A Model for the Assessor Bias in Second Language Pronunciation Assessment

Jose Antonio Lopez Saenz

Machine Intelligence for Natural Interfaces (MINI) Lab,
Speech and Hearing (SPandH) Research Group,
Department of Computer Science
University of Sheffield

This dissertation is submitted on April 2023 for the degree of
Doctor of Philosophy

To the tired, the poor, the huddled masses
yearning to breathe free...

Abstract

In pronunciation assessment (PA) of second language (L2) speech, it is known that similarity to a native accent is desired, yet not crucial. There are certain variations in pronunciation which do not interfere with communication. It is up to the listener to decide whether a pronunciation differs from the one of so-called *canonical reference*. The subjectivity in pronunciation assessment can be referred to as the assessor bias.

A computer-assisted pronunciation assessment is subject to the effects of assessor bias. The disagreement between assessors causes inconsistencies in the data used to build models for the assessment task. A model for the bias itself, however, would help build a general reference for a proficient L2 speaker as well as an impartial PA.

This thesis proposes a model for the assessor bias to be included as part of a model for a pronunciation assessor. The assessor model consists of an ideal assessor-independent scoring function for PA, modified by an additive term specific to the assessor. The latter term is referred to as bias. The research for the model resulted in four original contributions. All contributions were tested on data from L2 speech from young learners of English in the Netherlands. Each recording was annotated for mispronunciation at the phoneme level by three trained phoneticians. Overlapping annotation made the data the best fit for a consistent model of inter-assessor disagreement.

A first contribution is a novel approach for detecting mispronunciations without the need for a precise phoneme alignment, which outperformed a baseline of pronunciation correctness scores based on phoneme alignments. The second contribution is a study of the effect of speaker metadata on learning a pronunciation reference. Models trained on different assessors were proven to be sensitive to different speaker information. The third contribution was the proposal and implementation of the assessor model. Two deep networks combine a bidirectional long short-term memory module with self-attention and a feed-forward classifier to estimate the probabilities of phonemes being pronounced correctly. Both networks were trained jointly to estimate the observed pronunciation labels. Only one network was modelled on the assessor's identity. The fourth contribution consists of methods for increasing the specialisation of the bias networks by reducing its cosine similarity and co-dependence with respect to the assessor-independent network. Using cosine similarity and a contrastive log-ratio upper bound for mutual information, it was possible to both reduce the correlation and dependency between the two networks. The bias network managed to increase its dependence on assessor identity and speaker factors. The mutual information between the assessor and the bias output was useful to illustrate disagreement, as well as which assessors and phonemes were the most prone to the bias.

Declaration

This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except where specified in the text. This dissertation is not substantially the same as any that I have submitted for a degree or diploma or other qualification at any other university.

Jose Antonio Lopez Saenz
August 2022

Acknowledgements

I want to extend my gratitude to my supervisor, Prof. Thomas Hain, for his unconditional support and guidance. From the first lecture in the module on speech technology to the pints shared after the lockdown, not a single conversation has been in vain.

I need to acknowledge my family back home for always supporting my urge for doing something else, even when they would not know what I am all about. To see both my parents dedicated to research and the education of others, makes me realize that I need to do my best so others can also do their best.

This has been a difficult time far from home, with financial struggle, and uncertainty. It would have been even more difficult without the support of my partner Elysa and my new family from this side of the world, the Ioannou and the Woolley.

The sleep deprivation from the last couple of weeks allows me only to mention that this thesis was possible with the support from CONACYT and the program for overseas scholarships with fellowship number 661687.

Contents

List of Figures

List of Tables

Nomenclature

A list of the variables and notation used in this thesis is defined below. The definitions and conventions set here will be observed throughout unless otherwise stated. For a list of acronyms, please consult page 151.

α	Empirical coefficient
$\alpha_{c,t}$	Attention weight from attention component c at time step t
$\angle$	Angle between two vectors
β	Empirical coefficient
$\boldsymbol{\alpha}$	Attention weight vector
$\boldsymbol{\Lambda}$	Parameters of an acoustic model
$\boldsymbol{\mu}_{i,m}$	Mean vector for component m of GMM i
$\boldsymbol{\mathcal{V}}^{w}$	Categorical confusion matrix for worker w
$\boldsymbol{\nu}_{k}^{w}$	Confusion vector for class k given worker w
$\boldsymbol{\pi}$	A possible sequence path
$\boldsymbol{\pi}$	Dirichlet priors
$\cdot$	Inner product between two vectors
Δ	Differential or gap
ϵ	A small scalar
η	Assessor identity tag
$\hat{\boldsymbol{\Lambda}}$	Parameters of an acoustic model that maximize its likelihood
$\hat{l}_i$	The correctness label estimate for the i^{th} phoneme in $\mathbf{r}$
κ	Cohens kappa

$\mathbb{E}[x]$	Expected value of a random variable x
$\mathbb{E}_{p(x,y)}[x]$	Expected value of a random variable x given the joint probability distribution of random variables x and y
$\mathbb{E}_{p(y)}[x]$	Expected value of a random variable x g given the probability distribution of a random variable y
$\mathbb{E}_{Y}[x]$	Expected value of a random variable x given variable y.
$\hat{\mathbf{l}_\mathbf{A}}$	The assessor-independent correctness labels estimate associated to sequence $\mathbf{r}$
$\hat{\mathbf{l}_\mathbf{b}}$	The assessor bias correctness labels estimate associated to sequence $\mathbf{r}$
$\hat{\mathbf{l}}$	The correctness labels estimate associated to sequence $\mathbf{r}$
$\mathbf{\Sigma}_{i,m}$	Covariance matrix for component m of GMM i
$\mathbf{b}_f$	Linear bias vector for LSTM forget gate f
$\mathbf{b}_h$	Linear bias vector for RNN hidden layer h
$\mathbf{b}_i$	Linear bias vector for LSTM input gate i
$\mathbf{b}_o$	Linear bias vector for LSTM output gate o
$\mathbf{b}_o$	Linear bias vector for output layer h
$\mathbf{b}_z$	Linear bias vector for LSTM input layer z
$\mathbf{c}$	A sequence of characters
$\mathbf{c}^{(t)}$	LSTM cell value at time t
$\mathbf{f}^{(t)}$	LSTM forget gate at time t
$\mathbf{h}$	A latent variable
$\mathbf{h}^{(l)}$	hidden state of the layer l in an ANN
$\mathbf{h}_{\mathbf{O}^{(w)}}$	BDLSTM hidden states for utterance $\mathbf{O}^{(w)}$
$\mathbf{i}^{(t)}$	LSTM input gate at time t
$\mathbf{K}$	A set of categories
$\mathbf{L}_\mathbf{A}$	Assessor-independent logit output
$\mathbf{L}_\mathbf{b}$	Assessor-specific logit output
$\mathbf{L}_{\mathbf{j}_n}$	Logit output from layer j

$\mathbf{l}$	A sequence of labels in $\mathbf{L}^T$
$\mathbf{l}$	The correctness labels associated to sequence $\mathbf{r}$
$\mathbf{L}^T$	A set of possible labels for a sequence of length T
$\mathbf{L}'^T$	A set of possible labels for a sequence of length T including a *blank* symbol
$\mathbf{L}_i^w$	Label for observation x_i provided by worker w
$\mathbf{o}^{(t)}$	LSTM output gate at time t
$\mathbf{O}^{(w)}$	Acoustic segment associated to w
$\mathbf{o}^t$	Acoustic observation at time t
$\mathbf{O}_s$	An utterance produced by a student
$\mathbf{O}_t$	An utterance produced by a teacher
$\mathbf{P}_t$	Phoneme sequence produced by a teacher
$\mathbf{p}_f$	Linear weight vector for LSTM forget gate f
$\mathbf{p}_i$	Linear weight vector for LSTM input gate i
$\mathbf{p}_o$	Linear weight vector for LSTM output gate o
$\mathbf{q}$	A finite sequence of events
$\mathbf{r}$	A phoneme sequence assumed to be the canonical pronunciation of w
$\mathbf{R}_f$	Linear weight matrix for LSTM forget gate f
$\mathbf{R}_h$	Linear weight matrix for RNN hidden layer h
$\mathbf{R}_i$	Linear weight matrix for LSTM input gate i
$\mathbf{R}_o$	Linear weight matrix for LSTM output gate o
$\mathbf{R}_z$	Linear weight matrix for LSTM input layer z
$\mathbf{s}$	A phoneme sequence uttered by the speaker to pronounce w
$\mathbf{W}$	Set of annotation workers
$\mathbf{W}^{(l)}$	Linear weights matrix of the layer l in an ANN
$\mathbf{W}_t$	A word produced by a teacher
$\mathbf{W}_f$	Linear weight matrix for LSTM forget gate f
$\mathbf{W}_h$	Linear weight matrix for RNN hidden layer h

$\mathbf{W}_i$	Linear weight matrix for LSTM input gate i
$\mathbf{W}_o$	Linear weight matrix for LSTM output gate o
$\mathbf{W}_o$	Linear weight matrix for output layer h
$\mathbf{W}_z$	Linear weight matrix for LSTM input layer z
$\mathbf{X}$	A set of observations
$\mathbf{Y}^*$	Set of latent true identity labels
$\mathbf{Z}$	A sequence of states
$\mathbf{z}^{(l)}$	Logit output of the layer l in an ANN
$\mathbf{z}^{(t)}$	LSTM input block at time t
$\mathcal{C}_j$	Contribution percentage from the output layer j to the total sum of squared logits.
$\mathcal{L}$	Loss function
$\mathcal{O}$	An utterance
$\mathcal{W}$	Sequence transcription of an utterance
$e_{c,t}$	Energy for attention component c at time step t
V_c	Weight matrix for attention component c
v_c	Weight matrix for attention component c
W_c	Weight matrix for attention component c
μ_p	Mean GOP score for phoneme p
$\mu_{\mathbf{L_A}}$	Overall mean $\mathbf{L_A}$
$\mu_{\mathbf{L_b}}$	Overall mean $\mathbf{L_b}$
$\odot$	Hadamard product
∂f	Partial derivative of function f
ϕ	Non-linear transformation
ψ	Context vector
σ	Non-linear transformation
σ	Sigmoid regularization function
σ_p^2	Variance of the GOP score for phoneme p

$EC(\mathbf{O}^{(w)})$	Sequential encoding for utterance $\mathbf{O}^{(w)}$
FFN_A	Assessor-independent feed-forward network
FFN_b	Assessor bias feed-forward network
sign	Sign function
$\mathrm{CS}(\mathbf{L_A}, \mathbf{L_b})$	Cosine similarity between $\mathbf{L_A}$ and $\mathbf{L_b}$
$\mathrm{MI}(x;y)$	Mutual information between random variables x and y
$\mathrm{MI}_C(x;y)$	Mutual information upper bound between x and y
$\mathrm{MI}_{C\theta}(x;y)$	Variational mutual information upper bound between x and y
$\mathrm{NF}(p)$	Number of frames for phoneme p
θ	Angle between two vectors
θ	Model parameters
$\theta_{\mathbf{O}}$	Current estimate of model parameters θ given the observation $\mathbf{O}$
$\top$	Matrix transpose
ϱ	Empirical coefficient
$A(\mathbf{O})$	Assessor-independent pronunciation scoring function for utterance $\mathbf{O}$
a^{ij}	The probability to reach state q_j from q_i
$A_\eta(\mathbf{O})$	Assessor-specific pronunciation scoring function for utterance $\mathbf{O}$
b	Scalar bias parameter
$b_\eta(\mathbf{O})$	Assessor-specific bias function for utterance $\mathbf{O}$
b_c	Linear bias vector for attention component c
$b_i(\mathbf{o}^t)$	Observation probability of $\mathbf{o}^t$ at state q_i
$c_{i,m}$	Scalar mixture weight for component m of GMM i
$corr(\mathbf{L_A}, \mathbf{L_b})$	Correlation between $\mathbf{L_A}$ and $\mathbf{L_b}$
d_l	Context vector for the lth character
$D_\theta \mathcal{L}$	Derivative of the loss function w.r.t parameters θ
E_x	Expected value of variable x
f	A function

$f(x)$	A function over variable x
$F1$	Harmonic mean of the precision and recall
g	Non-linear transformation
$GOP(p)$	Goodness of pronunciation for phoneme p
$h_{o_{t_0}^{(w)}}$	BDLSTM hidden state for utterance $\mathbf{O}^{(w)}$ at time t
k	Annotation category
l	The lth layer in an ANN
l_i	The correctness label for the i^{th} phoneme in $\mathbf{r}$
M	Sample size
N	Sample size
P	Precision or positive predictive value
$p(x)$	Probability distribution of a random variable x
Q	Phoneme set
q^0	Initial state of a Markov chain
q^j	The jth event in sequence $\mathbf{q}$
q^{T+1}	Final state of a Markov chain of length T
R	Length of phoneme sequence $\mathbf{r}$.
R	Recall or sensitivity
r_t^i	The i^{th} phoneme in $\mathbf{P}_t$
r_i	The i^{th} phoneme in $\mathbf{r}$
s^j	The jth state in sequence $\mathbf{q}$
$s_{\mathbf{L_A}}$	Overall standard deviation of $\mathbf{L_A}$
$s_{\mathbf{L_b}}$	Overall standard deviation of $\mathbf{L_b}$
T	Number of steps in sequence $\mathbf{q}$
t	A point in time
T_p	GOP score threshold for phoneme p

w	A known prompt
w	Annotation worker
w	Linear weights
$x^{(t)}$	Input at time t
x_i	The ith observation
$y^{(t)}$	Output at time t
y_i^*	True categorical label for an observation x_i
y_i^w	Categorical label for an observation x_i provided by worker w
z_i	The ith state
z_l	The lth character
$*$	Convolution
$\mathbf{Y}$	Hidden representation obtained from an encoder
$\|\rightarrow\|$	Euclidean norm of vector $\rightarrow$
$\mapsto$	Mapping function
$\overrightarrow{1}$	A vector with all components equal to 1
$\overrightarrow{\nu}$	A vector
aj	Assessor $j \in 1, 2, 3$
E1_aj	ASIM with output configuration E1 for assessor $j \in 1, 2, 3$
E1_AND0	ASIM with output configuration E1 for AND0 consolidated annotation
E1_AND1	ASIM with output configuration E1 for AND1 consolidated annotation
E1_MAX	ASIM with output configuration E1 for MAX consolidated annotation
E1	Output layer with a single binary output for detecting mispronunciations
E2_aj	ASIM with output configuration E2 for assessor $j \in 1, 2, 3$
E2_AND0	ASIM with output configuration E2 for AND0 consolidated annotation
E2_AND1	ASIM with output configuration E2 for AND1 consolidated annotation
E2_MAX	ASIM with output configuration E2 for MAX consolidated annotation

E2	Output layer with an output for every phoneme class being declared as correctly pronounced
E3_aj	ASIM with output configuration E3 for assessor $j \in 1, 2, 3$
E3_AND0	ASIM with output configuration E3 for AND0 consolidated annotation
E3_AND1	ASIM with output configuration E3 for AND1 consolidated annotation
E3_MAX	ASIM with output configuration E3 for MAX consolidated annotation
E3	output layer with two outputs for declaring either a correct or incorrect pronunciation for every phoneme class
GOP_aj	GOP baseline for assessor $j \in 1, 2, 3$
GOP_AND0	GOP baseline for AND0 consolidated annotation
GOP_AND1	GOP baseline for AND1 consolidated annotation
GOP_MAX	GOP baseline for MAX consolidated annotation
MF_aj	Most Frequent Label baseline for assessor $j \in 1, 2, 3$
STR_aj	Stratified baseline for assessor $j \in 1, 2, 3$
AND0	Consolidated annotation reference considering mispronunciation by unanimity
AND1	Consolidated annotation reference considering a correct pronunciation by unanimity
BP	Dutch province or country of birth
DIAL	Dutch dialect used daily
I	Inter-annotator agreement
L1	Native Language
MAX	Consolidated annotation reference obtained via majority voting
max	MAX function
MLH	Multilingual household
NND	Non-native Dutch speaker
NNP	Non-native Dutch speaking parents
SAL	Self assessed English proficiency level
SCH	School ID
YENG	Years of formal studies of English as L2

Chapter 1

Introduction

Language facilitates the transfer of information between people. Technically, a language is thereby a code that is understood by both interlocutors. Spoken language adds to this code the layer of sound, and to cope with real-world situations the code is fault-tolerant and allows for numerous variations. It is standard to analyse spoken words in terms of a sequence of sounds, so-called phones, and many allophones can relate to the same phoneme, a unit which is represented in dictionaries to describe the pronunciation of words. Learning the language code would be straightforward if there was only one true sound sequence that represents a word. But due to the aforementioned redundancy in practice, one can observe many variations. Intra-personal variations can be profound due to context and speaker state, but variations between speakers encompass many factors, due to physical differences and the different sociolinguistic backgrounds of a speaker. Accents and dialects exist, and while traditional pronunciation training would make use of an undisputed reference (a role model of pronunciation), a more modern approach considers that even assessors have specific perceptions. Rather than looking at defining a unique language reference, the work presented here understands any Pronunciation Assessment (PA) as unique, having its own emphasis, i.e., bias. In this thesis, models for this assessor bias are developed and analysed in the context of Second Language (L2) learners of English.

1.1 Pronunciation Assessment

1.1.1 The Assessor Bias

In PA, a listener declares the proficiency of a speaker in communicating using a reference that is assumed to be canonical. Strictly speaking, any change to a canonical phone sequence is considered a *mispronunciation*. However, such changes must be perceived by a listener in their role of assessor. It is known that not every phone variation affects communication if the same linguistic structure is kept. This is the case for *accents*, different pronunciations of the same language reference. A L2 learner is often encouraged to imitate a native accent, although this is not a required condition for speaking any language. When two speakers of the same

Native Language (L1) show different accents, none of them is considered incorrect. This is not the case for L2 speakers as their phoneme variations are more noticeable than the ones of L1 speakers.

A listener previously exposed to a particular accent is likely to overcome consistent variations. The listener must perceive minimal differentiation between phonemes to avoid confusion about the message. If an L2 speaker can produce key phoneme contrasts, their accent should not matter much. However, the accent of an L2 speaker can cause prejudice against them. The personal experience of the listener, their linguistic background and the perceived identity of the speaker have been noted to influence the identity of speech sounds recognized by a language assessor (Carey et al., 2011, Kartushina and Frauenfelder, 2014, Witteman et al., 2014). This work assumes the effect has a major role in the subjectivity of L2 PA; therefore, it is referred to as the *assessor bias*.

Less than 30 years ago, the similarity between L2 pronunciation and an ideal L1 accent was a condition for declaring a proficient speaker. The perception of a L2 accent was an undesired feature in L2 learning. Terms such as *natural pronunciation* and the *presence of a L2 accent* were still used recently to define levels of L2 proficiency in a grading scale (Harding, 2017). Research on L2 speech has started to change the need for replicating L1 accents. Formal PA now focuses more on speaker's ability to differentiate and articulate phonemes as required by the language reference (Soproni, 2020). This change of criteria makes the relevance of an accent less explicit; however, the decision of whether a speaker manages to produce the right articulation is still dependent on the assessor's perception (Harding, 2017, Kuiken and Vedder, 2014). Any preconception about the speaker can affect the listener's perception. Section 2.1.2 covers more in depth the possible causes of assessor bias. From speaker identity, previous exposure to similar accents and the accent of the listener influence PA (Harding, 2017, Lindemann, 2017, Winke et al., 2012).

1.1.2 Agreement Across Assessors

The variability in assessor perception is an obstacle to a fair and consistent PA. Recall that the pronunciation reference is an ideal representation of a particular accent. Since perception bias is specific to the listener, so is PA. A complete agreement across pronunciation assessors is not guaranteed. It is often assumed agreement can be inferred via a consensus or the decision of the majority. A consolidated annotation via majority voting or the average of joint assessments is often assumed to be a representation of inter-assessor agreement. Institutions in charge of authoritative L2 tests such as the IBT TOEFL train their assessment staff towards the same reference (Wei and Llosa, 2015). To do so represents an agreement on the bias rather than an assessment free of bias. Overall, the bias caused by the pronunciation reference prevails.

1.1.3 The Pronunciation Reference

The selection of a speech standard is rather arbitrary. A speech reference is typically used by a group of people concentrated in a particular geographical area. Said reference is called a dialect. When a dialect is used in a broader form outside the region it comes from, both its spoken and written uses are formally defined by a set of rules, i.e., a *standard*. A particular standard can be retained for status or societal motivations and becomes part of the identity of a nation (Finegan, 2014). The speech standard in particular is then known as a language; hence the famous phrase from linguist Max Weinreich: *"A language is a dialect with an army"* (Weinreich, 1945). A deviation from the language reference is stigmatized as *incorrect* (Finegan, 2014).

Only deviations in the pronunciation reference are considered in this thesis for the sake of a simpler definition of the problem. The fact that a L2 speaker is required to imitate a certain accent from all existing native variations of a language is often overlooked. The selection of a particular accent over other native accents contributes to the bias in the assessment of L2 proficiency.

1.2 Computer-Assisted Pronunciation Assessment

1.2.1 A Reference with a Bias

Early attempts for automatic PA using Machine Learning (ML) were carried out based on L1 speech examples assumed to be correct (Litman et al., 2018). It is no surprise that the resulting models fell short when tested on real L2 learner data. ML for PA is used for inferring a pronunciation reference from a set of speech examples paired with their respective assessment annotations. A biased selection of speech examples, along with a biased assessment of such examples, will always yield a biased model for PA.

The bias in any annotation task is often dealt with by using joint assessments. If there are enough annotations to break a tie on each example, the decision of the majority is typically used. A consolidated annotation, however, makes for an inconsistent property of the labels. The data could also be labelled with non-overlapping annotations from multiple assessors, causing an even less consistent reference. These and other problems related to annotation methods for mispronunciations are explained in more detail in Section 2.5.1.

1.2.2 The Alignment Problem.

An additional assumption often used for CAPA is related to the misconception of the phoneme as a defined acoustic event. A widely used method for CAPA is based on detection, isolation and comparison of phoneme examples against a reference to produce a score for similarity (Witt and Young, 2000). This method suffers from what in this thesis is called the *alignment problem*. Since speech is a continuous signal, the idea of finding the boundaries in time for a phoneme is subject to inconsistencies. A single phoneme identity is assumed for each segment in the alignment. Such assumption clashes with the variability and co-articulation, particularly of L2 speech (Dudy et al., 2018). The phoneme alignment of L2 speech also implies a segmentation into phonemes which the speaker might have not attempted to produce at all. The insertion and deletion of phonemes in L2 speech make the comparison against a canonical sequence not trivial; this is without mentioning that a phoneme boundary is meaningless. Section 2.4.2 illustrates the *alignment problem* under the possible cases defined by (Dudy et al., 2018), and how it affects CAPA.

1.3 Motivation

In short terms, PA requires for a listener to judge how a speaker communicates using a language reference. The criteria used for assessment are used in the context of the reference internalised by the assessor. A method for PA free of bias, or at least with a reduced bias effect, would represent a more reliable and fair proficiency test for L2 speakers. Consider, the life and career of a L2 speaker could be subject to obtaining a specific L2 certification. The elimination of the bias could also help define a metric for which any listener would declare a speaker either competent or incomprehensible.

A pronunciation assessor immune to bias is unfeasible. The inherent bias in any speech-related activity makes it impossible to conceive accent-free speech. A ML approach might offer a solution to this problem. Since speech is not constant, it can be seen as a random process parameterised by latent factors related to the experience of the speaker, the pronunciation reference held, the social context of the speaker, and so on. Intra-speaker factors such as focus, tiredness, or illness also add variation to speech (Isaacs and Harding, 2017). On the assumption that it is possible to model a speaker based on their latent parameters, these can be manipulated to control their effect. Similarly, assuming a model for the L2 assessor is found, the parameters with an effect on the bias can be identified. Once the effect of the bias is understood and further reduced, what is left could be understood as a PA model free of bias. Therefore, it is the aim of this thesis to find a way to model the assessor bias.

A model of the assessor bias could also help create unbiased references for CAPA. One of the main problems with CAPA is that it will always be limited by the inconsistency of the data chosen as the reference. This thesis challenges the assumption that the chosen data assumes the pronunciation reference is consistent and free of bias. To better capture the assessor bias, the reference should be determined only from the annotations available. The less additional bias imposed by the construction of the model, the closer it can get to the real assessor's perception.

Another common assumption in CAPA challenged in this thesis are phonemes represented as discrete acoustic units. The assessment of phoneme production does not require the precise time of occurrence of the phoneme. A pronunciation assessor listens to an utterance first, and then compares it against a reference. Such interpretation of PA already clashes with phoneme alignment-based metrics for CAPA (Chu et al., 2020, Lin et al., 2020, Witt and Young, 2000, Wu et al., 2012, Yang, 2015). PA at phoneme level can be accomplished without subjecting it to the alignment problem (see Section 2.4.2) nor a reference model for each phoneme class. To avoid the incorporation of bias unrelated to the assessor, an alternative is to detect the presence of phonemes directly from wider speech segments. In essence, the phoneme reference from an assessor is determined directly from utterances with a wider phoneme context.

1.4 Contributions

The work presented in this thesis includes the following contributions:

1. **a segment-based approach for mispronunciation detection:** a novel method for estimating the presence of mispronunciations in short utterances given an expected phoneme sequence (Chapter 3).

2. **the use of speaker metadata for improving automatic PA:** information about the linguistic background of the speaker was used for improving automatic PA (Chapter 4).

3. **a model for the assessor bias:** a model for PA was defined as the interaction of an assessor-independent scoring function modified by an assessor-specific function, referred to as the bias (Chapter 5).

4. **methods for encouraging bias estimate:** the cosine similarity and mutual information between the assessor independent and the bias functions were minimised to make the components of the assessor model less redundant (Chapter 6).

1.4.1 Segment-Based Approach for Mispronunciation Detection

The objective of the work proposed here was to find a new method for PA without the need for a precise phoneme alignment. Inspired by the fact that a listener does not need time information about the phonemes recognized, the presence of mispronunciations is estimated directly from a short speech segment without the need for timing information. The task was carried out by generating a sequential encoding using a combination of Bidirectional Long Short-Term Memory (BDLSTM) module and a self-attention mechanism. The resulting encoding was passed through a FFN with outputs for both correct and incorrect pronunciations of each phoneme class. The output posteriors were learned as a multi-label classification problem; hence there was no need for the definition of phoneme boundaries. The model was called ASIM. The model was tested on L2 speech data of young learners of English from the Netherlands. Each was annotated for mispronunciation at phoneme level by three trained phoneticians. An experiment compared the ASIM against an implementation of the GOP (see Section 2.4.1) which served as a baseline. The segment-based mispronunciation detection outperformed the GOP for learning the annotation reference of each assessor, as well as three formats of consolidated annotation. It was noted that the weights in the attention module aligned with the phoneme boundaries used for the GOP baseline, without any timing information used during training.

Relevant publication : Saenz, J. A. L., Jalal, M. A., Milner, R., & Hain, T. (2021, December). Attention based model for segmental pronunciation error detection. In 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) (pp. 725-732). IEEE.

1.4.2 Use of Speaker Metadata for Improving Automatic Pronunciation Assessment

This work aimed to offer an option to combat the known lack of L2 speech data annotated for mispronunciation. It is known that prior experience with accents and the perceived identity of the speaker can cause preconceived notions about the speaker. If proven useful, speaker metadata could alleviate the need for out-of-domain data to train a model for L2 PA. Information related to the linguistic background of the speaker was used along with the acoustic examples to improve the performance of the ASIM. The data set of L2 learners of English in the Netherlands used previously also includes metadata provided by the students, regarding L1, dialects used, multilingual households, and so on. The speaker factors from the metadata were anonymised and concatenated, with fixed dimensions at the acoustic features. Various ASIMs were trained for each of the three assessors in the dataset, and a consolidated reference via majority. The models were trained on the acoustic segments along with either single or combinations of speaker factors. Models trained on different assessors responded differently to the factors used. Some combinations showed an improvement in performance for learning certain assessors. The finding confirmed that speaker information could influence the bias of each assessor differently. It was observed that the balance of the metadata was crucial for it to cause an improvement in the ASIM. Further experiments on the ASIM trained on a more diverse and balanced speaker sample are required to help determine which factors affect the assessor bias the most. A model for the role of speaker factors in PA would make it easier to isolate the effect the bias has on assessment.

Relevant publication : Saenz, J. A. L., & Hain, T. (2021, November). Use of Speaker Metadata for Improving Automatic Pronunciation Assessment. In International Conference on Statistical Language and Speech Processing (pp. 61-72). Springer, Cham.

1.4.3 A Model for the Assessor Bias

After finding a method for learning a reference for PA, the next step was to isolate the bias contribution to the assessment. In this Section, a model for the assessor bias was proposed. The model defines PA as the interaction between an assessor-independent scoring function and an assessor-specific additive term referred to as the bias. The proportion in which each component of the assessor model contributes to the observed assessment is not known; hence both components must be learned simultaneously. The assessor model was implemented using the summation of the logits coming from two ASIMs, each corresponding to a component of the assessor model. The two ASIMs setup was called the DASIM. Both subnetworks observe the same acoustic input. Only the bias subnetwork was made sensitive to assessor identity by concatenating the assessor tag as a constant dimension along the acoustic features. The assessor-independent subnetwork learns each observation normalised across assessors. The DASIM was trained on L2 speech data of young learners of English from the Netherlands. The DASIM was trained on data from three assessors simultaneously. It was shown that the bias subnetwork was sensitive to assessor identity. The attention curves from the bias subnetwork could indicate points of inter-assessor disagreement on each segment. The DASIM was improved further by reducing the number of parameters. The optimised ASIM used a single BDLSTM and self-attention module for sequential encoding, and two FFNs for the assessor-independent and bias logits.

Relevant publication : Saenz, J. A. L., & Hain, T. (2022, May). A Model for Assessor Bias in Automatic Pronunciation Assessment. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7267-7271). IEEE.

1.4.4 Methods for Encouraging Bias Specialization

The logits corresponding to the components of the assessor model had similar behaviour. The work in this section focused on reducing the similarity and co-dependence between the assessor-independent and bias logits. The cosine similarity (CS) and the contrastive log-ratio upper bound for mutual information (CLUB) (Cheng et al., 2020) were used as part of the loss function to learn the assessor reference. Due to the unknown behaviour of the components of the assessor model, the criteria used were based on information theory, and vector similarity instead of a consolidated inter-assessor agreement inferred from the annotation. The effect of CS and CLUB on the ASIM behaviour was tested using the L2 speech data of young learners of English from the Netherlands, used previously. The model was trained on data from three assessors. The use of CS alone did reduce the correlation between the logits by making them perpendicular pairwise, with no negative effect on the detection of mispronounced segments. The use of CLUB alone was found to make the logits more independent with respect to each other and increased the mutual information (MI) between the bias logits at the assessor identity. It was also found that by including speaker factors in the bias input, CLUB increased the MI between the bias and the assessor identity even further. The MI values were useful to illustrate how relevant was the assessor identity for the bias output, offering a metric for comparing how the bias influences annotators and phoneme classes.

1.5 Organisation

The rest of the thesis is organised as the following: Chapter 2 contains background information for understanding the problem of PA, including the assessor bias, formal PA and opportunity areas in the widely used alignment-based scores for PA. Chapter 3 introduces the segment-based approach for mispronunciation detection and the ASIM. Chapter 4 explores the combination of speaker factors that have an effect on the learning of a pronunciation reference. The effect of the speaker factors is observed for both individual assessors and a consolidated reference. The assessor model and its implementation using the DASIM is shown in Chapter 5. The subsequent search for a leaner ASIM is also included in Chapter 5. Chapter 6 presents the reasoning and desired effect of the CS penalty and the MI reduction between the components of the assessor model. Finally, Chapter 7 summarises the findings of the thesis and sketches out a plan for the further direction of this research.

Chapter 2

Background

2.1 Speech, Sounds and Phonemes

Speech serves a purpose in communication by encoding meaning in a predominantly acoustic signal. For this, a speaker generates a range of sounds which can be classified based on spectral characteristics. Nonetheless, the relationship between acoustic features and meaning varies considerably (Holmes, 2001). From a linguistic point of view, the message in speech can be segmented into discrete units such as sentences, words, and syllables. This work focuses instead on smaller meaningful acoustic units put together to produce speech, which, in comparison to other linguistic units, a clear-cut segmentation of the continuous speech signal turns out to be more complicated.

The speech signal consists of a sequence of sounds produced by successive actions of the human vocal system, consisting of the lungs, larynx, pharynx, nose, and various movable organs in the mouth (Holmes, 2001). The different states of the vocal system can be identified; this would seem enough to map the acoustic signal to its corresponding linguistic units if it were not for the fact that changes in the vocal tract occur continuously without a clear separation between the different states associated to a particular sound (Jones, 1976). This situation can be explained using the idea of concrete and abstract sounds. A concrete sound corresponds to the actual sound produced at a given moment in the acoustic signal, while an abstract sound is common to multiple utterances which are said to have the same sound. In other words, a concrete sound corresponds to the manifestation of an abstract sound (Jones, 1976).

In linguistics, it is practical to assume a spoken language uses a reduced set of consistent sounds, i.e., abstract sounds. The sounds of this reduced set are understood as acoustic units which define the meaning of the utterance. It is then implied that if one of the abstract sounds changes, the meaning of the message does as well (Holmes, 2001). The abstract sounds are called *phonemes*. It is necessary to distinguish the mental construction of the phoneme from the real sound uttered by a speaker. The word *phone* is used to refer to said acoustic realisations (Holmes, 2001, Jones, 1976). Phonemes are often misunderstood as consistent concrete sounds (Moore and Skidmore, 2019). If this was the case, any language would be spoken without variation across all speakers and no evidence of accents could be observed. A clear distinction between phonemes and phones is essential to understand how variations in pronunciation could be considered *errors*, even if they do not affect the intended meaning in any way.

A set of phonemes is defined as a language based on the contrastive function of their phones. When two words in a language differ in one sound only, the choice of such sound sets the meaning of the utterance and are both considered to be different phonemes (Giegerich, 1992, Kortland, 2017). A speaker needs to efficiently produce the corresponding contrastive sounds to successfully transmit the intended meaning (Brown, 1995, International Phonetic Association., 1999).

The phonemes preceding and succeeding a particular phoneme directly influence transitions between different positions of the vocal articulators, e.g., tongue, lips, and teeth. This coarticulation effect is noticeable, as the muscles in the vocal system do not cause an immediate change in the sound wave. When different phones are used to represent the same phoneme, these phones are called *allophones* (Holmes, 2001). Various authors coincide in explaining the phoneme as a contrastive sound unit that characterises words and defines the meaning of the utterance. Allophones on the other hand, can be seen as non-distinctive sounds, yet the question of how much a phone could vary before being considered the allophone of a different phoneme remains (Giegerich, 1992, Holmes, 2001, International Phonetic Association., 1999, Jones, 1976, Kortland, 2017).

Although allophones can be perceived, they do not necessarily represent confusion over which phoneme they represent. Allophones are important when explaining speech accents of the same language. Although accents can differ acoustically when representing the same phonemes, the nearly identical underlying linguistic structure makes confusion between L1s speakers to be a rare event (Holmes, 2001). If the listener perceives a completely different phoneme from a speaker with a different accent, the listener can adapt their perception to said accent as long as the speaker is consistent and can properly produce contrasting phonemes to define the meaning of their speech (Lindemann, 2017, Witteman et al., 2014).

2.1.1 A Pronunciation Reference

A language is not completely constrained by a formal set of rules, and it is acknowledged that multiple factors influence how a certain population uses a language (Lindemann, 2017). It is common that from all the existing variations of a language, a particular style rises as a canonical reference. This usually comes from some entity located at the centre of power. This creation of a more *legitimate* form of the language can be understood as the search for uniformity and identity in a population (Milroy, 2001). For a language, this uniformity process usually involves institutions such as the language academies in charge of defining a *correct* dictionary, orthography and grammar for a language (Elizaincín, 2016). The creation of language reference is a more subjective process which could hold more inconsistencies compared to designing norms for concepts such as mass or distance (Milroy, 2001).

The English Language is the most spoken language worldwide; it is also recognized as an official language or L1 in more than 40 countries and territories around the world (Formentelli, Myers, 2015). Because of all the variations of English that exist, the idea of a unique standard has been discarded centuries ago (McArthur and Lam-McArthur, 2018). The concept of national standards for languages spoken over more than one country is a more accepted term, yet not even this assumption of uniformity is reflected in how L1 speakers use the language (Lindemann, 2017, McArthur and Lam-McArthur, 2018).

The assumption of uniformity in a language implies the legitimization of a particular pronunciation or accent and the stigmatisation of others (Milroy, 2001). The speakers of English from the BBC and Voice of America are examples of *good pronunciation*, according to the British Council, despite it being known that not even the population of London use the same accent (McArthur and Lam-McArthur, 2018, The British Council, 2018). A pronunciation reference is an accent chosen almost arbitrarily to be labelled as *normal* or *natural*, labels commonly used as descriptors to assess a *correct* pronunciation. Any deviation from the pronunciation reference can be considered *incorrect* by the listener, i.e., any perceived substitution, deletion or insertion of the phonemes dictated by the reference.

2.1.2 Perception Bias

Analogous to a speaker producing phones to represent phonemes, the listener assigns identities to the perceived sounds to decode a message. Whether a listener perceives phones either as allophones or as different *phonemes* is subject to preconceived notions of how their own language should sound like. These preconceptions can stigmatise any form of speech that sounds different to what a listener could consider a reference, even if they are listening to the same language, for example, a Spanish L1 speaker from Colombia listening to a Spanish L1 speaker from Puerto Rico (Lindemann, 2017). The listener's perception can be influenced by their phonological background, i.e., their own L1, and previous experience with other accents, along with the perceived identity of the speaker (Galaczi et al., 2011, Harding, 2017, Lindemann, 2017). L2 accents are easily noted by L1 speakers due to particular phonemic cues which may interfere with the realisation of contrastive sounds in the target language. Therefore, the accent has an effect on speech perception causing a bias towards the speaker (Carey et al., 2011, Kartushina and Frauenfelder, 2014, Witteman et al., 2014). The bias in speech perception becomes relevant, considering guidelines used for L2 PA using descriptors such as *A foreign accent is sometimes evident* or *It shows a natural intonation* (Harding, 2017). The influence of the perception bias in PA is discussed in more depth in Section 2.2.

Theories developed to explain how a speaker learns to differentiate phonemes which are not part of their L1 differ mainly on the relevance a previously acquired language has on the acquisition of another. On one extreme side of L2 acquisition theory, the process of language learning is the same whether a person learns any L1 or L2. Theories on the opposite side explain L2 acquisition based on the structure of a language previously learned in a process of *transferring* abilities (Klein, 1986). A discussion between various views on L2 acquisition theory goes beyond the scope of this work, yet it is important to acknowledge that an undeniable correlation between L1 and L2 has been used for explaining L2 accents and label some particular phoneme realisations as pronunciation errors (Flege, 1995, Klein, 1986, Winke et al., 2012).

2.2 Formal Pronunciation Assessment

A usual approach for grading L2 proficiency is to test the communicative competence of a subject through various tasks within a context (Council of Europe. Council for Cultural Co-operation. Education Committee. Modern Languages Division (Strasbourg), 2001). Among the various competence tested, pronunciation and conversation present some concern regarding how the perception of the rater can influence the evaluation of the L2 speaker. In PA of L2 speakers, a listener in the role of an assessor declares the proficiency of a speaker in communicating using a canonical reference. As mentioned before, the assessor's perception plays a key role in determining the identity of the phonemes produced by the speaker. Recall that any substitution, deletion, or insertion of a phoneme different from the reference can be labelled as a mispronunciation.

The seemingly inherent bias is easy to acknowledge considering it is more likely that a L1 speaker will understand a given L2 speech easily if they have been exposed to such an accent before (Harding, 2017, Levis, 2010, Lindemann, 2017). Consider a L1 speaker who is exposed daily to various L2 speakers from the same linguistic background. What this L1 speaker could understand for a *heavy accent* will not be the same as what a more naive L1 speaker with less to no contact with foreign accents would consider a heavy accent. In a situation like this, the more naive L1 speaker could penalise the presence of an accent even when this does not prevent communication (Lindemann, 2017, Munro, 2018). On the other hand, the L1 speaker with previous exposure to the same L2 accent might inadvertently ignore when the speaker fails to differentiate some phonemes. Since L1 speakers can adapt to a consistent accent, any previous knowledge of other accents can help overcome perceived mispronunciations (Lindemann, 2017).

There are certain word pairs that differ from each other not only in their meaning, but also on a single phoneme. These word pairs are called *minimal pairs* and require a clear differentiation when spoken to avoid confusion (Brown, 1995). A L2 speaker could have problems producing certain phones, which may affect the intended meaning. Since the minimal pairs affected by this might occur with a low frequency in the language, the context could be enough to convey the message successfully (Brown, 1995, Munro, 2018). Therefore, if speech does not rely only on pronunciation, the question of how much an accent really affects the performance of L2 speakers remains (Harding, 2017, Levis, 2010).

When an assessor compares an accent against a pronunciation reference, a phoneme differentiation particular to said accent may be interpreted as a mispronunciation. It is also known that variations in phoneme realisation have no effect on the meaning of the utterance (Harding, 2017, Holmes, 2001, Kuiken and Vedder, 2014, Lindemann, 2017, Suzukida and Saito, 2022). Therefore, bias in assessment should be acknowledged to avoid artificially deflated test scores when a L2 speaker can communicate without difficulties. The ideal PA would declare if a speaker can be understood regardless of the similarity of their accent to L1 pronunciation, yet the idea of perception free of bias is unrealistic. However, an evaluation for L2 proficiency needs not only to declare whether a speaker can replicate a given pronunciation reference; other communication competencies must be assessed as well.

Multiple authoritative tests for language proficiency exist; since English remains one of the most spoken languages in the world; this work focuses on tests for English as L2 (Eberhard et al., 2022). Different proficiency tests exist for different conventions of standard English, the most widely used being the Educational Testing Service (ETS) Test of English as a Foreign Language (TOEFL) for American English and the IELTS for British English. Both TOEFL and IELTS are of high importance as they are often requirements for immigration, job positions and to enrol in schools in many countries and territories where English is an official language. The reliability and fairness of any L2 proficiency test is a major concern, as the life and career of a L2 speaker could be subject to them achieving a minimum score.

Pronunciation is not the only competence scored in an L2 formal assessment, yet it is the one most subject to a biased perception. Disagreement across assessors is more common in PA compared to tasks involving grammar, orthography, or any aspect of language for which a set of rules are defined (Harding, 2017, Levis, 2010, Lindemann, 2017). A bias-free assessment is unfeasible, hence the importance for language assessors to follow the same pronunciation reference consistently. A group of listeners with the same L2 pronunciation reference represents the same bias in perception of the L2 accent. The challenge then consists of defining a set of descriptors for different proficiency levels of L2 pronunciation.

Different types of assessment define the requirements for the descriptors of L2 proficiency. An assessment with a unidimensional *fail-or-pass* decision lacks information and practicality. An evaluation of the current state of the speaker's skills needs to cover multiple competencies to generate adequate feedback for the test subject. If no feedback is provided, it would take longer for an L2 speaker to improve their pronunciation. Each stage in the speaker's proficiency needs to be clear about the minimum required skills the speaker needs to show. For example, vocabulary size, correct intonation, problems in phoneme differentiation, no grammatical errors and so on. The definition of different levels of L2 speech proficiency is implemented in the scoring rubrics used by the assessors. Nonetheless, the design of the rubrics and further training of the assessors using them are still subject to the effects of the perception bias, particularly for pronunciation (Harding, 2017, Kuiken and Vedder, 2014, Winke et al., 2012).

2.2.1 Pronunciation Scoring Scales and Rubrics

It has been found that certain features of L2 speech change according to the amount of training a speaker has; therefore, these changes can be discriminated alongside a range of proficiency levels (Baker, 2012, Galaczi et al., 2011). To classify pronunciation skills, graders rely on scales mainly based on the assumption of a uniform or *natural* accent (Harding, 2017). The use of the words *natural* or *normal* to describe pronunciation often causes confusion between assessors, especially when these are L2 speakers of the language they are marking (Harding, 2017). In various studies on the assessor's experience using common frameworks for grading L2 proficiency, the subjects often feel disoriented as their individual experience with different accents make them ask for clarification regarding the descriptors describing a correct or erroneous pronunciation (Harding, 2017, Kuiken and Vedder, 2014, Wei and Llosa, 2015).

Assessors point out that grading rubrics using descriptors such as *natural*, *clear* or *normal*, allow personal interpretations of what is a proficient L2 accent (Harding, 2017). Said descriptors for different levels of speech proficiency used in formal L2 PA do not come from unified assessment criteria. A global authority for language learning or assessment does not exist; however, a guideline developed by the Council of Europe is being used worldwide and keeps updating to better achieve multilingualism in Europe. The Common European Framework of Reference for Languages: Learning, teaching, assessment (CEFR) was designed as a template for teaching syllabi and examinations for L2 learning in Europe (Council of Europe. Council for Cultural Co-operation. Education Committee. Modern Languages Division (Strasbourg), 2001).

Among the different tools designed for the CEFR, the most relevant for this work are the definition of the levels of L2 proficiency along with their descriptors. The CEFR uses six levels to describe proficiency in a language, from the lowest to the most skilled the levels are: A1, A2, B1, B2, C1 and C2. The CEFR levels were introduced using a set of descriptors for communicative tasks a speaker could accomplish at a particular level. The descriptors for particular competence are task oriented. In the original CEFR published in 2001, the framework assesses pronunciation using the illustrative scale called *Phonological control* shown in Table 2.1. In this scale, the term *presence of a foreign accent* is used to describe the less proficient levels; on the other hand, the acknowledgement of an accent is replaced with descriptors for intonation and stress in speech for the more proficient levels. Annotators have declared that this sudden change of criteria in the scales adds inconsistency to the scores (Harding, 2017).

The CEFR keeps updating to include new research, particularly new descriptors offered by language teaching institutions. In the most recent CEFR companion volume published in 2020 (Soproni, 2020), the Phonological Control descriptors have been replaced completely as it were the least successful of the original descriptors and the progression they reflect was deemed *unrealistic*. The 2020 CEFR acknowledges that aiming to imitate an ideal L1 accent limits pronunciation teaching. The new Phonological Control scale focuses on the articulation of phonemes, prosody, *accentedness* and intelligibility. The higher proficiency levels in the 2020 Phonological Control scale now include the perceived influence of other languages in pronunciation, with the condition of not affecting intelligibility.

There is no unique way to measure intelligibility as it is often defined as *the extent to which an acoustic signal, generated by a speaker, can be correctly recovered by a listener* (Kent et al., 1989).

Table 2.1: Criteria for the assessment of phonological control as described initially for the Common European Framework of Reference for Languages. The different levels of proficiency go from A1 being the lowest up to C2 being the most skilled. Table reproduced from (Council of Europe. Council for Cultural Co-operation. Education Committee. Modern Languages Division (Strasbourg), 2001).

	PHONOLOGICAL CONTROL
C2	As C1
C1	Can vary intonation and place sentence stress correctly in order to express finer shades of meaning.
B2	Has acquired a clear, natural, pronunciation and intonation
B1	Pronunciation is clearly intelligible even if a foreign accent is sometimes evident and occasional mispronunciations occur.
A2	Pronunciation is generally clear enough to be understood despite a noticeable foreign accent, but conversational partners will need to ask for repetition from time to time.
A1	Pronunciation of a very limited repertoire of learned words and phrases can be understood with some effort by native speakers used to dealing with speakers of his/her language group.

Intelligibility assessment is often carried as orthographic transcriptions of phonemes meant to produce an existing word (Xue et al., 2021). Word-level intelligibility can also be measured on single-syllable rhyming words of the form consonant-vowel-consonant, i.e., minimal pairs (Holmes, 2001). However, transcriptions for word intelligibility are both costly, and time-consuming and often show low inter-assessor agreement. Therefore, numerical scales are used for how intelligible an utterance is perceived. Such scales include but are not limited to, the Likert scale and the mean opinion score (see Section 5.2.2). The discussion of whether transcriptions or numerical scales offer better inter-assessor reliability continues (Xue et al., 2021).

