate confusion	Confusion affricates for each other. Common in L1s with fewer affricates than English.	SH <=> CH JH <=> SH	cheap => sh-eep	French Thai Vietn. Spanish Arabic Dutch German

Table K.5 Other consonant substitution error types (Vietn. = Vietnamese, All = French, Chinese, Vietnamese, Thai, Spanish, Russian, Dutch, Arabic, Polish and German)

Appendix L

Error annotation interfaces

Each row of the table below contains a recording of a speaker speaking three words and a transcription of what we believe they said. Please listen to each recording and, focusing on the second to last word, answer:

1. Whether the second to last word (in **bold and red**) is indeed the word spoken by the speaker (Yes or No).

2. Which of the given pronunciations represents the way the speaker pronounced that word?

e.g.

Audio	Words	Correct?		Pronunciation			
		Yes	No	Phone	Correct	Incorrect	Other Incorrect
▶ 0:00	BECAUSE OF THE DEMAND **THEY** HAVE	◉	○	th in th-ey	○ like th in that	◉ like d in dog	○ Other

Audio	Words	Correct?		Pronunciation			
		Yes	No	Phone	Correct	Incorrect	Other Incorrect
▶ 0:00	FOR TO SIDE TO **SIDE** AIRPORTS	○	○	de in si-de-	○ like d in dog	○ like t in top	○ Other
▶ 0:00	PARTS HAVE SOME SAME **SAME** ON	○	○	s in s-ame	○ like s in sad	○ like sh in sheep	○ Other

Fig. L.1 AMT Hit collecting data on pronunciation errors

Audio	Words	Correct?		Pronunciation		
		Yes	No	Correct	Incorrect	Other Incorrect
▶ 🔊 ⬇	THE PRODUCTS PURCHASE AND **PURCHASE** PROBLEM	◉	○	○ p-UR-chase	◉ purch-A-se	○ Other

Audio	Words	Correct?		Pronunciation		
		Yes	No	Correct	Incorrect	Other Incorrect
▶ 🔊 ⬇	EVENT MANAGEMENT **EVENT** MANAGEMENT	○	○	○ ev-E-nt	○ -E-vent	○ Other
▶ 🔊 ⬇	DON'T USE MOBILE PHONES **MOBILE** PHONES	○	○	○ m-O-bile	○ mob-I-le	○ Other
▶ 🔊 ⬇	TWO HUNDRED FOUR **HUNDRED** SIX	○	○	○ h-U-ndred	○ hundr-E-d	○ Other

Fig. L.2 AMT Hit collecting data on stress errors