The new descriptors in the CEFR describe the ability of the speaker of producing the required sounds based on the perceived familiarity with phonological features and the exploitation of prosodic features. This change of focus is important as pronunciation assessors usually mention the vagueness of rubrics when describing different levels of L2 speakers (Baker, 2012, Harding, 2017, Kuiken and Vedder, 2014, Wei and Llosa, 2015). Evaluations such as TOEFL or IELTS are showing a shift of focus for PA like the CEFR. Both TOEFL and IELTS entities released part, if not all, of their research and the effect of the assessor bias in PA has been acknowledged. The influence of accents in the final decisions of the assessors is often mentioned in published studies about assessment methods and criteria (Ockey and French, 2016, Seedhouse and Satar, 2021). However, pronunciation holds a different relevance in the IELTS and TOEFL. Cambridge English Language Assessment, co-owner of IELTS, has considered pronunciation part of the criteria for L2 assessment way before the publication of the CEFR. In contrast, the TOEFL only considered pronunciation when the internet-based TOEFL was launched in 2005 (Isaacs et al., 2015). The pronunciation scale from IELTS is part

Table 2.2: Public version of the IELTS Pronunciation descriptors for each of the 9 proficiency bands with 9 being the most proficient (IELTS, 2019).

Band	Pronunciation
9	• uses a full range of pronunciation features with precision and subtlety • sustains flexible use of features throughout • is effortless to understand
8	• uses a wide range of pronunciation features • sustains flexible use of features, with only occasional lapses • is easy to understand throughout; L1 accent has minimal effect on intelligibility
7	• shows all the positive features of Band 6 and some, but not all, of the positive features of Band 8
6	• uses a range of pronunciation features with mixed control • shows some effective use of features but this is not sustained • can generally be understood throughout, though mispronunciation of individual words or sounds reduces clarity at times
5	• shows all the positive features of Band 4 and some, but not all, of the positive features of Band 6
4	• uses a limited range of pronunciation features • attempts to control features but lapses are frequent • mispronunciations are frequent and cause some difficulty for the listener
3	• shows some of the features of Band 2 and some, but not all, of the positive features of Band 4
2	• speech is often unintelligible
1	• no communication possible • no rateable language

of the speaking assessment criteria using the descriptors shown in Table 2.2.

It is complicated to gain access to the actual rubrics used by assessors of authoritative L2 certifications, although some public versions exist to aid test candidates prior to the evaluation. Considering the descriptors in Table 2.2 released to the public, the definition of *limited*, *wide* and *full* range of pronunciation features may be insufficient for the self-assessment of an IELTS candidate. The assessors' version of the pronunciation descriptors includes descriptors for rhythm, stress, the intonation of both words and phonemes, and *chunking*, which refers to a combination of pauses and stressed words in sentences (IELTS Australia, 2019). However, bands 3, 5, and 7 are defined vaguely in relation to other bands (Isaacs et al., 2015). In the case of the TOEFL, the published rubric for the speaking tasks uses four score levels with four dimensions: General Description, Delivery, Language Use, and Topic Development (ETS, 2014). The dimension called Delivery is the one with descriptors for pronunciation and has no mention of the effects or traces of the speaker's L1. The descriptors used in Delivery focus more on intelligibility and fluency, while pronunciation is paired with intonation and its description for the different score levels is based on the amount of difficulty the listener requires to fully understand the speaker (ETS, 2014, Kang and Ginther). For example, the descriptor used in Delivery for the highest score level is:

Generally well-paced flow (fluid expression). Speech is clear. It may include minor lapses, or minor difficulties with pronunciation or intonation patterns, which do not affect overall intelligibility

, while the descriptor of Delivery for the second highest level is:

Speech is generally clear, with some fluidity of expression, though minor difficulties with pronunciation, intonation, or pacing are noticeable and may require listener effort at times (though overall intelligibility is not significantly affected).

It usually remains unclear, even for trained assessors, how to label L2 speech proficiency if there are no serious obstacles to understanding the speaker (Harding, 2017, Isaacs et al., 2015). For aspects of language such as grammar and orthography, there are rules that are consistent, while PA remains considerably subjective. To capture this attitude towards speech, the Likert scale is widely used (Harding, 2017). Consider the Likert scale as a more elementary scoring scale than the scoring bands in Table 2.2, with a range of numbers to represent two extremes of an opinion, which in this context could be a *correct* or an *incorrect* pronunciation. This scale captures both the direction and intensity of an opinion towards an event or idea and can represent it as a more discrete metric (Albaum, 1997). The resolution of the scale is defined by the intermediate values used, which also can influence the behaviour of the user. An important characteristic of this scale is that people tend to avoid extreme values making it complicated for the scale to report on its entire resolution and making it subject to the individual interpretation of the assessors (Albaum, 1997, Kuiken and Vedder, 2014).

Since uncertainty in assessment criteria allows a greater influence of the assessor's bias towards a speaker, authors have expressed the need for reducing this effect. Some of the suggestions for improving the consistency of PA mentioned by trained assessors include a more explicit focus on comprehensibility, a revision of the effects of using Likert scales and the use of less ambiguous descriptors for defining each individual level of L2 proficiency (Harding, 2017, Isaacs et al., 2015, Kuiken and Vedder, 2014). Ambiguity in the descriptors could be solved with the proper training; nonetheless, variation in assessment from different professional examiners over the same observation and even a shift in speech features they focus on during the evaluation still occurs when using the latest evaluation rubrics (Inoue et al., 2021, Yates et al., 2011).

2.3 Acoustic Modelling

A widely used approach for CAPA consists of detecting phoneme mispronunciations (Arora et al., 2018, Chu et al., 2020, Dudy et al., 2018, Huang et al., 2017a, Laborde et al., 2016, Song et al., 2010, Witt and Young, 2000). Recall phonemes are an ideal representation of meaningful sounds, which are perceived from a continuous acoustic signal. To treat phonemes as categories differentiable from one another, multiple assumptions are required. The most important assumption, or maybe a *misinterpretation*, is the one of the phoneme being a consistent acoustic phenomenon (Moore and Skidmore, 2019). Some phonemes may be more consistent than others, as a difference in the number of corresponding allophones. Phonemes with a reduced variation are easy to represent and classify, yet this is not the case for phonemes causing disagreement across assessors.

A parametric model could be built to represent a phoneme class; however, the model will also be subject to the same perception bias imprinted in the data chosen. The reference data used for phoneme modelling could consist of speech from L1 speakers, L2 speakers, or both. The selection of said data and its further labelling as either *correct* or *incorrect* examples replicates the bias from all human intervention in the process (Dudy et al., 2018). Regardless of the assessment consisting of a binary classification of phoneme classes or a larger speech example, the bias implied by the reference remains.

Before getting into detail about how CAPA is implemented, it is important to introduce how speech models are built, particularly for creating a pronunciation reference. The speech signal can be characterised as a sequence of observations, each following a continuous distribution. Multiple models capable of describing the temporal aspects and variable nature of speech exist and can be used to infer the content of an utterance. In simple terms, any Acoustic Model (AM) for speech is trained to estimate the most likely sequence transcription $\mathcal{W}$ of an utterance $\mathbf{O}$. Therefore for a set of models each with its own set of parameters $\mathbf{\Lambda}$ and associated $\mathcal{W}$, the most likely sequence is found by solving:

$$\hat{\mathbf{\Lambda}} = \arg\max_{\mathbf{\Lambda}} p(\mathbf{O}|\mathcal{W}, \mathbf{\Lambda}) \tag{2.1}$$

The sequence $\mathcal{W}$ often corresponds to sub-word units, say a phoneme, to ease the difficulty of creating a recording example for every possible utterance (Holmes, 2001). Acoustic phoneme modelling, although useful for Automatic Speech Recognition (ASR), could also bring up multiple contradictions when used to score L2 pronunciation for correctness, as mentioned in Section 2.4.2.

Models of speech acoustics are essential in any speech-related application. Particularly for CAPA, the difference between an AM for L1 pronunciation and one AM for L2 pronunciation could be associated with human scores on PA. In current-day ASR, a conventional AM uses a Hidden Markov Model (HMM) to learn sequential information and a generative model to learn the Probability Density Function (PDF) of the observations in the speech signal (Yu and Deng, 2016). As new developments in ML come to light, new methods for AM and CAPA appear as well.

Deep Neural Networks (DNNs) have proven useful for improving the performance of ASR when these were used to learn the observations' PDF in an HMM-based AM (Hinton et al., 2012). Nowadays, entire AMs can be built using only DNNs designed for sequential modelling. This section provides a summary of the HMM-based conventional AM as well as other AMs based on Artificial Neural Networks (ANNs) considered relevant for this work.

2.3.1 HMM based Speech Modelling

The combination of a HMM with generative model results in the ideal to learn both temporal aspects and the variable nature of speech. For this, speech is seen as a Markov chain, a stochastic sequential event in which the probability of the following step depends only on the current state in the sequence (Yu and Deng, 2016). The HMM represents a finite sequence $\mathbf{q}$ of T states s^j, where $j = \{1, 2, \ldots, N\}$. The sequence $\mathbf{q} = \{q_1, q_2, \ldots, q_T\}$ is defined by the state transition probabilities a^{ij}. Hence, the probability to reach state q_j from q_i is represented as:

$$a^{ij} = p(q_t = s^j | q_{t-1} = s^i) \tag{2.2}$$

Each state s^i in the Markov chain is associated with the observation distribution $p(\mathbf{o}^t|s^i)$ for the observed spectral or frequency domains in an element in the signal $\mathbf{O}^T = \{\mathbf{o}^t; t = 1, \ldots, T\}$ at time t. If $\mathbf{o}^t$ belongs to a continuous probability distribution, a PDF is used to characterise each state in the HMM. The most common generative model to represent the observable PDF is a multivariate Gaussian Mixture Model (GMM). Therefore, the observation probability for state i is defined as:

$$b_i(\mathbf{o}^t) = p(\mathbf{o}^t|s^i) \tag{2.3}$$

$$b_i(\mathbf{o}^t) = \sum_{m=1}^{M} \frac{c_{i,m}}{(2\pi)^{D/2}|\mathbf{\Sigma}_{i,m}|^{1/2}} \exp[-\frac{1}{2}(\mathbf{o}^t - \boldsymbol{\mu}_{i,m})^\top \mathbf{\Sigma}_{i,m}^{-1}(\mathbf{o}^t - \boldsymbol{\mu}_{i,m})] \tag{2.4}$$

The parameters for the GMM with M components shown in Equation (2.4) are the scalar mixture weights $c_{i,m}$, the component mean vectors $\boldsymbol{\mu}_{i,m}$ and the covariance square matrices $\mathbf{\Sigma}_{i,m}$.

Recall from Equation (2.1), the HMM parameters which maximise the likelihood of the observed utterance, indicate the most likely transcription sequence $\mathcal{W}$. The computation of the likelihood of a finite T length state sequence $\mathbf{q}^T$ and the observable sequence $\mathbf{O}^T$ given an HMM with parameters set $\mathbf{\Lambda}_m$ is found from the observation and transition probabilities.

$$p(\mathbf{O}^T, \mathbf{q}^T|\mathbf{\Lambda}_m) = p(\mathbf{O}^T|\mathbf{q}^T, \mathbf{\Lambda}_m)p(\mathbf{q}^T|\mathbf{\Lambda}_m) \tag{2.5}$$

The transition probability assumes any state q^t depends only on q^{t-1}; this is the Markov assumption shown in Equation (2.2). The Markov chain also uses an initial state q^0 and a final state q^{T+1} with no observation distribution; hence the transition probability is defined as:

$$p(\mathbf{q}^T|\mathbf{\Lambda}_m) = p(q^1|q^0)\prod_{t=1}^{T} p(q^{t+1}|q^t) \tag{2.6}$$

The observation probability assumes conditional independence between observations. Each element in $\mathbf{O}^T$ depends only on the current state q^t and its correspondent PDF. The observation likelihood for sequence $\mathbf{O}^T$ is shown in Equation (2.7) as the product of the individual observation probabilities.

$$p(\mathbf{O}^T|\mathbf{q}^T,\mathbf{\Lambda}_m) = \prod_{t=1}^{T} b_{q^t}(\mathbf{o}^t) \tag{2.7}$$

2.3.2 The EM algorithm

It is complicated to maximise the likelihood of models with hidden random variables, such as the state sequence of an HMM. The Expectation-Maximization (EM) algorithm is an efficient method to learn latent variables from data in an iterative manner. The main problem consists in estimating an unknown parameter θ, to maximise the log-likelihood of an observed utterance $\log p(\mathbf{O};\theta)$.

Consider a latent variable $\mathbf{h}$ used to help explain the *complete* data $\mathbf{y} = \{\mathbf{O},\mathbf{h}\}$, such that the PDF is easier to express in closed form (Yu and Deng, 2016). The use of the latent $\mathbf{h}$ implies that the observed $\mathbf{O}$ is insufficient to explain $\mathbf{y}$. The problem now is that regardless of $\mathbf{h}$, $\mathbf{y}$ is not available, making it impossible to estimate $\log p(\mathbf{y};\theta)$ directly.

Assume a good estimate for θ is found. Therefore, the expected value E for $\log p(\mathbf{y};\theta)$ conditioned on $\mathbf{O}$ and θ is

$$\mathcal{Q}(\theta|\theta_{\mathbf{O}}) \quad = E_{h|o}[\log p(\mathbf{y};\theta)|\mathbf{O};\theta] \tag{2.8}$$

$$\mathcal{Q}(\theta|\theta_{\mathbf{O}}) \quad = E[\log p(\mathbf{O},\mathbf{h};\theta)|\mathbf{O};\theta], \tag{2.9}$$

where θ is the next best estimate and $\theta_{\mathbf{O}}$ is the current estimate given the observed data. For the case of $\mathbf{h}$ being a continuous distribution, Equation (2.9) becomes

$$\mathcal{Q}(\theta|\theta_{\mathbf{O}}) = \int p(\mathbf{h}|\mathbf{O};\theta_{\mathbf{O}}) \log p(\mathbf{O},\mathbf{h};\theta) d\mathbf{h} \tag{2.10}$$

The EM algorithm up to Equation (2.9) corresponds to the E-step. The following M-step consists in finding the parameters θ that maximize the function $\mathcal{Q}(\theta|\theta_{\mathbf{O}})$, meaning:

$$\theta = \arg\max_{\theta} \mathcal{Q}(\theta|\theta_{\mathbf{O}}) \tag{2.11}$$

A series of E and M steps are guaranteed to increase the likelihood. The EM iterations are repeated until the model convergences in a local maximum.

Limitations of the HMM as a generative model and the EM algorithm cause weaknesses on the HMM-based AM. For example, the selection of the data used for training the model is arbitrary; hence only a local maximum likelihood dependent on the observed data is expected. As with most iterative processes, the performance of the model is subject to the initial estimate $\theta_{\mathbf{O}}$. Although the Markov assumption simplifies the model of speech, it also weakens longer temporal dependencies. The independent assumptions of the GMMs used for each HMM state have also pushed for their replacement with more realistic, temporally correlated dynamic systems (Yu and Deng, 2016).

2.3.3 Deep Learning in Acoustic Modelling

The rise in popularity of ANNs in the late 1980s reflected the integration of ANNs into ASR. At the time, ANNs could not model speech signals directly, as they worked on fixed-length inputs. However, a framework of HMM-ANN models managed to combine the temporal features of the Markov chain and the advantages of ANNs for classification (Trentin and Gori, 2001, Yu and Deng, 2016).

The HMM-ANN framework used a stack of two layers of logistic regression models. The stack configuration is known as a FFN. The FFN is used to estimate the observation PDFs $b_i(\mathbf{o}^t)$ (Equation 2.3) (Bishop and Nasrabadi, 2006). Initially, the HMM-ANN models slightly outperformed the conventional HMM-GMM in some tasks. More significant improvements on ASR would come along with the development of ANNs such as the ability to train models with more layers and capable of greater context modelling. Recurrent Neural Networks (RNNs) would later be capable of modelling temporal dependencies longer than the ones allowed in a Markov chain.

Feed-Forward Networks

The FFN consists of a sequence of functions with parameters θ arranged as layers which jointly learn the function $f^*(x;\theta) = y$. A network with L layers can be written as

$$y = f^{(L)}(f^{(\dots)}(f^{(2)}(f^{(1)}(x)))) \tag{2.12}$$

, where $f^{(1)}$ corresponds to the function of the first layer, $f^{(2)}$ to the second layer, and so on. The number of layers determines the depth of the model. The first layer in a network is known as the input layer, while the output layer is the L layer, yielding y. The remaining layers between the input and output layers are called the hidden layers, as no clear meaning can be obtained from their outputs (Goodfellow et al., 2016).

Each layer consists of a fix amount of parallel units each performing the linear operation

$$f(x;w,b) = \mathbf{x}^\top \mathbf{w} + \mathbf{b} \tag{2.13}$$

, where the input $\mathbf{x}$ is multiplied by a set of linear weights w and shifted by a scalar bias parameter b. The units in layer l can be grouped in the same linear operation by stacking the corresponding weights and biases in the matrix $\mathbf{W}^{(l)}$ and vector $\mathbf{b}^{(l)}$ respectively. In the Equation (2.14) for a layer l, $\mathbf{z}^{(l)}$ is the logit output of the layer l and $\mathbf{h}^{(l-1)}$ the output, or *hidden* state, of the previous layer.

$$\mathbf{z}^{(l)}(\mathbf{h}^{(l-1)}) = \mathbf{h}^{(l-1)^\top} \mathbf{W}^{(l)} + \mathbf{b}^{(l)} \tag{2.14}$$

A non-linear transformation ϕ is applied element-wise to each $\mathbf{z}^{(l)}$ such as

$$\mathbf{h}^{(l)} = \phi(\mathbf{z}^{(l)}(\mathbf{h}^{(l-1)})) \tag{2.15}$$

Table 2.3: Commonly used activation functions.

Name	Function
ReLU	$\phi(x) = \max\{0, x\}$
Sigmoid	$\phi(x) = \frac{1}{1+e^{-x}}$
Tanh	$\phi(x) = \frac{1-e^{-x}}{1+e^{-x}}$
Softmax	$\phi(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$

The transformation ϕ is also known as the activation function since it defines the value of every unit in each layer. Function ϕ maps the logits within a certain range, making the model easy to train using the back-propagation algorithm (Rumelhart et al., 1986a). The output layer often uses an activation function different to the rest of the layers in the model. Some of the most used activation functions are shown in Table 2.3. The choice of activation function depends on the mapping of the logits required by the task performed by the layer. For example, the Softmax function maps the k outputs of a layer into a probability distribution over a discrete variable with k possible values, while the Rectifier Linear Unit (RELU) function is mainly used to allow consistent and larger gradients through active units during training as explained in the following section.

Optimization via Gradient-Descent

Most of the algorithms for training ANNs are based on the optimization of a function which depends on the model parameters θ and the observed data $\mathbf{X}$. The goal is to find the parameters θ^* that maximise or minimise a function $\mathcal{L}(\mathbf{X};\theta)$, called *loss* or *objective* function.

The loss function provides an overall measure of a loss in the approximation to the real $y = f(x)$ learned by the ANN. Multiple examples of a loss function exist, yet they serve the same purpose of measuring how close the ANN gets towards a desired function or criteria.

Since both the parameters θ and the loss function output are real numbers, the derivative $D_\theta\mathcal{L}(\mathbf{X};\theta)$ with respect to any point θ can be found. The derivative indicates the magnitude and direction of change in the loss function given a change in θ. On the assumption that $\mathcal{L}(\mathbf{X};\theta)$ is convex, for a small enough scalar ϵ,

$$\mathcal{L}(\mathbf{X};\theta - \epsilon \operatorname{sign}(D_\theta\mathcal{L}(\mathbf{X};\theta))) < \mathcal{L}(\mathbf{X};\theta) \tag{2.16}$$

The optimization of a function given small changes in the opposite side of its derivative is called Gradient-Descent (Goodfellow et al., 2016). The model parameters θ can be learned via error-backpropagation (Rumelhart et al., 1986b). For this, recall the derivative $D_\theta\mathcal{L}(\mathbf{X};\theta)$ can be defined with respect to any particular parameter in the model via the chain rule. The chain rule explains a given rate of change as a product of two or more related rates of change.

For example, $D_\theta \mathcal{L}(\mathbf{X};\theta)$ with respect to the l-th layer parameters $\theta^{(l)}$ in an FFN is shown in Equation (2.17), where $\phi^{(j)}$ corresponds to the non-linear transformation at layer j and $z^{(j)}$ is the affine transformation at layer j.

$$\frac{\partial \mathcal{L}(\mathbf{X};\theta)}{\partial \theta^{(l)}} = \left(\frac{\partial \mathcal{L}(\mathbf{X};\theta)}{\partial \phi^{(L)}}\right)\left(\frac{\partial \phi^{(L)}}{\partial z^{(L)}}\right)\left(\frac{\partial z^{(L)}}{\partial \phi^{(L-1)}}\right)\left(\frac{\partial \phi^{(L-1)}}{\partial z^{(L-1)}}\right)$$
$$\cdots\left(\frac{\partial \phi^{(l)}}{\partial z^{(l)}}\right)\left(\frac{\partial z^{(l)}}{\partial \theta^{(l)}}\right) \tag{2.17}$$

The derivative $D_\theta \mathcal{L}(\mathbf{X};\theta)$ is used to create a gradient at time t averaged across an observation sample of size M. Therefore, a parameter $\theta^{(l)}$ can be updated as

$$\theta^{(l)}_{t+1} \leftarrow \theta^{(l)}_{t} - \epsilon \frac{1}{M} \sum_{m=1}^{M} D_{\theta^{(l)}_t} \mathcal{L}(\mathbf{X};\theta) \tag{2.18}$$

The reduction of the loss function will ideally stop when $D_\theta \mathcal{L}(\mathbf{X};\theta) = 0$, meaning a critical point in the function was reached. A critical point could refer to a local minimum, a local maximum, or a saddle point which is neither a maximum nor minimum in a function. Gradient-Descent does not guarantee a global-optimal value for $\mathcal{L}(\mathbf{X};\theta)$.

Gradient-Descent allows training ANNs which, given enough layers and hidden units, perform as universal approximators. In practice, the performance of a trained network depends on the nature of real-world data and the limitations of the technology available. Therefore, ANNs have been modified to better model a problem in the way information flows internally. This is the case for using ANNs for sequential modellings, such as required in speech. The following section presents ANNs for learning temporal dependencies by implementing recurrent connections.

Recurrent Networks

As mentioned before, FFNs lack the ability to model temporal dependencies. When the HMM-ANN framework started being used, FFNs were limited in the complexity of the functions they could learn, since only models with a small number of layers could be properly trained.

The HMM-ANN framework is useful for modelling speech; however, the Markovian and observation independence assumptions still limit the models to relatively short temporal dependencies. The creation and development of RNNs aims to sort out problems of sequence modelling. These networks are called *recurrent* from using cyclical directed connections between internal units, which are technically layers on their own. The recurrence created by said connections is time-delayed and creates an effect of memory, as explained in this section (Yu and Deng, 2016).

A basic RNN layer performs an affine transformation over an input $\mathbf{x}^{(t)}$ at a time point t, followed by a non-linear transformation σ similar to an FFN layer defined in Equation (2.13). Equation (2.19) defines the hidden state of the RNN layer at time t, $\mathbf{h}^{(t)}$, where $\mathbf{W}_h$ and $\mathbf{R}_h$ are weight matrices, and $\mathbf{b}_h$ is a linear bias vector. Notice the previous hidden state $\mathbf{h}^{(t-1)}$ is an

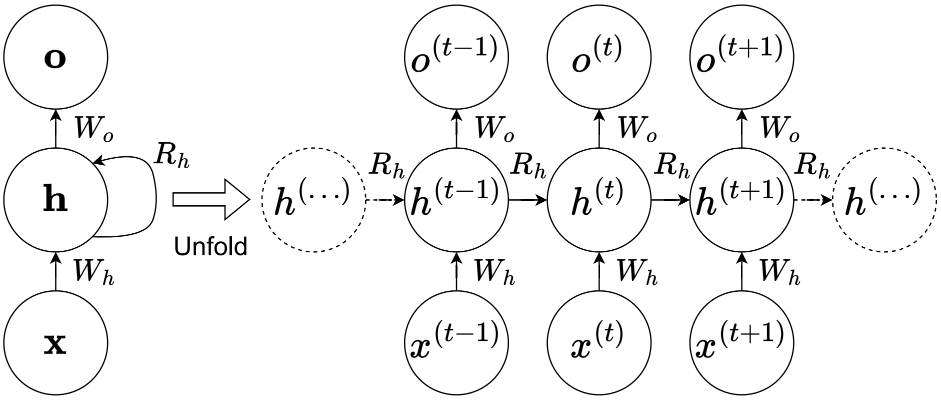


Figure 2.1: Computational graph for a RNN. Based on (Goodfellow et al., 2016).

input additional to $\mathbf{x}^{(t)}$. The non-linearity σ is often a *Tanh* or a RELU function. An additional layer shown in Equation (2.20) estimates the output at time t, $\mathbf{o}^{(t)}$ from $\mathbf{h}^{(t)}$, the weight matrix $\mathbf{W}_o$ and the bias vector $\mathbf{b}_o$. The computational graph for the RNN is shown in Figure 2.1. The unfolded graph on the right side shows the information flow through time.

$$\mathbf{h}^{(t)} = \sigma(\mathbf{W}_h^\top \mathbf{x}^{(t)} + \mathbf{R}_h^\top \mathbf{h}^{(t-1)} + \mathbf{b}_h) \tag{2.19}$$

$$\mathbf{o}^{(t)} = g(\mathbf{W}_o^\top \mathbf{h}^{(t)} + \mathbf{b}_o) \tag{2.20}$$

The recurrent connection in Equation (2.19) only observes the previous hidden state, somewhat similar to the Markovian assumption. In practice, the basic RNN cannot look back far into temporally extended patterns or perform well on detecting events separated by long time windows (Yu and Deng, 2016). The problem would be solved by implementing gating mechanisms defined as a Hadamard product to control the flow of information within the network. Said gates allow the network to *remember* or *forget*, resulting in the Long Short-Term Memory (LSTM).

Long Short-Term Memory

First introduced in (Hochreiter and Schmidhuber, 1997), LSTM aims to solve the vanishing gradient problem typical of RNNs when learning long-term relationships in a data element. The sequential nature of speech makes this model optimal for capturing the context in which acoustic events are associated with phonemes.

An LSTM model is defined as a set of recurrently connected subnetworks referred to as *memory blocks*. Each block is meant to maintain its state over time t and regulate the flow of information by controlling a set of gating functions (Van Houdt et al., 2020). The computation process, i.e., forward-pass, of the LSTM block updates each of the subnetworks and inputs as follows. First, the block input $\mathbf{z}^{(t)}$ combines the current input $\mathbf{x}^{(t)}$ and the previous output $\mathbf{y}^{(t-1)}$ of the same block. The combination is carried out via a linear transformation using the weight matrices $\mathbf{W}_z$, $\mathbf{R}_z$, and the bias weight vector $\mathbf{b}_z$. A regularisation function g, generally *tanh*, is used to obtain $\mathbf{z}^{(t)}$ as shown in Equation (2.21).

$$\mathbf{z}^{(t)} = g(\mathbf{W}_z\mathbf{x}^{(t)} + \mathbf{R}_z\mathbf{y}^{(t-1)} + \mathbf{b}_z) \tag{2.21}$$

The input gate $\mathbf{i}^{(t)}$ combines $\mathbf{x}^{(t)}$ and $\mathbf{y}^{(t-1)}$ with the previous cell value $\mathbf{c}^{(t-1)}$. Equation (2.22) defines $\mathbf{i}^{(t)}$ as the output of a transformation using the associated matrices $\mathbf{W}_i$ and $\mathbf{R}_i$, a pointwise product ($\odot$) between weights $\mathbf{p}_i$ and value $\mathbf{c}^{(t-1)}$ and the bias weight $\mathbf{b}_i$. The *sigmoid* function σ is used for regularisation this time. The resulting $\mathbf{i}^{(t)}$ controls which values of $\mathbf{z}^{(t)}$ could contribute to the current cell value $\mathbf{c}^{(t)}$.

$$\mathbf{i}^{(t)} = \sigma(\mathbf{W}_i\mathbf{x}^{(t)} + \mathbf{R}_i\mathbf{y}^{(t-1)} + \mathbf{p}_i \odot \mathbf{c}^{(t-1)} + \mathbf{b}_i) \tag{2.22}$$

The *forget* gate $\mathbf{f}^{(t)}$ is analogous to $\mathbf{i}^{(t)}$, with the difference of $\mathbf{f}^{(t)}$ selecting which values should be discarded from $\mathbf{c}^{(t-1)}$. Equation (2.23) shows the computation of $\mathbf{f}^{(t)}$, where $\mathbf{W}_f$ and $\mathbf{R}_f$ are weight matrices, $\mathbf{p}_f$ is multiplied point-wise with $\mathbf{c}^{(t-1)}$ and $\mathbf{b}_f$ is a bias vector. The *sigmoid* function is also used for $\mathbf{f}^{(t)}$.

$$\mathbf{f}^{(t)} = \sigma(\mathbf{W}_f x^{(t)} + \mathbf{R}_f\mathbf{y}^{(t-1)} + \mathbf{p}_f \odot \mathbf{c}^{(t-1)} + \mathbf{b}_f) \tag{2.23}$$

The input and forget gates control the contributions of $\mathbf{z}^{(t)}$ and $\mathbf{c}^{(t-1)}$ for the current $\mathbf{c}^{(t)}$ as:

$$\mathbf{c}^{(t)} = \mathbf{z}^{(t)} \odot \mathbf{i}^{(t)} + \mathbf{c}^{(t-1)} \odot \mathbf{f}^{(t)} \tag{2.24}$$

A final output gate $\mathbf{o}^{(t)}$ is used to generate the block output $\mathbf{y}^{(t)}$ from the cell state $\mathbf{c}^{(t)}$. The operation to obtain $\mathbf{o}^{(t)}$ is similar to the ones of the other gates, with the difference of it acting on the current $\mathbf{c}^{(t)}$ rather than the previous one. Equation (2.25) uses the weight matrices $\mathbf{W}_o$ and $\mathbf{R}_o$ associated to $\mathbf{x}^{(t)}$ and $\mathbf{y}^{(t-1)}$ respectively, the weight vector $\mathbf{p}_o$ for a point-wise multiplication with $\mathbf{c}^{(t)}$ and the bias vector $\mathbf{b}_o$. Finally, $\mathbf{y}^{(t)}$ is obtained by multiplying $\mathbf{o}^{(t)}$ point-wise with the $\mathbf{c}^{(t)}$ value after passing it through the *tanh* regularisation function as shown in Equation (2.26). The computational graph for a single LSTM cell is shown in Figure 2.2. The rectangles in the graph correspond to layer-wise operations, and the circles correspond to element-wise operations.

$$\mathbf{o}^{(t)} = \sigma(\mathbf{W}_o x^{(t)} + \mathbf{R}_o\mathbf{y}^{(t-1)} + \mathbf{p}_o \odot \mathbf{c}^{(t)} + \mathbf{b}_o) \tag{2.25}$$

$$\mathbf{y}^{(t)} = g(\mathbf{c}^{(t)}) \odot \mathbf{o}^{(t)} \tag{2.26}$$

The different subnetworks in the LSTM cell allow the preservation of relevant information and the disposal of less important features given the input pattern. The memory gates also improve the training of the LSTM as they ensure the loss gets propagated from the output to the memory subnetworks.

The recurrent connections in the LSTM do not need to be unidirectional, meaning the network can process a sequence both back and forward. The bidirectional property of RNNs is exploited using two simultaneous hidden units, one for each direction. The transformations

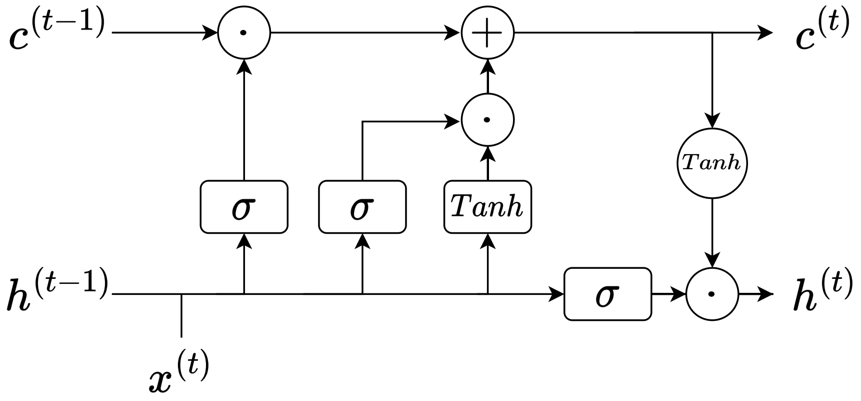


Figure 2.2: Computational graph for a LSTM cell.

for the different directions are often expressed with overset arrows to indicate either a forward (→) or a backward (←) pass. The BDLSTM (Graves and Schmidhuber, 2005) outperformed unidirectional RNNs on tasks of speech recognition, handwriting recognition, and keyword spotting, among others (Yu and Deng, 2016).

The combination of recurrent connections, LSTM and bidirectional sequence processing would change the standards of acoustic modelling and ASR. However, the assumptions about speech and L2 pronunciation already mentioned in this chapter remain.

2.4 Computer-Assisted Pronunciation Assessment

The use of computers for teaching, practising and the assessment of L2 speech has kept increasing since the last few decades (Chapelle and Chung, 2010, Chapelle and Voss, 2016, Eskenazi, 2009, Golonka et al., 2014, Isaacs and Harding, 2017). Computer Assisted Language Learning (CALL) has been used to create guidelines for learning more suprasegmental features of L2, as well as vocabulary and grammar. Regarding pronunciation, the more noticeable advances happened until recently. Although pronunciation is considered essential for comprehensible speech (Neri et al., 2002), pronunciation training has been overlooked until the end of the late 20th century. It used to be considered that getting rid of the accent of a L2 speaker was neither possible nor necessary (Neri et al., 2002). Regardless of the unnecessary goal of making a L2 speaker sound exactly like a native speaker, achieving a clear differentiation between minimal pairs in the target language is crucial for L2 speakers (Lindemann, 2017).

The use of computers for teaching and assessing more *receptive* language tasks such as reading comprehension, listening, and the acquisition of vocabulary has been found to be beneficial and sometimes more effective than traditional L2 teaching (Golonka et al., 2014). One must consider the task a language learner needs to perform for PA, as it defines what speech aspects are going to be assessed and which technology is required to do so. Consider for example, if a speaker performs a highly interactive task such as keeping up a dialogue or just the pronunciation of a list of selected words or sentences. For tasks that require more *active* language skills, such as sustaining conversations, it can represent a greater technological challenge (Dodigovic, 2009). The use of ASR in any CALL needs to be done carefully due to assumptions on how L2 acquisition occurs.

A common approach for the automation of PA consists of first identifying the content of an utterance in order to compare it to a reference using a metric to allocate the speaker's performance in a scale (Huang et al., 2017a, Kanters et al., 2009, Neri et al., 2002, Witt and Young, 2000). The reference could consist of task-related speech examples marked by an assessor given specific criteria. The metric chosen to judge a speaker can come from either a single or a multidimensional analysis of the speaker's performance.

In early attempts for integrating ASR into CALL, the systems did not achieve satisfactory results, mostly because the ASR systems used were originally designed to work with L1 speakers (Litman et al., 2018). A system for ASR uses various sources of knowledge, the most common being the AM, the lexicon, and the language model. The models comprising an ASR represent a particular speaker sample; this choice of sources influences the performance of the final product, as the intended users might not fit the data domain the ASR was originally built for (Litman et al., 2018). A model trained only on ideal L1 pronunciation will have difficulties recognizing L2 speech or utterances with a considerable amount of phoneme repetitions, substitutions, and deletions.

The use of CALL is not limited to measuring performance; it should also provide the test subject with adequate feedback to help them improve (O'Brien et al., 2018). Most *errors* in L2 pronunciation remain unnoticed by the speaker until they are pointed out in the form of feedback; otherwise, it is unlikely that a L2 student could learn on their own an adequate phoneme differentiation. The likelihood of a speaker achieving a more functional L2 phoneme realization is heavily influenced by the quality of the feedback and how it is delivered. For the case of students learning from a L2 tutor who is not a L1 speaker, any accent from the teacher will likely be replicated by the students as this is the main reference available.

Computer-assisted teaching and assessment of L2 overlap in methods. Assessment could even be considered part of learning, as it is necessary for generating feedback for the student to learn and improve. In the case of pronunciation, the same questions and biases over a *good* pronunciation mentioned in Section 2.2 remain in any automation attempt of the task. To build a good CAPA tool, the definition of all processing steps, inputs, outputs, and interpretations of the results are crucial. For pronunciation, it is obvious that the input must be speech. However, to limit the input to just the sound wave might be insufficient to perform PA without even thinking in solving the problem caused by the bias. Similarly, in what concerns a feedback output, acoustic representations via spectrograms or even a plot of the sound wave have been found to be difficult to interpret by language students (Luo, 2016, Neri et al., 2002). To map acoustic features into a proficiency scale and then generate meaningful feedback is not trivial, therefore, many strategies remain available to be explored and tested to build useful tools for language learning and assessment.

As mentioned in Section 2.2.1, there are multiple aspects of pronunciation used in the descriptors of the proficiency bands for L2 speech. CAPA requires finding out which segmental and supra-segmental features of speech correlate to different levels of pronunciation proficiency (Suzukida and Saito, 2022). Not all assessors even focus on the same speech features (Yates et al., 2011). Therefore, there is not unique way to try to replicate a human assessor of L2 speech.

Publications about speech technology used for L2 PA often keep the focus on the similarity between a L2 speaker and an arbitrary L1 reference. The likelihood-based scores for PA are perhaps the most published and one of the earliest CAPA frameworks (Chen and Li, 2016). In (Kim et al., 1997), it was found that HMM-based log-posterior probabilities for phoneme segments averaged across multiple utterances of a speaker, strongly correlated with human scores for speech proficiency. In essence, likelihood-based CAPA measures the difference between the likelihood of a canonical phoneme sequence and the likelihood of a free phoneme recognition. The more similar a speaker gets to the reference; the likelihood gap closes. The following section explains the standard algorithm for likelihood-based CAPA.

2.4.1 Goodness of Pronunciation Algorithm

The GOP algorithm (Witt and Young, 2000) is a well-established method for detecting mispronunciations at the phoneme level which remains widely used for research in L2 CAPA (Arora et al., 2018, Chu et al., 2020, Duan and Chen, 2020, Dudy et al., 2018, Huang et al., 2017a, Laborde et al., 2016, Song et al., 2010). The GOP is a likelihood-based method which generates a score for how likely a phoneme has been mispronounced. Before the GOP, word and sentence-based CAPA existed and were deemed to be heavily text dependent. Pronunciation *correctness* scores for full sentences were obtained using ASR. Although there were approaches for detecting mispronunciations at the phoneme level before, these were based on the recognition of phonemes different from a reference without being related to human judgement (Eskenazi, 1996, Witt and Young, 2000). The main contribution of the GOP algorithm was a score for the similarity between L1 and L2 pronunciation while incorporating *tolerance* thresholds from human assessors.

In its original form, GOP is a score for the *quality* of pronunciation equivalent to the posterior probability of an expected phone q being present in a corresponding acoustic segment $O^{(q)}$. The likelihood $p(O^{(q)}|q)$ is computed using a set of previously trained HMMs with GMMs, assuming the transcription of the acoustic segment is known. The GOP score for any phoneme p from a set Q is then defined as the length normalized log-posterior $P(p|O^{(p)})$ as shown in Equations 2.27 and 2.28, where $NF(p)$ is the length of $O^{(q)}$ in frames.

$$\mathbf{GOP}(p) \equiv \frac{1}{NF(p)} \left| \log(P(p|O^{(p)})) \right| \tag{2.27}$$

$$\mathbf{GOP}(p) = \frac{1}{NF(p)} \left| \log\left(\frac{p(O^{(p)}|p)P(p)}{\sum_{q \in Q} p(O^{(p)}|q)P(q)} \right) \right| \tag{2.28}$$

Additional assumptions were used to simplify and estimate the GOP score. First, all Q phonemes are considered equally likely. Next, the denominator in Equation (2.28) is approximated by the largest contributor, i.e., the phoneme q with the highest likelihood as shown in Equation (2.29).

$$\mathbf{GOP}(p) = \frac{1}{NF(p)} \left| \log \left(\frac{p(O^{(p)}|p)P(p)}{\max_{q \in Q} p(O^{(p)}|q)P(q)} \right) \right| \tag{2.29}$$

The use of the maximum phoneme likelihood as the denominator for $\mathbf{GOP}(p)$ turns the score into a log-likelihood ratio between the expected phoneme p and the most likely phoneme q. If $q = p$ the GOP equals 0 meaning a small GOP indicates a high similarity to the pronunciation reference for phoneme p. The segment lengths and the likelihoods required for GOP are obtained using Viterbi alignments (Forney, 1973). The phoneme based HMMs are trained on L1 data and have their Gaussian means adapted to L2 speech using Maximum Likelihood Linear Regression.

The phoneme-specific thresholds for declaring a mispronunciation are obtained from GOP score statistics on the training data. For the phoneme p with GOP score mean μ_p and variance σ_p^2, the threshold T_p is defined as

$$T_p = \mu_p + \alpha\sigma_p^2 + \beta \tag{2.30}$$

, where α and β are coefficients empirically determined to better match the human annotation. The authors of GOP claim the errors from the HMMs can be reduced by averaging the GOP scores.

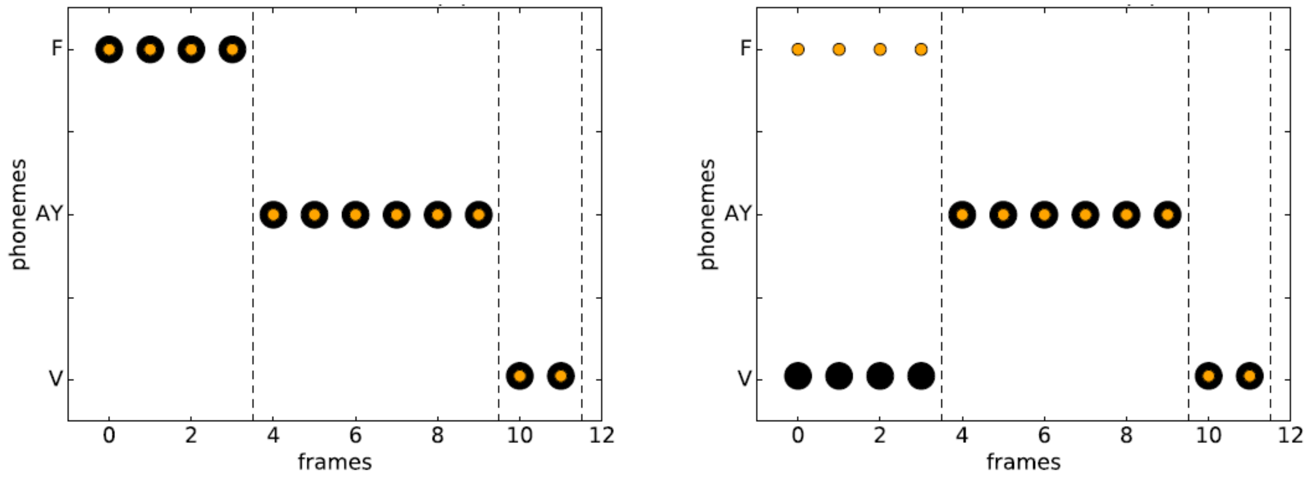
When GOP was first published, all the assumptions used to simplify the computation of GOP yielded improvements in the performance of the score. Said simplifications could have been necessary due to the technology available at the time; nonetheless, multiple authors kept working on GOP and implemented further adjustments for its improvement (Chen et al., 2019, Chu et al., 2020, Huang et al., 2017a,b, Lin and Wang, 2021, Lin et al., 2020, Shi et al., 2020, Song et al., 2010, Sudhakara et al., 2019a,b, Wu et al., 2012, Yang, 2015). Regardless of how realistic or not the original GOP might be, the algorithm pushed research in the direction of replicating human assessment of phoneme segments.

2.4.2 The Alignment Problem

An implicit assumption necessary for the GOP is the one of a precise phoneme segmentation. Although phones can be identified and associated with phonemes, a frame-level segmentation required for GOP turns problematic. The GOP strongly depends on the segment boundaries, which define the identity of each frame. The likelihood contributions in Equation (2.28) will produce different ratios given the alignment. For the case of L2 speech, the variability in pronunciation adds difficulty to the task. If the speakers assessed are children students of L2, it does not get any easier (Dudy et al., 2018, Witt and Young, 2000).

In (Dudy et al., 2018), the alignment problem is skilfully illustrated using four cases of the computation of Equation (2.29) for the pronunciation of the word *five*:

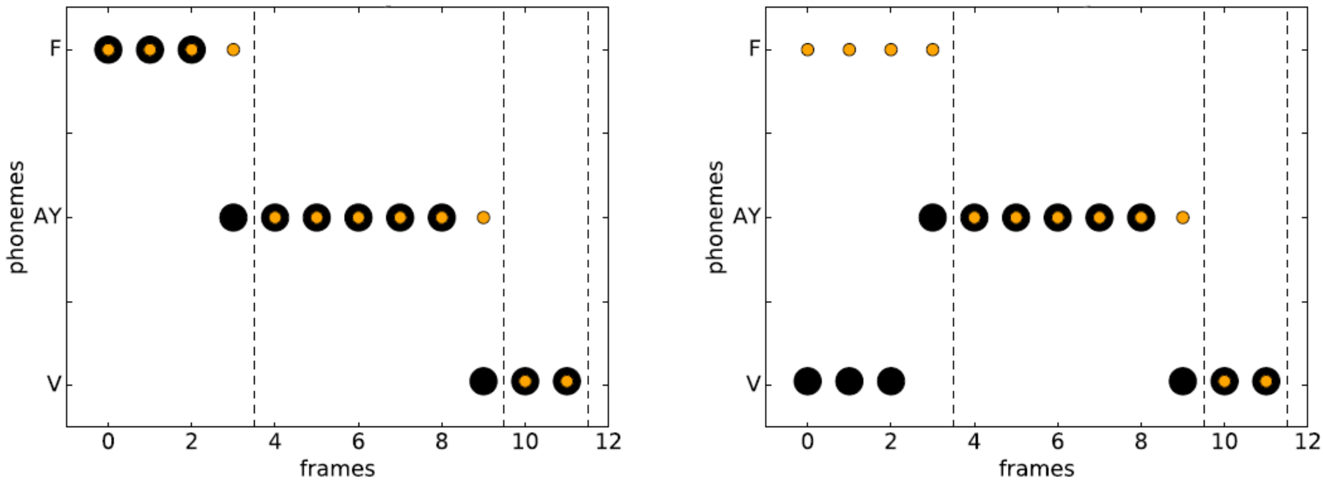
- **Case 1**: The speaker correctly pronounces the word and the alignment fits the path with the highest likelihood (Figure 2.3, plot *a*).
- **Case 2**: The speaker mispronounces the word as *vive* and the alignment fits the path with the highest likelihood (Figure 2.3, plot *b*).
- **Case 3**: The speaker correctly pronounces the word and the alignment does not fit the path with the highest likelihood (Figure 2.4, plot *a*).
- **Case 4**: The speaker mispronounces the word as *vive* and the alignment does not fit the path with the highest likelihood (Figure 2.4, plot *b*).



(a) Case 1

(b) Case 2

Figure 2.3: Representation of the segmentation in (a) with the phoneme boundaries marked with a dotted line. Case 1 is shown in (b) with solid black circles marking the maximum likelihood path. Case 2 is shown in (c) with its corresponding maximum likelihood path. Plots reproduced from (Dudy et al., 2018).



(a) Case 3

(b) Case 4

Figure 2.4: Representation of Case 3 in (a) and Case 4 in (b). Phoneme segmentation is marked with yellow dots. The maximum likelihood path is marked with solid black circles. Plots reproduced from (Dudy et al., 2018).

The plots in Figure 2.3 illustrate the alignment for cases 1 and 2, while Figure 2.4 does it for cases 3 and 4. The plots show the forced alignment for the phonemes on the y-axis and the acoustic frames on the x-axis. The phoneme segment boundaries are marked with vertical dotted lines. The maximum likelihood paths are indicated by solid black circles. The yellow dots indicate the frames contributing to the numerator of Equation (2.29); i.e., the frames for the expected phoneme p. The path of highest likelihood in Case 1 matches the yellow dots, meaning a small GOP score for the phonemes F, AY, and V. For Case 2, the phoneme sequence V-AY-V shows the highest likelihood. Since the initial phoneme F in Case 2 is not part of the most likely path, the GOP score should exceed the threshold value to declare the segment mispronounced. Cases 3 and 4 in Figure 2.4 show phoneme boundaries not matching the segments marked by the solid black circles, affecting the likelihood ratio for GOP. The mismatch in alignment is often the case for non-standard pronunciation as it could be the one of a child, an L2 speaker, or a person with a speech disorder (Dudy et al., 2018).

The performance of GOP and any alignment-based method for detecting mispronunciation rely strongly on the performance of the alignment. The implementation of GOP needs a precise alignment, as the algorithm assumes that the observations corresponding to a well-defined phoneme class are available. The erroneous labelling of speech frames brings noise to a metric with the resolution of the GOP. Modifications for the GOP have been developed ever since to better tackle the alignment problem and improve its robustness. The Lattice GOP (Song et al., 2010) for example, allows the maximum phoneme class on each frame to contribute to the denominator easing the mismatch between phoneme boundaries and the most likely path as seen in Cases 3 and 4. Another alternative is to learn the most likely pronunciation errors and include them in the GOP score in the form of Extended Recognition Networks (ERNs) (Arora et al., 2018, Chu et al., 2020), weighted phoneme confidence scores (Doremalen et al., 2013) and L2 adaptation of the ASR used in the alignment (Huang et al., 2017a, Witt and Young, 2000)

The original GOP assumes a phoneme is always present and remains consistent for its entire duration. Such views oversimplify L2 speech, particularly speech from early learners. As shown in Figures 2.3 and 2.4, the definition of precise start and end points for phonemes yields a noisy metric for CAPA.

2.4.3 Other Likelihood-Based Methods in CAPA

Not all likelihood-based CAPA is GOP, even if the assessment is performed on phoneme level. In (Nicolao et al., 2015) for example, a language student is considered proficient if they replicate the same phoneme sequence $\mathbf{P}_t$ a teacher does for a word $\mathbf{W}_t$. Using a recording from the student $\mathbf{O}_s$ and a recording from the teacher $\mathbf{O}_t$, the probability that a student mimics the teacher's reference is expressed as the probability of the teacher's pronunciation is a good predictor of the student utterance:

$$p(\mathbf{O}_s|\mathbf{O}_t) = \frac{p(\mathbf{O}_s, \mathbf{W}_t, \mathbf{P}_t, \mathbf{O}_t)}{p(\mathbf{W}_t, \mathbf{P}_t, \mathbf{O}_t)} \tag{2.31}$$

It is expected that both the word and phoneme sequence of the student and teacher are the same. $P(\mathbf{W}_t, \mathbf{P}_t, \mathbf{O}_t)$ only depends on the utterance $\mathbf{O}_t$. Therefore,

$$\frac{p(\mathbf{O}_s, \mathbf{W}_t, \mathbf{P}_t, \mathbf{O}_t)}{p(\mathbf{W}_t, \mathbf{P}_t, \mathbf{O}_t)} \propto p(\mathbf{O}_s, \mathbf{W}_t|\mathbf{P}_t)p(\mathbf{P}_t) \tag{2.32}$$

The prior for phoneme sequence $\mathbf{P}_t = \{r_t^i\}$ is assumed constant. With the help of alignment information, $p(\mathbf{O}_s, \mathbf{W}_t|\mathbf{P}_t)$ can be computed as the product of phoneme paired segments:

$$p(\mathbf{O}_s, \mathbf{O}_t|\mathbf{P}_t)p(\mathbf{P}_t) = \prod_i p(\mathbf{O}_s^i, \mathbf{O}_t^i|r_t^i) \tag{2.33}$$

Each acoustic phoneme segment is interpolated to a fixed length L. Each element of the product in Equation (2.33) is modelled as a phoneme specific GMM. For a paired segment $\mathbf{O}^i = [\mathbf{O}_s^i, \mathbf{O}_t^i]$, it is considered correctly pronounced (C = Correct) when the following is true:

$$\frac{p(\mathbf{O}^i|C = \text{Correct})}{p(\mathbf{O}^i|C = \text{Error})} > \frac{p(C = \text{Correct})}{1 - p(C = \text{Correct})} \tag{2.34}$$

The ratio in Equation (2.34) eases part of the alignment problem (Section 2.4.2). Assuming the utterance $\mathbf{O}_t$ comes from the same person who annotated the data, the ratio should be a good representation of the reference held by the assessor.

2.4.4 Classification-Based Framework

A limitation of the GOP and any other likelihood-based CAPA is that it cannot identify the identity of the mispronunciation (Chen and Li, 2016). To overcome this problem, a classifier-based framework can identify the identity of an acoustic example. However, this poses the challenge of labelling the true identity of the sounds. Additionally, mispronunciations are not highly frequent, resulting in quite unbalanced labels. For example, the Business Language Testing Service (BULATS) corpus of L2 speech collected by Cambridge Assessment English shows mispronunciations only in 9.7% out of 61,722 manually annotated words (Kyriakopoulos et al., 2020).

There is a reduced number of publications which aim to classify specific pronunciation errors particular to a L2 accent. In (Truong et al., 2004), three decision tree classifiers were built for the correct detection of phonemes /ɑ/, /ʏ/ and /x/, respectively. Each of the three classifiers was trained to detect a correct pronunciation of the chosen phoneme from a single type of phoneme substitution, meaning each classifier made a binary decision between two phonemes which presented confusion to L2 speakers of Dutch. All the phoneme classes used for building the classifiers were selected due to their relatively higher count in the data

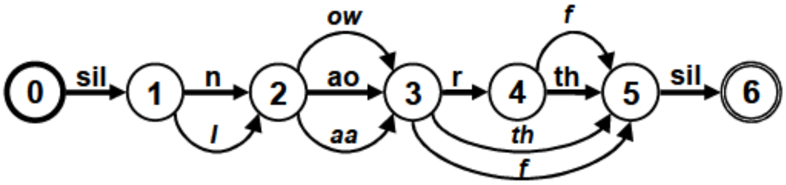


Figure 2.5: Example of an ERN for different pronunciations of the word *North*. The canonical pronunciation is indicated by the path of straight arrows. Figure reproduced from (Lo et al., 2010)

available.

Unsupervised pronunciation error discovery has been used to sort the lack of annotated speech from L2 learners (Chen and Li, 2016). Common mispronunciations can be modelled using an ERN, a data-driven sequence representation that allows the ASR to identify variations in pronunciation (Lo et al., 2010). Any change to a canonical sequence could be considered a mispronunciation, or it may not even be noticed by the annotators. A benefit of ERNs is that mispronunciations can be both detected and defined. Knowledge about the actual mispronunciation can be used to give feedback to the speaker.

The classification of speakers into groups or proficiency bands is an alternative to classifying the phoneme identity of the utterances. It can be as simple as classifying whole utterances into a five-band proficiency scale using only an FFN trained using i-vectors (Dehak et al., 2010) as done in (Takai et al., 2020). The broader the classes and the scope of analysis, the less information a model for L2 CAPA can offer.

2.4.5 End-to-End ASR for PA

An alternative to scoring pronunciation using a precise phoneme alignment is to make free-phoneme recognition on an utterance for then compare it against a phoneme sequence serving as a reference; in simple words, an ASR for L2 speech. A free-phoneme recognition eliminates the need for precise alignment. On the assumption of knowing the true phoneme sequence, it should be easy to detect when an L2 speaker produces a mispronunciation. The original GOP (Section 2.4.1) has a free recognition stage, which affects the denominator in Equation (2.29). However, the decision of correct pronunciation for GOP is based on the likelihood ratio between the expected and the most likely phoneme.

The recognition of non-standard speech, e.g., L2 speech, children's speech, and pathological speech remains a challenging task (Dudy et al., 2018). An ASR for mispronunciation detection must not be confused with robust accented ASR. The intention is to catch the variations in pronunciation rather than generalise them out. Mispronunciations occur sparsely and sporadically, hence the lack of examples for ASR to train on (O'Brien et al., 2018, Shi et al., 2020). However, dynamic models such as HMMs and finite state transducers have been used to model phoneme sequences observed in L2 speech (Arora et al., 2018, Chu et al., 2020). As RNNs became widely used in acoustic modelling, these were also used for mispronunciation detection (Chen et al., 2018, Feng et al., 2020, Fu et al., 2021, Lin and Wang, 2021, Zhang et al., 2020).

State-of-the-art ASR uses considerably deep ANNs capable of sequence modelling to infer labels directly from a speech segment. The use of a single ANN trained on all the required steps to obtain a final desired output directly from the observation is often called End-to-End (E2E). A combination of RNNs and attention mechanisms (Bahdanau et al., 2015) are trained to map a sequence of speech features $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ into a sequence $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_S\}$, where $S \leq T$. An usual way to train an ANN to do sequence decoding is via the Connectionist Temporal Classification algorithm (CTC) (Graves et al., 2006).

An ANN with parameters θ trained on CTC uses a softmax output (See Table 2.3) for each possible label $\mathbf{L}$ and one *blank* symbol, representing a change in the sequence or no label at all. The space of $\mathbf{Z}$ is the same as $\mathbf{L}$ plus the *blank* symbol. For every instance $\mathbf{x}_t$ at time t, the network outputs a vector $\mathbf{y}_t$ containing the probability for all possible labels in the set $\mathbf{L}'^T = \{\mathbf{L} \cup \mathit{blank}\}$ of T sequences. The probability of a path π in $\mathbf{L}'^T$ is then:

$$p(\pi|\mathbf{X}) = \prod_{t \in T} \mathbf{y}_t^{\pi_t} \in \mathbf{L}'^T \tag{2.35}$$

A map $\mathcal{B} : \mathbf{L}'^T \mapsto \mathbf{L}^{\leq T}$ is used to eliminate any repeated label and *blank* symbol in π. The frame-alignment is practically discarded at this stage as the label sequence $\mathbf{l}$ is sufficient information about the content of $\mathbf{X}$. The conditional probability of a label sequence $\mathbf{l} \in \mathbf{L}^{\leq T}$ is obtained by adding the probabilities of all its corresponding paths:

$$p(\mathbf{l}|\mathbf{X}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} p(\pi|\mathbf{X}) \tag{2.36}$$

Equation 2.36 can be approximated using the Viterbi algorithm (Forney, 1973), as done for HMMs. Hence, the ANN parameters θ can be trained via Backpropagation (Rumelhart et al., 1986b) to maximize the likelihood:

$$\mathcal{L}((\mathbf{X}, \mathbf{Z}), \theta) = - \sum_{(\mathbf{X}, \mathbf{Z})} \log(p(\mathbf{Z}|\mathbf{X}, \theta)) \tag{2.37}$$

Note the explicit independence between y_t over time in Equation (2.35). Therefore, the ANN trained with CTC needs constraints for adequate decoding. For example, an RNN-based language model is often used during CTC training (Sak et al., 2015). The RNN can learn a wider context than HMMs and alleviate the assumption of independence between outputs y_t.

The ANNs trained with CTC could combine different modules given the task at hand. For the case of CAPA, in (Zhang et al., 2020) an E2E network is trained for phoneme recognition of L2 speech. The architecture consists of a BDLSTM encoder, a CTC decoder, and an attention-based decoder. The attention decoder combines content and location attention mechanisms. The content attention mechanism scores each element in a representation $\mathbf{Y}$ generated by the encoder and then normalizes the scores to generate the attention weights $\boldsymbol{\alpha}$ (Chorowski et al., 2015). In (Zhang et al., 2020), the attention decoder consists on an RNN which outputs the character or phoneme sequence $\mathbf{c} = \{c_1, c_2, \ldots, c_L\}$. To output the character c_l, first the attention mechanism obtains the energy $e_{l,t}$ using the decoder's previous hidden state h_{l-1} as shown in Equation (2.38). The symbols ω, W, V and b in Equation (2.38) are all weight matrices. The $e_{l,t}$ is then normalized over l to obtain the attention weights in Equation (2.39).

$$e_{l,t} = \omega^{\top} \tanh(Wh_{l-1} + V\mathrm{Y}_{l,t} + b) \tag{2.38}$$

$$\alpha_{l,t} = \frac{\exp(e_{l,t})}{\sum_l (e_{l,t})} \tag{2.39}$$

A shortcoming of the content attention is that similar events are weighted the same regardless of their location in time, hence time information about the events needs to be included (Chorowski et al., 2015). The previous attention weights $\boldsymbol{\alpha}_{t-1}$ are convoluted $(*)$ with a weight matrix F to generate vector f_t as shown in Equation (2.40). The vectors f_t are included in Equation (2.38) with an additional weight matrix U to model time information about events in $\mathbf{Y}$. The new attention component with both content and location information is shown in Equation (2.41).

$$f_t = F * \boldsymbol{\alpha}_{t-1} \tag{2.40}$$

$$e_{l,t} = \omega^{\top} \tanh(Wh_{l-1} + V\mathrm{Y}_{l,t} + Uf_{l,t} + b) \tag{2.41}$$

The weights $\boldsymbol{\alpha}$ serve as a *soft alignment* between output c_l and $\mathbf{Y}_t$. The RNN decoder uses the previous output character c_{l-1}, its previous hidden state vector $\mathbf{h}_{l-1}$ and the hidden representation $\mathbf{d}_l$ obtained from the weighted sum of $\mathbf{Y}_t$ shown in Equation (2.42), also referred to as the *context* vector.

$$\mathbf{d}_l = \sum_t \boldsymbol{\alpha}_{l,t} \mathbf{Y}_t \tag{2.42}$$

The network in (Zhang et al., 2020) uses both CTC and attention decoders to estimate $P(c_t | c_{t-1}, c_{t-2}, \ldots, c_1, \mathbf{X})$. The network is trained using a multi-objective function $\mathcal{L}_{mult}$ combining the CTC loss $\mathcal{L}_{ctc}$ and the attention-based decoder $\mathcal{L}_{att}$ balanced by a hyperparameter ϱ as shown in Equation (2.43).

$$\mathcal{L}_{mult} = \varrho \mathcal{L}_{ctc} + (1 - \varrho) \mathcal{L}_{att} \tag{2.43}$$

The main objective of the E2E ASR is to generate sequence labels; however, more information and constraints are added to the model to improve its performance at detecting mispronunciation. In (Zheng et al., 2021), in addition to a CTC-based decoder previously trained, the model then trains two more classifiers based on Transformer (Vaswani et al., 2017) and Feed-Forward layers. The classifiers output a phoneme label and a binary *error state* label respectively. The *error state* classifier uses the output of the phoneme classifier and a previously assembled context vector as an input. The losses for the decoder and two classifiers are combined in an empirically weighted sum to use as a multi-objective loss function without any interference on the original CTC decoder, nor the other way around. The outputs of the *error state* classifier in (Zheng et al., 2021) are used as safety checks for the CTC decoder. If the decoder outputs the same phoneme label as the reference, yet the *error state* posterior is greater than 0.5, it is considered the decoding is not reliable.

An additional technique for E2E systems for mispronunciation detection is to include an encoding of the reference phoneme sequence (Feng et al., 2020, Fu et al., 2021, Lo et al., 2020, Zheng et al., 2021). All the different layers combinations used for encoding, processing, and further decoding of the identity of the phonemes uttered by the speaker are not as relevant as the annotation of the L2 data itself. As discussed more extensively in (O'Brien et al., 2018), an important problem is the scarcity of annotated mispronunciation examples. It is also common practice to first train an AM or E2E ASR on exemplar L1 speech, then to fine-tune it using L2 data which might not show the same level of annotation, the same recording conditions, speaker population and so on. Therefore, the data annotation format is also a relevant factor to consider.

2.5 Data Annotation Formats

A constant problem for any of the CAPA frameworks mentioned so far is the definition of the reference. The building of an arbitrary ideal pronunciation reference is often not even considered as a source of noise and bias. The scarcity of annotated speech from L2 learners and the sparsity of mispronunciations constrain the construction of models for L2 CAPA (O'Brien et al., 2018). In (Cucchiarini et al., 1998) the question of what is a L2 accent brings up the need for some sort of *anti-model* for L1 pronunciations.

The selection of a pronunciation reference needs to be trustworthy; hence *expert* annotators are asked to provide both phonetic transcription and an assessment of correctness. The need for larger amounts of data to build deep models with large amounts of parameters calls for many annotators. It is at this moment when subjectivity in PA is evidenced by how much the annotators agree with each other (Loukina et al., 2015).

When professional annotators are trained to follow the same reference, it is expected to see an agreement in 90% of the cases (Chambers and Ingham, 2011). However, the real agreement reported for various data corpus of L2 learners'speech differs from the ideal level of inter-assessor agreement. A corpus of English as L2 spoken by L1 speakers of Japanese was presented in (Franco et al., 2010). A team of 7 L1 speakers of american English annotated the overall *pronunciation quality* of 4,652 sentences using a scale ranging from 1 to 5. The reported correlation coefficient was $r = 0.8$. The ISLE speech corpus, on the other side, reported an agreement of 64% in the error location from 5 annotators (Bonaventura et al., 2000). The maximum level of inter-assessor agreement in a corpus serves as an upper bound in the performance of a model built using such data.

It is better for the modelling of disagreement to infer it from many annotations for the same observation. A common observation serves as a reference point for comparing multiple annotation references and finding a common ground for a minimum approval criterion. Usually, when more than one label is collected for each observation, the disagreements are relabelled as the average of the annotated values or as the label decided by most annotators (Loukina et al., 2015). The new consolidated annotation is held as an ideal reference or at least one that represents agreement the best. However, the cap on the real levels of assessor agreement remains in the reference. A pronunciation model trained on consolidated reference would be affected by ambiguous cases in which different labels were assigned to similar observations. Consolidated references also treat all levels of disagreement equally, meaning a model would be trained to miss-classify observations (Loukina et al., 2015).

Numerous annotators could reduce the levels of disagreement and the number of ambiguous cases. Many professionals come with a high price; therefore, an alternative is to rely on a larger number of non-professional annotators instead. *Crowdsourced annotation* allows the collection of multiple judgements via online platforms from people who are not necessarily professionals. Crowdsourced annotation has been proven useful for highly subjective labelling tasks like the detection of grammar errors and phonetic transcription (Loukina et al., 2015).

Another argument in favour of crowdsourcing annotation for L2 PA is that the phoneme-specific annotation for pronunciation errors results in numerous false positives. A way to reduce false positives in L2 PA data is to ask the assessors to focus only on pronunciations affecting intelligibility, adding more subjectivity to the task. The large number of annotators via crowdsourcing often reach a higher agreement than a reduced number of professionals (Loukina et al., 2015). A validation method is required to eliminate inconsistent annotators and reduce noise in the labels. An ASR hypothesis or example cases marked by a professional are often used to compare against non-professional annotators (Loukina et al., 2015, Van Dalen et al., 2015). Done right, crowdsourcing annotation does not differ largely from the views of professional annotators (Loukina et al., 2015).

A reduction in the joint annotation of different assessors increases the difficulty of modelling the individual bias. Instead of using examples to infer the bias, the common factor used as a reference across assessors needs to scale up. A similarity measure for unsupervised data selection (Park et al., 2022) allows the comparison of observations which do not overlap. Similar examples could be phoneme representations, speech from the same speaker, or even a similar representation of a speaker. Multiple formats of speaker representation could be used, say, speaker embeddings (Sztahó et al., 2019) or speaker-pairs similarity scores (Saito et al., 2021).

The worst-case scenario for modelling the bias as well as for the hope of training a consistent and fair model for L2 CAPA is having a single annotation per observation. Either the annotation comes from a single person or multiple annotators took part in it. With no available information on the number of participants or which examples each worked on, there is no easy way to infer the annotation bias. The bias is defined in part by contrast with other judgements. Without any fairground to compare, an analysis of the consistency of the annotation corresponds to intrapersonal variability for the case of a single speaker. Different resolutions can be used to observe the consistency of the annotation, say at phoneme level, speaker level, word level, or accents or other criteria for grouping speakers like accent, rate of speech or proficiency level in L2 speech.

2.5.1 Available Corpora for L2 PA

The collection of L2 speech data and further annotation for mispronunciation is complicated due to many reasons listed in this chapter. However, there are still various data corpora available. Most of the corpora are task-oriented; for example, read speech, conversational speech, interviews, and so on. The speaker population is also a relevant factor when designing a corpus. Most corpora are collected with an adult speaker population, yet many people start learning a L2 from a young age at school.

A group of available data corpora for L2 language learning are listed below. The main features of each corpus are shown for comparison.

L2-ARCTIC

- **Language:** English
- **Speech type:** Prompted L2 speech
- **Length:** 24 Hours
- **Speakers:** 24 speakers with various L1
  - 2 Hindi
  - 2 Korean
  - 2 Mandarin
  - 2 Spanish
  - 2 Arabic
  - 2 Vietnamese
- **Annotation:** Annotation for mispronunciation at word and phoneme level from 1 of the 3 assessors.
- **Note:** The data uses the prompts from the CMU ARCTIC corpus (Kominek and Black, 2004). The L2-ARCTIC aims to benchmark algorithms for mispronunciation detection.
- **Reference:** G. Zhao, S. Sonsaat, A. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna. L2-ARCTIC: A Non-Native English Speech Corpus. Perception Sensing Instrumentation Lab, jan 2018.

JASMIN-CGN

- **Language:** Dutch
- **Speech type:** L1 and L2 Prompted and human-machine interaction speech
- **Length:** 90 hours from which 44 hours correspond to L2 speakers.

- **Speakers:** L1 and L2 Dutch children, adults and seniors. The L1s in the corpus are:
  - Dutch
  - Turkish
  - Moroccan
  - French
- **Annotation:** Orthographic transcription from 2 annotators. A L2 Dutch lexicon was obtained by reviewing an automatic phonetic transcription.
- **Note:** The corpus was created as an extension of the Spoken Dutch Corpus (Oostdijk, 2000). The objective of JASMIN-CGN was to include speech from children, L2 speakers and the elderly into a reference for contemporary spoken Dutch.
- **Reference:**
  - C. Cucchiarini, H.V. Hamme, O.v. Herwijnen, and F. Smits. Jasmin-cgn: Extension of the spoken dutch corpus with speech of elderly people, children and non-natives in the human-machine interaction modality. 2006.
  - C. Cucchiarini, J. Driesen, H.V. Hamme, and E. Sanders. Recording speech of children, non-natives and elderly people for hlt applications: the jasmin-cgn corpus. 2008.

CHILDES English-L2 Paradis Corpus

- **Language:** English
- **Speech type:** L2 conversational children speech
- **Length:** 5 rounds of 45 minutes per speaker.
- **Speakers:** 25 children with various L1
  - Mandarin
  - Farsi
  - Spanish
  - Korean
  - Japanese
  - Cantonese
  - Arabic
- **Annotation:** The data is annotated for morphemes. A failure to use a target morpheme in each obligatory context was coded as an error of either omission or commission.
- **Note:** The longitudinal nature of the corpus aims to determine the similarities and differences in acquisition patterns between monolingual and multilingual children speakers of English.
- **Reference:**J. Paradis. Grammatical morphology in children learning English as a second language. 2005.

ISLE Speech Corpus

- **Language:** English
- **Speech type:** Prompted L2 speech
- **Length:** 17 Hours
- **Speakers:** 46 teenage learners of English as L2
  - 23 German
  - 23 Italian
- **Annotation:** The recordings are marked for stress and mispronunciation at word level. A phoneme alignment was generated automatically. The alignment for each recording was reviewed by 1 of the 6 annotators available.
- **Note:** Only 2/3 of the recordings from each speaker were annotated. The annotators were L1 British English speakers.
- **Reference:** P. Bonaventura, P. Howarth, and W. Menzel. 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, 2000.

Spoken CALL Shared Task

- **Language:** English
- **Speech type:** L2 speech collected using the language learning software CALL-SLT (Rayner et al., 2010).
- **Length:** 6 Hours
- **Speakers:** Swiss German teenage students of English as L2
- **Annotation:** The recordings were transcribed and stored with an associated prompt from the speaker. Each pair of recording and prompts is marked *correct* if the meaning answers an original question presented by the software. The prompt is marked for vocabulary, grammar, and meaning.
- **Note:** The corpus is used for the CALL task challenge
- **Reference:**
  - C. Baur, J. Gerlach, E. Rayner, M. Russell, and H. Strik. A shared task for spoken call? 2016.
  - M. Qian, X. Wei, P. Jancovic, and M.J. Russell. The university of birmingham 2017 slate call shared task systems. In SLaTE, pages 91–96, 2017.

ITSLANG Corpus

- **Language:** English
- **Speech type:** Prompted L2 speech
- **Length:** 80 hours from which 6 hours are annotated at phoneme level and other 6 hours annotated at word level.

- **Speakers:** Over 230 young learners of English in the Netherlands. Not all speakers are L1 Dutch speakers.
- **Annotation:** Joint annotation from three trained phoneticians at word and phoneme level.
- **Note:** The corpus provides the individual annotations from the three raters. The annotations overlap completely.
- **Reference:** M. Nicolao, A.V. Beeston, and T. Hain. Automatic assessment of English learner pronunciation using discriminative classifiers. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5351–5355. apr 2015.

Speechocean762

- **Language:** Language
- **Speech type:** Prompted L2 speech
- **Length:** 6 hours
- **Speakers:** 250 learners of English as L2. All speaker have Mandarin as L1. The speakers'age range from young children to adults
- **Annotation:** Phoneme accuracy, word accuracy, word stress, sentence accuracy, sentence completeness, sentence fluency and sentence prosody. Each aspect is marked with a numerical value by 5 experts.
- **Note:** Each speaker recorded 20 sentences themselves with their own phone at 20cm from their mouths. The individual annotations are provided as metadata.
- **Reference:** Zhang, J., Zhang, Z., Wang, Y., Yan, Z., Song, Q., Huang, Y., ... & Wang, Y. (2021). speechocean762: An open-source non-native english speech corpus for pronunciation assessment. arXiv preprint arXiv:2104.01378.

Business Language Testing Service by Cambridge Assessment English (BULATS)

- **Language:** English
- **Speech type:** Long spontaneous utterances of L2 speech
- **Length:** 108 hours
- **Speakers:** 1075 Gujarati L1 speakers of English
- **Annotation:** ASR and Crowdsourced transcription. The speakers are scored using a numerical scale which can be binned into the CEFR scale.
- **Note:** The data is used to infer an overall performance score for the speaker rather than to identify a particular mispronunciation.
- **Reference:** Wang, Y., Gales, M. J. F., Knill, K. M., Kyriakopoulos, K., Malinin, A., van Dalen, R. C., & Rashid, M. (2018). Towards automatic assessment of spontaneous spoken English. Speech Communication, 104, 47-56.

Most of the mentioned data corpus are publicly available for academic work. However, various research publications on L2 LA and CAPA rely on private datasets, which are not usually available. The lack of a bench-marking corpus for L2 CAPA makes it difficult to compare algorithms, as corpus are usually developed with a specific task in mind.

Chapter 3

Attention-Based Method for Automatic Pronunciation Assessment

3.1 Introduction

Section 2.2.1 explains how the Common European Framework of Reference for Languages: Learning, teaching, assessment (CEFR) keeps updating the criteria about how to describe the pronunciation of a proficient Second Language (L2) speaker. In various case studies about L2 Pronunciation Assessment (PA), both naive and experienced assessors mention the importance of a clear definition of *good* and *bad* L2 pronunciation (Harding, 2017, Kuiken and Vedder, 2014, Wei and Llosa, 2015). The instruction for focusing L2 PA in *"How easy is the speaker to understand"* does not reduce the subjectivity of assessment (Loukina et al., 2015).

The competence of a speaker is often tested on well-defined language tasks such as reading comprehension and sustaining interviews about a particular topic (Dodigovic, 2009). The definition of the task directs the focus of the assessor on a particular speaking skill rather than a global assessment of proficiency. The aspect of L2 speech that concerns this work is *pronunciation correctness*. Given the reviewed materials for existing methods for Computer Assisted Pronunciation Assessment (CAPA) (Section 2.4.4), one of the most used methods for assessment is based on phoneme scores obtained from alignment. However, a particular interpretation of the phoneme influences the interpretations of metrics used for assessment. In the widely cited Goodness of Pronunciation (GOP) algorithm (Witt and Young, 2000) (see Section 2.4.1), the phoneme is assumed as a well-defined acoustic building block. As mentioned in Section 2.4.2, the definition of an acoustic example that represents a single phoneme class is not trivial, nor even guaranteed to exist. Nonetheless, a phoneme-based approach for CAPA seems useful to better learn acoustics the assessor associates with each ideal sound in their own pronunciation reference. Therefore, it is desired that an association between phoneme identities and, possibly a range, of acoustic observations is achieved.

An additional factor which seems to limit performance in various CAPA publications is the assumption of ground truth in pronunciation. The selection of a pronunciation reference is done arbitrarily. The descriptors for the different levels of speaker performance are also based on interpretations of what consists of a proficient speaker, hence a universal agreement across assessors is not usual (see Section 2.2.1). Methods for creating consolidated annotation such as majority voting or averaging over a small sample of annotators also create inconsistencies which are undesirable when training a machine learning model for CAPA (Lin and Wang, 2021, Lin et al., 2020, Zhang et al., 2020).

This chapter introduces a new method to differentiate L2 utterances labelled as either correct or incorrect given a particular pronunciation assessor. The task is carried out via the analysis of short speech segments expected to contain at least one phoneme. The proposed method relies on the probability of the assessor identifying a phoneme identity, rather than comparing a speech segment to an Acoustic Model (AM) built on exemplar Native Language (L1) speaker data. It is crucial to not assume a ground truth, as the ideal examples selected might not best represent the annotator's own pronunciation reference. Therefore, only annotated data is considered, meaning the use of prior knowledge is completely discarded.

The segment-based method for L2 CAPA presented in this section is compared against an implementation of the classic GOP algorithm (Section 2.4.1). Both implementations were tested on learning the annotation for recordings of prompted speech from young learners of English as L2. The annotation format tested consists of the annotation from each individual assessor and consolidated annotation formats as representations of agreement across annotators.

3.2 Segment-Based Approach for Mispronunciation Detection

The main limitation for CAPA based on a precise phoneme alignment is that non-canonical pronunciation is difficult to model (Dudy et al., 2018). Mispronunciations occur with a relatively low frequency and these are usually not as simple as a well-defined deletion, substitution, or insertion of a particular phoneme. Mispronunciation can also be subtle, meaning the uttered sound would not be necessarily confused with a different phoneme class. As a summary of the alignment problem explained in Section 2.4.2, in order to locate the exact time boundaries of an acoustic segment representing a single ideal phoneme and compare it to an arbitrary template, requires consistent and perfectly distinct every phoneme realisation. Said implications seem unnatural for L2 speech; therefore, it seems better to avoid such metrics based on isolated phoneme segments.

Assessors of L2 speech focus on defining which phones correspond to a mispronunciation instead of its precise location in time (Baker, 2012, Carey et al., 2011, Harding, 2017, Kartushina and Frauenfelder, 2014, Kuiken and Vedder, 2014, Wei and Llosa, 2015, Witteman et al., 2014). A pronunciation assessor does not worry about finding the start and ending times for each phoneme identified. Instead, they listen to a segment and then compare it against a reference. Therefore, a segment-based approach to perform CAPA based on short utterances seems logical.

The key difference between an alignment-based CAPA and the segmental-based approach presented here is that *it is not important when a phoneme was uttered, but which phoneme was perceived*. For this, assume a prompt w which can be defined using the phoneme sequence $\mathbf{r} = \{r_i; i = 1, \ldots, R\}$. Sequence $\mathbf{r}$ is assumed the canonical pronunciation for w. Additionally, a sequence of *correctness* labels $\mathbf{l} = \{l_i; i = 1, \ldots, R\}$ is associated with $\mathbf{r}$ by a pronunciation assessor η. In $\mathbf{l}$, $l_i = 1$ if assessor η considers that the corresponding phoneme r_i has been produced correctly; otherwise, $l_i = 0$. Therefore, for the acoustic segment $\mathbf{O}^{(w)}$ associated with a known prompt w, a correct pronunciation is declared only if all the corresponding elements l_i are equal to 1. The probability of correct pronunciation is defined in Equation (3.1), where $\mathbf{l} = \overrightarrow{1}$ represents $l_i = 1 \forall i \in \{1, \ldots, R\}$.

$$P(\mathit{Correct\ Pronunciation}|\mathbf{O}^{(w)}, \eta) = P(\mathbf{l} = \overrightarrow{1}|\mathbf{r}, \mathbf{O}^{(w)}, \eta) \tag{3.1}$$

The probability of labelling a pronunciation error in $\mathbf{O}^{(w)}$ corresponds to the probability of any r_i having a corresponding label $l_i = 0$. This is simply the complement to Equation (3.1).

$$P(\mathit{Pronunciation\ Error}|\mathbf{O}^{(w)}, \eta) = 1 - P(\mathbf{l} = \overrightarrow{1}|\mathbf{r}, \mathbf{O}^{(w)}, \eta) \tag{3.2}$$

The model is kept simple by assuming independence between phonemes. Therefore, the probability of a segment labelled as correctly pronounced can be expressed as:

$$P(\mathbf{l} = \overrightarrow{1}|\mathbf{r}, \mathbf{O}^{(w)}, \eta) = \prod_i (l_i = 1|\mathbf{r}, \mathbf{O}^{(w)}, \eta) \tag{3.3}$$

The independence between phonemes allows two types of equivalences: a focus on a given phoneme being present (Equation 3.4), or one to obtain information about a phoneme segment (Equation 3.5). The latter one, where $\mathbf{O}_i^{(w)}$ denotes the audio segment associated with phoneme r_i, is then equivalent to the GOP estimate as outlined in Section 2.4.1. Other implications of the assumption of independence were softened with the architecture chosen to implement the model as defined in Section 3.3.

$$P(l_i = 1|\mathbf{r}, \mathbf{O}^{(w)}, \eta) \equiv P(l_i = 1|r_i, \mathbf{O}^{(w)}, \eta) \tag{3.4}$$
$$P(l_i = 1|\mathbf{r}, \mathbf{O}^{(w)}, \eta) \equiv P(l_i = 1|r_i, \mathbf{O}_i^{(w)}, \eta) \tag{3.5}$$

It is assumed the sequence $\mathbf{r}$ is always known. Said assumption corresponds to a listener holding a pronunciation reference for w and reflects common tasks in language learning such as pronunciation training by repetition conducted by a teacher. It is possible that the speaker has produced a phoneme sequence $\mathbf{s} = \{s_i; i = 1, \ldots, S\}$ different from $\mathbf{r}$. The association of the perceived $\mathbf{s}$ with $\mathbf{r}$ given the assessor is not trivial in practice. The impact of $\mathbf{s}$ in Equation (3.3) is:

$$
\begin{aligned}
P(\mathbf{l} = \overrightarrow{1}|\mathbf{r}, \mathbf{O}^{(w)}, \eta) &= \sum_{\mathbf{s}} P(\mathbf{l} = \overrightarrow{1}, \mathbf{s}|\mathbf{r}, \mathbf{O}^{(w)}, \eta) && (3.6) \\
&= \sum_{\mathbf{s}} P(\mathbf{l} = \overrightarrow{1}|\mathbf{s}, \mathbf{r}, \mathbf{O}^{(w)}, \eta) P(\mathbf{s}|\mathbf{r}, \mathbf{O}^{(w)}, \eta) && (3.7) \\
&\approx \sum_{\mathbf{s}} P(\mathbf{l} = \overrightarrow{1}|\mathbf{s}, \mathbf{O}^{(w)}, \eta) P(\mathbf{s}|\mathbf{r}) && (3.8)
\end{aligned}
$$

Since the focus of this approach is to detect whether the correct phoneme was perceived by the annotator, the recognition of the uttered **s** is not necessary when estimating the correctness labels **l**. The component $P(\mathbf{s}|\mathbf{r})$ in Equation (3.8) can be seen as the speaker bias and may lead to re-weighting the model with prior information on typical errors. The model of common errors associated with a given linguistic profile, say L1, age or years of formal L2 training, is the object of continuous research on CAPA (Arora et al., 2018, Chu et al., 2020). However, the summation over all possible confusing sequences may be impractical and likely to require more data than the one available. Therefore, no previous knowledge is assumed in this model, and it is assumed the most likely confusions will be observed while learning $P(\mathbf{l} = \overrightarrow{1}|\mathbf{r}, \mathbf{O}^{(w)}, \eta)$.

Notice so far, no timing information has been required. The relationship between the labels and sequence **r** is only affected by the acoustic segment $\mathbf{O}^{(w)}$. The resulting model is implemented as a combination of sequential encoding, self-attention, and multi-label classification as outlined in the following section.

3.3 Attention-Based Segmental Incorrectness Model

The implementation of the segment-based approach for error detection assumes that humans perform PA once an entire utterance within a context has been heard, without consciously performing tasks of acoustic modelling and decoding of an unknown message. This implementation estimates $P(\mathbf{l}|\mathbf{r}, \mathbf{O}^{(w)}, \eta)$ in three stages: sequential encoding using a Bidirectional Long Short-Term Memory (BDLSTM), self-attention and segment classification using an Feed-Forward Network (FFN).

The initial BDLSTM stage aims to reduce the dependence on precise alignment boundaries and exploit acoustic long-time dependencies. Recent publications on CAPA exploit the internal memory representation of the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) (see Section 2.3.3) to build systems which can analyse entire utterances using sequential encoding. It is known the encoding of relevant acoustic and linguistic information can be used to obtain a score which can be linked to human annotation for PA (Chen et al., 2018, Fu et al., 2021). The LSTM also allows the building of audio-to-text models, which can be used to detect differences between the canonical **r** and the uttered phoneme sequence **s** (Zhang et al., 2020); however, this work avoids recognition of the uttered sequence.

Sequential encoding based on LSTM alone might not be enough to get rid of the assumption of equal gravity, meaning that all observations are equally relevant for PA. Therefore, a self-attention mechanism is attached to the LSTM. Attention is used to allocate resources to avoid information overload by using low-level features to determine potential salient regions within an observation (Niu et al., 2021). Attention mechanisms were originally designed for sequence-to-sequence tasks, such as machine translation (Bahdanau et al., 2015). However, it was found in (Milner et al., 2019) that attention benefits sequential encoding for further inferring aspects of the data from temporal dependencies.

Additive attention as defined in (Bahdanau et al., 2015) is applied on the BDLSTM hidden states $\mathbf{h}_{\mathbf{O}^{(w)}} = \{h_{o_{t_0}^{(w)}}, \ldots, h_{o_T^{(w)}}\}$ for the utterance $\mathbf{O}^{(w)}$. The self-attention mechanism (Bahdanau et al., 2015) computes the energy $e_{c,t}$ defined as

$$e_{c,t} = v_c \odot \tanh(W_c^\top h_{o_t} + V_c^\top h_{o_t} + b_c) \tag{3.9}$$

,where $\odot$ stands for the Hadamard product, v_c, W_c and V_c are weight matrices and b_c is a linear bias vector. The energy is normalised over time to calculate the attention weights $\boldsymbol{\alpha} = \{\alpha_{c,t}\}$ as

$$\alpha_{c,t} = \frac{\exp(e_{c,t})}{\sum_{k=0}^{T}(e_{c,k})} \tag{3.10}$$

The $\boldsymbol{\alpha}$ weights and states $\mathbf{h}_{\mathbf{O}^{(w)}}$ are used to get the context vector

$$\psi = \boldsymbol{\alpha} \odot \mathbf{h}_{\mathbf{O}^{(w)}} \tag{3.11}$$

A residual connection is implemented by adding $\mathbf{h}_{\mathbf{O}^{(w)}}$ to ψ easing the flow of the gradient (He et al., 2016, Vaswani et al., 2017). A normalization layer (Ba et al., 2016) and regularization via dropout ($p = 0.1$) are applied before the final classification stage. The resulting encoding $EC(\mathbf{O}^{(w)})$ is passed through an FFN to output the state of the correctness labels $\mathbf{l}$. The final output layer in the FFN changes depending on the annotation level used to infer Equation (3.3). The overall architecture for this Attention-Based Segmental Incorrectness Model (ASIM) is shown in Figure 3.1.

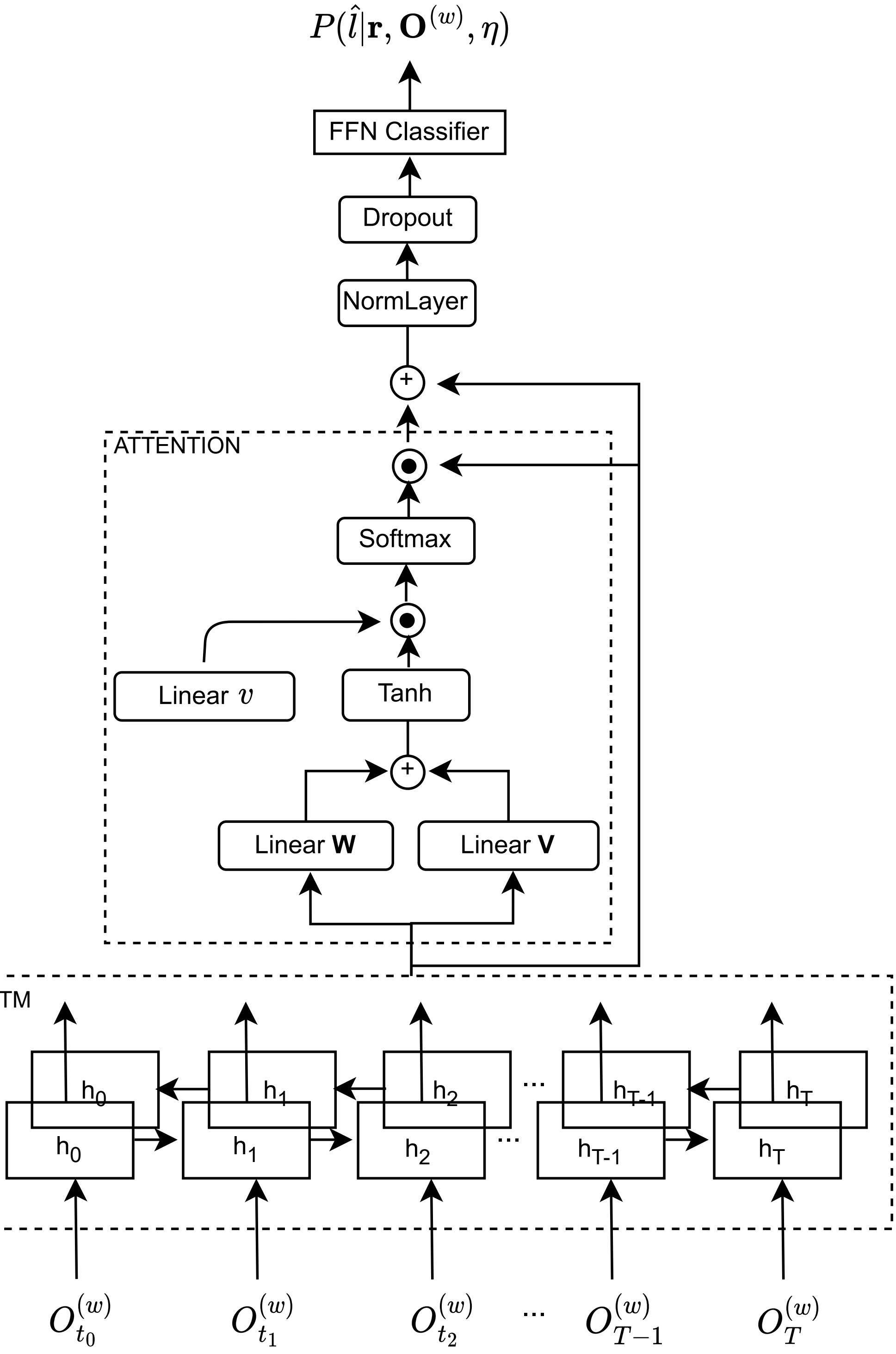


Figure 3.1: Diagram for the Attention Based model for mispronunciation detection. The acoustic input $\mathbf{O}^{(w)}$ ranges from frames t_0 to T. The phonemes $\mathbf{r}$ are used as a scoring condition rather than an actual input for the model.

3.4 Output Layer Configuration for the Scoring of Segments.

Three different configurations for the output of the FFN classifier are proposed for different interpretations of the annotations:

- *E1*: A single binary output to estimate the left side of Equation (3.2).
- *E2*: The output layer contains an output for every r_i to estimate $P(\hat{l}_i = 1|r_i, \mathbf{O}^{(w)}, \eta)$.
- *E3*: For every r_i, there are 2 outputs for estimating either a correct or incorrect pronunciation label by learning $P(\hat{l}_i|r_i, \mathbf{O}^{(w)}, \eta)$.

The *E1* layout uses no information about $\mathbf{r}$ nor w whatsoever. The entire network performs binary classification for whether segment $\mathbf{O}$ was pronounced correctly, regardless of the expected w. Configuration *E1* aims to show the benefits phonemic context has on CAPA compared to using isolated phoneme-based scores. It is assumed that there are acoustic sequences which can be labelled as incorrect given the observed occurrences in each spoken language. Therefore, a model without an explicit reference built on phoneme identities can still build an AM for correct pronunciations.

In *E2*, the model is a classifier for ideal phonemes according to the assessor. Equation (3.3) is estimated as the Log-summation of the posteriors for each l_i. The outputs for phoneme classes not expected in the phoneme sequence $\mathbf{r}$ are not considered. A small set of posteriors makes for better representative scores.

The *E3* configuration differs from *E2* by using two outputs per phoneme class, corresponding to the occurrence being marked either correct or incorrectly pronounced. *E3* allows to modelling non-canonical pronunciations with a higher precision than *E2*. The model can in this way learn to detect phoneme utterances which may not be confused with a different phoneme identity yet can still be differentiated from correct pronunciations. Like the scoring used for *E2*, the outputs associated with phonemes not in $\mathbf{r}$ are not included in the score of a correct segment computed as

$$P(\mathbf{l} = \overrightarrow{1}|\mathbf{r}, \mathbf{O}^{(w)}, \eta) \cong \frac{\sum_{i=1}^{R} P(l_i = 1|\mathbf{r}, \mathbf{O}^{(w)}, \eta)}{\sum_{i=1}^{R} P(l_i|\mathbf{r}, \mathbf{O}^{(w)}, \eta)} \tag{3.12}$$

Notice, this model does not perform speech recognition, but a direct association between $EC(\mathbf{O}^{(w)})$, and the probability of the correctness labels. The ASIM presented here does not focus on sequence recognition, as the expected phonemes defined in $\mathbf{r}$ are not even subject to sequential order. The condition $\mathbf{r}$ is only involved in the scoring and the error propagation during training.

Regardless of the output layer configuration, the model is trained to minimize the Binary Cross-Entropy (BCE) between the annotation l_i and the model θ distribution $\hat{l}_i = p_\theta(l_i|r_i, \mathbf{O}^{(w)}, \eta)$. The loss function is defined for an observation sample N as:

$$\mathrm{BCE}(\mathbf{l}, \hat{\mathbf{l}}) = -\frac{1}{N} \sum_{i=1}^{N} \left[l_i \cdot \log \hat{l}_i + (1 - l_i) \cdot \log(1 - \hat{l}_i) \right] \tag{3.13}$$

Table 3.1: Inter-annotation agreement (I) and Cohen's kappa (κ) for the assessors in INA

vs.			I	κ
a1		*a2*	0.871	0.349
a2		*a3*	0.770	0.254
a3		*a1*	0.808	0.446
a1	*a2*	*a3*	0.725	0.331

The combination of phoneme-level annotation to infer the probability of correctness for a segment $\mathbf{O}^{(w)}$ aims for a metric more robust than building models for individual phoneme classes. The ASIM constructs a pronunciation reference given the observed annotation without the help of any additional data. Therefore, the resulting $p_\theta(\mathbf{l}|\mathbf{r}, \mathbf{O}^{(w)}, \eta)$ depends more on the observed sequences. This segment-based analysis is also expected to alleviate the difficulty of learning to identify phonemes with low occurring frequencies, as it is also a downside in the computation of the GOP decision thresholds.

3.5 ITSLanguage Corpus of Dutch Learners of English as L2

All the main experiments for this work were tested on the INA set of the ITSLanguage (ITSL) corpus, provided by ITSLanguage BV (Nicolao et al., 2015). The corpus consists of recordings of prompted speech from young learners of English as L2 in the Netherlands.

The data was collected in classrooms using an online learning tool developed by ITSLanguage BV. The students recorded themselves reading from an ordered list of 193 short sentences and isolated words. The students could re-record each prompt until they were satisfied with their performance. The recordings were carried out using headset microphones and were stored in MS-WAVE format of 22.05 kHz and 16-bit. Environment noise and speech from other students are present in the recordings since many students performed the task simultaneously in the same classroom.

A total of 80 hours of data were collected. Recordings with high levels of distortion were filtered out using clipping detection. An initial forced alignment for the expected prompt using an AM for British English and a multi-pronunciation dictionary was used to discard recordings with missing, partial or nonsense speech. A total of six hours corresponding to over 230 students were selected to form the INA set, annotated for mispronunciation.

A team of three trained phoneticians (*a1*, *a2*, *a3*) marked the INA set for pronunciation errors at the phoneme level. INA considers a mispronunciation any phoneme substitution, deletion and insertion differing from the canonical sequence obtained via the pronunciation dictionary used in the initial forced alignment. Every recording in INA was marked by all three assessors. The assessors did not identify the true phoneme produced in the case of mispronunciation. The inter-annotation agreement (I) and Cohen's kappa (κ) between the assessors are shown in Table 3.1.

The INA speakers range in age, L2 proficiency level and Dutch dialect used; not all students in the corpus are L1 Dutch speakers. The speaker variability is useful for observing and modelling perception bias. The data contains joint individual annotations which show the disagreement between assessors for a wide range of pronunciations. Until now, there is no publicly available nor benchmark data corpus for L2 CAPA which also provides the individual annotation from multiple assessors for this amount of speakers.

3.6 Experiment on Mispronunciation Detection

The ASIM was tested for detecting mispronounced segments with at least one phoneme present. Additionally, a GOP baseline was tested as well to compare it to the proposed segment-based approach. The objective is to show the benefits of distancing from alignment-based scores towards the detection of features associated with phoneme identities. The task was carried out using the three output configurations listed in Section 3.4 and different annotation formats. Each ASIM and the GOP baseline were scored for precision (P), recall (R), and F1 score in detecting segments with at least one phoneme marked as incorrectly pronounced. The reliability between the models and the given annotation reference was also scored using Cohen's kappa (κ).

3.6.1 Experiment Dataset

The INA set was split into 85% and 15% for Train and Test, respectively. The split was balanced for sex, age, and L2 proficiency level with no speaker overlaps. From a total of 238 INA speakers, 215 were used for training and 23 were left out for testing. All speakers recorded the same number of prompts.

The data is processed as short acoustic segments containing a reduced number of phonemes; this is to avoid confusion in the ASIM. Recall the label probability $\hat{l}_i$ is learned as a non-sequential multi-label classification problem. The acoustic segments $\mathbf{O}^{(w)}$ are obtained using a sliding window 0.5s long with 0.05s stride.

The canonical phoneme reference $\mathbf{r}$ for each $\mathbf{O}^{(w)}$ comes from forced-aligning the INA set using a triphone-based Gaussian Mixture Model (GMM)-Hidden Markov Model (HMM) AM trained on WSJCAM0 (Robinson et al., 1995) and 46 hours of ITSL data which do not overlap with INA. The rest of the ITSL data was not annotated for mispronunciation. The AM used for alignment was built using HTK v3.4.1 (Young et al., 2002). To consider a phoneme r_i to be present in $\mathbf{O}^{(w)}$, the alignment has to allocate r_i entirely within at least 2 frames from the edges of the sliding window. Any r_i not fulfilling said alignment condition was left out of $\mathbf{r}$ to help the GOP baseline get the most out of the alignment and to allow the ASIM to overcome alignment errors. The aligned segments contained a mean of 3.46 phonemes with a standard deviation of 1.54.

3.6.2 Annotation Formats for Mispronunciation Detection

Thanks to the joint assessment available for INA, it is possible to implement multiple consolidated annotation formats. Besides, from the labels provided by each assessor, three interpretations of a consolidated annotation were used for the phoneme correctness labels. Each annotation format used corresponds to different interpretations of the bias and ways to *reduce* its noise on the labels. A segment $\mathbf{O}^{(w)}$ is considered mispronounced if at least one of the corresponding phoneme labels $l_i = 0$. The following list briefly describes the different annotation formats used in this experiment.

- *a1*: The labels from a assessor *a1*.
- *a2*: The labels from a assessor *a2*.
- *a3*: The labels from a assessor *a3*.
- *MAX*: The decision with the most votes from all assessors.
- *AND0*: A phoneme realization is considered incorrect only if all assessors considered it incorrect.
- *AND1*: A phoneme realization is considered correct only if all assessors considered it correct.

3.6.3 GOP Baseline

A basic implementation of the GOP as defined in (Witt and Young, 2000) was used as a baseline. The expected phoneme sequence $\mathbf{r}$ for each segment $\mathbf{O}^{(w)}$ was obtained via the forced alignment of the INA recordings with their respective prompt. The Grapheme to Phoneme (G2P) conversion was obtained from the pronunciation lexicon for the British Received Pronunciation accent of English, Combilex (Richmond et al., 2009). If a word was not included in Combilex, its pronunciation was inferred from the Phonetisaurus toolkit for G2P using Weighted Finite-State Transducers (Novak et al., 2016). The forced alignment was carried out using HVite from the HTK toolkit for building HMMs (Young et al., 2002). The triphone HMMs for the alignment were trained using WSJCAM0 and Dutch-accented speech from ITSLANG not included in INA.

The phoneme posteriors for the GOP score were estimated using a four-layer deep FFN built using Tnet (Veselỳ et al., 2010). The network was trained on the same data as the alignment HMMs. The network input consists of a 15-frame span vector with 23 filter bank coefficients for each frame. The acoustic features were obtained using a 25ms window size with a 10ms frame rate. The network outputs 144 monophone states, this is 3 states for each of the 48 phonemes classes. A bottleneck layer of size 26 was implemented in the network to extract features which were also used to train the AM for the alignment. The monophone states were combined to obtain the GOP scores.

The HMMs for the alignment and the network for phoneme posteriors were originally trained as part of a discriminative phoneme classifier for CAPA on ITSL (Nicolao et al., 2015). The GOP baseline considered a segment to be mispronounced if the GOP scores of any of the expected phonemes crossed its corresponding threshold T_p (Equation 2.30). The α and β coefficients for T_p were adjusted to find the Equal Error Rate (EER) on the training examples. Since the GOP scores were based on the alignment of **r**, both GOP and ASIM scored the same number of phonemes.

3.6.4 Model Training Setup

The configuration of the ASIM incorporated a single BDLSTM of size 64, an additive self-attention module with linear weights of size 128, and a 4-layer FFN classifier of size 1024. A different ASIM was trained for each combination of output configuration and annotation format (Section 3.6.2).

The ASIMs were trained using perceptual linear prediction coefficients (Hermansky, 1990) known for their known noise robustness (Yu and Deng, 2016). Vectors of 13 coefficients with their 1^{st} and 2^{nd} order differentials were extracted using a sliding window size of $25ms$ and a stride of $10ms$. The models were trained using the Adam optimizer (Kingma and Ba, 2014) and a BCE loss (Equation 3.13). The decision for declaring $\mathbf{O}^{(w)}$ as mispronounced was based on the point of EER on the ASIM posteriors for the Train data given the output layer configuration **E1**, **E2**, or **E3**.

3.6.5 Performance on Learning a Single Annotator

The performance metrics for each model configuration and GOP baseline are grouped for each assessor in Table 3.2. A pair of dummy classifiers were also used as additional baselines on each assessor reference. The strategies for the dummy outputs were theMost Frequent Label (MF) and Stratified (STR). The MF classifier always outputs the class label with the highest frequency in the Train set. The STR classifier samples a label from the empirical label class distribution of the Train set. All the ASIM and GOP performed better than the dummy classifiers, except for GOP for *a3* (GOP_a3) on Test. The MF baseline for *a3* (MF_a3) showed a relatively high F1 score due to its recall of 1.0 and a precision greater than 0.0 as it is for the rest of the assessors. This is explained by *a3* being the only assessor from all 3 who marked more than half of the speech segments as *mispronounced*. In Table 3.3, the percentage of segments marked either as *Error* or *Correctly pronounced* are shown for all the annotation formats. The percentage of segments marked with a mispronunciation by each assessor is shown in Table 3.3. The results of the STR baseline reflect the percentages in Table 3.3. The effect the severity of each assessor had on the ASIMs is discussed in more depth ahead in this section.

In most configurations, ASIM outperformed the GOP baseline. Only the **E1** configuration of the ASIM showed an F1 score lower than the GOP. However, the large difference between P and R, indicates the baseline outputs many false positives. Notice also the κ for the GOP is like the values of the dummy classifiers, meaning the agreement between the GOP and the reference occurred mostly at random. The results for GOP were not a surprise since the GOP

Table 3.2: For each output configuration and the GOP baseline trained on each INA assessor, the table shows the precision (P), recall (R), F1 score and Cohen's kappa (κ) for detecting segments with mispronunciation given each assessor. Two dummy baselines were also used : *most frequent* (MF) and *stratified* (STR).

	Train				Test			
Model	P	R	F1	κ	P	R	F1	κ
MF_a1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
STR_a1	0.4412	0.4409	0.4411	0.0023	0.3961	0.4437	0.4186	0.0080
GOP_a1	0.4372	0.9806	0.6048	-0.0083	0.3896	0.9839	0.5581	-0.0040
E1_a1	0.6253	0.6796	0.6513	0.3563	0.5112	0.5870	0.5465	0.2199
E2_a1	0.6951	0.7437	0.7185	0.4880	0.5179	**0.7641**	0.6173	0.2865
E3_a1	**0.7053**	**0.7529**	**0.7283**	0.5021	**0.5984**	0.6994	**0.6450**	0.3857
MF_a2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
STR_a2	0.2583	0.2586	0.2585	0.0028	0.2093	0.2585	0.2313	0.0024
GOP_a2	0.2543	0.9780	0.4036	-0.0050	0.2058	0.9818	0.3403	-0.0027
E1_a2	0.3670	0.6268	0.4630	0.2065	0.2799	0.5180	0.3634	0.1286
E2_a2	**0.4781**	**0.7267**	**0.5767**	0.3914	0.2961	**0.7347**	0.4221	0.1827
E3_a2	0.4466	0.7008	0.5455	0.4492	**0.3741**	0.6762	**0.4817**	0.3338
MF_a3	0.6325	1.0000	0.7749	0.0000	0.6121	1.0000	0.7594	0.0000
STR_a3	0.6325	0.6329	0.6327	-0.0001	0.6120	0.6323	0.6220	-0.0003
GOP_a3	0.6303	0.9840	0.7684	-0.0094	0.6104	0.9879	0.7545	-0.0005
E1_a3	0.8403	0.7534	0.7945	0.4887	0.7437	0.7023	0.7224	0.3153
E2_a3	0.8642	0.7870	0.8238	0.5562	0.7815	**0.8123**	**0.7966**	0.4591
E3_a3	**0.8725**	**0.7991**	**0.8342**	0.5806	**0.8239**	0.7542	0.7875	0.4879

score depends on the likelihood ratio between $\mathbf{O}^{(r_i)}$ and the AM, which might be unrelated to the annotation reference. Recall GOP is a single-dimension measure for the fit of alignment. Given the distribution of labels l_i, it is assumed the GOP score range can be split optimally into score bands for *good* and *bad* examples. The ASIM on the other hand, builds a pronunciation reference directly and only from the paired observations and labels. The generally higher κ shown by the ASIMs indicates the combination of BDLSTM and self-attention preferable to GOP for learning the behaviour of the annotator.

There is a slight gain in performance over the baseline from using output layer *E1*. The ASIM with a single binary output uses the least linguistic information from the labels. The *E1* models only observe the acoustic sequence, showing the benefits of sequential encoding over a precise alignment. The only case in which *E1* did not generalise better than GOP was *E1_a2*. When looking at the results in Table 3.2 across all assessors, the models for *a2* are more prone to return a high number of false positives. This behaviour for *a2* might indicate a considerable inconsistency in this assessor, which is not obvious from the correlation coefficients in Table 3.1. On the other hand, the models trained on *a3* show the best performance metrics, even on the GOP baseline. The results on *a3*, may be only useful to conclude that the decision boundaries of this assessor are relatively simple with little confusion when annotating similar observations.

Due to the subjectivity of L2 PA and the scarcity of skilled annotators, usual methods for

Table 3.3: Percentage (%) of segment labels marked either as Errors or Correct pronunciations for each annotation format in both INA Train and Test subsets.

	a1	*a2*	*a3*	MAX	AND0	AND1
Train - Error	44	26	63	42	17	69
Train - Correct	56	74	37	58	83	31
Test - Error	39	21	61	37	14	66
Test - Correct	61	79	39	63	86	34

detecting malicious workers based on a majority vote are not viable for the current state of ITSL. Deeper analysis for anomaly detection would also require a model for the acoustics of L2 phoneme realisation, which is close to what this project aims to explain. So far, there is no strong evidence to entirely discard *a2*; similarly, *a3* cannot be considered a ground truth for INA. From these initial results, *a2* could either be considered inconsistent in their perception or a person with more complex decision boundaries, which cannot be determined with the amount of data available.

With respect to the output layer configuration, the performance metrics in Table 3.2 increase with the number of outputs in the model. The decomposition of the phoneme classes into correct and incorrect occurrences allows ASIM to better estimate the probability of a segment labelled as mispronounced. The binary correctness label l_i ignores any possible similarity between what the annotator considers correct and incorrect realisations of the same phoneme. The extended number of classes in *E3* forces the model to identify pronunciations which might not be different enough to be confused with the replacement or deletion of a phoneme.

3.6.6 Performance for Learning Consolidated Annotations

The results of the experiments for the consolidated annotation formats *MAX*, *AND1* and *AND0* are grouped in Table 3.4. For this set of experiments, ASIM outperformed GOP except for E1_AND1, for which GOP performed slightly better on the Test set. However, the low κ shown by the baseline also indicates its similarity to a dummy classifier. Similarly, to the models trained on individual assessors, the use of more outputs (*E3*) did improve performance. However, there is an evident relationship between the results and the consolidated format. An insight into the consolidated annotations is required to better explain the levels of inconsistency or *noise* in the resulting labels.

The reduced annotator set for INA allows simple analysis of the consolidated formats. The Cohen's kappa (κ) between each consolidated annotation format and the INA assessors is shown in Table 3.5. The different κ values indicate how much assessors agree with each consolidated annotation. The format *MAX* is the annotation with a more even distribution of κ across assessors. Both *AND0* and *AND1* show heavily unbalanced κ values, indicating their proximity to a particular assessor.

The *AND0* annotation ignores mispronunciation labels, which are not unanimous. This

Table 3.4: For each output configuration and the GOP baseline trained on each consolidated annotation format, the table shows the precision (P), recall (R), F1 score and Cohen's kappa (κ) for detecting segments with mispronunciation given each annotation reference.Two dummy baselines were also used : *most frequent* (MF) and *stratified* (STR).

	Train				Test			
Model	P	R	F1	κ	P	R	F1	κ
MF_MAX	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
STR_MAX	0.4216	0.4211	0.4214	0.0031	0.3695	0.4196	0.3930	0.0013
GOP_MAX	0.4174	0.9812	0.5857	-0.0069	0.3665	0.9839	0.5341	-0.0035
E1_MAX	0.6609	0.7290	0.6933	0.4519	0.4898	0.6012	0.5398	0.2248
E2_MAX	0.6900	0.7547	0.7209	0.5029	0.5011	**0.7730**	0.6081	0.2907
E3_MAX	**0.7087**	**0.7708**	**0.7385**	0.5351	**0.6006**	0.7167	**0.6535**	0.4213
MF_AND0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
STR_AND0	0.1740	0.1745	0.1742	0.0020	0.1392	0.1723	0.1540	0.0028
GOP_AND0	0.1706	0.9744	0.2904	-0.0039	0.1359	0.9794	0.2387	-0.0020
E1_AND0	0.2619	0.6291	0.3698	0.1671	0.1863	0.4805	0.2685	0.0884
E2_AND0	0.3492	0.7204	0.4703	0.3102	0.2053	**0.7326**	0.3208	0.1357
E3_AND0	**0.4027**	**0.7640**	**0.5274**	0.3897	**0.2926**	0.6827	**0.4097**	0.2695
MF_AND1	0.6932	1.0000	0.8188	0.0	0.6633	1.0000	0.7975	0.0
STR_AND1	0.6933	0.6935	0.6934	0.0002	0.6610	0.6916	0.6759	-0.0073
GOP_AND1	0.6908	0.9837	0.8116	-0.0123	0.6613	0.9874	0.7921	-0.0012
E1_AND1	0.8772	0.7594	0.8141	0.4788	0.7841	0.7193	0.7503	0.3166
E2_AND1	0.9020	0.8029	0.8496	0.5667	0.8178	**0.8253**	**0.8215**	0.4652
E3_AND1	**0.9031**	**0.8049**	**0.8512**	0.5707	**0.8536**	0.7657	0.8073	0.4831

could be interpreted as only recognizing strong examples of mispronunciation, for which all assessors agree. However, the results for *AND0* in Table 3.4 are not the best. When looking back at the reliability κ, *AND0* agrees the most with *a2* and disagrees the most with *a3*. Once the similarity between *AND0* and *a2* is noted, the performance of the models trained on these references seems to be related as well. The fact that both ASIM and GOP baseline struggle to learn an annotation reference that disagrees with the majority of the assessors, could be a sign of a lack of consistency for *a2*.

The *AND1* annotation keeps every label for mispronunciation from the individual assessors. Opposite to *AND0*, the *AND1* could be considered to acknowledge strong examples of correct pronunciation. The high F1 values for E3_AND1 in Table 3.4 indicate ASIM has less trouble learning these annotations. The agreement between AND1 and the INA assessors in Table 3.5 shows $\kappa = 0.9134$ for *a3* and $\kappa = 0.2601$ for *a2*. The AND1 format replicates most of *a3* criteria, meaning this assessor declares a high amount of mispronunciations compared to the other two annotators.

A look at class imbalance offers more information on the behaviour of the models given the annotation. The INA Train and Test subsets hold a total of 285,991 and 52,461 segments respectively. Refer again to Table 3.3 with the percentages of segments each assessor marked as pronunciation errors. The first thing to notice is how each assessor varies on their *strictness* for declaring a correct pronunciation. The assessors ranked by the number of incorrectly pronounced segments labelled are *a3*, *a1* and *a2*. The more strict an assessor works, the annotation classes becomes more balanced.

Table 3.5: Cohen's kappa between the consolidated annotation formats and the INA assessors.

vs.	*a1*	*a2*	*a3*
MAX	0.8182	0.5547	0.5707
AND0	0.3814	0.7613	0.1971
AND1	0.5612	0.2601	0.9134

The lack of mispronounced examples is a known problem in CAPA (Chen and Li, 2016), particularly errors which are not systematic given the L1 of the speaker. An assessor, which can clearly differentiate pronunciation errors consistently, contributes to well-defined decision boundaries for annotation models. If the assessor groups a wide range of features under the same label, say, assuming *a2* has been careless, it is more difficult for ASIM to tell the observations apart, causing a high recall and a low precision as seen for both *a2* and *AND0*. It may be possible to better learn the criteria of *a2* using more data. However, these findings are still no solid evidence to assume any assessor represents a ground truth more than the others.

It is evident that a simple arithmetic aggregation is not sufficient to get rid of inter-annotator disagreements. The methods for inferring *true* labels from the individual annotators available are heavily dependent on the sample, especially for cases where only a reduced number of skilled workers are available. Each of the consolidated formats seems to *agree* the most with a particular assessor. Therefore, when looking at κ and the class imbalance on each annotation format, it is no surprise to see similar results in both individual and consolidated annotations.

3.6.7 Observations on the Alignment Problem

In Section 2.4.2, the alignment problem was discussed based on different cases of alignment of non-canonical pronunciation, as presented in (Dudy et al., 2018). As a short summary, the alignment problem states the difficulties and inconsistencies arising from the assumption of a clear phoneme segmentation. It is often the case that more than one phoneme identity is detected on a given segment. The estimated phoneme boundaries also interfere with which acoustic frames contribute to each segment. As a result, scores based on precise alignments add even more noise than the ones already present in the annotation.

The GOP baseline was no exception to the alignment problem. A portion of the alignment for the pronunciation of the word *Inheritable* is shown in Figure 3.2. The plot shows the expected phonemes left-to-right in the x-axis, while the detected phonemes are shown on the y-axis. The phoneme classes with the maximum posterior on each frame are marked with a magenta dot. The green dots indicate the phoneme with the maximum GOP in a segment.

All three assessors agreed on the annotation of the example in Figure 3.2, marking only the phonemes /t/ and /ax/ as mispronounced. At first glance, a mismatch between the phoneme with the maximum posterior and the phoneme with the best GOP occurs for almost every segment; however, only a few were labelled as errors. The /ax/ segment shows insertion

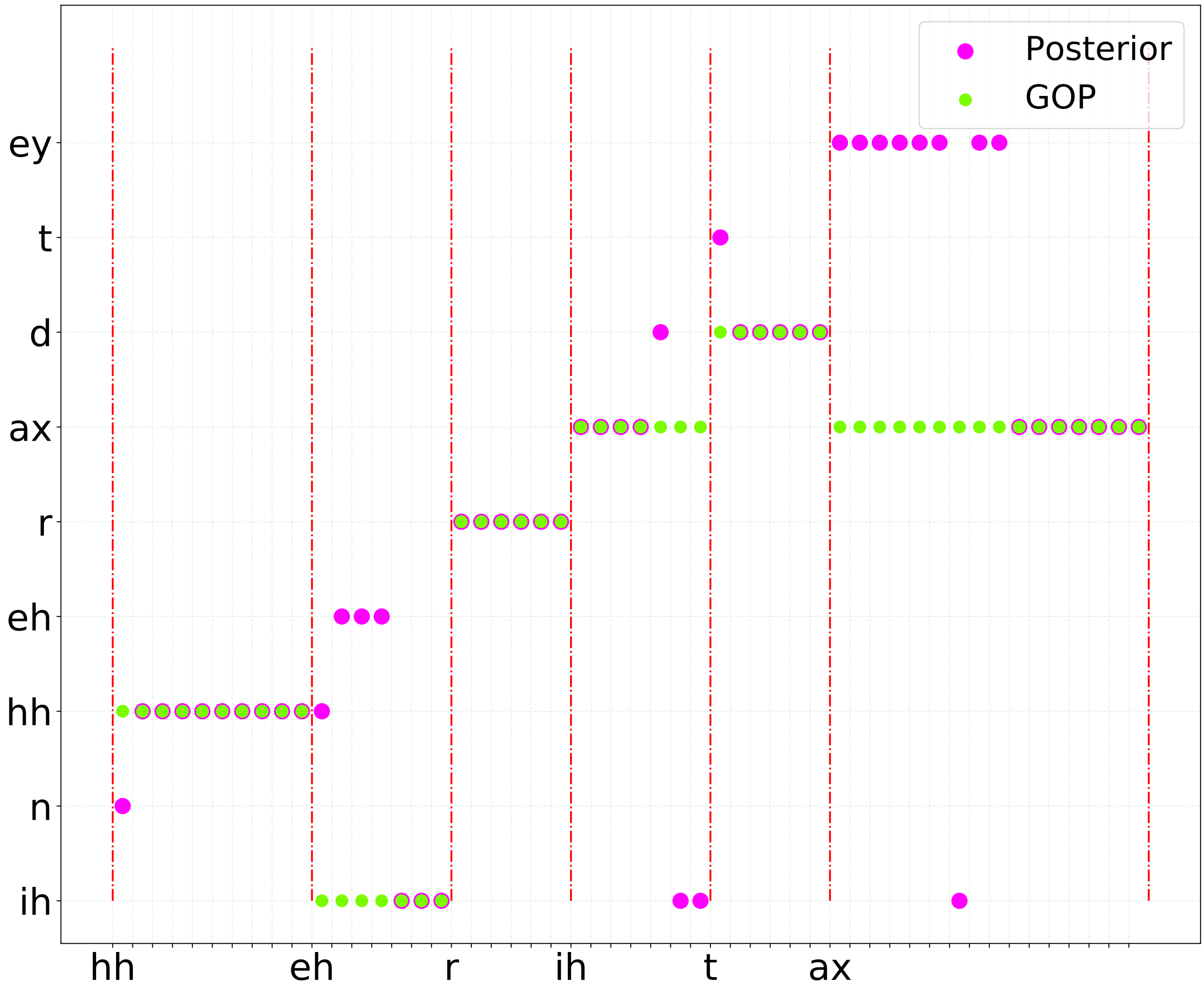


Figure 3.2: Maximum posterior per frame and best GOP score per segment for the alignment of the word *Inheritable*.

of the phoneme /ey/, yet /ax/ still managed to come up as the phoneme with the best GOP score. The /t/ segment also considered mispronounced, clearly shows the speaker's confusion between /t/ and /d/. However, such discrepancies between phonemes with the maximum posterior and the ones with the best GOP also occur for /eh/ and /ih/, which were considered *correct*. The decision thresholds used for GOP aim to map scores obtained from an AM to the criteria of the assessors, although these are not strictly related. The over 40 hours of L2 speech data available would not be enough to train the entire AM from scratch, yet the use of external data in the role of an ideal pronunciation raises the same agreement problems observed when combining different annotators.

There is an effort from the GOP to reduce the discrepancy between the alignment and the pronunciation reference held by the annotators. Recall the phone-specific GOP threshold T_p in Equation (2.30) that comes from said phoneme global score statistics and a pair of hyper-parameters which empirically amplify the variance and shift the threshold to better fit the

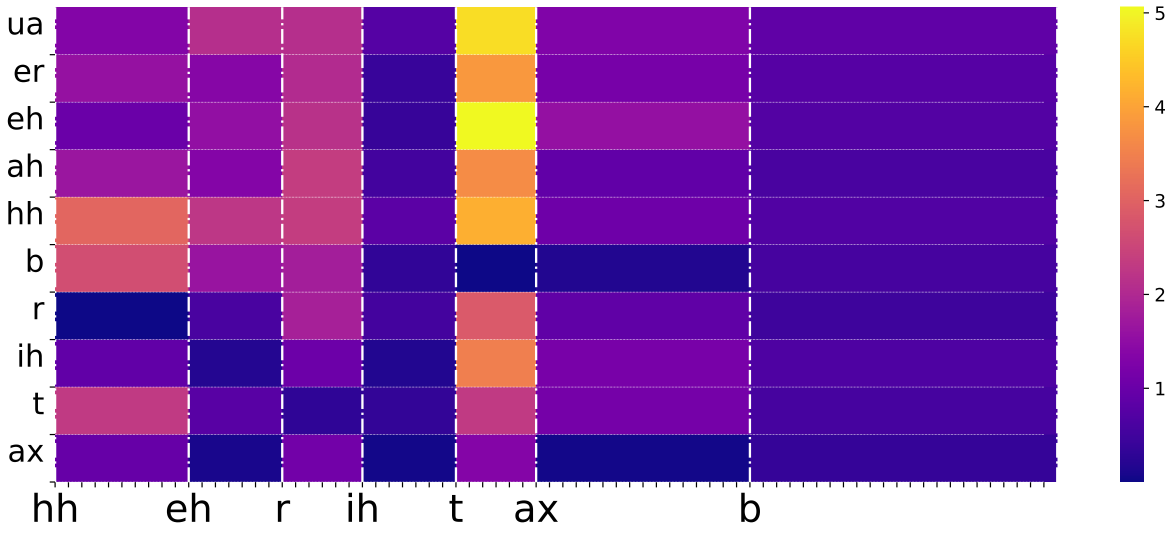


Figure 3.3: Heat-map for the GOP score for for the alignment of the word *Inheritable*.

annotation. However, the distribution of alignment posteriors does not ease the assumptions of equal gravity and uniformity as could an attention mechanism.

It was also noted that the normalisation of the GOP score (Equation 2.27) over the segment length in frames (NF) could artificially disguise the final scores as correct pronunciations. When a segment grows in duration, the denominator NF does as well, reducing the final GOP score. The heat-map in Figure 3.3 shows the GOP for various phoneme classes indicated in the y-axis for the same example of the word *Inheritable* used in Figure 3.2. The different segment lengths can be appreciated in the frame alignment on the x-axis. A GOP score close to zero is equivalent to the ratio of the expected phoneme and the actual phoneme to be close to 1. The colour gradient in the plot indicates a dark colour for values close to zero. The difference in GOP scores across phoneme classes is larger in the relatively short phoneme segments compared to the longer ones, say /ax/ and /b/. The last segment being the longest shows a more even distribution of GOP scores. This artificial deflation of the GOP scores influence the estimation of the decision thresholds and allows slow-paced speakers to obtain better scores.

It is known the attention mechanism helps a Deep Neural Network (DNN) focus on relevant events for the task at hand, in this case, to determine the *correctness* of phonemes. It was expected that the behaviour of the attention weights could show which frames in a segment are the most important when declaring a mispronunciation. It was also expected the visualisation of any behaviour on the attention weights to help with the interpretability of the ASIM.

The attention weights α (Equation 3.10) were normalized across time and vector components. The normalized α was then plotted along the observed speech segment. The alignment used for both the GOP and the creation of the reference $\mathbf{r}$ for the speech segments (see Section 3.6.1) is also included as a visual aid.

A particular phenomenon was noticed in the self-attention mechanism by observing multiple examples. The three plots in Figure 3.4 show the normalised attention weights from E2_AND1 in blue for three different pronunciations of the word *Inheritable* each from a different speaker. The phoneme boundaries used for the GOP baseline are marked by coloured vertical dotted lines matching the phoneme labels on the x-axis. The annotation for mispronunciation is indicated by the orange line. A high position in the orange line indicates the phoneme segment is marked as correctly pronounced, and a low position of the line represents a pronunciation error.

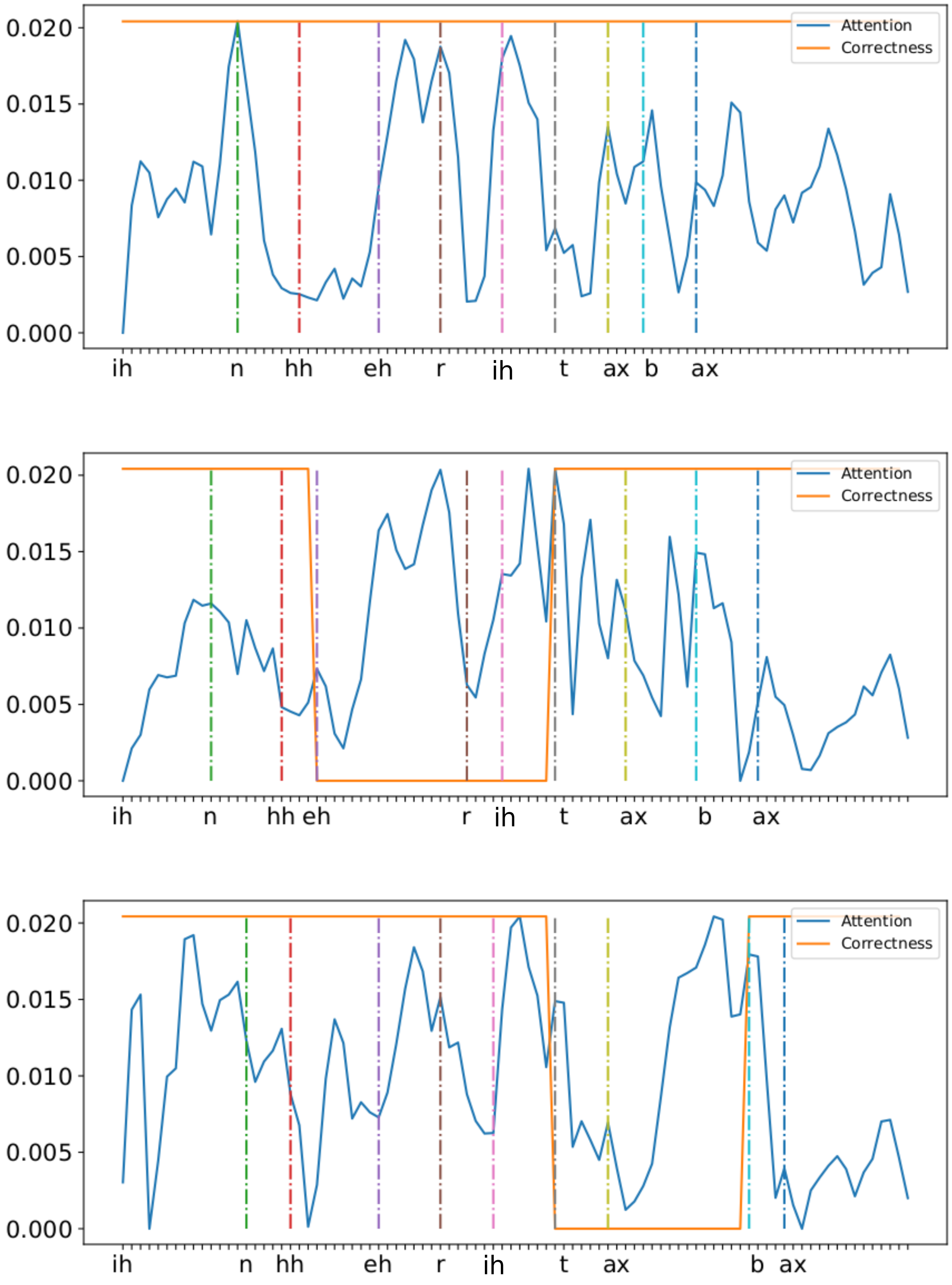


Figure 3.4: Attention curves for 3 examples from different speakers of the word *Inheritable* annotated using AND1. A high correctness curve indicates the segment is labelled as correct, while the curve being low indicates the opposite.

The first thing to notice in the plot is the different spikes. The larger the spike, the larger the effect the corresponding encoded feature has in the following layers of the DNN. Meanwhile, features diminished by a small α have a reduced relevance. The fact that the attention curve consists of multiple spikes rather than small slopes or sustained values, indicates only certain transitions are relevant for the ASIM. Some of the spikes occur in the vicinity, if not in the actual phoneme boundary determined by the GOP alignment. The alignment-like behaviour was unexpected since the ASIM does not receive any sequential information from the labels. Recall the ASIM is trained to learn co-occurrent phoneme classes as a multi-label classification problem. All information associated with phoneme boundaries must be inferred by the BDLSTM and attention ensemble from the acoustic features.

The plot at the top of Figure 3.4 shows no mispronunciations. Most of the phoneme boundaries in the top plot match a spike from the attention curve. The plot in the middle of Figure 3.4 contains a mispronunciation at the first half of the word, and different to the other two examples, it lacks the initial spike marking the $/n/$ segment. The bottom example shows a deep decline followed by a rise which seems to match the position of the mispronounced $/ax/$. There are other spikes occurring elsewhere besides the alignment boundaries. Since the boundaries are obtained from a forced alignment of expected phonemes, the acoustic segment could contain other variations which are less relevant for the final classifier. The decay in the magnitude of the spikes occurring after the last $/ax/$ segment in all the plots might indicate events that would have a higher relevance in a different location in time.

3.6.8 Summary

The experiment presented in this section aimed to show the advantages of the segment-based approach for detecting mispronunciations over phoneme-based scores dependent on precise phoneme alignments. The new approach was implemented on the ASIM, a deep Artificial Neural Network (ANN) using spatiotemporal modelling and saliency region selection based on attention weights. Different levels of annotation were tested to estimate the probability that all the phonemes expected in a short utterance were marked as correct. The ASIM was trained on prompted speech recordings from young Dutch learners of English as L2 and compared against a GOP baseline on detecting mispronounced segments with at least one phoneme present. The ASIM outperformed the baseline, which showed signs of the alignment problem. The behaviour of the baseline was close to the one of a dummy classifier. It was found that the attention weights aligned themselves with the phoneme labels and were useful to locate mispronunciations in the acoustic sequence. The combination of sequential encoding and self-attention allowed the interpretation of the internal representation in ASIM.

3.7 Conclusion

In this chapter, a novel method for automatic mispronunciation detection was proposed. The new strategy consisted of observing short speech segments to detect the presence of acoustic features associated with a *correct* pronunciation, regardless of their exact location in time. The motivation behind this was to ease the dependency on a precise alignment and to avoid inconsistencies raised from misinterpreting phonemes as consistent acoustic events.

The segment-based mispronunciation detection was carried out using sequential encoding via BDLSTM with an additive self-attention module. The resulting encoding was then passed through a deep FFN classifier to estimate the probability of the correctness label. The deep architecture was trained on the accented speech from young learners of English as L2 using different annotation references. The trained model outperformed a GOP reference based on DNN-HMM alignment and monophone posteriors previously used for detecting mispronounced phonemes in the same data.

The proposed method and ASIM outperformed the baseline. The ASIM also showed to agree better with the annotation reference. However, the performance of the models was subject to the strictness of the annotator. It was also found that models, trained on a consolidated annotation reference, would behave like models trained on the assessor with the highest correlation to the current reference. Effects from the alignment problem were observed in the baseline. Nonetheless, the attention weights in the deep architecture were, in some measure, aligned with the phoneme boundaries of the baseline, even when no linguistic nor timing information was present.

Chapter 4

Speaker Metadata for Improving Automatic Pronunciation Assessment

4.1 Introduction

The proficiency of a speaker declares their capability of reproducing a language reference considered to be *correct*. In the case of pronunciation, said reference represents a set of sounds defined by the word or meaning intended to communicate. Pronunciation is not exactly consistent across speakers, yet some pronunciations are legitimized over others by being designated as a standard (Lindemann, 2017). The Pronunciation Assessment (PA) of a speaker is carried out by a listener who judges how much the speaker reassembles a particular pronunciation standard. However, it is the listener's perception, the one who defines the identity of the sounds produced by the speaker, hence interfering with the perceived proficiency of the speaker. It is known to be a relationship between the perception of speech and any previous linguistic experience of the listener; the later one referring to any previous exposure to an accent or to other speakers of an assumed similar linguistic identity (Lindemann, 2017, Winke et al., 2012).

The effect of the listener's bias on assessment can be reduced, or at least made consistent over different assessors via previous training on an adapted Second Language (L2) pronunciation standard (Harding, 2017, Wei and Llosa, 2015, Witteman et al., 2014). This situation brings up the problem of reference for Computer Assisted Pronunciation Assessment (CAPA), particularly when L2 speech data is usually scarce or not annotated jointly by multiple assessors. As previously mentioned, conventional Automatic Speech Recognition (ASR) systems for CAPA result problematic due to the need for a proper Acoustic Model (AM) for L2 speech (Dudy et al., 2018). These AMs were mainly built using Native Language (L1) data to be later adapted to L2 pronunciation to some extent using available L2 data and techniques such as Maximum Likelihood Linear Regression (Chu et al., 2020, Dudy et al., 2018, Huang et al., 2017a, Witt and Young, 2000). In this chapter, a speaker representation based on metadata is tested as an additional input to improve the performance of the Attention-Based Segmental Incorrectness Model (ASIM). The goal is to find an alternative to the more complex AM adaptation and representation techniques when the target data available is limited.

4.2 Representation for the Speaker Linguistic Identity

Recall, a mispronunciation occurs if the perceived sounds are different to the ones defined by the reference. Proficient L2 speakers, however, do not need to imitate an L1 accent perfectly, yet some minimal similarity is needed to achieve a functional phoneme differentiation and to convey information. Since PA is not completely objective, what in this work is called *the assessor bias*, plays a crucial role. The bias is relevant particularly for L2 PA, as descriptors such as *the presence of an accent* or a pronunciation *close to natural* could be used to define different levels of pronunciation proficiency (Harding, 2017, Trofimovich and Isaacs, 2017).

As it has been pointed out in multiple study cases of L2 PA, assessors are prone to a bias particular to the perceived identity of the speaker (Galaczi et al., 2011, Harding, 2017, Lindemann, 2017, Ockey and French, 2016). Said bias can bring up stereotypes about L2 accents, social background, education, etc. A shift in perception can occur even when the speaker identity presented to the assessor is not the true one (Lindemann, 2017).

A speaker representation would be useful to provide more information about the speaker, to help the ASIM infer a more precise correctness reference from the observed labels. Speaker representations such as the i-vector have been proven useful for detecting accents and identifying speakers. However, i-vectors do not perform well for short segments, usually blamed on a low phoneme count (Verma and Das, 2015). An alternative is to use speaker information associated with their linguistic background. Speaker metadata could give additional structure to the data when more complex representations are not viable.

Not all assessors necessarily focus on every aspect of the background of a speaker. Some features such as L1 or birthplace might have a higher effect on assessor bias compared to, say, the number of years of formal training in L2. Therefore, an empirical study can offer some insight into what speaker information is relevant to a pronunciation assessor.

Such a strategy is of course subject to the type of data available. Particularly for ITSLanguage (ITSL), additional information about the linguistic background of the student was provided via questionnaire. This metadata is useful for building a speaker profile which could be associated with pronunciation tendencies in students and might be useful to help explain the perception of L2 pronunciation. The factors collected grouped as either Categorical or Binary data are listed below:

- Categorical:
  - **BP**: Dutch province or country of birth.
  - **DIAL**: Dutch dialect used daily.
  - **L1**: Native Language.
  - **SAL**: Self assessed English proficiency level.
  - **SCH**: School ID.
  - **YENG**: Years of formal studies of English as L2.

- Binary:
  - **MLH**: Multilingual household.
  - **NND**: Non-native Dutch speaker.
  - **NNP**: Non-native Dutch speaking parents.

The questionnaires were answered by the students as part of ITSL without any teacher or tutor supervision. The speaker factors listed above were anonymized for the protection of the speakers'identity.

4.3 Experiment on the Use of Metadata to Improve Mispronunciation Detection

On the assumption of an assessor bias linked to experiences with speakers of different linguistic backgrounds, the metadata available for ITSL was tested as part of the input for the ASIM. A speaker representation $\boldsymbol{\lambda}$ was built by performing a one-hot encoding of the classes in each speaker factor in ITSL listed in Section 4.2. The different speaker factor encodings were concatenated as needed for $\boldsymbol{\lambda}$ to represent multiple speaker dimensions. An ASIM trained without $\boldsymbol{\lambda}$ was used as a baseline to compare against different configurations of $\boldsymbol{\lambda}$. The performance of the models was tested on the *MAX* vote annotation reference, as well as for each INA assessor.

4.3.1 Experiment Dataset

The INA set was used for this experiment as defined in Section 3.6.1 for the segment-based approach for the mispronunciation detection experiment. The data was split 85% for Train and 15% for Test with no speaker overlap and balanced for sex, age and L2 proficiency level. The short segments $\mathbf{O}^{(w)}$ and their respective phoneme sequences $\mathbf{r}$ were the same obtained via the alignment for the Goodness of Pronunciation (GOP) baseline in Section 3.6.3.

4.3.2 Model Training Setup

The ASIM kept its original configuration of a single Bidirectional Long Short-Term Memory (BDLSTM) with 64 hidden units; an additive self-attention module with linear weights of size 128, and a deep Feed-Forward Network (FFN) classifier with 4 layers each of size 1024. The classifier used the *E3* output configuration, meaning two outputs for each phoneme class corresponding to either a correct or incorrect pronunciation. The probability of a mispronounced segment is estimated from the normalized probability of all expected phonemes being marked as correct, as shown in Equation (3.12).

The models were trained using the first 13 perceptual linear prediction coefficients with their 1^{st} and 2^{nd} order time differentials as acoustic features. The different representations $\boldsymbol{\lambda}$ were concatenated as constant dimensions to each acoustic frame passed to the BDLSTM layer

Table 4.1: Performance results for the ASIM matching the MAX reference. The number of classes (C), F1 score, Cohen's Kappa (κ) and p-value (p) for the McNemar's test is shown for every speaker factor λ.

		Train		Test		
λ	C	F1	κ	F1	κ	p
None	-	0.7385	0.5351	0.6535	0.4213	-
BP	14	0.7422	0.5218	0.6564	0.4138	0.006
DIAL	10	0.7407	0.5391	0.6515	0.4199	0.150
L1	6	0.7293	0.5183	0.6483	0.4120	0.180
MLH	2	0.7333	0.5256	0.6510	0.4177	0.157
NND	2	0.7319	0.5230	0.6526	0.4186	0.245
NNP	2	0.7316	0.5225	0.6507	0.4161	0.502
SAL	5	0.7332	0.5255	0.6537	0.4237	0.003
SCH	7	0.7325	0.5242	0.6530	0.4232	0.059
YENG	21	0.7321	0.5234	0.6494	0.4183	0.074

in the ASIM. All models were trained using the Adam optimizer and a Binary Cross-Entropy (BCE) (Equation 3.13) to learn the correctness of phoneme labels **l**. The posterior threshold for declaring a mispronunciation corresponds to the Equal Error Rate (EER) point on the train set.

4.3.3 Performance on Isolated Speaker Factors

The objective of this experiment was to test the capability of ASIM for associating metadata to a pronunciation reference. It is important to declare that the speaker factors listed in Section 4.2 were not the only ones tested, but the ones showing the most effect in the performance of the ASIM. All the models were scored for F1 score and Cohen's Kappa (κ) on detecting mispronounced segments given the reference. A McNemar's test (McNemar, 1947) was also conducted between each ASIM trained using a speaker factor and the baseline ASIM using none. The McNemar's test is a non-parametric method to detect a change in the distribution of responses on paired binary data, hence the null hypothesis of the test is that there is no difference between the outputs of two ASIMs. The p-value (p) is included along the results of each speaker factor λ tested.

The performance metrics for the models trained for *MAX* using isolated speaker factors are shown in Table 4.1. The isolated factors used for λ did not improve the performance of the ASIM. The DIAL factor was the only one showing a positive effect, at least on the Train set, yet this was not even greater than 0.01 on F1. The results on the individual INA assessors should be more informative, as a consolidated reference is not the best example of a consistent bias. The results in Table 4.2 corresponds to the ASIM trained on assessor *a1*, Table 4.3 does it for assessor *a2*, and Table 4.4 does it for assessor *a3*. Each row in Table 4.2, 4.3, and 4.4 corresponds to the ASIM learning each assessor using a single speaker factor $\boldsymbol{\lambda}$. The first thing to notice is that for *a1*, Table 4.2 shows no increase in metrics for any $\boldsymbol{\lambda}$ regardless of its apparent similarity to *MAX*. Both *a1* and *a2* show a slight decay in performance for any

Table 4.2: Performance results for the ASIM matching assessor *a1*. The F1 score, Cohen's Kappa (κ) and p-value (p) for the McNemar's test is shown for every speaker factor λ.

	Train		Test		
λ	F1	κ	F1	κ	p
None	0.7283	0.5021	0.6450	0.3857	
BP	0.723	0.4932	0.6416	0.3817	0.424
DIAL	0.7232	0.4926	0.6428	0.3843	1.0
L1	0.7229	0.4920	0.6419	0.3820	0.458
MLH	0.7223	0.4908	0.6422	0.3830	0.720
NND	0.7227	0.4915	0.6434	0.3837	0.677
NNP	0.7230	0.4922	0.6400	0.3804	0.360
SAL	0.7256	0.4969	0.6416	0.3853	0.435
SCH	0.7211	0.4885	0.6431	0.3878	0.140
YENG	0.7240	0.4941	0.6406	0.3847	0.414

Table 4.3: Performance results for the ASIM matching assessor *a2*. The F1 score, Cohen's Kappa (κ) and p-value (p) for the McNemar's test is shown for every speaker factor λ.

	Train		Test		
λ	F1	κ	F1	κ	p
None	0.6167	0.4492	0.5034	0.3338	-
BP	0.6127	0.4430	0.5034	0.3267	0.187
DIAL	0.6075	0.4350	0.5012	0.3223	0.0002
L1	0.6244	0.4610	0.5068	0.3314	1.0
MLH	0.6118	0.4417	0.5022	0.3232	1.4E-4
NND	0.6269	0.4648	0.5108	0.3371	0.094
NNP	0.6277	0.4661	0.5066	0.3316	0.648
SAL	0.6168	0.4494	0.5046	0.3295	0.610
SCH	0.6110	0.4404	0.5006	0.3233	0.099
YENG	0.6135	0.4442	0.5053	0.3317	0.032

$\boldsymbol{\lambda}$. Meanwhile, Table 4.4 for *a3* showed a slight improvement in generalization for various speaker factors. Only MLH did show improvement for the *a3* model in both Train and Test. The enhanced performance for *a3* was not surprising considering the previous analysis in Section 3.6.5. The baseline results for *a3* in Table 4.4 are likely higher than the other assessors due to the strictness of *a3*. Recall that from all assessors, *a3* marked the largest amount of mispronounced examples (see Table 3.3).

A further observation of speaker metadata revealed heavily unbalanced classes within the speaker factors. A set of bar plots in Figure 4.1 show the class distribution for the individual speaker factors used in this experiment. Almost every plot in Figure 4.1 is heavily skewed, except for SCH, as the schools had a similar number of students. The class imbalance affects the performance of the models (Bishop and Nasrabadi, 2006). For a heavily unbalanced $\boldsymbol{\lambda}$ such as L1, MLH or NND, the model may ignore an input dimension which is mostly constant. Meanwhile, classes with relatively low occurrences could be treated as noise in features.

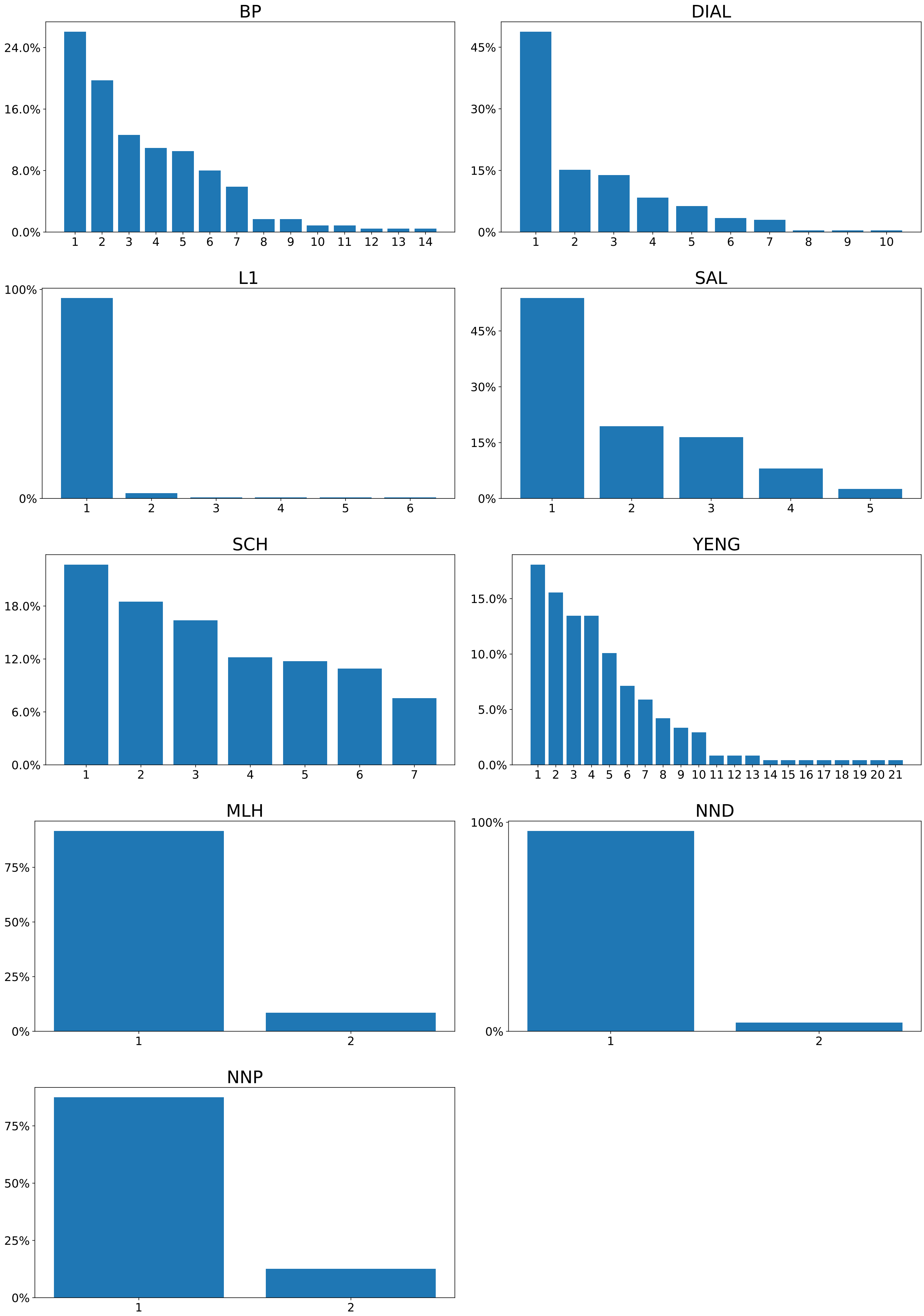


Figure 4.1: Anonymized class distribution within the speaker factors listed in the INA set.

Table 4.4: Performance results for the ASIM matching assessor *a3*. The F1 score, Cohen's Kappa (κ) and p-value (p) for the McNemar's test is shown for every speaker factor λ.

	Train		Test		
λ	F1	κ	F1	κ	p
None	0.8342	0.5806	0.7875	0.4879	-
BP	0.8326	0.5768	0.7896	0.4908	0.278
DIAL	0.8315	0.5742	0.7885	0.4901	0.503
L1	0.8307	0.5724	0.7886	0.4902	0.485
MLH	0.8412	0.5972	0.7887	0.4915	0.331
NND	0.8317	0.5746	0.7891	0.4908	0.332
NNP	0.8326	0.5769	0.7882	0.4883	0.813
SAL	0.8320	0.5754	0.7892	0.4921	0.219
SCH	0.8322	0.5758	0.7875	0.4904	0.639
YENG	0.8318	0.5749	0.7879	0.4912	0.468

4.3.4 Performance on Combined Speaker Factors

Various speaker factors were combined (concatenated) as the λ representations used to train the ASIMs. It is expected the metadata combinations result in a more informative and diverse λ. The combinations reported in this section correspond to the ones showing the most effect on the performance of the ASIM. Table 4.5 shows the performance for the ASIM trained on the MAX reference using the combined speaker factors, with ALL standing for the 9 listed factors combined. Overall, Table 4.5 shows better performance on detecting mispronounced segments compared to the models using single speaker factors as λ (Table 4.1).

The improvement effect had an optimal point given the combination of speaker factors. As an example, the ASIM trained on MAX using BP as λ increased over-fitting with a difference in κ from the baseline of 0.0133 and -0.0075 for Train and Test respectively. MLH also decreased κ by -0.0095 for Train and -0.0036 for Test. When BP was combined with MLH, the gain in κ was 0.0168 for Train and 0.0121 for Test. The gain from BP.MLH was the largest on both Train and Test for the MAX models. More factors did not improve the model further; for example, the addition of NNP to BP.MLH did not keep increasing κ. The smaller improvement from BP.MLH.NNP could indicate redundancy in certain factors impacting the growing number of classes. The original number of classes for BP is 14 and 2 for MLH. The number of classes for BP.MLH is 21, meaning more diversity in the labels. The 29 classes in BP.MLH.NNP would cause problems for the ASIM due to the class imbalance. As λ becomes more specific and sparser, it behaves similarly to a speaker ID. Notice the combination ALL has 173 classes for a set of 238 speakers. The class with the largest proportion of ALL represents 2.5% of the speakers. The result from using ALL in Table 4.5 shows over-fitting by increasing κ by 0.0441 for Train and decreasing by 0.0078 in Test.

There were also cases in which a more balanced λ did not cause the most improvement. The BP.MLH combination does not make for a class distribution more balanced than the one of BP or MLH. Another example is NND.MLH, with only 4 classes remain heavily skewed.

The ASIM trained on MAX did not improve from using NND.MLH; however, the ASIM for *a3* showed the largest gain in performance across all individual assessor models. Figure 4.2 shows the class distribution for the combination BP.MLH on the left and NND.MLH is on the right.

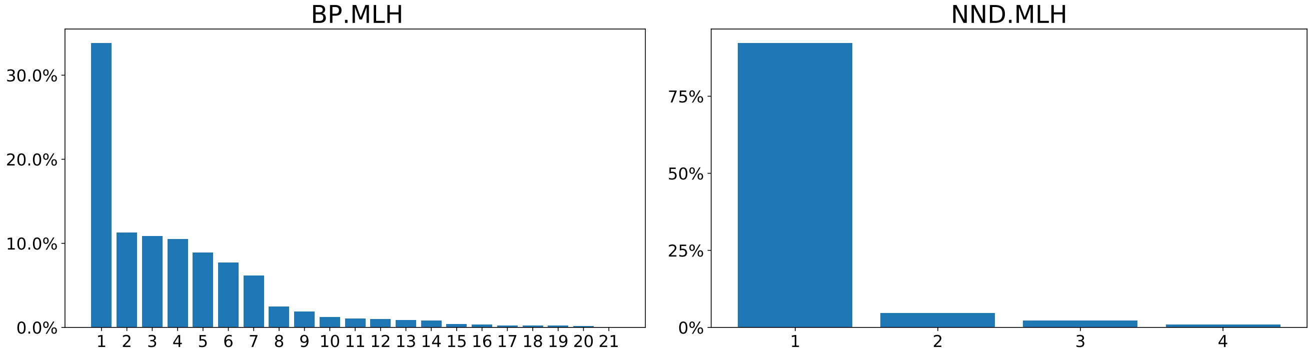


Figure 4.2: Anonymized class distribution for the BP.L1 (left) and BP.MLH (right) in the INA set.

As mentioned in Section 4.3.3 about the models trained on single speaker factors, the results on a single assessor reference should be more informative due to their bias being likely more consistent than MAX. The results from training ASIMs for each INA assessor using combined speaker factors as $\boldsymbol{\lambda}$, appear in Tables 4.6 through 4.8. The results in Table 4.6 corresponds to the ASIM trained on assessor *a1*, Table 4.7 does it for assessor *a2*, and Table 4.8 does it for assessor *a3*. Each row in Tables 4.6, 4.7, and 4.8 corresponds to the ASIM learning each assessor using a different combination of speaker factors. Compared to the results on MAX (Table 4.5), the effect of BP.MLH was not consistent across the individual assessor models. Only *a3* showed a slight improvement in κ of 0.0263 in Train and 0.0016 in Test from using BP.MLH. Each assessor responded differently to the combined speaker factors. In summary, the *a1* models had a slight improvement in generalization from all the $\boldsymbol{\lambda}$s in Table 4.6. Only SCH.DIAL did improve the learning of *a1* showing an increase in κ for both Train and Test by 0.0160 and 0.0099 respectively. The *a2* models did not show improvement on the Test set for any $\boldsymbol{\lambda}$, except for ALL. Only *a2* showed a slight increase in κ for the Test set by using ALL. Again, *a3* exhibited the most improvement from almost every $\boldsymbol{\lambda}$. The ASIM for *a3* trained using NND.MLH showed a gain in κ for Train and Test of 0.0157 and 0.0121 respectively, the largest improvement observed for the Test set across all assessor models.

A particular combination of speaker factors could improve the effect of $\boldsymbol{\lambda}$ on the ASIM by improving the class balance of more meaningful metadata. For example, the *a3* model using SCH.DIAL increased the κ for Test by 0.0092. The *a3* model trained using either SCH or DIAL alone, had gain in κ for Test of 0.0025 and 0.0022 respectively. Similarly, the model for *a1* using SCH.DIAL increased κ by 0.0099 on Test. The *a1* models using single speaker factors changed their κ on Test by 0.0021 using SCH and -0.0095 using DIAL. Recall, SCH is the most balanced speaker factor, which was crucial for the improvement on the *a1* and *a3* models using SCH.DIAL. This conclusion comes from comparing the results for *a1* and *a3* using SCH.DIAL against BP.DIAL. The *a1* model using BP had a decrease in performance.

Table 4.5: Performance results for the ASIM matching the MAX reference. The number of classes (C), F1 score, Cohen's Kappa (κ) and p-value (p) for the McNemar's test is shown for the speaker factor combinations used as λ

		Train		Test		
λ	C	F1	κ	F1	κ	p
None	-	0.7385	0.5351	0.6535	0.4213	-
BP.DIAL	31	0.7440	0.5452	0.6561	0.4257	0.427
BP.L1	24	0.7500	0.5563	0.6563	0.4269	0.204
BP.MLH	21	0.7476	0.5519	0.6602	0.4334	0.003
BP.NND	21	0.7346	0.5281	0.6527	0.4204	0.606
NND.MLH	4	0.7346	0.5281	0.6490	0.4136	0.014
NND.YENG	28	0.7358	0.5301	0.6515	0.4230	0.277
SCH.DIAL	25	0.7450	0.5469	0.6491	0.4185	0.844
BP.MLH.NNP	29	0.7463	0.5494	0.6570	0.4287	0.060
BP.NND.NNP	27	0.7451	0.5473	0.6562	0.4263	0.297
MLH.NND.YENG	34	0.7370	0.5324	0.6543	0.4262	0.104
SCH.DIAL.YENG	104	0.7464	0.5498	0.6506	0.4236	0.096
ALL	173	0.7625	0.5792	0.6448	0.4135	0.281

Table 4.6: Performance results for the ASIM matching assessor *a1*. The number of classes (C), F1 score and Cohen's Kappa (κ) is shown for speaker factor combinations used as λ.

		Train		Test		
λ	C	F1	κ	F1	κ	p
None	-	0.7283	0.5021	0.6450	0.3857	-
BP.DIAL	31	0.7295	0.5044	0.6470	0.3925	0.015
BP.L1	24	0.7260	0.4977	0.6480	0.3929	0.019
BP.MLH	21	0.7248	0.4955	0.6458	0.3892	0.19
BP.NND	21	0.7244	0.4947	0.6460	0.3884	0.347
NND.MLH	4	0.7242	0.4944	0.6455	0.3890	0.180
NND.YENG	28	0.7242	0.4944	0.6462	0.3950	1.4E-4
SCH.DIAL	25	0.7368	0.5181	0.6472	0.3956	2.3E-4
BP.MLH.NNP	29	0.7244	0.4947	0.6413	0.3788	0.047
BP.NND.NNP	27	0.7255	0.4969	0.6466	0.3906	0.083
MLH.NND.YENG	34	0.7265	0.4986	0.6451	0.3926	0.003
SCH.DIAL.YENG	104	0.7249	0.4957	0.6434	0.3921	9.8E-4
ALL	173	0.7794	0.5977	0.6344	0.3855	0.010

The *a3* model using BP had a gain in κ of 0.0029 for the Test set. The BP.DIAL combination showed better performance than BP, and DIAL individually, yet it was not greater than the models trained with SCH.DIAL. The gain in κ on Test from BP.DIAL was 0.0068 for *a1* and 0.0034 for *a3*. The *a2* model also had a better performance using SCH.DIAL compared to BP.DIAL, although neither managed to improve the baseline.

The SCH or BP factors alone might not be the most relevant information for the assessors, considering the mobility any student from this corpus could have. However, the addition

Table 4.7: Performance results for the ASIM matching assessor *a2*. The number of classes (C), F1 score and Cohen's Kappa (κ) is shown for speaker factor combinations used as λ.

		Train		Test		
λ	C	F1	κ	F1	κ	p
None	-	0.6167	0.4492	0.5091	0.3338	-
BP.DIAL	31	0.6161	0.4482	0.5057	0.3305	0.728
BP.L1	24	0.6212	0.4561	0.5039	0.3277	0.509
BP.MLH	21	0.6179	0.4510	0.5058	0.3300	0.711
BP.NND	21	0.6172	0.4499	0.5034	0.3267	0.257
NND.MLH	4	0.6180	0.4512	0.5053	0.3291	0.471
NND.YENG	28	0.6164	0.4488	0.5031	0.3290	0.077
SCH.DIAL	25	0.6166	0.4491	0.5063	0.3314	0.608
BP.MLH.NNP	29	0.6166	0.4490	0.5038	0.3260	0.011
BP.NND.NNP	27	0.6167	0.4492	0.4988	0.3191	1.5E-5
MLH.NND.YENG	34	0.6194	0.4534	0.5070	0.3333	0.040
SCH.DIAL.YENG	104	0.6167	0.4492	0.5043	0.3320	2.2E-4
ALL	173	0.6783	0.5427	0.5020	0.3357	1.2E-20

Table 4.8: Performance results for the ASIM matching assessor *a3*. The number of classes (C), F1 score and Cohen's Kappa (κ) is shown for speaker factor combinations used as λ.

		Train		Test		
λ	C	F1	κ	F1	κ	p
None	-	0.8342	0.5806	0.7875	0.4879	-
BP.DIAL	31	0.8401	0.5945	0.7884	0.4913	0.389
BP.L1	24	0.8416	0.5981	0.7915	0.4947	0.0229
BP.MLH	21	0.8454	0.6069	0.7881	0.4895	0.664
BP.NND	21	0.8379	0.5894	0.7897	0.4923	0.172
NND.MLH	4	0.8409	0.5963	0.7929	0.5000	2.7E-4
NND.YENG	28	0.8345	0.5813	0.7882	0.4945	0.149
SCH.DIAL	25	0.8435	0.6026	0.7900	0.4971	0.018
BP.MLH.NNP	29	0.8414	0.5977	0.7894	0.4895	0.452
BP.NND.NNP	27	0.8404	0.5953	0.7870	0.4855	0.568
MLH.NND.YENG	34	0.8331	0.5781	0.7882	0.4905	0.517
SCH.DIAL.YENG	104	0.8304	0.5717	0.7855	0.4885	0.716
ALL	173	0.8521	0.6229	0.7760	0.4711	2.5E-8

of SCH did influence the class distribution in λ. Figure 4.3 shows the class distribution for SCH.DIAL on the left and BP.DIAL on the right. The plot for SCH.DIAL shows a smaller tail on the right, the product of a smaller number of classes compared to BP.DIAL. The percentages on the y-axis in the plot for SCH.DIAL indicate the classes are more balanced than the ones of BP.DIAL. The individual plots for BP and DIAL in Figure 4.1 indicate BP is more balanced than DIAL, yet not as much as SCH.

Class balance in λ is no more important than the relevance the metadata could have for the assessor. The perception of the speaker is not affected in the same way for every lis-

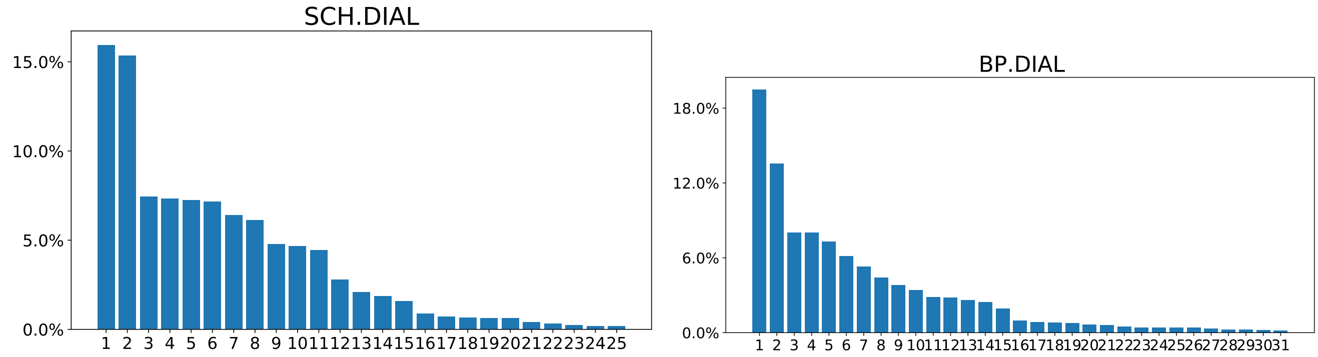


Figure 4.3: Anonymized class distribution for the SCH.DIAL (left) and BP.DIAL (right) in the INA set.

tener. Said effect is confirmed by the change in performance results given different speaker metadata shown in this experiment. The gain in performance from using metadata was not overwhelming, yet the ASIMs did manage to increase their agreement with the reference given the composition of $\boldsymbol{\lambda}$. Aside from the speakers in INA not being notably diverse, the effect of the metadata in ASIM comes from information provided by the students without any proof or supervision. A more efficient and meaningful speaker representation should be designed. Such a claim is backed by the particular case of the 4 skewed classes in NND.MLH shows the largest improvement for *a3*, as well as the reduction of improvement by adding more metadata to $\boldsymbol{\lambda}$.

4.4 Summary and Conclusions

This work presented a study to improve CAPA by using speaker metadata. The segmental based approach was used to train an ASIM instead of a conventional ASR pipeline which may depend on assumed L1 and L2 references. The ASIM outperformed a standard GOP implementation and learned a pronunciation reference using only L2 speech data and the annotation available. A speaker representation λ was built using different combinations of speaker metadata which was concatenated to the acoustic inputs. Different speaker metadata showed a positive effect in the performance metrics of the ASIM given the sparsity and class balance of λ. The findings in this work confirmed how speaker information affects the bias of each assessor differently. The speaker metadata comes from a sample in which most of the speakers share L1, dialects and various linguistic characteristics. A more balanced and diverse speaker sample is required to determine which speaker factors affect the assessor bias the most. If the assessor's behaviour can be correlated to the background of the speaker, it would be easier to isolate the contribution of the bias to the assessment.

Chapter 5

A Model for the Assessor Bias

5.1 Introduction

As mentioned in Section 2.1.2, perception bias is inherent in PA. The effects of said bias have been acknowledged, along with multiple ideas on the mechanisms it works on (Winke et al., 2012). The most mentioned ideas to reduce the bias in Pronunciation Assessment (PA) focus on means to make the bias consistent. Ideas such as a clear descriptor for different proficiency bands (See Section 2.2.1) and training the assessors on designated pronunciation references with scored examples are implemented with the intention of increasing the reliability of an authoritative test. It is important to acknowledge that a high inter-assessor agreement, does not mean the assessment is free of bias.

Like an Second Language (L2) certification body setting, a pronunciation reference to be learned by the assessors, an assessor model for Computer Assisted Pronunciation Assessment (CAPA) aims for the *ideal* most reliable assessor. In the early stages of CAPA, the main focus was on computing metrics for a difference between a pronunciation example and a reference, then tuning said metrics to find decision thresholds which matched real-world assessment.

The inter-rater and intra-rater variability in the annotation represents a cap on the performance of any CAPA. To increase the consistency of the reference, multiple annotators are used to weighing out disagreements in assessment, which are seen as noise in the data. Again, building a consolidated reference from the decisions backed by most of the annotator sample can only be considered a bias closer to most of the assessor sample. The obstacle of correlating metrics with an inconsistent reference remains; however, machine learning allows the construction of models from real labelled data using a higher complexity than decision thresholds over single dimension values such as Goodness of Pronunciation (GOP).

5.2 Existing Models for Annotator Bias

A model for the bias in annotation might not be the priority when trying to build a consistent CAPA tool. Any annotation is subject to bias; hence a model for PA will replicate said bias. Most CAPA related publications focus on better matching their choice of reference, which is expected to show a high level of inter-rater agreement. Again, since disagreement is always present, a consensus between the annotators is simulated. It is usual to ignore the bias present in annotation disagreements, yet the observations which raise conflicts will likely increase the perplexity of a model for PA.

Parametric models that use some form of interpretation of the annotator bias do exist for tasks involving human annotation. The field of Crowd-Sourced Annotation (CSA) as well as automatic labelling requires methods for assuring their reliability. In CSA, mainly non-professional annotators take part in small labelling tasks resulting in larger volumes of annotated data, lower costs, and results not too different from expert annotation (Loukina et al., 2015, Van Dalen et al., 2015). While different methods can be used to rectify the non-expert annotation, e.g., to include the output of an Automatic Speech Recognition (ASR) transcription (Shashidhar et al., 2015, Van Dalen et al., 2015) and the removal of observations with high disagreement of the data (Jamison and Gurevych, 2015), this section focuses mainly on a Bayesian approach for the aggregation of CSA.

Like the consolidated annotation mentioned in Section 2.5, CSA is based on the rationale of the law of large numbers. To assign a label to an observation, the simplest aggregation technique for determining the true identity of an observation is to perform Max-voting on the observed labels. However, relying only on a label count for ground truth could not be the best strategy for tasks such as recognition of mispronounced phonemes. Therefore, methods to better estimate ground truth and the likelihood of annotation errors are required.

An interesting take from CSA is that a true identity label exists and can be inferred from the variation in the label distribution observed. It is assumed, an annotator can be defined by a function or a distribution which provides a noisy annotation dependent on a true label y_i^* associated with a class prior probability (Ramakrishna et al., 2020). The annotator w assigns the categorical label y_i^w to an observation x_i; the label y_i^* can be inferred from the annotations across a set of **W** workers as

$$y_i^* = \arg\max_{y_i^*} \prod_{w \in \mathbf{W}} p(y_i^w | y_i^*) p(y_i^* | x_i) \tag{5.1}$$

The true label result of Equation (5.1) depends on the choice of model for the annotators' *noisy* behaviour. The inclusion of the label preferences specific to annotator w as, say, a set of weights or a confusion matrix, upholds some similarities to the definition of the bias used in this work. Additionally, the dependence of the prior probability y_i^* on the observed data could represent a *common ground* across the annotator set. In practical terms, y_i^* remains subject to the set of labels as Equation (5.1) is usually obtained using Maximum A Posteriori estimation (Ramakrishna et al., 2020).

5.2.1 Bayesian Approach for Crowd-sourcing Aggregation.

The Bayesian approach naturally allows the integration of the prior y_i^* conditioned on the data. Additionally, the interpretability of the annotation model is also kept over implementations fully built on Deep Neural Networks (DNNs) with further human tuning (Li et al., 2021). The annotation model (Equation 5.1) can be expanded as needed in regard to the elements involved in the relationship between the latent ground truth and the observed labels.

A good illustrative example of the definition of an annotation model is the one of (Li et al., 2021). The setup consists of a size N data set of observations $\mathbf{X} = \{x_1; i = 1, \ldots, N\}$ annotated by W workers. The label for the x_i example annotated by the worker w is represented as $\mathbf{L}_i^w = k$. The label corresponds to class k, with $k = 0$ reserved for when a worker annotation is missing. The latent true identity labels $\mathbf{Y}^* = \{\mathbf{y}^*{}_i; i = 1, \ldots, N\}$ are estimated using the generative model $p(\mathbf{L}_i^w | \mathbf{y}^*{}_i, \boldsymbol{\mathcal{V}}^w)$. The model represents the annotator inconsistencies in the form of the independent confusion matrix $\boldsymbol{\mathcal{V}}^w = \{\boldsymbol{\nu}_k^w; k = 1, \ldots, K\}$ for worker w, with vectors $\boldsymbol{\nu}_k^w = \{\nu_{kl}^w; l = 1, \ldots, K\}$.

Once the prior distribution for the latent labels $\boldsymbol{\pi}$ has been defined, the probability of the annotation model given the observations is

$$p(\mathbf{L}, \mathbf{Y}^*, \boldsymbol{\mathcal{V}}, \boldsymbol{\pi} | \mathbf{X}) = p(\boldsymbol{\pi}) p(\mathbf{Y}^* | \boldsymbol{\pi}) p(\mathbf{Y}^* | \mathbf{X}) p(\mathbf{L} | \mathbf{Y}^*, \boldsymbol{\mathcal{V}}) p(\boldsymbol{\mathcal{V}}) \tag{5.2}$$

, where

$$p(\mathbf{L} | \mathbf{Y}^*, \boldsymbol{\mathcal{V}}) = \prod_{i=1}^{N} \prod_{w=1}^{W} p(\mathbf{L}_i^w | \mathbf{y}^*{}_i, \boldsymbol{\mathcal{V}}^w)_{\mathbf{L}_i^w \neq 0} \tag{5.3}$$

$$p(\boldsymbol{\mathcal{V}}) = \prod_{w=1}^{W} \prod_{k=1}^{K} p(\boldsymbol{\nu}_k^w) \tag{5.4}$$

Both $\mathcal{V}$ and $\boldsymbol{\pi}$ are assumed to come from Dirichlet distribution as it is a conjugate prior to the multinomial distribution. The parameters $\mathcal{V}$, the prior $\boldsymbol{\pi}$ and $p(\mathbf{L} | \mathbf{Y}^*, \boldsymbol{\mathcal{V}})$ are found by maximizing the likelihood of the observed labels.

The Bayesian approach for modelling CSA allows an easy interpretation of the factors influencing the deviation between the biased and the latent variables. The intra-annotator variation $\boldsymbol{\nu}_k^w$ could be seen as the annotator bias, as it captures the confusion across labels and influences the observed annotation. On the other hand, the latent labels $\mathbf{Y}^*$ cannot exactly be considered a global agreement nor to be free of bias, as their role in Equation (5.2) is to help explain the observed labels given the observed data.

A downside of Bayesian inference is the lack of methodology for defining the label prior $\boldsymbol{\pi}$ and practically any of the distributions involved; this adds unexplained bias to an interpretation of true identity labels. Another problem with Bayesian inference is that it alone does not necessarily mitigate the effects of a small sample size (McNeish, 2016). Inference methods for addressing small samples exist; however, the number of professional annotators involved in L2 learning corpus is usually no more than three to perform Max-voting aggregation. Additional verification of the annotators would help to validate an authoritative L2 corpus annotated for mispronunciation using CSA.

Authors using CSA for L2 CAPA usually aim for annotation which agrees across the most consistent workers or annotation showing similarity to a *gold standard* task (Shashidhar et al., 2015, Wang et al., 2013). Annotations coming from multiple, yet similar workers retain the bias of the population taking part in CSA, making the interpretation of the inferred labels as the *true identity* of the data quite circumstantial.

5.2.2 Models for the Prediction of the Mean Opinion Score for Synthetic Speech.

Another annotation task suffering from a lack of reference is the assessment of the quality of synthetic speech, for which the Mean Opinion Score (MOS) is used. The International Telecommunication Union defines the opinion score as the value on a predefined scale that an annotator assigns to their opinion of the performance of a system; the average of said values across annotators from the MOS (Streijl et al., 2014). As with many other labelling tasks, the cost, time, and availability of annotators and data samples are limiting factors for collecting MOS that help evaluate the quality of media content produced. Previous works have tried to infer MOS using statistical models on crafted features. Deep learning has also been used to obtain high correlations between observed MOS and raw data inputs. This subsection focuses only on a small set of models which consider the variation across annotators to construct an MOS generator (Huang et al., 2022).

An important work on learning MOS for synthetic speech was done using the deep model MOSnet (Lo et al., 2019). Said network performs sequential encoding on the acoustic features of an utterance using either convolutional layers, Bidirectional Long Short-Term Memory (BDLSTM) layers or a combination of both. The encoded frames are then passed through a Fully Connected (FC) network, uses a combination of convolutional which outputs a numerical value per frame. The MOS for the entire utterance is determined by averaging the frame-level scores. In (Lo et al., 2019), MOSnet is trained to minimize the mean squared error between the averaged output and the observed MOS. The individual opinion scores from each annotator are not considered by MOSnet; however, better results would be achieved by modelling the score of each assessor for every utterance.

The network MBnet (Leng et al., 2021) uses two subnetworks to learn both MOS and the individual Listener Dependant (LD) score. Like MOSnet, MBnet uses a combination of convolutional layers, BDSLTM layers and an FC network with a single output corresponding to the observation's MOS; this subnetwork is called MeanNet. The second subnetwork, BiasNet, also performs sequential encoding to obtain a single output score as MeanNet, with the difference of concatenating an encoding for the listener identity to the output of its first convolutional layer. Additionally, the output of MeanNet is added to the output of BiasNet to obtain the LS score for the utterance. Both MeanNet and BiasNet are trained to learn MOS and the LD score, respectively. After training, only MeanNet is used to obtain MOS.

The inclusion of the LD score did improve the correlation of MOS with the acoustic features. A further improvement for the learning of both MOS and LD scores would come as LDNet-MN (Huang et al., 2022). Based on the claim that MBnet was inefficient by having two subnetworks performing sequential encoding simultaneously, LDNet-MN was designed

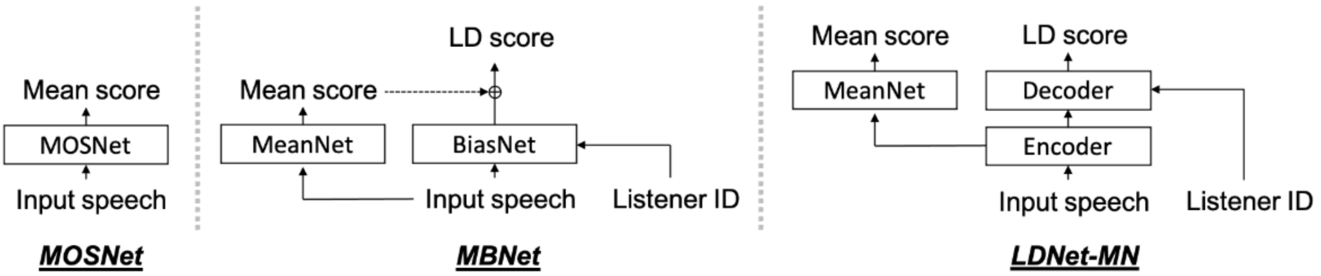


Figure 5.1: Diagrams of various networks used to learn MOS. From left to right: MOSNet, MBNet, and LDNet-MN. Modified from (Huang et al., 2022).

using a single sequential encoder connected to both an FC network acting as MeanNet and a BDLSTM-based decoder which outputs the LD scores. Compared to MBnet, LDNet-MN concatenates the listener identity to the decoder in charge of the LD scores, keeping the initial encoding listener-independent. The outputs of LDNet-MN are combined exactly as MBnet does and are trained simultaneously using empirical hyperparameters when combining the losses of both subnetworks. Figure 5.1 shows diagrams for MOSNet, MBNet, and LDNet-MN.

Notice how, even when MOS is a consolidated method, its modelling improves when considering all available variations in annotation for the same observation. However, MOS is still far from what could be considered an annotation free of bias, as it is an arithmetic method for consolidating joint annotations. The objective of this work is to find a model for the individual bias given both the annotator and observation, in order to infer labels with strong arguments to claim the bias is at least kept at the minimum possible.

5.3 Proposed Model for the Assessor Bias in Pronunciation Assessment

The previous work on learning valuable information from annotation disagreement aims to better model an interpretation of *true labels* or a statistical value chosen to represent annotation samples. Both models described in the previous section rely on the law of large numbers. When the data of the population is available, it is possible to assume small variations as noise in the data; this is not the case for PA. The assessors taking part in authoritative language tests are specifically trained to replicate a particular perception bias for the sake of consistency. However, the assessor sample is often too diverse or not large enough to obtain statistics with relatively low variations. Consider the case of CSA, for which algorithms have been developed to ensure the reliability of the inferred labels even when the assessor sample could have a size of multiple thousands. Case studies and available data corpus of L2 PA do not usually count with many assessors. Labelled data for PA usually range in the number of annotators involved from one to three in order to have a tiebreaker, this even annotators did even judge the same data.

The Bayesian approach used for CSA (Section 5.2.1) allows an interpretation of the annotator bias as prior probabilities towards classes. Meanwhile, the distribution referred to as the *true* labels, still depend on other hand-crafted prior distributions picked arbitrarily. Said model is designed to explain the observed data via latent random variables. The model proposed in this work focuses instead on separating the individual assessor bias from an ideal assessor-independent scoring function.

Consider an assessor η who performs PA of an utterance $\mathbf{O}$. It is assumed this task is performed using an η specific scoring function $A_\eta(\mathbf{O})$, which outputs the probability of η declaring the *correctness* of the utterance. An additional function $D_\eta(A_\eta(\mathbf{O}))$ acts as a decision threshold for declaring the segment as *mispronounced* or not. It is also assumed an ideal bias-free scoring function $A(\mathbf{O})$ exists. Said function is independent of any annotator and can be considered as the starting point of PA, meaning this function alone could output *universal* agreement or the score for a minimum proficiency required to be understood. The relationship between the two scoring functions $A_\eta(\mathbf{O})$ and $A(\mathbf{O})$ is an additive *corrupting* term responsible for every deviation from the assumed universal correctness score. Any factor or parameter causing a preference in the phoneme perception of the listener, hence in PA, is modelled by the function $b_\eta(\mathbf{O})$ and can be referred to as the *bias*. Therefore, the function for an observed PA score is defined as

$$A_\eta(\mathbf{O}) = A(\mathbf{O}) + b_\eta(\mathbf{O}) \tag{5.5}$$

No constraints are defined for Equation (5.5); neither assumptions about any priors nor their output distributions. It is possible to learn $A_\eta(\mathbf{O})$ from the observed annotation provided by η. On the other hand, to find both the assessor independent and bias components, hereby A and b_η respectively, represent a greater challenge. Given it is possible to estimate b_η, its contribution to PA can be isolated or *subtracted* from $A_\eta(\mathbf{O})$, hereby A_η, to obtain a bias-free PA score. Once both A_η and b_η are obtained from an adequate large assessor population H, the bias-free assessment function can be approximated as the average of all the corrected scoring functions:

$$\frac{1}{H}\sum_{\eta\in H} A_\eta(\mathbf{O}) = \frac{1}{H}\sum_{\eta\in H}[A(\mathbf{O}) + b_\eta(\mathbf{O})] \tag{5.6}$$

$$A(\mathbf{O}) = \frac{1}{H}\sum_{\eta\in H}[A_\eta(\mathbf{O}) - b_\eta(\mathbf{O})] \tag{5.7}$$

Although simple, Equation (5.5) allows a clear interpretation of the role of the bias in PA. Both components of $A_\eta(\mathbf{O})$ can be augmented as needed in order to make the model more realistic. For example, to assume the intended prompt w associated with $\mathbf{O}$ is known is equivalent to the condition of supervised pronunciation training. Any factors which could possibly contribute to the perception of the listener given both the speaker and the pronunciation reference can be tested in this model.

5.4 The DASIM for Modelling the Assessor Bias

Previously, in Section 3.3, the Attention-Based Segmental Incorrectness Model (ASIM) was defined as a DNN for estimating the probability of a phoneme correctness label from the sequential encoding of a short speech segment. The segment-based approach for mispronunciation detection implemented in the ASIM aims to reduce inconsistencies caused by the dependence on a precise phoneme alignment.

The experiment in Section 3.6 showed ASIM could outperform an aligned-based score without using any additional linguistic information or speech data other than the one observed by the annotators. The same approach for mispronunciation detection is selected to learn both assessor-independent and bias terms in Equation (5.5). In simple words, two ASIMs are trained to contribute to the final assessor η dependent observed score, A_η.

The ASIM design remains practically the same as in its original definition: a setup of BDLSTM, additive self-attention, and a deep Feed-Forward Network (FFN) classifier. Recall, the objective of this work is to learn the bias component b_η. Since a single ASIM estimates the probability of the correctness labels as $P(\hat{\mathbf{l}}|\mathbf{r},\mathbf{O}^{(w)},\eta)$, this can be split between two simultaneous networks to estimate each component of Equation (5.5) as

$$P(\hat{\mathbf{l}}|\mathbf{r},\mathbf{O}^{(w)},\eta) = P(\hat{\mathbf{l}_\mathbf{A}}|\mathbf{r},\mathbf{O}^{(w)}) + P(\hat{\mathbf{l}_\mathbf{b}}|\mathbf{r},\mathbf{O}^{(w)},\eta) \tag{5.8}$$

, where $\hat{\mathbf{l}_\mathbf{A}}$ and $\hat{\mathbf{l}_\mathbf{b}}$ are the estimates for the η independent and η specific contributions to the annotation labels estimate $\hat{\mathbf{l}}$ respectively. Since the contributions of both $\hat{\mathbf{l}_\mathbf{A}}$ and $\hat{\mathbf{l}_\mathbf{b}}$ to $\hat{\mathbf{l}}$ are not known, both DNNs are trained simultaneously.

The interaction between the two ASIM models needs to generate a probability distribution for each phoneme correctness label. Therefore, the outputs of each ASIM are added up and passed through a sigmoid layer for regularization. More constraints for the combination of $\hat{\mathbf{l}_\mathbf{A}}$ and $\hat{\mathbf{l}_\mathbf{b}}$ can be designed to strengthen the claim of the Dual Attention-Based Segmental Incorrectness Model (DASIM) learning the bias component in the annotation. The combination of both outputs may look simple at this stage, yet it is a starting point for an interpretable model design.

In accordance with Equation (5.8), both ASIMs process the same acoustic features for $\mathbf{O}^{(w)}$. Only the ASIM corresponding to $\hat{\mathbf{l}_\mathbf{b}}$ observes the assessor identity η concatenated as a constant dimension to each acoustic frame O_t. It is expected that the ASIM for $\hat{\mathbf{l}_\mathbf{A}}$ generalizes over η while $\hat{\mathbf{l}_\mathbf{b}}$ adjusts the output to better predict each assessor. This DASIM setup is shown in Figure 5.2.

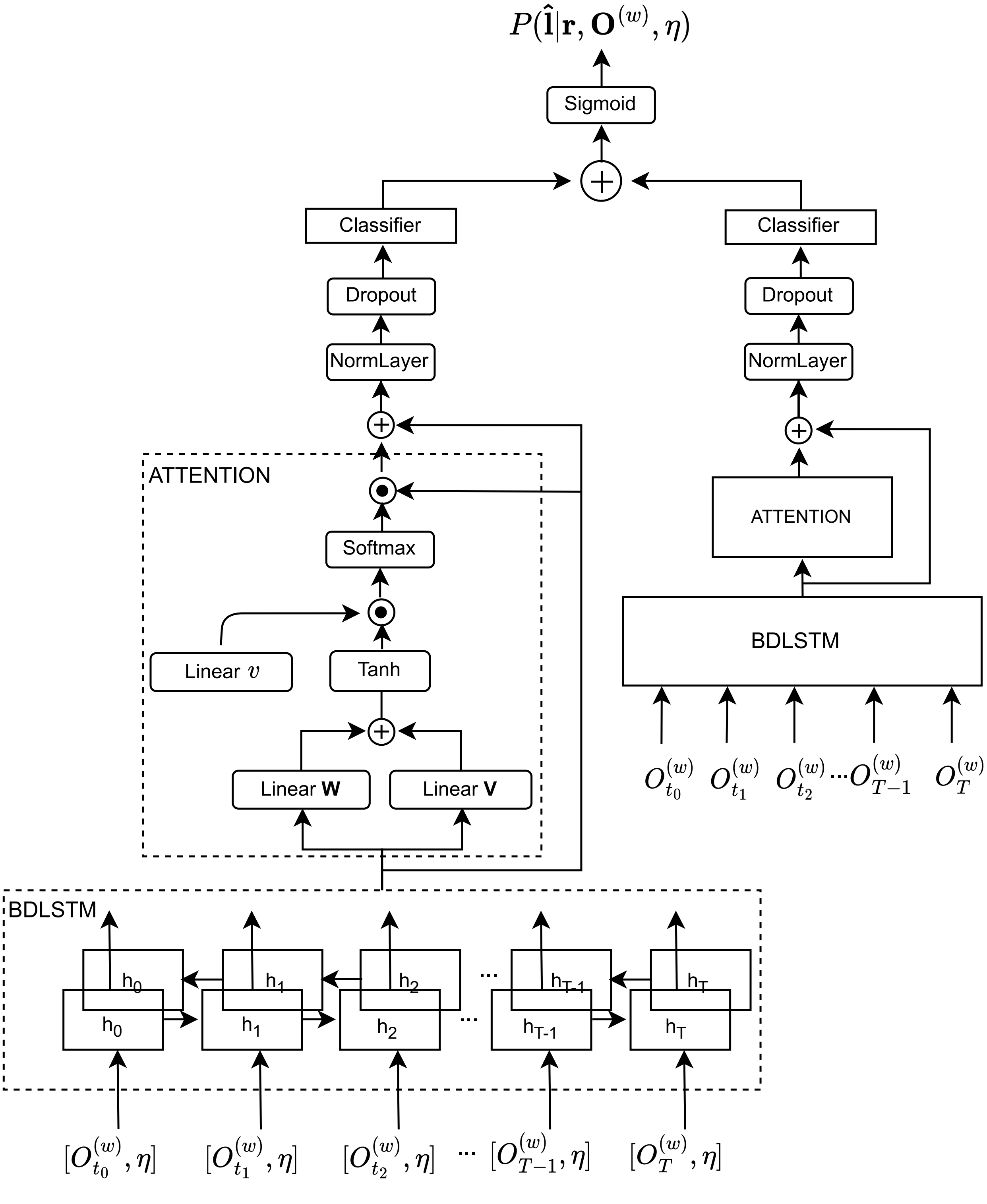

$P(\hat{\mathbf{l}}|\mathbf{r}, \mathbf{O}^{(w)}, \eta)$
Sigmoid
Classifier
Dropout
NormLayer
ATTENTION
Softmax
Linear v
Tanh
Linear **W**
Linear **V**
Classifier
Dropout
NormLayer
ATTENTION
BDLSTM
$O_{t_0}^{(w)}$ $O_{t_1}^{(w)}$ $O_{t_2}^{(w)}$ ... $O_{T-1}^{(w)}$ $O_{T}^{(w)}$
BDLSTM
h_0 h_1 h_2 ... h_{T-1} h_T
$[O_{t_0}^{(w)}, \eta]$ $[O_{t_1}^{(w)}, \eta]$ $[O_{t_2}^{(w)}, \eta]$ ... $[O_{T-1}^{(w)}, \eta]$ $[O_{T}^{(w)}, \eta]$

Figure 5.2: Diagram for the DASIM setup. The left path observes the assessor tag η concatenated to the input **O** of length T.

5.5 Experiment on the DASIM for learning Multiple L2 Pronunciation Assessors

The DASIM was trained to learn the observed phoneme correctness labels given assessor η on the INA dataset of L2 speech annotated for mispronunciation (See Section 3.5). The goal was to approximate the assessor independent and bias components in function A_η (Equation 5.5) via the network output logits $\mathbf{L_A}$ and $\mathbf{L_b}$. The DASIM was trained on the three INA assessors, *a1*, *a2* and *a3*, and scored for precision (P), recall (R) and F1 score on detecting mispronounced segments given each individual assessor. The reliability between the models and the given annotation reference was also scored using Cohen's kappa (κ). The DASIM was compared against two single ASIMs BASE-S and BASE-M; to observe the effects of splitting the task between two subnetworks. The models BASE-S and BASE-M consist of each of a single ASIM as defined in Section 5.5.1. The model BASE-S was trained on a single INA assessor, while BASE-M was trained on all INA assessors simultaneously, similar to the left subnetwork in Figure 5.2. The results of BASE-M would reflect the benefits, if any, from the network learning parameters common to multiple pronunciation references.

The contribution from $\mathbf{L_A}$ and $\mathbf{L_b}$ to the final output logits was also observed. The $\mathbf{L_b}$ logits should be on average smaller than those of $\mathbf{L_A}$ since the bias is considered a deviation from the *ideal* independent scoring function A in Equation (5.5). The sigmoid-normalized $\mathbf{L_A}$ were used for scoring each assessor and a MaxVote consolidated annotation in order to test which reference is the closest to the assessor-independent subnetwork. The relevance of the assessor tags η for $\mathbf{L_b}$ was also tested by scoring each annotation reference using mismatching assessor tags.

5.5.1 Model Training Setup

The DASIM consists of two subnetworks for the encoding and classification of an acoustic segment. Each ASIM subnetwork used a 6-layer BDLSTM of size 64, an additive self-attention module with linear weights of size 128 and a 6-layer deep FFN of size 1024 for the correctness posteriors. The output layer configuration implemented two binary outputs for each phoneme class, corresponding to either a correct or incorrect pronunciation (see output layer configuration *E3* in Section 3.4). The models BASE-S and BASE-M used the same architecture of a single ASIM subnetwork.

The first 13 Perceptual Linear Prediction (PLP) coefficients with their first and second-order time differentials were used as feature vectors. The DASIM was trained for each assessor annotation jointly, meaning each $\mathbf{O}^{(w)}$ was observed three times due to the three assessors. A one-hot encoding of η was concatenated to the acoustic features for the path of $\mathbf{L_b}$ as indicated in Figure 5.2.

5.5.2 Experiment Dataset

The INA set was split the same way as in the initial experiment on the ASIM detecting mispronounced segments reported in Section 3.6.1. The data was split, 85% for Train and 15% for Test. The split is balanced for sex, age, L2 proficiency level and has no speaker overlap. The short acoustic segments $\mathbf{O}^{(w)}$ were defined using a sliding window of 0.5s with a 0.05s stride. The pronunciation reference $\mathbf{r}$ is the same obtained from the forced alignment used for the GOP baseline in Section 3.6.3. Refer to Table 3.1 for the inter-assessor agreement in INA and

Table 5.1: Precision (P), Recall (R), F1 score and Cohen's kappa (κ) for the DASIM on detecting mispronounced segments for each assessor η.

		Train				Test			
Model	η	P	R	F1	κ	P	R	F1	κ
BASE-S	*a1*	0.7459	0.7888	0.7667	0.5741	0.6246	0.6941	0.6575	0.4172
	a2	0.5904	0.8071	0.6819	0.5482	0.4312	0.6637	0.5228	0.3623
	a3	0.8928	0.8287	0.8595	0.6407	0.8337	0.7563	0.7931	0.5047
BASE-M	*a1*	0.7447	0.7878	0.7656	0.5721	0.6441	0.6709	0.6572	0.4290
	a2	0.5797	0.8001	0.6723	0.5337	0.4602	0.6322	0.5327	0.3849
	a3	0.8913	0.8265	0.8577	0.6364	0.8508	0.7706	0.8087	0.5425
DASIM	*a1*	0.7553	0.7971	0.7756	0.5907	0.6444	0.6636	0.6539	0.4256
	a2	0.5967	0.8111	0.6876	0.5566	0.4645	0.6111	0.5278	0.3820
	a3	0.8938	0.8302	0.8608	0.6439	0.8482	0.7643	0.8041	0.5333

Table 3.3 for a perspective on the strictness of each assessor.

5.5.3 Performance on Learning Individual Assessors

The BASE-S, BASE-M and DASIM scores for detecting mispronounced segments given to each assessor in INA are shown in Table 5.1. The difference in results between the three models is not particularly large. First, the improvement in generalization BASE-M showed compared to BASE-S was not greater than 1.8% for both *a2*, and *a3*. For the case of *a1*, BASE-M had an even smaller decay of less than 0.05%. The gain from training on multiple annotation references simultaneously was not outstanding. Similarly, the slight difference in results between the BASE models and the DASIM is relatively small. The DASIM decrease F1 for *a1*, yet the decay was smaller than the gain for *a2* and *a3*. Table 5.1 shows that there is no severe change in performance from using two subnetworks to estimate the final correctness posteriors. (Equation 5.8). The assessors ranked by performance result in both BASE models and the DASIM are *a3*, *a1* and *a2*. This trend in performance, given the assessors, was first observed in the original ASIM experiment for detecting mispronunciations discussed in Section 3.6.5. Both the BASE models and the DASIM did improve the previous ASIM results on INA (Table 3.2) most likely due to their larger amount of parameters.

The stricter *a3* seems to provide enough examples for the model to better distinguish between correct and incorrect pronunciations. In the case of *a2*, the assessor is still showing the lowest metrics in Table 5.1. However, there is an increase of almost 20% in all the performance metrics for *a2*, compared to the best single ASIM results in Table 3.2. It is likely that the general improvement shown by the DASIM comes straight from the larger amount of parameters.

5.5.4 The Effect of the Assessor Identity on the Bias Output

The claims of the assessor model proposed in Equation (5.5) need to be verified from both the $\mathbf{L_A}$ and $\mathbf{L_b}$ outputs. The sensitivity of $\mathbf{L_b}$ to the tag η was tested. For this, the same DASIM

Table 5.2: F1 score for the DASIM on detecting mispronounced segments for each assessor reference giving a different tag η.

	Reference					
	Train			Test		
η	*a1*	*a2*	*a3*	*a1*	*a2*	*a3*
a1	**0.7756**	0.5919	0.8280	**0.6539**	0.4508	0.7604
a2	0.7058	**0.6876**	0.7967	0.5788	**0.5278**	0.7185
a3	0.7034	0.5489	**0.8608**	0.6251	0.4394	**0.8041**

Table 5.3: Contribution percentages from the assessor independent **A** or the bias **b** logits in the DASIM.

	Train		Test	
η	A	b	A	b
a1	72.2	27.8	71.7	28.3
a2	69.3	30.7	68.6	31.4
a3	73.2	26.8	72.6	27.4

was scored on detecting mispronounced segments given each annotator reference; only this time mismatched assessor tags would be used in the input.

The results in Table 5.2 correspond to the F1 scores on detecting mispronounced segments for each of the combinations of the tag η and each annotation reference. The highest F1 scores in bold correspond to the match of η with its corresponding annotation reference. When the inappropriate tag η is used, the performance metrics can decrease from 3% to 20%. The difference in how much the performance decay reflects the inter-assessor agreement.

The observations made from Table 5.2 indicate the tag η adjusts the model to *compensate* or *rectify* $\mathbf{L_A}$, which remains independent of the assessor identity.

It is expected the agreement between assessors is larger and more consistent than their disagreement; therefore, the bias should not be the main contributor to the final assessment score. The DASIM reflected this assumption by relying on more in $\mathbf{L_A}$ than in the $\mathbf{L_b}$ output to better learn the annotation reference. A simple metric $\mathcal{C}_j$ illustrates the contribution from each output layer to the final estimate $P(\hat{\mathbf{l}}|\mathbf{r}, \mathbf{O}^{(w)}, \eta)$. Equation (5.9) defines $\mathcal{C}_j$ as the contribution percentage from the output layer j to the total sum of squared logits. The output j is either the assessor independent **A** or the bias **b**. The contribution percentages for j are averaged across all N segments in the data. Equation (5.9) uses the squared logits since outputs close to zero will have little effect on the final combined output.

$$\mathcal{C}_j = \frac{1}{N}\sum_{n}^{N}\left[\frac{\mathbf{L_j}_n^2}{\mathbf{L_A}_n^2 + \mathbf{L_b}_n^2}\cdot 100\right] \tag{5.9}$$

The contribution percentages from $\mathbf{L_A}$ and $\mathbf{L_b}$ are shown for each assessor in Table 5.3. The assessor-independent logits A are the larger contributors to the final correctness posteriors for all assessors. The bias output intervenes efficiently as its effect on average is relatively small. The percentages in Table 5.3 are also consistent with the inter-assessor agreement. The b

contribution for *a2* is the largest of all assessors, which is not surprising given the lower agreement coefficients in Table 3.1.

It is assumed the closest $\mathbf{L_b}$ gets to model every inter-assessor disagreement caused by an offset term as defined in Equation (5.5), the $\mathbf{L_A}$ output should be free of all bias. It is complicated to make sense empirically of an unbiased scoring function for L2 PA. The $\mathbf{L_A}$ outputs might not be used directly for scoring L2 performance, as a reference is still needed for a decision threshold. However, a claim on the model of assessor bias could be made if a considerable contribution of $\mathbf{L_b}$ is observed for particular speakers or phonemes.

5.5.5 Observations on the Assessor Independent Output

The $\mathbf{L_A}$ outputs were used to score each INA assessor as well as the MAX voting consolidated reference. On the assumption of the DASIM allocating the A and b contributions in an optimal way, it is expected that $\mathbf{L_A}$ represents a level of agreement. The scoring results for subnetwork A shown in Table 5.4 are consistent with the inter-assessor reliability. The best performance was not for scoring the MAX reference. This was not unexpected as it was found out before the similarities between *a1* and MAX. The *a3* reference seemed to be the closest one to A, which again could be related to the strictness and the number of error examples present in the annotation.

As mentioned in Section 3.6.2, assumptions are always used when trying to obtain a consolidated annotation reference with the least presence of disagreement. Although the amount of annotated data is limited, the subnetwork responsible for $\mathbf{L_A}$ must learn a starting point for the estimate of the correctness labels, which results optimal for the model.

5.5.6 Interpretations on the Attention Mechanism

It was found the original ASIM performed a form of phoneme detection on the sequence encoding. Therefore, the same attention weight normalization was used to observe the behaviour of the self-attention modules. The normalized attention weights for $\mathbf{L_A}$ and $\mathbf{L_b}$ are plotted in Figure 5.3 for the same example of the word *January*. The top plot corresponds to the $\mathbf{L_A}$ subnetwork. The attention curve in blue is plotted along the phoneme alignment at frame level on the x-axis marked with vertical dotted lines. The orange line corresponds to the MAX annotation reference, for which a high position indicates the current phoneme was marked as correct and a low position means otherwise. The plot in the middle of Figure 5.3 corresponds to the attention weights of the $\mathbf{L_b}$ subnetwork for *(a2)*. Finally, the plot at the bottom shows the attention weights and annotation for *(a3)*.

The first thing to observe in Figure 5.3 is that phoneme segmentation is not as evident as in the original ASIM design (Figure 3.4). The attention curves for $\mathbf{L_A}$ and $\mathbf{L_b}$ follow a similar trend, with spikes indicating the subnetworks focus differently on the same observation. The similarity in trends indicates redundancy in the networks. Since both sequential encoders observe the same features and have no more interaction with each other than a summation

Table 5.4: Precision (P), recall (R), F1 score and Cohen's Kappa (κ) for A on detecting segments with mispronunciation for each assessor η and the MAX reference.

	Train				Test			
η	P	R	F1	κ	P	R	F1	κ
a1	0.5249	0.5844	0.5530	0.1668	0.4592	0.5873	0.5154	0.1353
a2	0.3406	0.5998	0.4345	0.1598	0.2665	0.5929	0.3677	0.1139
a3	0.6815	0.5542	0.6113	0.1016	0.6498	0.5588	0.6008	0.0803
MAX	0.5234	0.6028	0.5603	0.2013	0.4526	0.6071	0.5186	0.1664

of their classifier output logits, the similarity can be caused by the data itself. This indicates the possibility of further improvements on the architecture to implement the assessor model in Equation (5.5).

The attention plots for $\mathbf{L_b}$ show points of disagreement between annotators for the same input. As an example, notice the difference in the orange line for the /uw/ segment for $\mathbf{L_b}^{(a2)}$ and $\mathbf{L_b}^{(a3)}$. For *a2*, /uw/ was correctly pronounced and a local maximum is present in the attention curve. On the other hand, the attention curve for *a3* shifted completely to a local minimum while the assessor marked a mispronunciation. Besides the location of the disagreement between *a2* and *a3*, the attention weights behave practically the same.

5.5.7 Summary

A model for a L2 pronunciation assessor was proposed and implemented using the DASIM. The assessor model consists of an assessor-independent scoring function affected by an additive assessor-dependent bias function. The DASIM comprised two ASIM networks trained simultaneously, each corresponding to the components of the assessor model. The INA set of L2 prompted speech was used, as each example was annotated for mispronunciation by three trained phoneticians. The bias subnetwork in the DASIM was sensitive to assessor identity. Also, the bias output was not the major contributor to the final correctness label posteriors. The contribution of each subnetwork reflected the inter-assessor reliability from the data. The assessor-independent output was closer to assessor *a3* than to the MAX annotation reference, often assumed to be assessor-independent. The attention weights in the subnetworks do not show a precise phoneme alignment as in the original ASIM, yet they indicate the location of assessor disagreements in a speech segment.

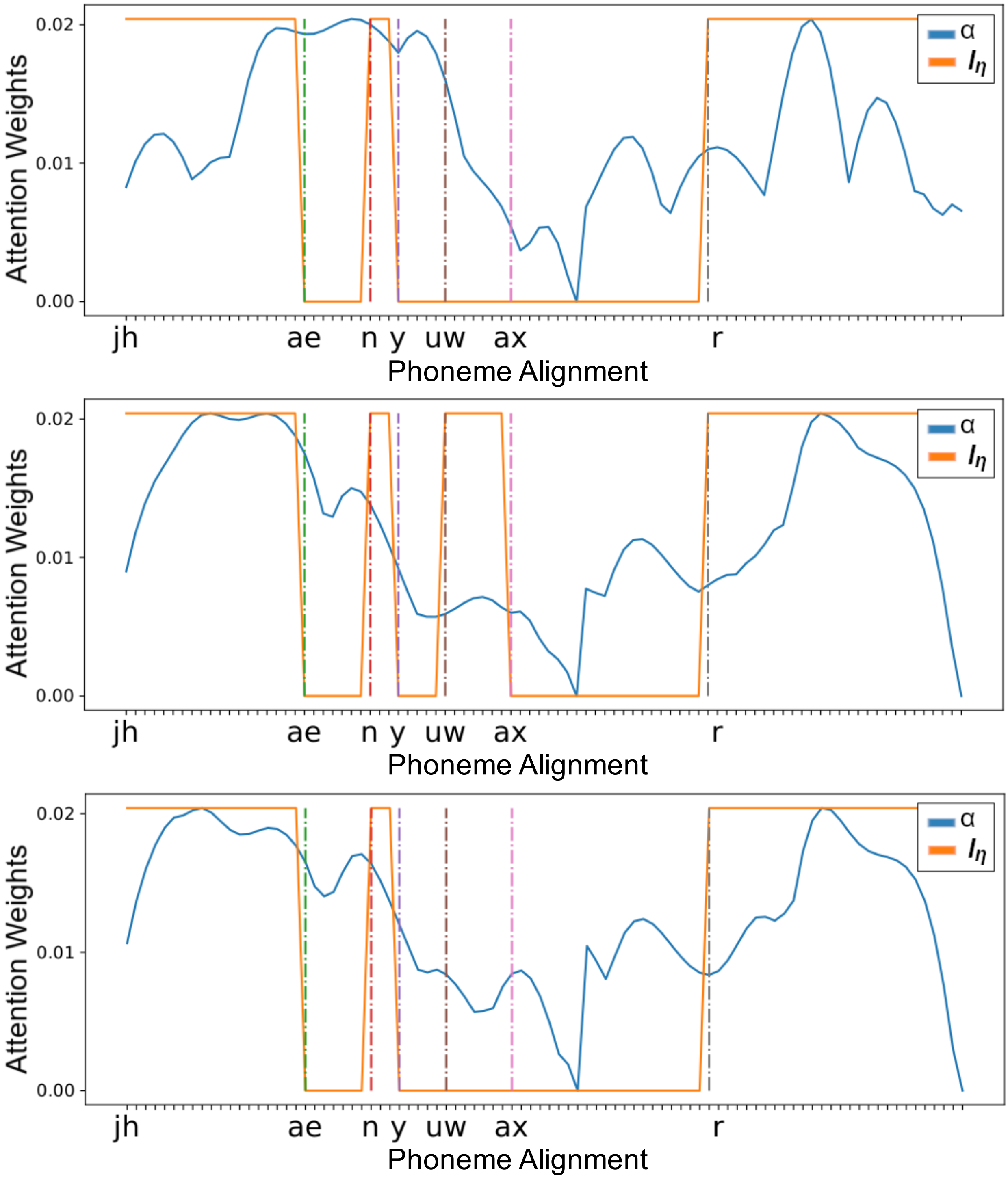


Figure 5.3: Normalized attention curves for $\mathbf{L_A}$ (top), $\mathbf{L_b}^{(a2)}$ (mid) and $\mathbf{L_b}^{(a3)}$ (bottom). The MAX reference is used for $\mathbf{L_A}$. The normalized attention weights α appear in blue while the orange line indicates the correctness label $\mathbf{l}$ given η.

5.6 An Efficient Architecture for the Assessor Model

The assessor model defined in Equation (5.5) was originally implemented using two simultaneous ASIMs. The two subnetworks for detecting mispronunciations did not interact with each other more than the arithmetic sum of their output logits before normalization, as shown in Figure 5.2. The DASIM showed some specialization of its assessor-independent logits $\mathbf{L_A}$ and the bias logits $\mathbf{L_b}$. While $\mathbf{L_A}$ is the main contributor to the correctness posterior $P(\hat{\mathbf{I}}|\mathbf{r}, \mathbf{O}^{(w)}, \eta)$, $\mathbf{L_b}$ adjusts the final logits given the assessor to better match the observed labels (see Table 5.3).

A close look into the attention mechanisms for both $\mathbf{L_A}$ and $\mathbf{L_b}$ revealed the DASIM was highly redundant. The behaviour of their attention modules shown in Figure 5.3 indicate both subnetworks process the same input similarly, hence a more efficient architecture should be designed.

The bias subnetwork in the DASIM processed the acoustic features of segment $\mathbf{O}^{(w)}$ and the assessor tag η as a standalone calculation. If $\mathbf{L_b}$ is also made dependent on the assessor-independent $\mathbf{L_A}$ besides $\mathbf{O}^{(w)}$ and η, the $\mathbf{L_b}$ subnetwork can take advantage of any prior processing done in the sequential encoding $EC(\mathbf{O}^{(w)})$. It is desired that most of the model for mispronunciation detection remains assessor-independent as the bias should not be the main driver in the final decision on correctness (Lindemann, 2017). The $\mathbf{L_b}$ logits are also likely to increase their specialization on annotation disagreements if the subnetwork can actually use $\mathbf{L_A}$ as a starting point. A more specialized and precise bias model means a more assessor-independent $\mathbf{L_A}$.

A reinterpretation of Equation (5.5) allows the bias component to observe the assessor-independent assessment. Any modification to the assessor model should not impose more assumptions on the assessor-independent scoring function A. Therefore, only a modification of how the bias component is generated is proposed in this section.

The bias function in the assessor model is augmented by using the independent assessment as input. Equation (5.5) is changed to

$$A_\eta(\mathbf{O}) = A(\mathbf{O}) + b_\eta(A(\mathbf{O})) \tag{5.10}$$

In regard to the ASIM architecture, Equation (5.10) can be implemented by using a single encoding $EC(\mathbf{O}^{(w)})$ and passing $\mathbf{L_A}$ to the bias FFN classifier FFN_b. A first modification to the DASIM uses the same $EC(\mathbf{O}^{(w)})$ for both assessor-independent FFN_A and bias FFN_b. This configuration is called Attention-Based Segmental Incorrectness Model - Configuration 1 (ASIM1), and it is shown in Figure 5.4. ASIM1 aims to reduce the number of parameters required to learn $P(\hat{\mathbf{I}}|\mathbf{r}, \mathbf{O}^{(w)}, \eta)$, yet it does not directly use $\mathbf{L_A}$.

The logits $\mathbf{L_A}$ are included along with the inputs for FFN_b in Attention-Based Segmental Incorrectness Model - Configuration 2 (ASIM2). Figure 5.5 shows ASIM2 using a connection from FFN_A to FFN_b, defining the bias posteriors as $P(\hat{\mathbf{I_b}}|\mathbf{r}, \mathbf{O}^{(w)}, \hat{\mathbf{I_A}}, \eta)$. In ASIM2, the FFN_b classifier still depends directly on $EC(\mathbf{O}^{(w)})$. The sequential encoding would be optimized to

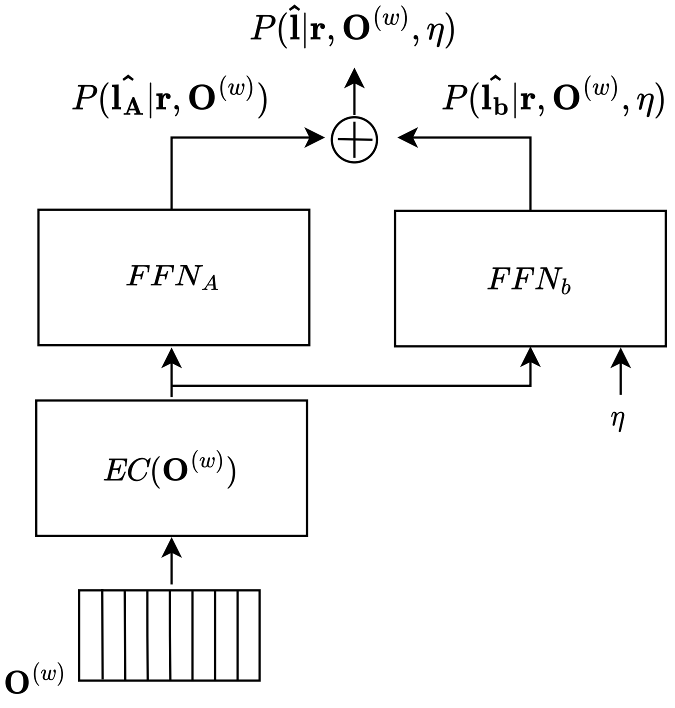

$P(\hat{\mathbf{l}}|\mathbf{r}, \mathbf{O}^{(w)}, \eta)$
$P(\hat{\mathbf{l}_\mathbf{A}}|\mathbf{r}, \mathbf{O}^{(w)})$
$P(\hat{\mathbf{l}_\mathbf{b}}|\mathbf{r}, \mathbf{O}^{(w)}, \eta)$
FFN_A
FFN_b
η
$EC(\mathbf{O}^{(w)})$
$\mathbf{O}^{(w)}$

Figure 5.4: ASIM1 architecture.

feed both $\mathbf{L_A}$ and $\mathbf{L_b}$.

The Attention-Based Segmental Incorrectness Model - Configuration 3 (ASIM3) model has the most assessor-independent $EC(\mathbf{O}^{(w)})$ by design. ASIM3 estimates the bias posterior as $P(\hat{\mathbf{l}_\mathbf{b}}|\mathbf{r}, \hat{\mathbf{l}_\mathbf{A}}, \eta)$. It is also likely $\mathbf{L_b}$ specializes more in adjusting the assessor-independent posteriors to match the observed labels. Notice none of the modified ASIM designs interfere with the dependencies of $\mathbf{L_A}$. The final interaction between $\mathbf{L_A}$ and $\mathbf{L_b}$ remains a simple addition for the sake of interpreting the bias as an offset value from the *ideal* assessor-independent PA.

5.7 Experiment on Multiple ASIM Designs for Mispronunciation Detection

The three modifications to the DASIM were tested for detecting mispronounced segments on the INA dataset of L2 speech from young learners of English as L2 in the Netherlands (see Section 3.5). The objective of this experiment was to observe if the proposed modifications to the DASIM could reach the same or better performance with fewer parameters. Therefore, all models were scored using the F1 score and Cohen's Kappa (κ) on declaring an acoustic segment mispronounced given the assessor reference. Additionally, any changes in the contribution of $\mathbf{L_A}$ and $\mathbf{L_b}$ to the final logits would be informative on how the bias interacts with the assessor-independent outputs.

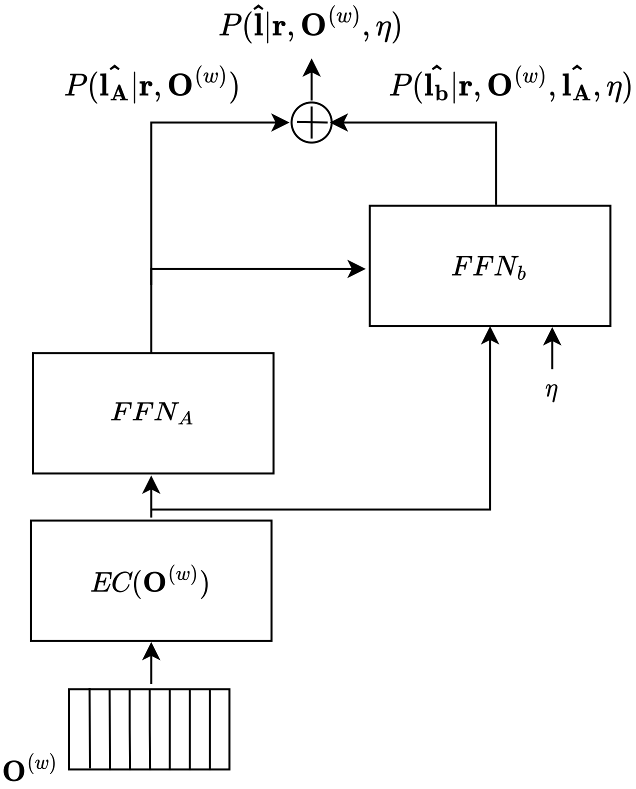

$P(\hat{\mathbf{l}}|\mathbf{r}, \mathbf{O}^{(w)}, \eta)$
$P(\hat{\mathbf{l_A}}|\mathbf{r}, \mathbf{O}^{(w)})$
$P(\hat{\mathbf{l_b}}|\mathbf{r}, \mathbf{O}^{(w)}, \hat{\mathbf{l_A}}, \eta)$
FFN_b
η
FFN_A
$EC(\mathbf{O}^{(w)})$
$\mathbf{O}^{(w)}$

Figure 5.5: ASIM2 architecture.

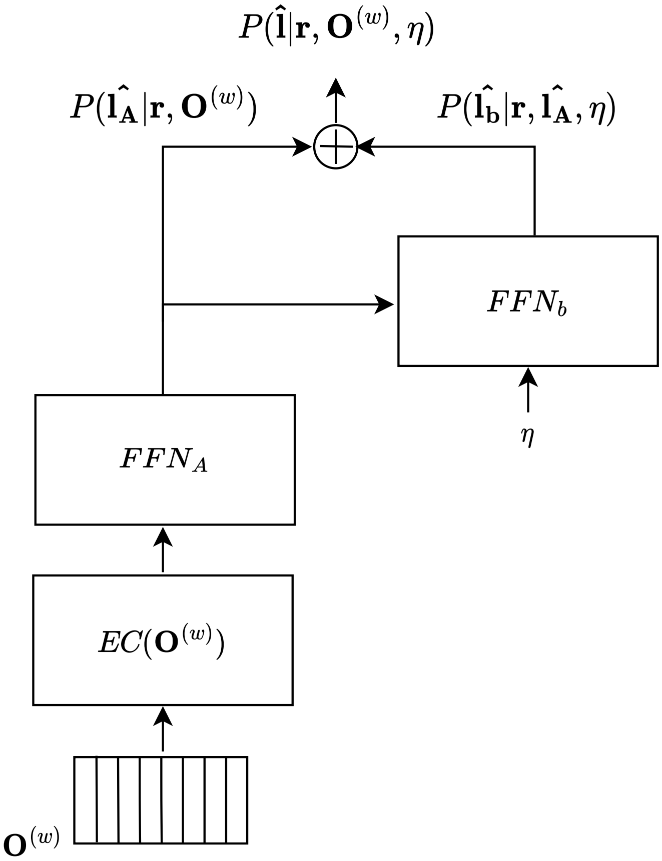

$P(\hat{\mathbf{l}}|\mathbf{r}, \mathbf{O}^{(w)}, \eta)$
$P(\hat{\mathbf{l_A}}|\mathbf{r}, \mathbf{O}^{(w)})$
$P(\hat{\mathbf{l_b}}|\mathbf{r}, \hat{\mathbf{l_A}}, \eta)$
FFN_b
η
FFN_A
$EC(\mathbf{O}^{(w)})$
$\mathbf{O}^{(w)}$

Figure 5.6: ASIM3 architecture.

5.7.1 Model Training Setup

The models ASIM1, ASIM2, ASIM3 have all similar components with differences only in how said components connect with each other. The main components for each model are the sequential encoding $EC(\mathbf{O}^{(w)})$ and the classifier *FFN*. All three ASIM variations have an $EC(\mathbf{O}^{(w)})$ consisting of BDLSTM layers of size 64 and an additive self-attention module with linear weights of size 128. The *FFN* are all FFN classifiers which estimate either $\mathbf{L_A}$ or $\mathbf{L_b}$. The output layer of every *FFN* holds two binary outputs representing either the correct or incorrect utterance of a phoneme class given the annotation reference.

A DASIM holding 6 BDLSTM layers for the encoding section and 6-layer deep FFNs was used as a baseline model, the same as the experiment at Section 5.5.1. The ASIM variations were tested with different amounts of layers for both the encoding BDLSTM and the *FFN* classifiers. The models ASIM1, ASIM2 and ASIM3 were tested each with 3, 6 and 8 layers for both BDLSTM and FFNs. Table 5.5 summarizes the parameter count for the different ASIM configurations. The models with the 8-layer configuration hold a parameter count greater than the baseline. The objective of the experiment is to reduce the number of parameters without sacrificing much of the model's performance. However, the 8-layer configuration shows if the new architectures benefit enough from their design to keep improving their performance via additional parameters.

Table 5.5: Number of parameters for the DASIM and the ASIM variations given the number of layers for both BDLSTM and FFNs.

Model	Layers		
	3	6	8
DASIM	-	24,729K	-
ASIM1	17,539K	24,135K	28,531K
ASIM2	17,635K	24,230K	28,628K
ASIM3	11,212K	17,808K	22,205K

All four models were trained using the first 13 PLP coefficients with their first and second-order time differentials as acoustic feature vectors. The one-hot encoding of the assessor tag η was used only for FFN_b concatenated to the input, as defined in each ASIM variation. The dDASIM kept η attached to the PLP input vectors as constant dimensions.

All models were trained on the three INA assessors simultaneously. The $EC(\mathbf{O}^{(w)})$ and FFN_A on the three ASIM variations did observe the same input three times as the bias component FFN_b did. The baseline kept an assessor-independent $EC(\mathbf{O}^{(w)})_A$ and a bias $EC(\mathbf{O}^{(w)}, \eta)_b$.

5.7.2 Experiment Dataset

This experiment on the ASIM variations used the same INA split as defined in Section 3.6.1. The speaker balance and non-overlap criteria are kept limiting any advantage to the models. The alignment for $\mathbf{r}$ is the same used for the original GOP baseline in Section 3.6.3.

Table 5.6: F1 score and Cohen's Kappa (κ) for the ASIM variations on detecting mispronounced segments given all assessors in the INA dataset.

		Train		Test	
Model	Layers	F1	κ	F1	κ
DASIM		0.8103	0.6536	0.7224	0.5322
ASIM1	3	0.8049	0.6435	0.7030	0.5003
	6	0.8117	0.6561	0.7243	0.5335
	8	0.7988	0.6321	0.7186	0.5280
ASIM2	3	0.8006	0.6355	0.7056	0.5032
	6	0.8032	0.6403	0.7249	0.5337
	8	0.8022	0.6408	0.7192	0.5312
ASIM3	3	0.8020	0.6381	0.7254	0.5338
	6	0.8119	0.6565	0.7434	0.5616
	8	0.8075	0.6484	0.7433	0.5602

5.7.3 Performance on Detecting Mispronounced Segments.

The models were scored on all three INA assessors combined to have an overview of the performance of the whole dataset. Table 5.6 shows the overall F1 score and Cohen's Kappa (κ) on detecting mispronounced segments given the three INA assessors. The models with 3 layers could not outperform the DASIM. Meanwhile, the 6-layer models remain the closest to the baseline. At first glance, the difference in performance was not prominently large. The reduction in parameters from using a single sequential encoder and attention module did not decrease the performance of all 6-layer networks below the baseline on the Test set. ASIM1 and ASIM2 with 6 layers showed the most similar results to DASIM. The 6-layer ASIM3 showed the most improvement on the Test set. All versions of ASIM3 improved generalization; however, the 6-layer ASIM3 managed to increase the performance metrics by 3% on the Test set while reducing by 30% the number of parameters used in the DASIM.

The models with 8 layers did not necessarily improve the performance on the Test set. In the case of ASIM1, the 8-layer model was outperformed by the baseline. For ASIM2 and ASIM3, the architectures did not improve the metrics of their 6-layer counterpart, although the 8-layer ASIM3 still outperformed the baseline. For further analysis of the ASIM variations, ASIM1, ASIM2 and ASIM3 hereby refer to their 6-layer configurations as these represent a peak in performance on the INA set.

5.7.4 Interaction Between Assessor-Independent and Bias Components

It is desired that the bias contributes the least possible to the assessment while better matching the observed annotation. The re-design of the DASIM aims to avoid an equal contribution between the logits $\mathbf{L_A}$ and $\mathbf{L_b}$, hence causing the specialization of the network components. To illustrate the behaviour of the assessor-independent and bias outputs of the DASIM and the 6-layer ASIM variations, both $\mathbf{L_A}$ and $\mathbf{L_b}$ for the INA set were passed through a sigmoid regularization (see Table 2.3) and averaged per output class. The naming convention for the output classes is $\{ph\}\_\{l\}$, where ph is the ARPABET phoneme label as defined in (Rice, 1976) and l is the binary correctness label where $l = 1$ means a correct pronunciation.

The averaged regularized outputs of the DASIM hold as a baseline are plotted in Figure 5.7. The plots show the mean regularized η-independent outputs ($\mu_{\hat{\mathbf{I_A}}}$) with their respective standard deviation ($s_{\hat{\mathbf{I_A}}}$) as a vertical black line. The corresponding mean bias outputs ($\mu_{\hat{\mathbf{I_b}}}$) are shown in the plot below as red bars, with their standard deviation ($s_{\hat{\mathbf{I_b}}}$) indicated by a black vertical line. Overall, $\mu_{\hat{\mathbf{I_b}}}$ showed greater values than $\mu_{\hat{\mathbf{I_A}}}$. The overall $s_{\hat{\mathbf{I_A}}} = 3.26$ was larger than the bias counterpart $s_{\hat{\mathbf{I_b}}} = 2.19$. The plots in Figure 5.7 show that $\hat{\mathbf{I_A}}$ and $\hat{\mathbf{I_b}}$ do not follow the same posterior distribution, confirming the relevance of η in FFN_b. The difference between Train and Test averaged outputs was not prominent. It can be concluded the zero-speaker overlap in the set did not affect the behaviour of the model considerably. Certain output classes did elicit either a far more positive or a more negative logit average than most class outputs. Most of the outputs with larger mean values correspond to English phonemes with a direct or similar equivalent; in Dutch, this is the case of /ax/, /ih/, /n/, /r/, /s/ and /t/ (Tops et al., 2001). Most of the averaged $\hat{\mathbf{I_b}}$ were larger than their assessor-independent counterparts. A shift in the logits with the larger positive logits was observed in the ASIM variations presented in Figures 5.7 to 5.10.

The mean and variance plots for the regularized outputs of ASIM1 appear in Figure 5.8. The phoneme classes with the highest logits were the same observed in the DASIM. The use of a shared sequential encoder $EC(\mathbf{O}^{(w)})$ did increase the number of output classes in FFN_A with the most positive logits between $\mathbf{L_A}$ and $\mathbf{L_b}$ from 0 to 31 out of 94. The absolute mean value in $\hat{\mathbf{I_A}}$ increased from 0.06 to 0.07 while the highest mean value in $\hat{\mathbf{I_b}}$ decreased from 0.14 to 0.08. Overall, a single $EC(\mathbf{O}^{(w)})$ did make the contribution of FFN_A and FFN_b more even. From the classes outputs with the highest positive logits in $\mathbf{L_A}$, 61% correspond to phonemes prone to mispronunciation by Dutch speakers of English as L2 such as /ah/, /ao/, /b/, /ch/, /dh/, /eh/, /hh/, /ih/, /jh/, /n/, /p/, /r/, /s/ and /t/ among others (Tops et al., 2001). It was likely said phonemes were mispronounced with such a distinction the network did not need to consider the tag η.

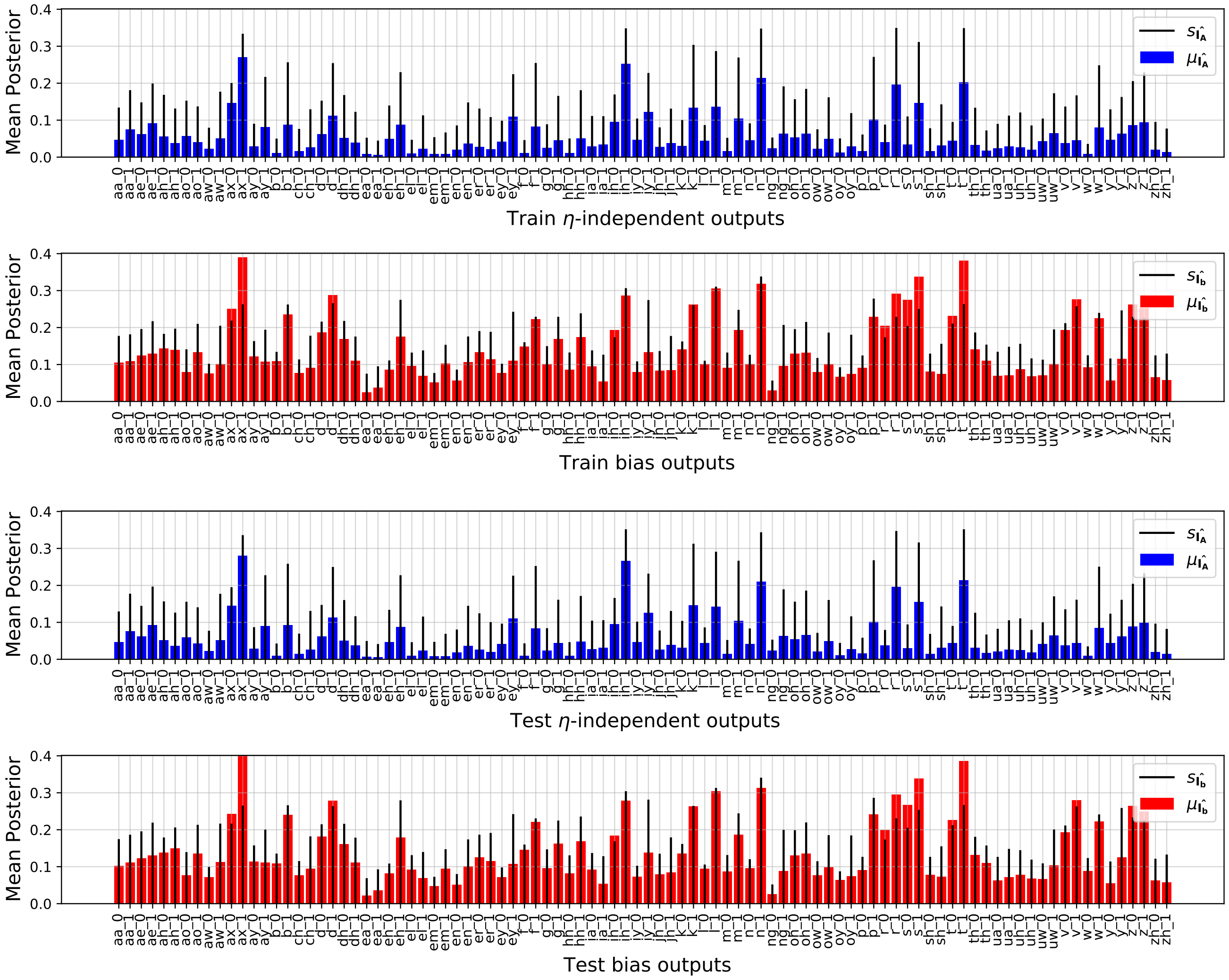


Figure 5.7: The η-independent mean outputs $\mu_{\hat{\mathbf{I_A}}}$ of the DASIM are shown in blue with their standard deviation $s_{\hat{\mathbf{I_A}}}$ as black lines. The mean bias outputs $\mu_{\hat{\mathbf{I_b}}}$ are shown as red bars with their standard deviation $s_{\hat{\mathbf{I_b}}}$ as black lines.

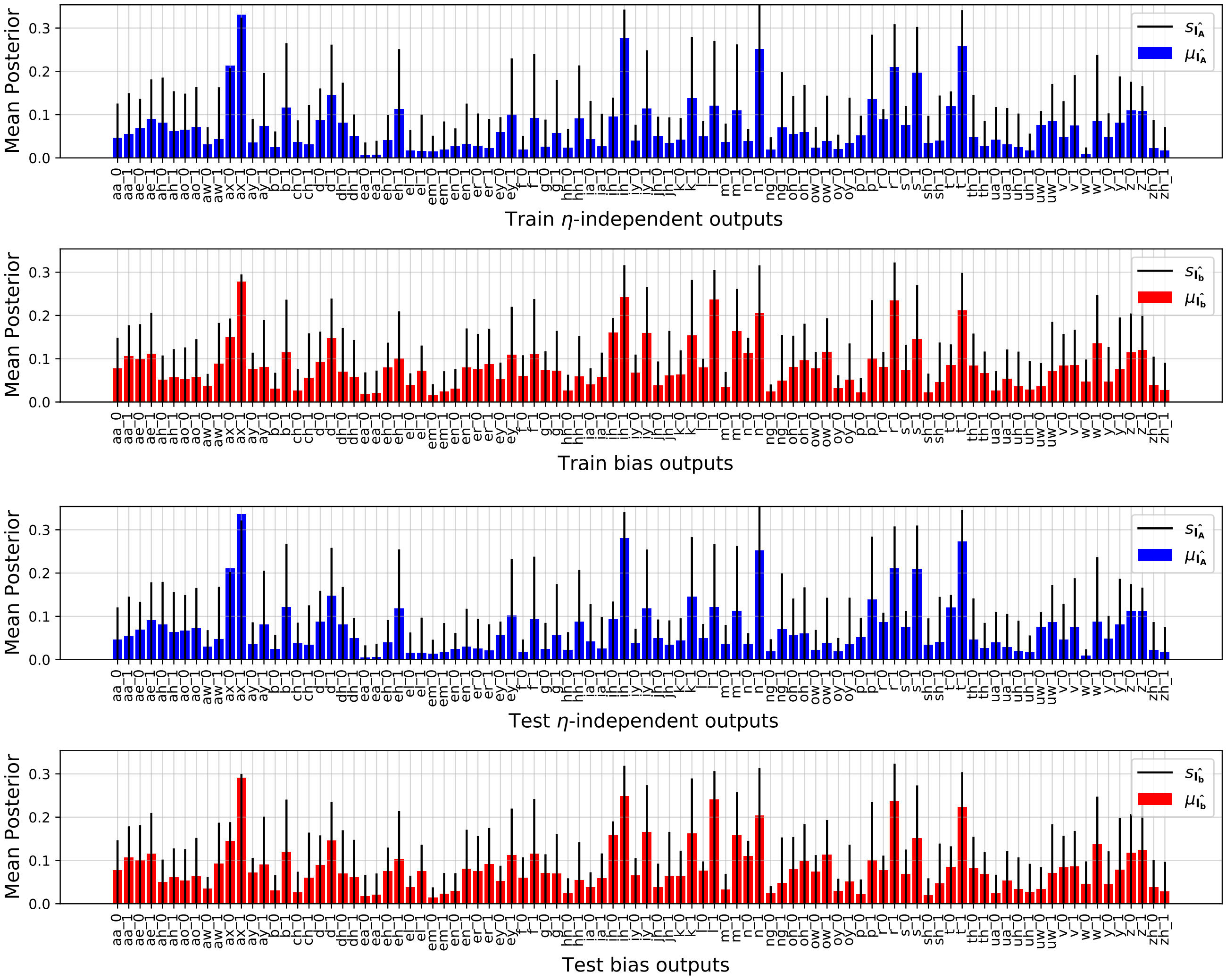


Figure 5.8: The η-independent mean outputs $\mu_{\hat{\mathbf{I_A}}}$ of the ASIM1 are shown in blue with their standard deviation $s_{\hat{\mathbf{I_A}}}$ as black lines. The mean bias outputs $\mu_{\hat{\mathbf{I_b}}}$ are shown as red bars with their standard deviation $s_{\hat{\mathbf{I_b}}}$ as black lines.

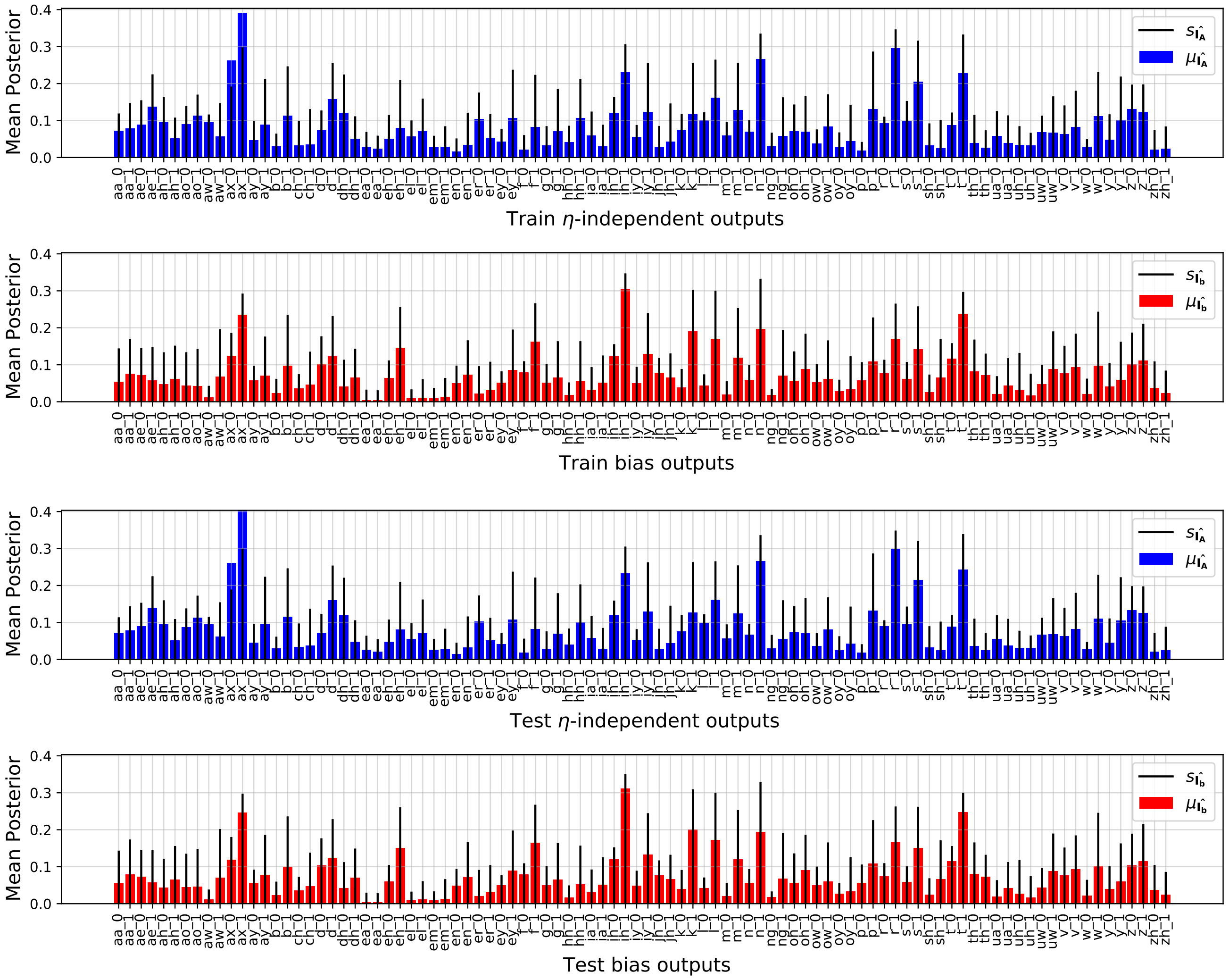


Figure 5.9: The η-independent mean outputs $\mu_{\hat{\mathbf{I}}_\mathbf{A}}$ of the ASIM2 are shown in blue with their standard deviation $s_{\hat{\mathbf{I}}_\mathbf{A}}$ as black lines. The mean bias outputs $\mu_{\hat{\mathbf{I}}_\mathbf{b}}$ are shown as red bars with their standard deviation $s_{\hat{\mathbf{I}}_\mathbf{b}}$ as black lines.

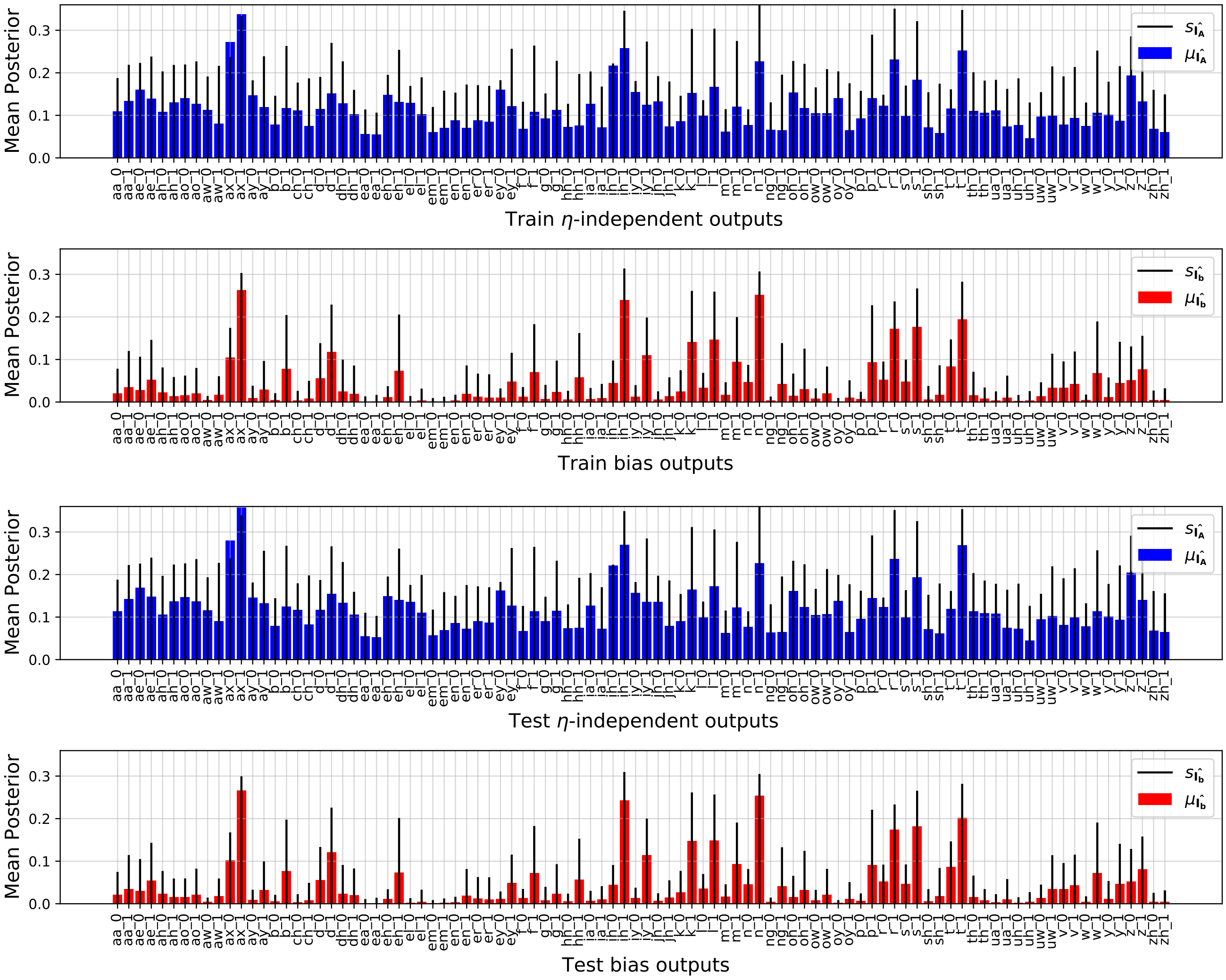


Figure 5.10: The η-independent mean outputs $\mu_{\hat{\mathbf{I_A}}}$ of the ASIM3 are shown in blue with their standard deviation $s_{\hat{\mathbf{I_A}}}$ as black lines. The mean bias outputs $\mu_{\hat{\mathbf{I_b}}}$ are shown as red bars with their standard deviation $s_{\hat{\mathbf{I_b}}}$ as black lines.

Table 5.7: Inter-annotation agreement (I) and Cohen's κ) for the highest mean correct phoneme outputs $\mathbf{L_b}$ in ASIM3.

Phoneme	IPA	I	κ	Count
ax	ə	0.57	0.21	677
ih	ɪ	0.81	0.45	515
k	k	0.89	0.53	238
l	l	0.92	0.19	264
n	n	0.95	0.45	375
r	r	0.70	0.14	364
s	s	0.87	0.48	327
t	t	0.78	0.23	479

The plots for the mean regularized outputs and variances for $\mathbf{L_A}$ and $\mathbf{L_b}$ in ASIM2 are shown in Figure 5.9. The ASIM2 variation kept decreasing the overall mean output $\mathbf{L_b}$. The number of output classes for which $\mathbf{L_A}$ was the major average contributor to the final $P(\hat{\mathbf{I}}|\mathbf{r}, \mathbf{O}^{(w)}, \eta)$ increased to 56. Looking back to the baseline mean output distribution in Figure 5.7, the reduction in the processing load of FFN_b did shift most of the contribution to the final decision to the assessor-independent $\mathbf{L_A}$. The proportion of phonemes prone to mispronunciation with $\mathbf{L_A}$ as their maximum contributor was kept at 62%. The network components FFN_A and FFN_b did not show a tendency for a higher response for either correct or incorrect phoneme outputs. Neither a clear correlation was found between the inter-reliability κ for a phoneme class and whether $\hat{\mathbf{I_A}}$ or $\hat{\mathbf{I_b}}$ showed a higher response for said phoneme.

The mean regularized output plots and variance for ASIM3 in Figure 5.10 showed the smallest average $\hat{\mathbf{I_b}}$ in this experiment. The output distribution for ASIM3 also showed large mean values for outputs corresponding to the correct pronunciation of /ax/, /ih/, /k/, /l/, /n/, /r/, /s/, /t/. The only similarity between the listed phonemes is that they show a relatively high count when the average phoneme class occurrence in INA is 148. Said phonemes also have a high inter-annotation agreement (I). However, a large I and a high count do not necessarily mean a high Cohen's κ as shown in Table 5.7. As a quick reference, the Pearson correlation coefficient of a linear regression between phoneme class occurrences in annotation and κ was 0.27. It is not easy to explain the output tendency in $\mathbf{L_b}$ based on counts or inter-reliability. Half of the phoneme classes in INA have less than 100 occurrences in the annotation, yet some of them still show κ values greater than 0.3.

The average logit contribution for all models in this experiment was computed using the $\mathcal{C}_j$ metric on the sum of squared logits as defined in Equation (5.9). The $\mathcal{C}_j$ percentages shown in Table 5.8 showed a major shift from $\mathbf{L_A}$ to $\mathbf{L_b}$ as the component with larger logits. The effect observed in Table 5.8 can be explained by observing the actual logit distribution on the ASIM variations.

The average logits from both $\mathbf{L_A}$ and $\mathbf{L_b}$ in the DASIM are shown in Figure 5.11. The average assessor-independent outputs are shown as blue bars, while the bias mean outputs are shown as red bars in the plot underneath. The standard deviation for each mean logit output is also shown as a vertical black line for each output class. In the same format, Figure

Table 5.8: Contribution percentages from the assessor independent $\mathbf{L_A}$ or the bias $\mathbf{L_b}$ logits in the ASIM variations.

	Train		Test	
η	$\mathbf{L_A}$	$\mathbf{L_b}$	$\mathbf{L_A}$	$\mathbf{L_b}$
DASIM	71.57	28.43	70.95	29.05
ASIM1	55.13	44.83	54.89	45.11
ASIM2	46.55	53.45	46.54	53.46
ASIM3	31.20	68.80	31.32	68.68

5.12 plots the output logits statistics for ASIM1. Figure 5.13 does it for ASIM2, and Figure 5.14 does it for ASIM3.

The red bar plots from Figures 5.11 to 5.14 show the $\mathbf{L_b}$ average outputs became progressively more negative while the average $\mathbf{L_A}$ became less negative. Table 5.9 shows both the overall mean (μ) and standard deviation (s) for $\mathbf{L_A}$ and $\mathbf{L_b}$ on every model in this experiment. The $s_{\mathbf{L_A}}$ and $\mu_{\mathbf{L_b}}$. The standard deviation in $\mathbf{L_b}$ did increase, while the deviation of $\mathbf{L_A}$ was reduced. The shift in behaviour between $\mathbf{L_A}$ and $\mathbf{L_b}$ shown particularly in ASIM3 indicates the bias acts as a gating function controlling an assessor-independent output given the assessor identity.

Since the logits are combined in a sum, the more negative $\mu_{\mathbf{L_b}}$ would likely push the final output closer to a probability of zero after the sigmoid regularization. The larger $s_{\mathbf{L_b}}$ also indicates the corresponding output is not pushing down $\mathbf{L_A}$ all the time. For the cases of the correct pronunciation outputs for the phonemes listed in Table 5.7, $s_{\mathbf{L_b}}$ affects them the least. The normalized logit plots in Figures 5.7 to 5.10 showed that there is rarely any positive logit output. Therefore, more than contributing to the final probability mass, the ASIM components are actually blocking the logits to reduce the final posteriors given the annotation reference.

Table 5.9: Standard deviation s and overall mean μ for $\mathbf{L_A}$ and $\mathbf{L_b}$ on each ASIM variation.

	Train				Test			
η	$\mu_{\mathbf{L_A}}$	$s_{\mathbf{L_A}}$	$\mu_{\mathbf{L_b}}$	$s_{\mathbf{L_b}}$	$\mu_{\mathbf{L_A}}$	$s_{\mathbf{L_A}}$	$\mu_{\mathbf{L_b}}$	$s_{\mathbf{L_b}}$
DASIM	-6.21	3.26	-3.15	2.19	-6.25	3.31	-3.23	2.23
ASIM1	-4.98	2.66	-4.32	2.51	-4.99	2.68	-4.36	2.54
ASIM2	-4.33	2.44	-4.84	2.70	-4.38	2.48	-4.89	2.73
ASIM3	-3.60	2.44	-6.59	2.94	-3.59	2.48	-6.60	3.01

5.7.5 Normalization of the Bias Inputs.

Recall, the input for the bias FFN_b consists of the logits $\mathbf{L_A}$ concatenated to the tag η. In the case of ASIM2, $EC(\mathbf{O}^{(w)})$ is also part of the bias input. The logits range $(-\inf, \inf)$, η is a one-hot encoding with a zero-mean normalization and $EC(\mathbf{O}^{(w)})$ is normalized via a normalization layer (Ba et al., 2016). The use of logits as input for FFN_b is equivalent to skipping a layer of Rectifier Linear Units (RELUs). Therefore, the high response observed in certain output classes in both FFN_A and FFN_b was likely caused by the already large and non-regularized $\mathbf{L_A}$.

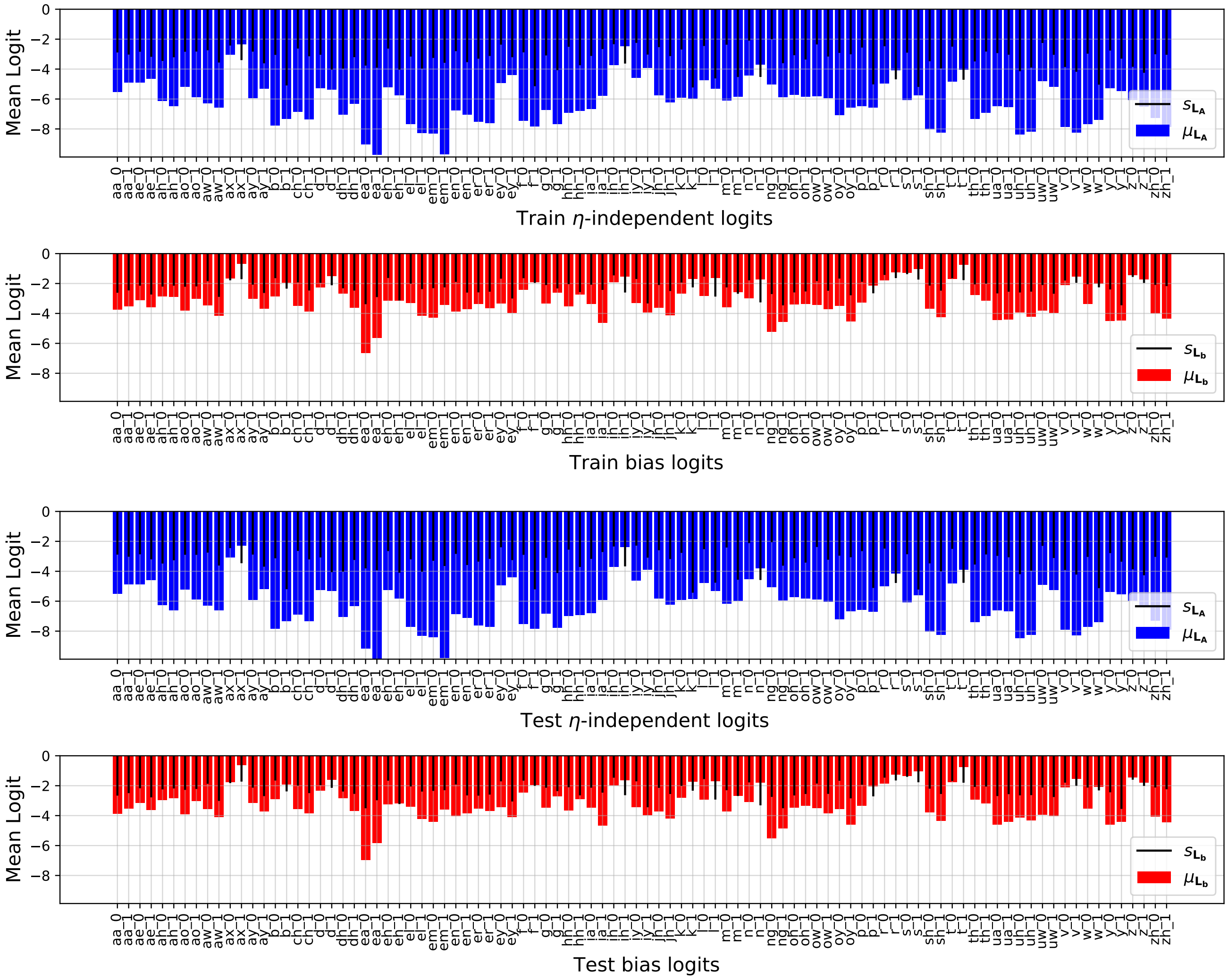


Figure 5.11: The η-independent mean logits $\mu_{\mathbf{L_A}}$ of the DASIM are shown in blue with their standard deviation $s_{\mathbf{L_A}}$ as black lines. The mean bias logits $\mu_{\mathbf{L_b}}$ are shown as red bars with their standard deviation $s_{\mathbf{L_b}}$ as black lines.

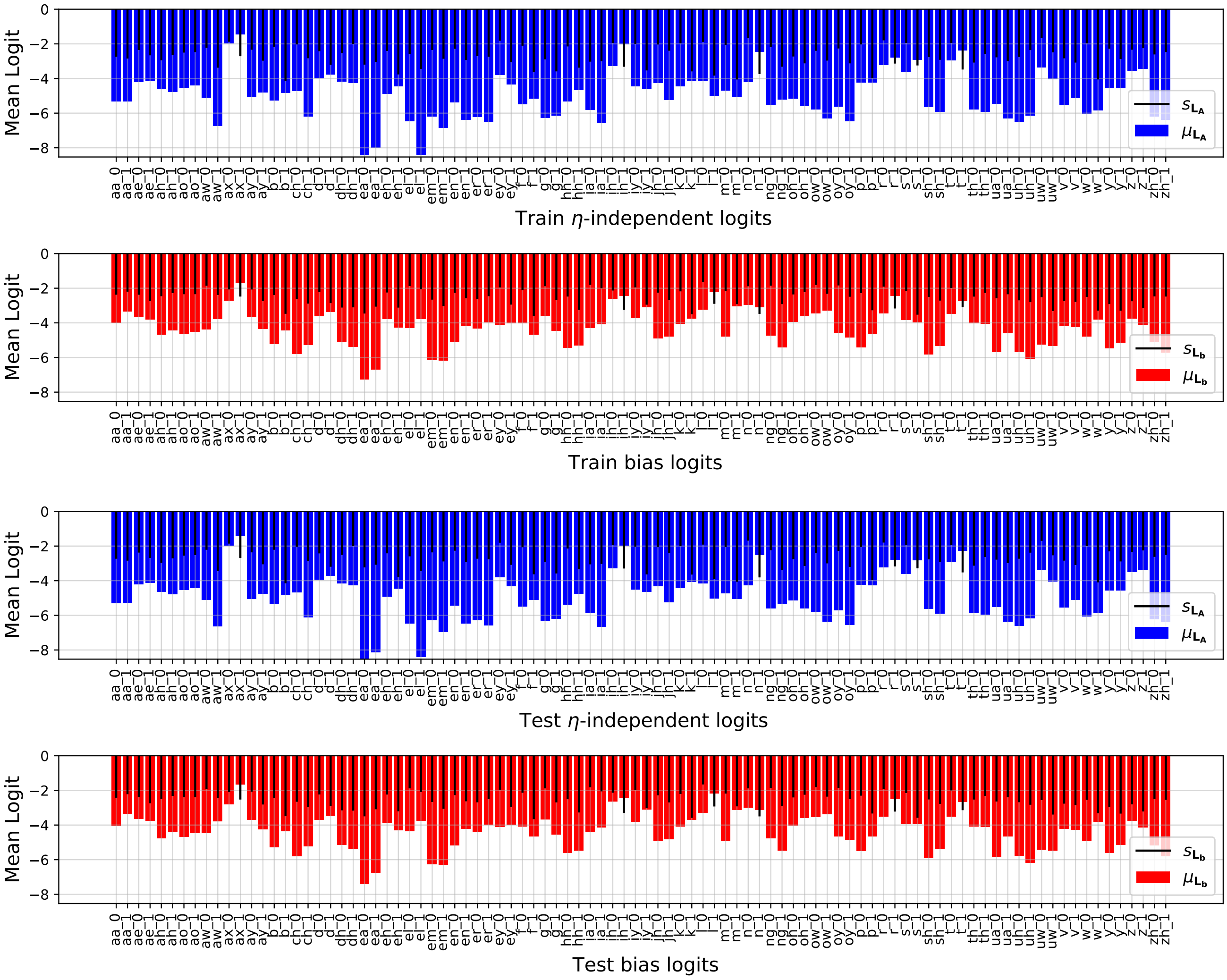


Figure 5.12: The η-independent mean logits $\mu_{\mathbf{L_A}}$ of the ASIM1 are shown in blue with their standard deviation $s_{\mathbf{L_A}}$ as black lines. The mean bias logits $\mu_{\mathbf{L_b}}$ are shown as red bars with their standard deviation $s_{\mathbf{L_b}}$ as black lines.

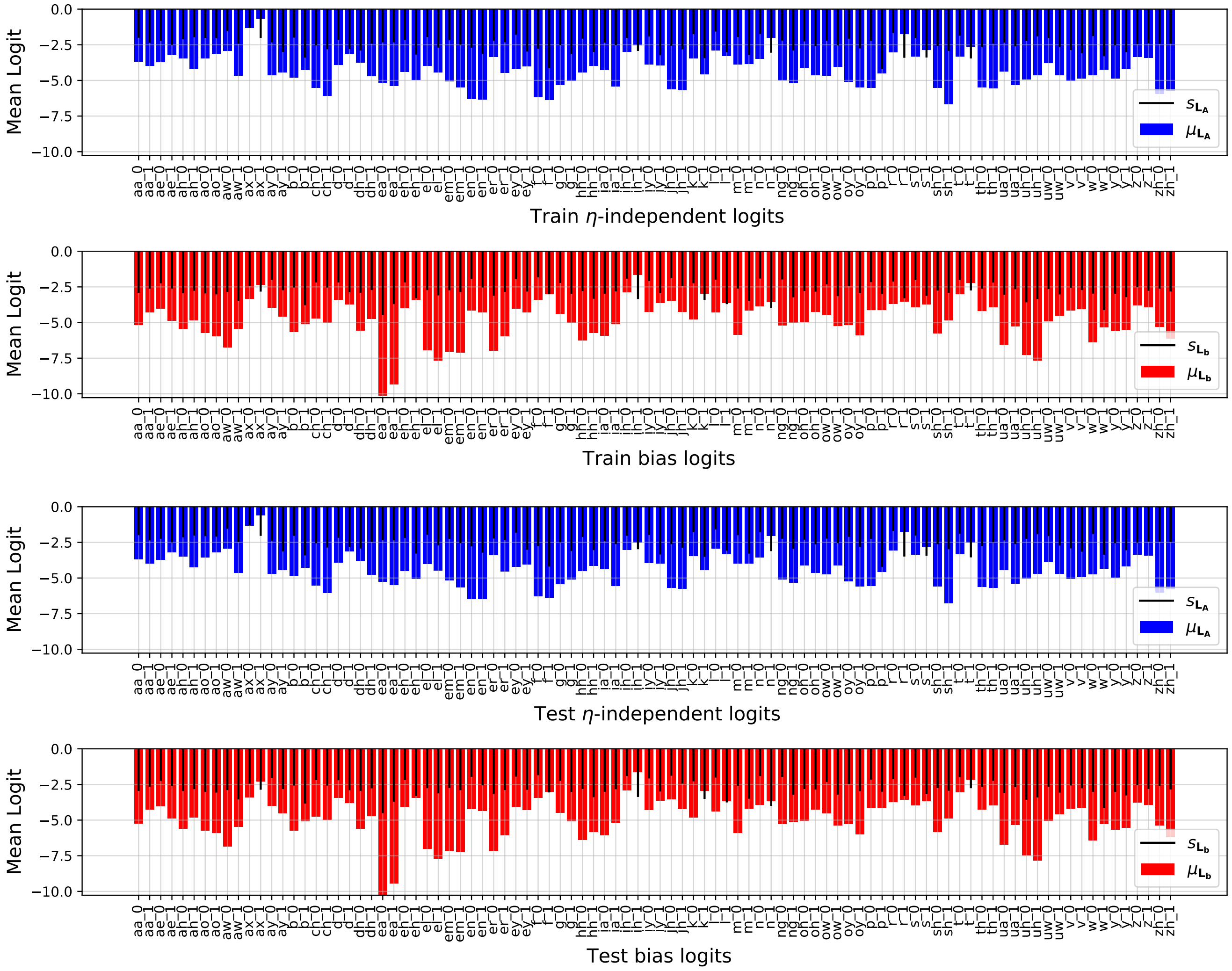


Figure 5.13: The η-independent mean logits $\mu_{\mathbf{L_A}}$ of the ASIM2 are shown in blue with their standard deviation $s_{\mathbf{L_A}}$ as black lines. The mean bias logits $\mu_{\mathbf{L_b}}$ are shown as red bars with their standard deviation $s_{\mathbf{L_b}}$ as black lines.

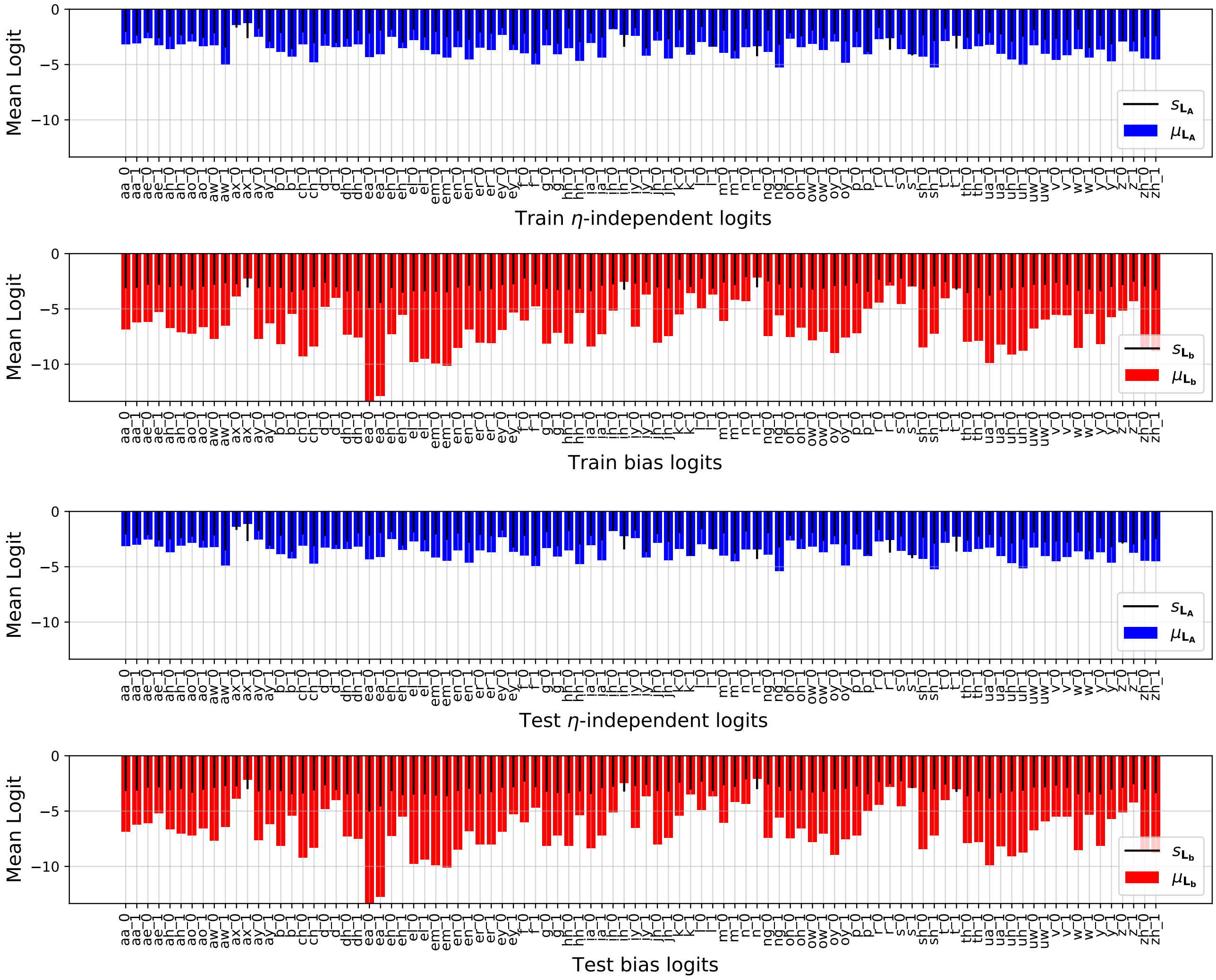


Figure 5.14: The η-independent mean logits $\mu_{\mathbf{L_A}}$ of the ASIM3 are shown in blue with their standard deviation $s_{\mathbf{L_A}}$ as black lines. The mean bias logits $\mu_{\mathbf{L_b}}$ are shown as red bars with their standard deviation $s_{\mathbf{L_b}}$ as black lines.

Table 5.10: F1 score and Cohen's Kappa (κ) for ASIM2 and ASIM3 variations on detecting mispronounced segments in the INA dataset.

	Train		Test	
Model	F1	κ	F1	κ
ASIM2	0.8032	0.6403	0.7249	0.5337
ASIM2N	0.8020	0.6381	0.7244	0.5326
ASIM2RN	0.6999	0.4462	0.6442	0.3832
ASIM3	0.8119	0.6565	0.7434	0.5616
ASIM3N	0.8038	0.6414	0.7387	0.5530
ASIM3RN	0.7118	0.4687	0.6550	0.4027

The architectures ASIM2 and ASIM3 were modified to use a normalization layer on $\mathbf{L_A}$ before being passed to FFN_b. The normalization occurred before concatenating $\mathbf{L_A}$ to η and $EC(\mathbf{O}^{(w)})$ as required. A RELU layer for $\mathbf{L_A}$ before its normalization was also tested with low expectations. The non-normalized $\mathbf{L_A}$ logits are still added to $\mathbf{L_b}$ as stated in the ASIM2 and ASIM3 architectures shown in Figures 5.5 and 5.6 respectively. The average logit plots in Figures 5.13 and 5.14 show the network components keeping mostly negative outputs. The use of a RELU layer before normalization could complicate the training of FFN_b as it will not observe the complete $\mathbf{L_A}$ output. Therefore, two variations for ASIM2 and ASIM3 were trained and evaluated using the same INA set with the purpose of observing a change in the behaviour of $\mathbf{L_A}$ and $\mathbf{L_b}$. Attention-Based Segmental Incorrectness Model - Configuration 2 with Normalization (ASIM2N) and Attention-Based Segmental Incorrectness Model - Configuration 3 with Normalization (ASIM3N) used only the normalization layer on $\mathbf{L_A}$. Attention-Based Segmental Incorrectness Model - Configuration 2 with Regularization and Normalization (ASIM2RN) and Attention-Based Segmental Incorrectness Model - Configuration 3 with Regularization and Normalization (ASIM2RN) used both a RELU and a normalization layer on $\mathbf{L_A}$ before being passed to FFN_b.

The modified ASIM2 and ASIM3 architectures were scored for detecting mispronounced segments using *F1* score and inter-reliability $\boldsymbol{\kappa}$. The performance metrics are shown in Table 5.10. The models without normalization or regularization layers are included in Table 5.10 as a baseline. The use of RELU in $\mathbf{L_A}$ logits decreased the performance of the model considerably, even below the performance of the DASIM (Table 5.6). The models using RELU were at a disadvantage since FFN_b had to complement a vector which was not able to observe, similar to the DASIM. The models using the normalization layer also showed a decrease in their metrics, yet it was less than 1%. The output contribution metric $\mathcal{C}_j$ (Equation 5.9) was also obtained for the models' $\mathbf{L_A}$ and $\mathbf{L_b}$. The contribution $\mathcal{C}_j$ is shown in Table 5.11 for the modified ASIM2 and ASIM3, along with their respective baselines. The models using only the normalized $\mathbf{L_A}$ kept $\mathbf{L_b}$ as the major contributor, similar to the original ASIM2 and ASIM3. Only ASIM3N showed a slightly more even $\mathcal{C}_j$. The use of RELU did make the models behave more similar to the DASIM (See Table 5.8).

The overall mean and standard deviation for both $\mathbf{L_A}$ and $\mathbf{L_b}$ in the normalized ASIM variations are shown in Table 5.10. The $\mathbf{L_A}$ normalization caused notable changes in the logit

Table 5.11: Contribution percentages from the η-independent $\mathbf{L_A}$ or the bias $\mathbf{L_b}$ logits in the ASIM2 and ASIM3 variations.

	Train		Test	
η	$\mathbf{L_A}$	$\mathbf{L_b}$	$\mathbf{L_A}$	$\mathbf{L_b}$
ASIM2	46.55	53.45	46.54	53.46
ASIM2N	44.26	55.74	44.14	55.86
ASIM2RN	61.51	38.49	61.17	38.83
ASIM3	31.20	68.80	31.32	68.68
ASIM3N	43.04	56.96	43.52	56.48
ASIM3RN	51.23	48.77	50.82	49.18

statistics, with little effect on the detection of mispronounced segments. The bias $s_{\mathbf{L_b}}$ was reduced in both ASIM2N and ASIM3N, indicating a more consistent bias output. Meanwhile, $s_{\mathbf{L_A}}$ remained close to the $s_{\mathbf{L_A}}$ of both ASIM2 and ASIM3 baselines. In the case of models using RELU regularization, $s_{\mathbf{L_A}}$ increased, yet $s_{\mathbf{L_B}}$ was reduced greatly. A bias vector with low variability could indicate a low correlation with the assessor η or a highly similar assessor sample. Since this is not the case of the INA set and from observing the low performance of ASIM2RN and ASIM2RN, it can be confirmed the RELU limited the usefulness of $\mathbf{L_A}$ as a starting point to learn the bias.

The results from ASIM2N and ASIM3N on the other hand, showed the models did find a different way of using $\mathbf{L_A}$ and $\mathbf{L_b}$. The plots for the average $\mathbf{L_A}$ and $\mathbf{L_b}$ outputs with their standard deviations are shown in Figure 5.15 for ASIM2N and Figure 5.16 for ASIM3N. Compared to the logit plots for ASIM2 and ASIM3 in Figures 5.13 and 5.14, the normalization of $\mathbf{L_A}$ did make the average logits more evenly distributed. Additionally, Table 5.12 shows the standard deviation $s_{\mathbf{L_A}}$ is still smaller than its $s_{\mathbf{L_b}}$ counterpart, indicating an average $\mathbf{L_A}$ more consistent than the bias logits. The η-independent logit plots for ASIM2N in Figure 5.15, kept a less negative mean output for most of the same phonemes, for which ASIM2 and ASIM3 also showed the least negative mean output. Said phoneme set was listed previously in Table 5.7, and it was noted to have a high inter-assessor agreement. The average logit barplots for ASIM3N in Figure 5.16 show more evenly distributed bias-free logits. Even the average logits for the phoneme set from Table 5.7 became more negative in ASIM3N. It looked like the regularization of $\mathbf{L_A}$ made the bias output of ASIM3N more similar to a gating function

Table 5.12: overall mean μ and standard deviation s for $\mathbf{L_A}$ and $\mathbf{L_b}$ on each ASIM2 and ASIM3 variations.

	Train				Test			
η	$\mu_{\mathbf{L_A}}$	$s_{\mathbf{L_A}}$	$\mu_{\mathbf{L_b}}$	$s_{\mathbf{L_b}}$	$\mu_{\mathbf{L_A}}$	$s_{\mathbf{L_A}}$	$\mu_{\mathbf{L_b}}$	$s_{\mathbf{L_b}}$
ASIM2	-4.33	2.44	-4.84	2.70	-4.38	2.48	-4.89	2.73
ASIM2N	-4.08	2.41	-4.95	2.54	-4.12	2.44	-5.01	2.57
ASIM2RN	-4.47	2.61	-3.31	2.61	-4.52	2.64	-3.37	1.70
ASIM3	-3.60	2.44	-6.59	2.94	-3.59	2.48	-6.60	3.01
ASIM3N	-4.21	2.46	-5.43	2.40	-4.21	2.51	-5.32	2.37
ASIM3RN	-4.69	3.00	-4.50	1.45	-4.68	3.05	-4.50	1.44

controlling a more consistent output.

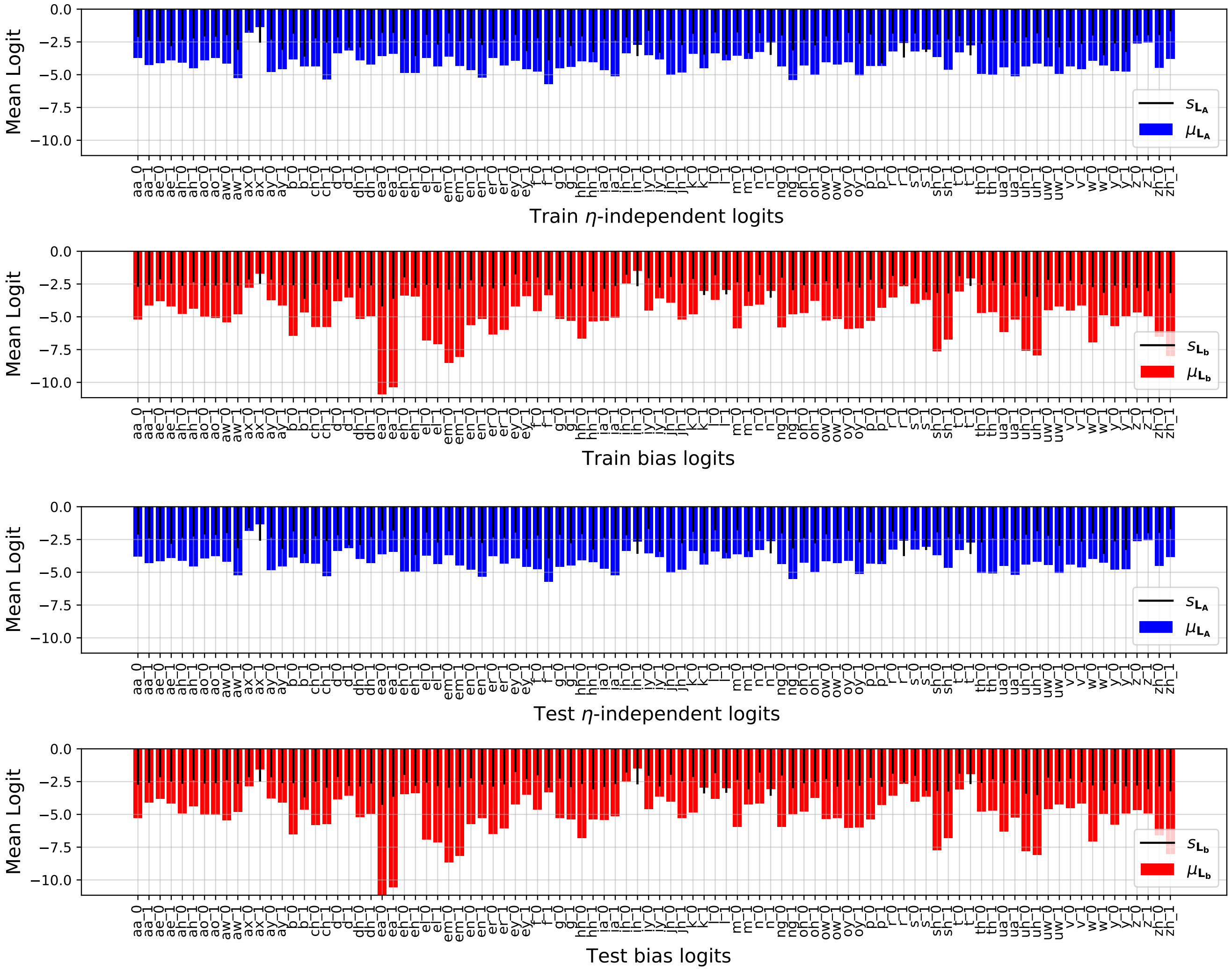


Figure 5.15: The η-independent mean logits $\mu_{\mathbf{L_A}}$ of the ASIM2N are shown in blue with their standard deviation $s_{\mathbf{L_A}}$ as black lines. The mean bias logits $\mu_{\mathbf{L_b}}$ are shown as red bars with their standard deviation $s_{\mathbf{L_b}}$ as black lines.

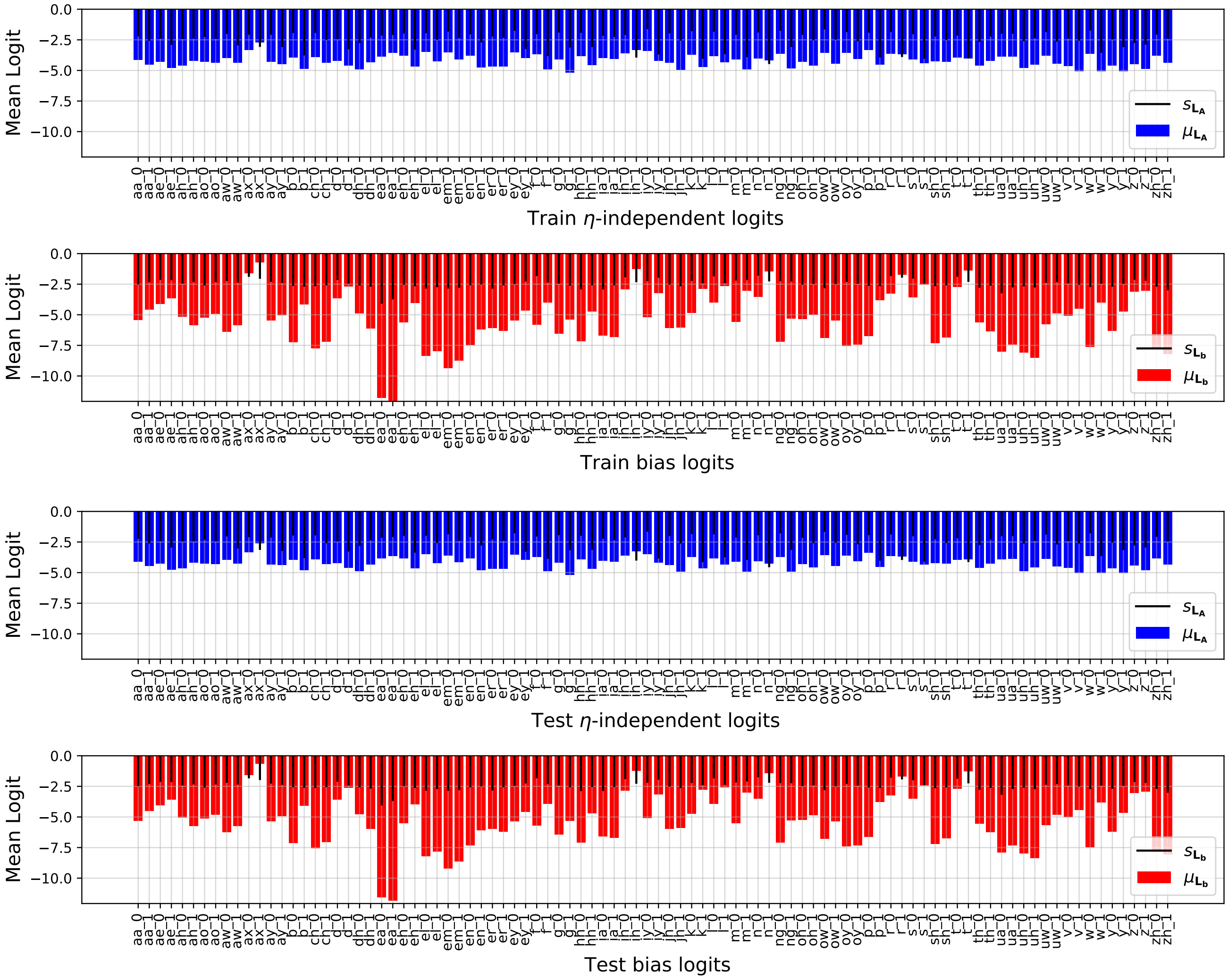


Figure 5.16: The η-independent mean logits $\mu_{\mathbf{L_A}}$ of the ASIM3N are shown in blue with their standard deviation $s_{\mathbf{L_A}}$ as black lines. The mean bias logits $\mu_{\mathbf{L_b}}$ are shown as red bars with their standard deviation $s_{\mathbf{L_b}}$ as black lines.

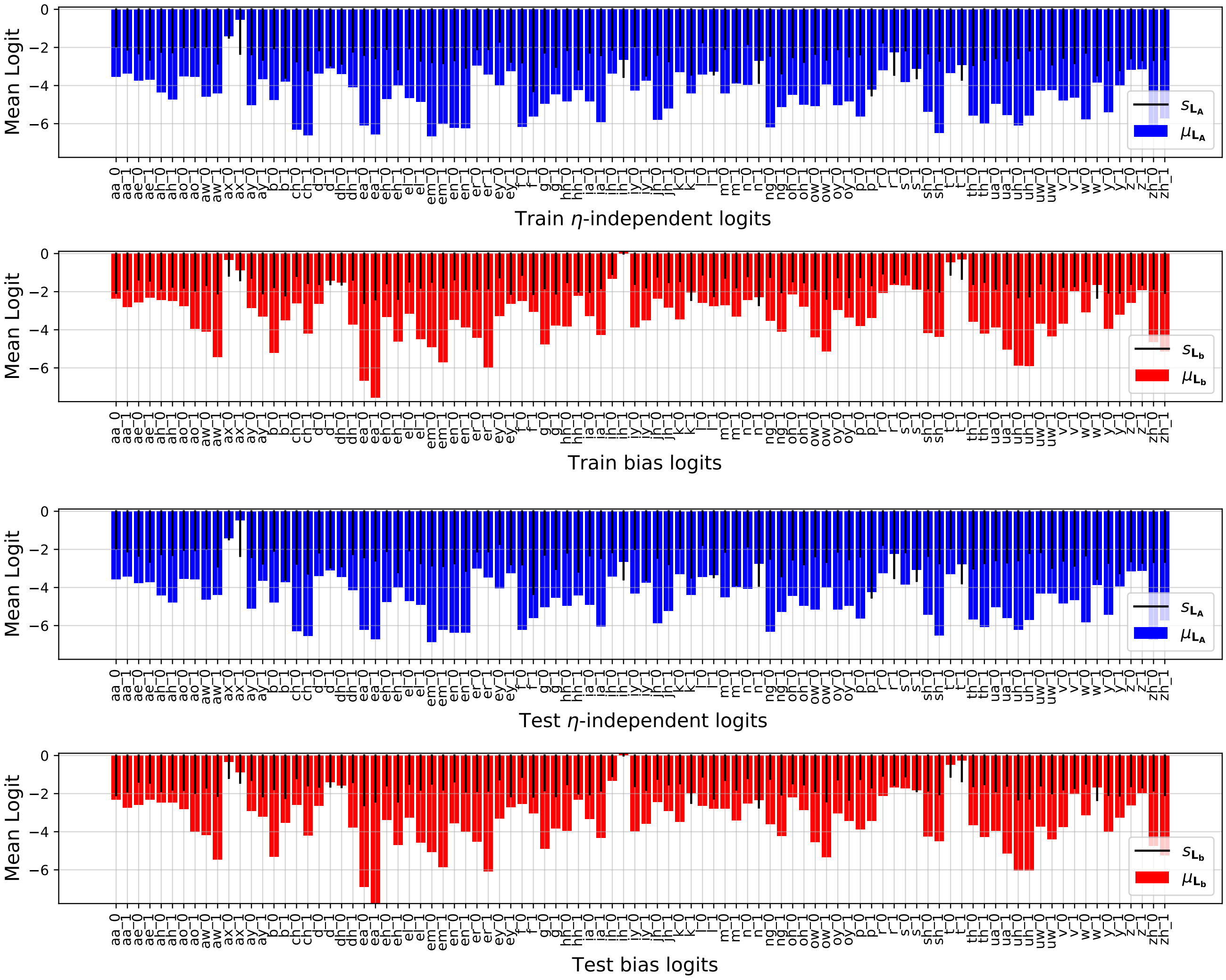
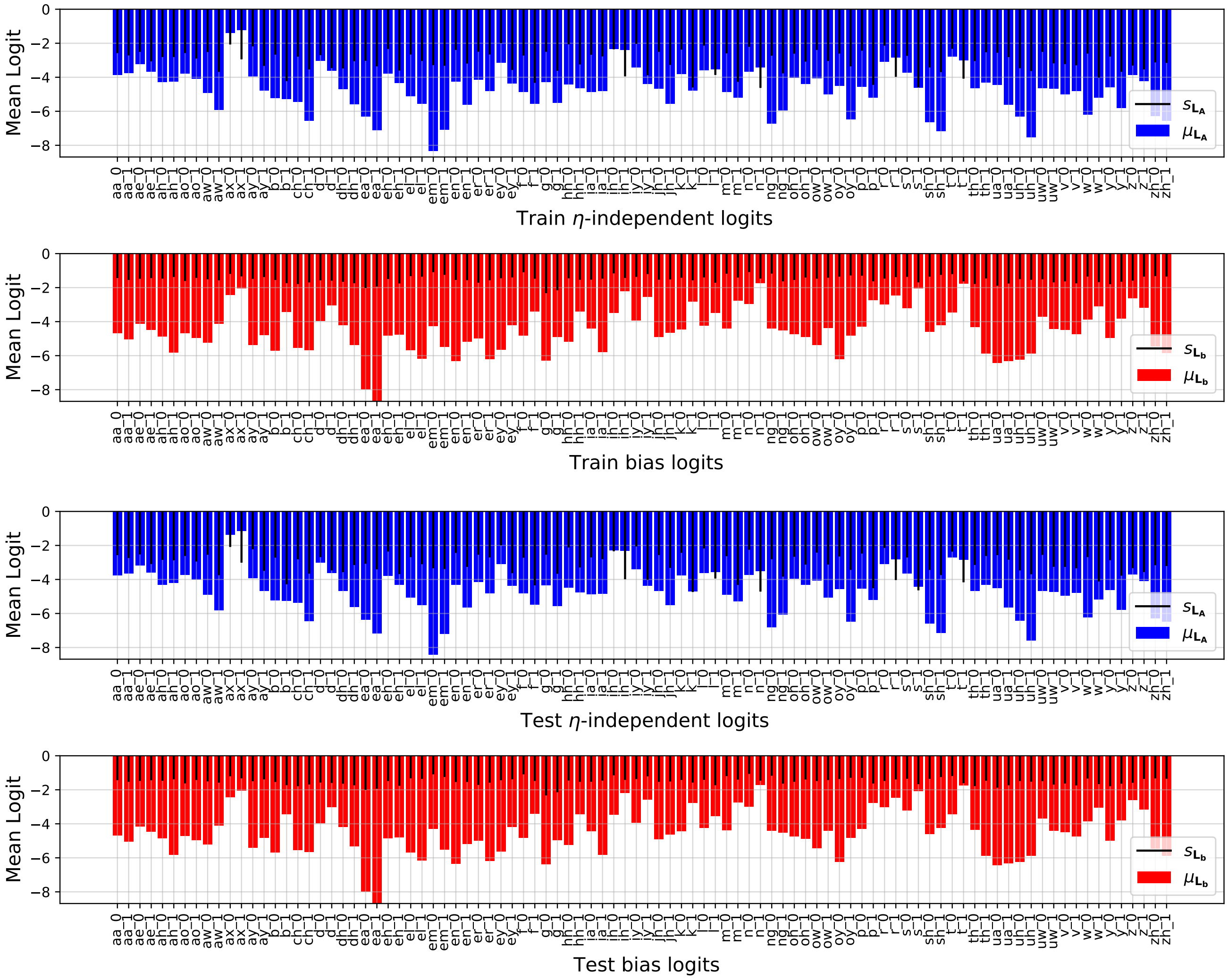


Figure 5.17: The η-independent mean logits $\mu_{\mathbf{L_A}}$ of the ASIM2RN are shown in blue with their standard deviation $s_{\mathbf{L_A}}$ as black lines. The mean bias logits $\mu_{\mathbf{L_b}}$ are shown as red bars with their standard deviation $s_{\mathbf{L_b}}$ as black lines.



Train η-independent logits

Train bias logits

Test η-independent logits

Test bias logits

Figure 5.18: The η-independent mean logits $\mu_{\mathbf{L_A}}$ of the ASIM3RN are shown in blue with their standard deviation $s_{\mathbf{L_A}}$ as black lines. The mean bias logits $\mu_{\mathbf{L_b}}$ are shown as red bars with their standard deviation $s_{\mathbf{L_b}}$ as black lines.

The plots for the average logit outputs for ASIM2RN and ASIM2RN are shown in Figures 5.17 and 5.18 respectively. In both models, the average $\mathbf{L_A}$ and $\mathbf{L_b}$ look similar to the ones of ASIM1 in Figure 5.12. Although ASIM2RN and ASIM2RN were not completely ignorant of $\mathbf{L_A}$, it seemed the bias FFN_b was made more consistent while its FFN_A counterpart had the most active role and a higher variation. From the results in Table 5.10, the behaviour of ASIM2RN and ASIM2RN do not reflect the behaviour of the assessors.

The use of a normalized $\mathbf{L_A}$ to assemble the input for FFN_b did make the bias component in the ASIM variations more similar to a gating element. The behaviour makes the architecture easier to interpret. For example, the fact that on average the $\mathbf{L_b}$ logits are far less evenly distributed than $\mathbf{L_A}$ indicates that FFN_b is the one component sensitive to phoneme identities and the level of disagreement across the assessors. Further work on the behaviour of the ASIM3 is required to confirm the model manages to separate assessor bias from an assessor-independent output that is also meaningful.

5.7.6 Summary

The DASIM presented in Section 5.4 was found it be able to learn multiple pronunciation references simultaneously using an assessor-independent and an assessor-bias subnetwork. It was found the elements for sequential encoding $EC(\mathbf{O}^{(w)})$ in the DASIM were redundant; therefore a more efficient architecture was designed. A reinterpretation of the assessor model in Equation (5.5) allowed redesign of the DASIM by using a single $EC(\mathbf{O}^{(w)})$ connected to two FFNs FFN_A and FFN_b. The new architectures for implementing Equation (5.5) included $\mathbf{L_A}$ to the input for the bias FFN_b to reduce its dependence on $EC(\mathbf{O}^{(w)})$ while making it more dependent on FFN_A. From the three new designs proposed, only ASIM3 improved mispronunciation detection on the INA dataset for both Train and Test sets. As FFN_b increased its dependence on the assessor-independent logits $\mathbf{L_A}$, the average bias logits $\mathbf{L_b}$ became more negative and showed a wider standard deviation. It was concluded the best performing model, the 6-layer ASIM3, used the bias FFN_b to learn a gating function for the assessor independent $\mathbf{L_A}$. The gating function would block an output class in $\mathbf{L_A}$ by making the final logit sum with $\mathbf{L_A}$ largely negative, so the sigmoid regularization gives a probability close to zero. The normalization of $\mathbf{L_A}$ for the input of FFN_b made the average $\mathbf{L_A}$ more evenly distributed, while $\mathbf{L_b}$ showed a wider variation across output classes.

5.8 Conclusion

A model for the assessor of L2 pronunciation was introduced in this chapter. The model considers an assessor-independent scoring function exists and it is affected by an assessor-specific additive bias term. The model was tested using the segment-based approach for detecting mispronunciations in short speech segments. The L2 assessor model implemented the DASIM, consisting of two ASIM networks observing the same acoustic features with their final output logits summed together and passed through a sigmoid regularization to obtain the estimates of the observed correctness labels. The subnetwork corresponding to the bias was made sensitive to the assessor identity via a one-hot encoding vector. The model design aims to be simple enough for interpretability and to perform well with the limited data available.

The DASIM outperformed the original ASIM design in detecting mispronounced segments given to each assessor. To test the claims of the assessor model in the DASIM, each of its subnetworks went through different tests. The bias subnetwork was proven sensitive to the assessor tag as it was found a mismatch between the identity vector and the annotation reference decreased the performance of the model. The assessor-independent subnetwork was the major contributor to the final logits behaving as a starting point in the assessment, which would later be shifted by the bias logits as the model requires.

The self-attention modules of both subnetworks show a similar trend, although the models have no interaction other than the summation of their final classifier logits. However, the normalized attention curves of the bias subnetwork can indicate points of disagreement in annotation given the assessor tag. A redundancy in the processing of the input is evident from the attention plots. Therefore, a re-interpretation of the assessor model in Equation (5.5) was the base for a re-design of the DASIM.

The different architectures for a more efficient ASIM were proposed. The new architectures did reduce the redundancy of the DASIM by achieving similar or better results while reducing the number of parameters. The new architectures differed in the inputs for the bias FFN FFN_b, shifting its dependency from the sequential encoding towards the assessor-independent FFN_A. All the ASIM variations proposed show a peak in performance at a 6-layer configuration for their encoder's BDLSTM and their FFNS. The ASIM3 model showed the best metrics for detecting mispronunciations in the INA dataset. ASIM3 was also the model with the least processing load for FFN_b since it received the assessor-independent logits $\mathbf{L_A}$ and the assessor tag η as input. It was found that ASIM3 used the bias logits $\mathbf{L_b}$ as a gating mechanism, which pushed the final output posteriors close to zero given the assessor identity. The average $\mathbf{L_A}$ and $\mathbf{L_b}$ did not show a clear preference for phonemes prone to mispronunciation by Dutch speakers of English as L2, nor to phonemes with no direct equivalent across Native Language (L1) and L2, nor to levels of inter-reliability. A set of phonemes with high counts listed in Table 5.7 for which the average $\mathbf{L_b}$ in ASIM3 tended to show a less negative and consistent output for their correct pronunciation.

The findings from ASIM3 did change the interpretability of the assessor model proposed, while still improving the detection of mispronounced segments. The follow-up work will consist of testing different constraints on the model to encourage the specialization of the network components to better separate the bias from an ideal assessor-independent PA.

Chapter 6

Methods for Encouraging Bias Specialization in ASIM.

6.1 Introduction

The efforts of this thesis aim for a model which can explain the assessor bias in Second Language (L2) Pronunciation Assessment (PA), as explained in Section 2.1.2. A model for the L2 assessor scoring process was proposed in Section 5.3. The assessor model was defined in Equation (5.5) as the output of an assessor-independent scoring function affected by an additive assessor-specific scoring term known as the bias. So far, related work has resulted in the Attention-Based Segmental Incorrectness Model (ASIM). The original design of the ASIM in Section 3.3 combined a Bidirectional Long Short-Term Memory (BDLSTM) and self-attention to perform sequence encoding of short speech segments. The resulting encoding was then passed through a Feed-Forward Network (FFN) with an output for each phoneme class labelled either as correctly or incorrectly pronounced. The experimental results in Section 3.6.5 showed ASIM is useful for learning a pronunciation reference via annotation without the need for a precise alignment or additional speech data.

The ASIM became the starting point for the implementation of the assessor model. First, two ASIMs were trained simultaneously on the same acoustic input, with one of the networks made sensitive to assessor identity by concatenating the assessor ID tag η to the input as a constant dimension. The two networks combined their output logits as an arithmetic sum before a final sigmoid normalization. Said arrangement was called the Dual Attention-Based Segmental Incorrectness Model (DASIM), and it is illustrated in Figure 5.2. The DASIM did show redundancies in its parameters and had both networks contributing similarly to the final output.

The main problem for implementing the assessor model from Equation (5.5) is that its components cannot be obtained directly from observed data. The model's terms must be learned jointly, without previous knowledge added. Rather than pushing examples to be assumed to be caused by the bias, it is better to motivate the overall behaviour of the model by design and training criteria. For example, it was desired that DASIM did most of the processing of the input using the assessor-independent subnetwork. Also, the bias component should not be the main drive in PA. The desired behaviour of the assessor model comes from the notion of a group of L2 pronunciation assessors trained on the same pronunciation reference and showing a high level of inter-assessor agreement. Recall, a high level of inter-assessor agreement is desired for the sake of consistency and fairness in assessment (Isaacs and Harding, 2017).

In Section 5.6, different interpretations of the assessor model did translate into a re-design of the DASIM. A key modification was to make the bias component more dependent on the assessor-independent component. The bias-free logits $\mathbf{L_A}$ were used as part of the input for the bias FFN_b. Particularly, the ASIM3 variation, shown in Figure 5.6, achieved the best performance on detecting mispronounced segments compared to the other ASIM re-designs and the DASIM. The ASIM3 design used only $\mathbf{L_A}$ concatenated to a one-hot encoding of the assessor identity.

The average output of the components of ASIM3 discussed in Section 5.7.5 showed the bias logits $\mathbf{L_b}$ would make an output class largely negative given the assessor identity. The bias $\mathbf{L_b}$ would act as a gating function, either blocking or having a minimal effect on the final combined logit. Both $\mathbf{L_A}$ and $\mathbf{L_b}$ did not show a tendency in response for phonemes with either high or low inter-annotator reliability κ. Not even a visible preference for phoneme classes known to be prone to mispronunciations by Dutch speakers of English was observed in either $\mathbf{L_A}$ or $\mathbf{L_b}$ in ASIM3. Without a clear role division between the ASIM components, it can be concluded that the optimization of the ASIM architecture should not be tuned to examples assumed to represent either agreement or an individual bias.

A particular set of phonemes was noted to elicit a less negative response on both $\mathbf{L_A}$ and $\mathbf{L_b}$ for multiple ASIM variations. Table 5.7 shows that said phoneme group have a high inter-annotator agreement percentage, yet not the largest κ. The major similarity seen in the phonemes listed in Table 5.7 was the relatively high counts in the annotation, as the mean occurrence per phoneme class in INA is 148.87 with a standard deviation of 139.81. A similar response from the assessor-independent and bias components to the same phoneme class could indicate that further specialization of the components is possible.

In this chapter, multiple criteria are tested for effect on the behaviour of the ASIM components. The criteria selected do not represent assumptions for cases based on agreement metrics. Instead, additional objective functions are used during training to push the model to make $\mathbf{L_A}$ and $\mathbf{L_b}$ different from each other. It is expected that decreasing the similarity between the bias and the assessor-independent components, will increase the bias dependency on the assessor identity.

6.2 Similarity Between the ASIM Components and Desired Behaviour of the Bias.

The use of the bias-free $\mathbf{L_A}$ as input for the bias FFN_b did improve the generalization of the ASIM (See Table 5.6). A similarity between the average $\mathbf{L_A}$ and $\mathbf{L_b}$ outputs was also observed, particularly for Attention-Based Segmental Incorrectness Model - Configuration 2 (ASIM2) and Attention-Based Segmental Incorrectness Model - Configuration 3 (ASIM3). The average logit plots in Figure 5.13 for ASIM2 and the plots in Figure 5.14 for ASIM3 showed the phonemes with the least negative $\mathbf{L_A}$ output also had the least negative $\mathbf{L_b}$. A similar behaviour was noted for some phonemes with average large negative outputs. A large response in $\mathbf{L_A}$ would propagate to the upcoming network layers of FFN_b. The observed similarity is a sign of the strong correlation between $\mathbf{L_A}$ and the input weights of FFN_b (Ba et al., 2016).

Similarity between $\mathbf{L_A}$ and $\mathbf{L_b}$ is not desired, since the objective is to isolate the effect of assessor bias on PA. Said similarity in trend can be reduced, or at least changed without affecting the performance of the ASIM. For example, the use of a normalization layer for the FFN_b input did make the mean $\mathbf{L_A}$ more similar across output classes. Also, using normalized inputs for FFN_b reduced the overall standard deviation of $\mathbf{L_b}$ yet kept it larger than the standard deviation of $\mathbf{L_A}$ (See Table 5.12). The $\mathbf{L_A}$ logits are meant to represent the agreement across all assessors; hence it is expected to be more consistent than $\mathbf{L_b}$. Since the bias logits can be made more variable with respect to the phoneme identity, more constraints can be used as part of the training setup of the ASIM. The following subsection introduces the cosine similarity as an additional cost on the ASIM training function. It is expected the assessor-independent and the bias component can be forced to show a behaviour that makes a stronger claim for the ASIM to be able to learn the components of the assessor model proposed in Equation (5.5).

6.3 Cosine Similarity Minimization

The Cosine Similarity (CS) between two vectors is measured using the cosine of the angle θ formed between them. Equation (6.2) defines the CS for two non-zero vectors $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$. The similarity consists of the dot product between the two vectors divided by the product of their Euclidean norms. The euclidean norm for a vector $\overrightarrow{v} \in \mathcal{R}^N$ is defined as $\|\overrightarrow{v}\| = \sqrt{\sum_{i=1}^{N} \overrightarrow{v}_i^2}$.

$$\theta = \angle(\overrightarrow{v_1}, \overrightarrow{v_2}) \tag{6.1}$$

$$\cos\theta = CS(\overrightarrow{v_1}, \overrightarrow{v_2}) = \frac{\overrightarrow{v_1} \cdot \overrightarrow{v_2}}{\|\overrightarrow{v_1}\|\|\overrightarrow{v_2}\|} \tag{6.2}$$

The cosine function is bounded between 1 and -1. Two vectors with the same direction yield $\cos 0^\circ = 1$. When the two vectors are collinear, $\cos 180^\circ = -1$. A cosine similarity of zero represents an orthogonality between $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$, meaning a null correlation between the vectors. The CS is a simple metric for learning similarities across data points. The cosine between two data representation vectors has been proven useful for facial verification (Nguyen and Bai, 2010), L2 PA (Wang et al., 2018b), speaker verification (Senoussaoui et al., 2013) and document clustering (Muflikhah and Baharudin, 2009); all activities in which the similarity across observations is crucial.

It was shown in Section 5.6 to 5.7.5 that the behaviour of the ASIM components can be changed without compromising the learning of the annotation reference. Therefore, the similarity between the ASIM logits $\mathbf{L_A}$ and $\mathbf{L_b}$ can be reduced beyond the results achieved from using a normalization layer for the FFN_b inputs.

Since $\mathbf{L_A}$ is part of the input to obtain $\mathbf{L_b}$, a level of correlation is expected. Said correlation should be kept at a minimum for a better separation of the bias function from the assessor-independent scoring function. Additionally, by decreasing CS between $\mathbf{L_A}$ and $\mathbf{L_b}$, it is expected to reduce confusion due to class co-occurrence (Pellegrini and Cances, 2019).

The original loss function for the ASIM defined in Equation (3.13) consists of the Binary Cross-Entropy (BCE) between the ASIM final output $\hat{\mathbf{l}}$ and the annotation $\mathbf{l}$. Recall the annotation vector $\mathbf{l}$ is a one-hot encoding indicating whether a phoneme class expected in an acoustic segment $\mathbf{O}$ was pronounced correctly or not. Recall, also that $\hat{\mathbf{l}}$ is equal to the arithmetic sum of logits $\mathbf{L_A}$ and $\mathbf{L_b}$ and a further sigmoid regularization:

$$\hat{\mathbf{l}} = \sigma(\mathbf{L_A} + \mathbf{L_b}) \tag{6.3}$$

The original loss function can be expanded to add $CS(\mathbf{L_A}, \mathbf{L_b})$ weighted by a hyper-parameter α. Hence, the ASIM loss over N examples becomes:

$$\mathrm{BCE}(\mathbf{l}, \hat{\mathbf{l}}, \mathbf{L_A}, \mathbf{L_b}) = -\frac{1}{N}\sum_{i=1}^{N}\left[\mathbf{l}_i \cdot \log \hat{\mathbf{l}}_i + (1-\mathbf{l}_i)\cdot \log(1-\hat{\mathbf{l}}_i) - \alpha \cdot \frac{\mathbf{L_A}_i \cdot \mathbf{L_b}_i}{||\mathbf{L_A}_i||\,||\mathbf{L_b}_i||}\right] \tag{6.4}$$

The new loss function in Equation (6.4) should reduce the similarity between the assessor-independent scoring function and the effect of the bias. However, a complete de-correlation between $\mathbf{L_A}$ and $\mathbf{L_b}$ would also cause problems with learning the bias. After the re-interpretation of the assessor model in Equation (5.10), the dependence $\mathbf{L_b}$ has on $\mathbf{L_A}$ became explicit. In (Pellegrini and Cances, 2019), only cases with a CS greater than zero contribute to the loss. Therefore, the loss function becomes:

$$\mathrm{MAXLoss}(\mathbf{l}, \hat{\mathbf{l}}, \mathbf{L_A}, \mathbf{L_b}) = -\frac{1}{N}\sum_{i=1}^{N}\left[\mathbf{l}_i \cdot \log \hat{\mathbf{l}}_i + (1-\mathbf{l}_i)\cdot \log(1-\hat{\mathbf{l}}_i) - \alpha \cdot \max\left(0, \frac{\mathbf{L_A}_i \cdot \mathbf{L_b}_i}{||\mathbf{L_A}_i||\,||\mathbf{L_b}_i||}\right)\right] \tag{6.5}$$

The reason for considering only positive CS values for the loss in (Pellegrini and Cances, 2019) was that it was expected for negative similarities to be on average smaller than the positive ones. For the case of the ASIM learning the bias, reducing the positive $CS(\mathbf{L_A}, \mathbf{L_b})$ is preferred to a complete de-correlation. The use of the positive CS as a penalty might not influence the capability of the ASIM to learn the annotation reference. Since a minimum redundancy between $\mathbf{L_A}$ and $\mathbf{L_b}$ is required, a negative CS could be informative enough to learn the assessor bias. A further look at the behaviour of the ASIM trained to reduce Equation (6.5) can offer new insights on how the assessor bias could affect an ideal bias-free PA.

6.4 Training of the ASIM with a CS Penalty

The results in Section 3.6.6 showed how assumptions over inter-assessor agreement via labels complicate the assessment of a model trained for PA. Therefore, the use of something such as explicit *bias* labels should be avoided. It is preferred for the model to optimize the contributions of its assessor-independent and assessor-specific components when learning a pronunciation reference. For this, MAXLoss in Equation (6.5) was tested for training the ASIM3 architecture as defined in Figure 5.6. The ASIM3 architecture was chosen as it outperformed the other ASIM variations (see Section 5.7.3). ASIM3 is also the architecture with the smallest parameter count, and it showed the smallest standard deviation for the $\mathbf{L_A}$ logits from all other ASIM variations proposed in Section 5.6 (see Table 5.9). Recall for ASIM3, the bias is assumed as $P(\hat{\mathbf{l}_\mathbf{b}}|\mathbf{r},\hat{\mathbf{l}_\mathbf{A}},\eta)$. The goal of this experiment was to reduce the similarity of the components of the assessor model proposed in Equation (5.10), for the sake of less redundancy in the model's components.

The minimization of $CS(\mathbf{L_A},\mathbf{L_b})$ should keep a minimum correlation $corr(\mathbf{L_A},\mathbf{L_b})$ to not decrease the ASIM3 performance on learning an annotation reference. The network configuration is defined in detail in Section 6.4.1. The network was trained on short segments of L2 speech, each marked for mispronunciation by three trained L2 assessors. The same architecture was trained using different values for the weight α in the loss function. The trained models were scored for detecting mispronounced segments given the reference. The F1 score and Cohen's κ were the metrics selected to assess the models' agreement with the annotation reference. Changes in $CS(\mathbf{L_A},\mathbf{L_b})$ were observed for the effect they could have on the performance and behaviour of ASIM3 and its components.

6.4.1 Model Training Setup

The architecture used to learn the assessor model was the Attention-Based Segmental Incorrectness Model - Configuration 3 with Normalization (ASIM3N), with a normalization layer for $\mathbf{L_A}$ before using it as an input for the bias classifier FFN_b. The ASIM3N was the same used for the experiment in Section 5.7.5, meaning a 6-layer BDLSTM of size 64 and an additive self-attention module with 128 linear weights as the encoding section EC. Both assessor-independent FFN_A and bias FFN_b classifiers consist of a 6-layer deep classifier with a layer size of 1024. The classifiers held two units for each phoneme class in their output layers, corresponding to each phoneme class being marked as either a correct or incorrect pronunciation.

Table 6.1: F1 score and Cohen's Kappa (κ) for the ASIM3N on detecting mispronounced segments across all assessors in the INA set. The average $CS(\mathbf{L_A}, \mathbf{L_b})$ is also shown. Each row corresponds to a different α weight for the CS penalty.

	Train			Test		
α	F1	κ	$CS(\mathbf{L_A}, \mathbf{L_b})$	F1	κ	$CS(\mathbf{L_A}, \mathbf{L_b})$
0	0.8086	0.6503	0.8478	0.7385	0.5543	0.8502
0.001	0.8044	0.6425	-0.0142	0.7417	0.5569	-0.0057
0.01	0.8113	0.6554	-0.0358	0.7433	0.5628	-0.0293
0.1	0.8124	0.6574	-0.0622	0.7434	0.5639	-0.0525
0.5	0.8118	0.6563	-0.0729	0.7426	0.5623	-0.0630
1.0	0.8110	0.6548	-0.0836	0.7402	0.5582	-0.0706

The ASIM3N was trained on the first 13 Perceptual Linear Prediction (PLP) coefficients with their first and second-order time differentials as the acoustic input for EC. The assessor tag η is a one-hot encoding concatenated to the FFN_A output after passing through the normalization layer. The ASIM3N was trained with the α weight for the $CS(\mathbf{L_A}, \mathbf{L_b})$ penalty set to 0.001, 0.01 and 0.1 in Equation (6.5). A model trained with $\alpha = 0$ was used as a baseline. All models were trained using the Adam optimizer (Kingma and Ba, 2014) until reaching 6 epochs without any improvement in the loss function. The version of the model scored corresponds to the one reaching the lowest loss on the Test set.

6.4.2 Experiment Dataset

The ASIM3N was trained on the INA split with zero-speaker overlap, used previously in this thesis for training all ASIM variations. The INA sit splits 85% of recordings from Train, leaving the remaining 15% for Test. The subsets are balanced for sex, age, and L2 proficiency level. The Train subset holds 215 speakers out of the 238 available. The Test subset contains 23 speakers. All speakers in INA supplied the same amount of recordings and all were annotated by the three trained phoneticians *a1*, *a2* and *a3*.

The recordings were used to create short acoustic segments using a sliding window 0.5s long with 0.05s stride. The pronunciation correctness labels $\mathbf{l}$ were force-aligned to the acoustic segment using the triphone-based DNN-Hidden Markov Model (HMM) Acoustic Model (AM) trained for (Nicolao et al., 2015) and used in all the experiments involving the INA set in this thesis. More information regarding the alignment for the INA set can be found in Section 3.6.1. A phoneme is assumed present in a segment if the alignment allocates it entirely within at least 2 frames from the edges of the sliding window. The aligned segments contained a mean of 3.46 phonemes with a standard deviation of 1.54.

6.4.3 Performance on Detecting Mispronounced Segments

The first test for models trained using the CS penalty was to detect mispronounced segments given to all three assessors. The results in Table 6.1 show the F1 score, κ and $CS(\mathbf{L_A}, \mathbf{L_b})$ for

Table 6.2: F1 score and Cohen's Kappa (κ) for the ASIM3N on detecting mispronounced segments across all assessors in the INA set. The models were trained using ABSLoss. The average $CS(\mathbf{L_A}, \mathbf{L_b})$ is also shown. Each row corresponds to a different α weight for the CS penalty.

	Train			Test		
α	F1	κ	$CS(\mathbf{L_A}, \mathbf{L_b})$	F1	κ	$CS(\mathbf{L_A}, \mathbf{L_b})$
0.001	0.8118	0.6563	0.0109	0.7444	0.5628	0.0242
0.01	0.8121	0.6569	0.0003	0.7426	0.5617	0.0054
0.1	0.8127	0.6579	-1.03E-6	0.7434	0.5636	0.0028
0.5	0.8118	0.6563	2.64E-6	0.7426	0.5623	0.0002
1.0	0.8110	0.6548	1.34E-6	0.7402	0.5582	0.0002

each model trained on the INA set while using a different CS penalty weight α. The baseline ($\alpha = 0$) shows the ASIM3N optimizes with a high $CS(\mathbf{L_A}, \mathbf{L_b}) = 0.8086$ in the Train set. Recall, the upper limit for CS is 1. The use of CS does affect $\angle(\mathbf{L_A}, \mathbf{L_b})$ with a small effect in the performance, similar to what occurred when the normalization layer for $\mathbf{L_A}$ was implemented in ASIM3. The model with the smallest $\alpha = 0.001$ achieved $CS(\mathbf{L_A}, \mathbf{L_b}) = -0.0142$ for Train and $CS(\mathbf{L_A}, \mathbf{L_b}) = -0.0057$ for Test. The $\alpha = 0.001$ managed to reduce $corr(\mathbf{L_A}, \mathbf{L_b})$ the most, yet it also showed a small decrease in F1 and κ compared to the baseline. The CS on the Test set from using $\alpha = 0.001$ is even smaller, yet the performance has a slight improvement, which was no greater than 0.47%.

When α increases, $CS(\mathbf{L_A}, \mathbf{L_b})$ is pushed towards negative values. Interestingly, as $CS(\mathbf{L_A}, \mathbf{L_b})$ becomes more negative, there is a small general gain in F1 and κ for both Train and Test. Said improvement on the metrics goes up to a maximum point to then slowly decay again. The highest improvement observed in Table 6.1 was at least 1% in κ for both Train and Test when $\alpha = 0.1$. The negative $CS(\mathbf{L_A}, \mathbf{L_b})$ observed after training in all cases listed in Table 6.1, were smaller than the baseline $CS(\mathbf{L_A}, \mathbf{L_b}) = 0.8478$ for Train and $CS(\mathbf{L_A}, \mathbf{L_b}) = 0.8502$ for Test.

The tendency of F1 and κ given α observed so far confirmed that reducing $CS(\mathbf{L_A}, \mathbf{L_b})$ does not aggravate the detection of mispronounced segments. The penalization of only the cases for which $CS(\mathbf{L_A}, \mathbf{L_b}) > 0$ leaves the negative correlations unaffected. All the models trained using $\alpha > 0$ showed a negative CS with a magnitude smaller than the CS of the baseline. The plots in Figure 6.1 show the $CS(\mathbf{L_A}, \mathbf{L_b})$ curves for each α during training. The baseline in blue indicates that if, let alone, $CS(\mathbf{L_A}, \mathbf{L_b})$ remains similar to how it started. As training progresses and the baseline model starts over-fitting, $CS(\mathbf{L_A}, \mathbf{L_b})$ also starts growing slowly yet constantly. The curves of the models with $\alpha > 0$ showed the penalty got rid of the positive CS early in the training. Figure 6.1 confirmed the claim of (Pellegrini and Cances, 2019) about negative similarities being constant and smaller than positive ones.

As the $CS(\mathbf{L_A}, \mathbf{L_b})$ values in Table 6.1 grew negative, both F1 and κ decreased from their peak values observed for $\alpha = 0.1$. Therefore, it can be inferred that an even smaller $corr(\mathbf{L_A}, \mathbf{L_b})$ could be informative enough for the model to learn the annotation reference. A further reduction of $corr(\mathbf{L_A}, \mathbf{L_b})$ could also better isolate the bias effect on ASIM3N. Besides, the smaller $CS(\mathbf{L_A}, \mathbf{L_b})$, the less redundant the model components will be.

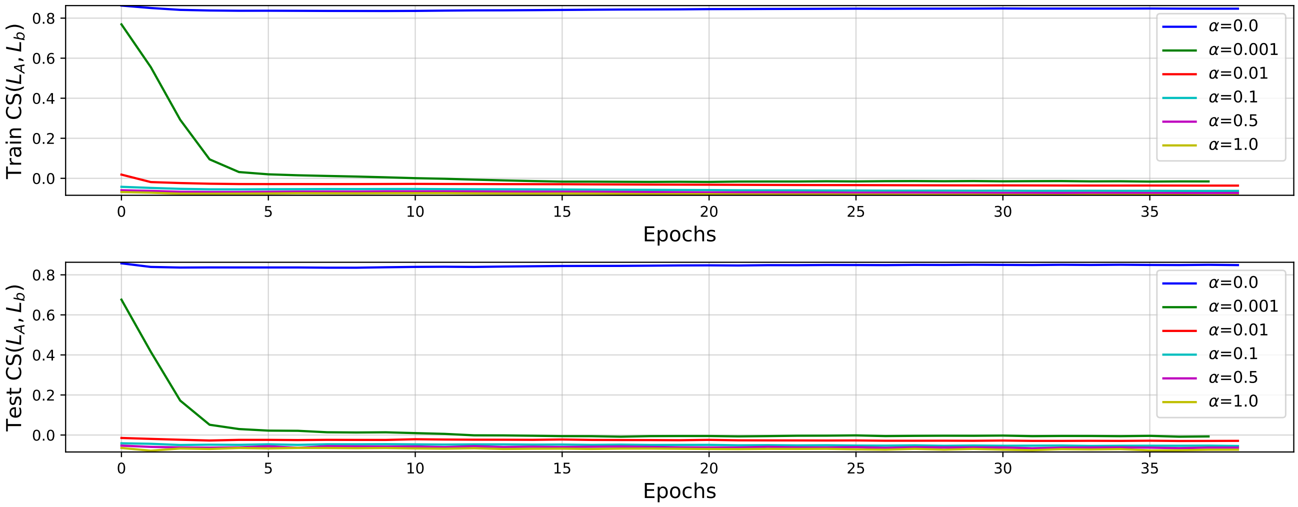


Figure 6.1: Curves for $CS(\mathbf{L_A}, \mathbf{L_b})$ given the weight α. The models were trained used the loss function in Equation (6.5).

An additional set of ASIM3N was trained using $\alpha \cdot |CS(\mathbf{L_A}, \mathbf{L_b})|$ as the similarity penalty, rather than only considering positive similarities as defined in Equation (6.5). The resulting loss function is referred to as ABSLoss, and it is defined as:

$$\text{ABSLoss}(\mathbf{l}, \hat{\mathbf{l}}, \mathbf{L_A}, \mathbf{L_b}) = -\frac{1}{N}\sum_{i=1}^{N}\left[\mathbf{l}_i \cdot \log \hat{\mathbf{l}}_i + (1 - \mathbf{l}_i) \cdot \log(1 - \hat{\mathbf{l}}_i) - \alpha \cdot \left|\frac{\mathbf{L_A}_i \cdot \mathbf{L_b}_i}{\|\mathbf{L_A}_i\| \|\mathbf{L_b}_i\|}\right|\right] \quad (6.6)$$

The ABSLoss pushes $\angle(\mathbf{L_A}, \mathbf{L_b}) \approx 90°$. The new goal was to keep the smallest $corr(\mathbf{L_A}, \mathbf{L_b})$ required to learn the annotation reference. The results in Table 6.2 correspond to the mispronunciation detection metrics for the models trained using ABSLoss. At first glance, the gain on F1 and κ is slightly greater than the models in Table 6.1. What matters most is that $CS(\mathbf{L_A}, \mathbf{L_b})$ decreased considerably from using ABSLoss. It can be assumed the remaining correlations between the $\mathbf{L_A}$ and $\mathbf{L_b}$ logits are strong.

6.4.4 Interaction Between Assessor-Independent and Bias Components

The changes in $CS(\mathbf{L_A}, \mathbf{L_b})$ from implementing CS as a penalty were notable. For a better visualization, the $CS(\mathbf{L_A}, \mathbf{L_b})$ from the baseline corresponds to an expected $\angle(\mathbf{L_A}, \mathbf{L_b}) = 30.03°$. Meanwhile, the model trained with ABSLoss and $\alpha = 0.1$ corresponds to $\angle(\mathbf{L_A}, \mathbf{L_b}) = 90.00006°$ on average. The perpendicularity between the logits was practically achieved, along with a small improvement in performance. The ASIM3N is capable of changing the behaviour of its classifiers FFN_A and FFN_b without a major change in its performance. Therefore, it is worth looking at the distribution of the correspondent logits.

The overall mean (μ) and standard deviation (s) for $\mathbf{L_A}$ and $\mathbf{L_b}$ were obtained for the models trained on either the loss from Equation (6.5), hereby referred to as MAXLoss, or ABSLoss in Equation (6.6). Table 6.3 shows the logit statistics given each α value used for

Table 6.3: Standard deviation s and overall mean μ for $\mathbf{L_A}$ and $\mathbf{L_b}$ on each ASIM variation.

	Train				Test			
α	$\mu_{\mathbf{L_A}}$	$s_{\mathbf{L_A}}$	$\mu_{\mathbf{L_b}}$	$s_{\mathbf{L_b}}$	$\mu_{\mathbf{L_A}}$	$s_{\mathbf{L_A}}$	$\mu_{\mathbf{L_b}}$	$s_{\mathbf{L_b}}$
0	-4.29	2.48	-5.60	2.45	-4.28	2.53	-5.49	2.42
	MAXLoss							
0.001	-1.51	2.17	-8.14	2.44	-1.5	2.47	-8.02	2.47
0.01	-8.37	3.47	-1.02	1.12	-8.38	3.55	-1.00	1.11
0.1	-9.09	3.59	-0.31	1.05	-9.09	3.66	-0.30	1.03
0.5	-9.14	3.56	-0.20	1.02	-9.14	3.64	-0.18	1.01
1.0	-9.14	3.56	-0.10	0.99	-9.16	3.64	-0.10	0.97
	ABSLoss							
0.001	-1.63	2.59	-8.48	2.56	-1.62	2.22	-8.33	2.59
0.01	-8.13	3.44	-1.24	1.13	-8.14	3.51	-1.21	1.12
0.1	-8.50	3.66	-0.95	0.95	-8.50	3.73	-0.93	0.94
0.5	-8.54	3.78	-0.77	0.74	-8.53	3.85	-0.76	0.73
1.0	-8.33	3.68	-0.78	0.72	-8.33	3.76	-0.77	0.71

training the model. The baseline statistics replicate the behaviour shown in Section 5.7.5, in which the bias logits act as a gating mechanism for $\mathbf{L_A}$ (See Figure 5.16). The CS seems to shift this behaviour considerably. The smallest $\alpha = 0.001$ seemed to keep the same behaviour as the baseline. As α grows, there is a major shift in the statistics of the logits. Both MAXLoss and ABSLoss pushed $\mu_{\mathbf{L_A}}$ towards large negative values and increased $s_{\mathbf{L_A}}$. Meanwhile, the bias $\mathbf{L_b}$ moved closer to zero, meaning little to no effect over $\mathbf{L_A}$. The overall $s_{\mathbf{L_b}}$ shrank considerably, indicating the bias became less dynamic as well.

The average logit output per label class was plotted for both $\mathbf{L_A}$ and $\mathbf{L_b}$ to observe their distribution. The bar plots in Figure 6.2 show the mean logit per class for the model trained using $\alpha = 0.001$ and MAXLoss. The standard deviation per class output is shown as a black line. The class naming convention consists of the 2-letter ARPABET (Rice, 1976) phoneme representation followed by an underscore and either a 0 for *mispronounced* or a 1 for *correctly* pronounced. The bar plots in blue correspond to the $\mathbf{L_A}$ logits independent of the assessor tag η. The bar plots in red correspond to the $\mathbf{L_b}$ bias logits. The wider variability and lower mean values in the bias outputs confirm the role of the bias acting as a gating mechanism. Figure 6.3 shows similar bar plots for the logits of the model trained using $\alpha = 0.001$ and ABSLoss. The use of ABSLoss did not cause a behaviour noticeably different to the one of the model trained using MAXLoss when $\alpha = 0.001$.

Recall models trained with an $\alpha > 0.001$ shifted the statistics of $\mathbf{L_A}$ and $\mathbf{L_b}$ completely in Table 6.3. Said change is shown in the corresponding average logit bar plots. Figure 6.4 shows the output statistics per class for the model trained using $\alpha = 0.1$ and MAXLoss. Notice the bias (red bar plot) kept on average most of its outputs close to zero, meaning a very limited effect over $\mathbf{L_A}$. The behaviour of the bias in Figure 6.4 matches the expectation of the bias not being the main drive of the assessor model (see Section 5.5). The bias bar plots also show an average positive output for a reduced set of output classes.

Table 6.4: Phoneme classes with a positive $\mu_{\mathbf{L_b}}$. The script (1) corresponds to the *correct pronunciation* label, the script (0) corresponds to the *mispronunciation* label. The inter-assessor agreement (I), Cohen's κ and counts in the annotation (N) are also shown.

Phoneme	IPA	$\mu_{\mathbf{L}_\mathbf{A}^{(1)}}$	$\mu_{\mathbf{L}_\mathbf{A}^{(0)}}$	$\mu_{\mathbf{L}_\mathbf{b}^{(1)}}$	$\mu_{\mathbf{L}_\mathbf{b}^{(0)}}$	I	κ	N
aw	aʊ	-9.59	-10.96	-1.07	0.94	0.96	0.54	103
ea	eə	-22.33	-22.76	8.08	8.91	0.0	-0.50	2
em	ɛm	-13.13	-20.36	1.21	7.42	0.87	0.63	53
ng	ŋ	-7.39	-13.45	-3.01	2.82	1.0	1.0	72
w	w	-8.86	-15.59	-0.65	4.48	0.93	0.14	149
uh	ʊ	-17.52	-7.54	4.82	-4.33	0.2	-0.16	30

Table 6.5: Phoneme classes with the most negative $\mu_{\mathbf{L_b}}$. The script (1) corresponds to the *correct pronunciation* label, the script (0) corresponds to the *mispronunciation* label. The inter-assessor agreement (I), Cohen's κ and counts in the annotation (N) are also shown.

Phoneme	IPA	$\mu_{\mathbf{L}_\mathbf{A}^{(1)}}$	$\mu_{\mathbf{L}_\mathbf{A}^{(0)}}$	$\mu_{\mathbf{L}_\mathbf{b}^{(1)}}$	$\mu_{\mathbf{L}_\mathbf{b}^{(0)}}$	I	κ	N
ch	tʃ	-10.16	-1.63	-1.00	-10.57	0.81	0.33	48
oy	ɔɪ	-8.13	-3.72	-2.93	-7.81	0.87	-0.05	53
p	p	-8.33	-0.05	-0.55	-11.06	0.94	0.33	189

Table 6.6: Phoneme classes with the smallest $\mu_{\mathbf{L_b}}$. The script (1) corresponds to the *correct pronunciation* label, the script (0) corresponds to the *mispronunciation* label. The inter-assessor agreement (I), Cohen's κ and counts in the annotation (N) are also shown.

Phoneme	IPA	$\mu_{\mathbf{L}_\mathbf{A}^{(1)}}$	$\mu_{\mathbf{L}_\mathbf{A}^{(0)}}$	$\mu_{\mathbf{L}_\mathbf{b}^{(1)}}$	$\mu_{\mathbf{L}_\mathbf{b}^{(0)}}$	I	κ	N
ay	eə	-7.64	-9.11	-1.67	-0.10	0.92	0.56	133
l	l	-6.72	-7.77	-0.43	0.01	0.92	0.19	264
n	n	-5.83	-7.03	0.04	-0.39	0.95	0.45	375
v	v	-9.60	-9.13	0.08	-0.27	0.43	0.08	109

The models trained with ABSLoss also showed $\mu_{\mathbf{L_b}} > 0$ for the same classes the MAXLoss model did. Figure 6.5 shows the bar plots for the mean logits and standard deviation per class for the model trained using ABSLoss with $\alpha = 0.1$. The phonemes for which both models showed $\mu_{L_b} > 0$ are listed in Table 6.4 with their respective μ_{L_A} and μ_{L_b} for both correctly and incorrectly pronunciation. The script (1) corresponds to the label of *correct pronunciation* and the script (0) corresponds to the label of *mispronunciation*. The inter-assessor agreement (I), Cohen's κ and phoneme occurrences in the annotation (N) are also shown in Table 6.4. The main observation about phonemes with $\mu_{L_b} > 0$ is that many show relatively low counts, considering a mean count of 148 per phoneme class in INA. The negative κ indicates the possibility that the inter-agreement for a phoneme is close to random guessing (McHugh, 2012). Said effect can be observed in /ea/ with a perfect disagreement, the smallest N, and very similar outputs for both correctness labels. Phoneme /ng/ is also worth mentioning since it has a low N and a perfect inter-assessor agreement, yet its bias output is not even the

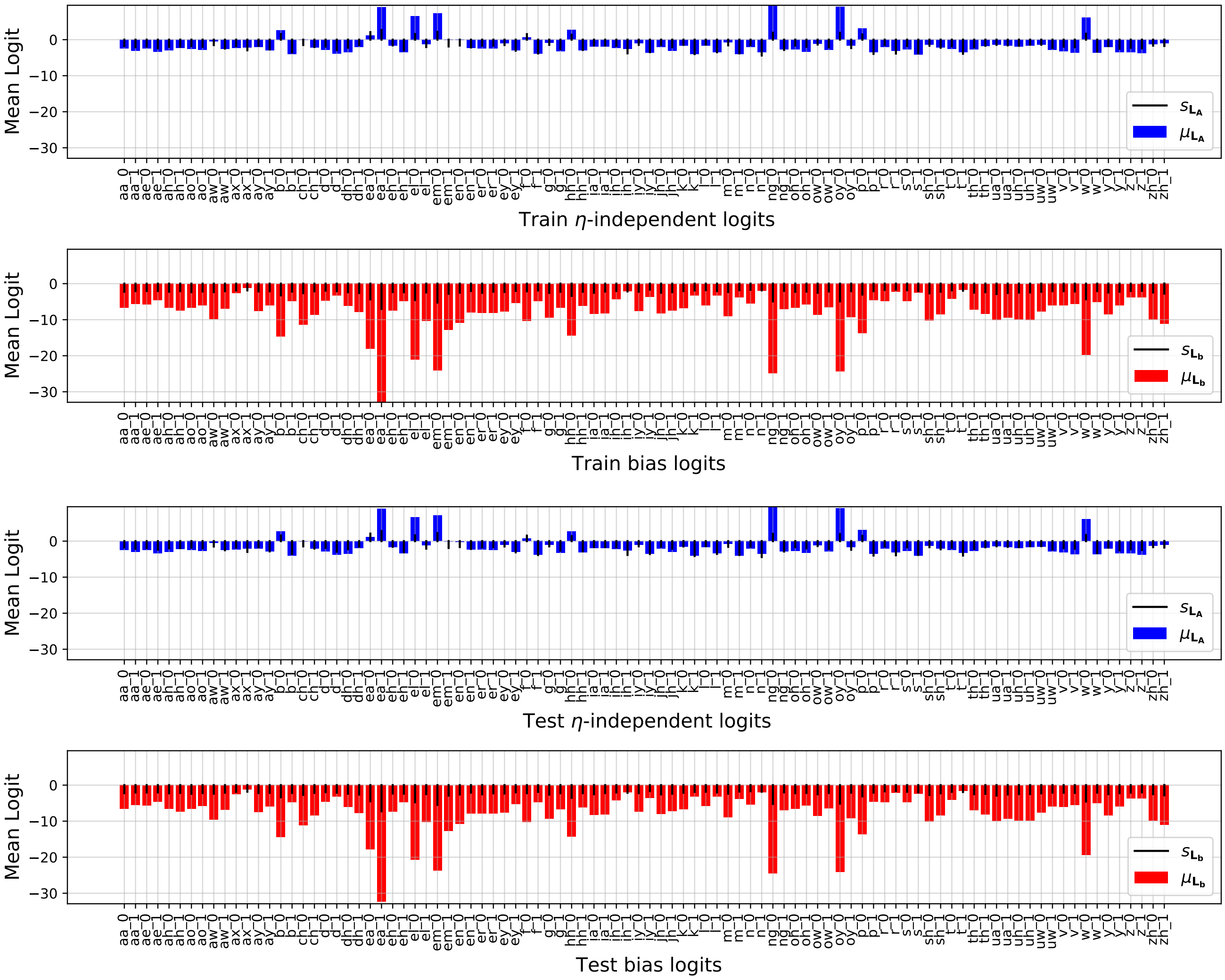


Figure 6.2: The η-independent mean logits $\mu_{\mathbf{L_A}}$ and standard deviation $s_{\mathbf{L_A}}$ of the model trained using $\alpha = 0.001$ and MAXLoss. The mean bias logits $\mu_{\mathbf{L_b}}$ with their standard deviation $s_{\mathbf{L_b}}$ are also plotted.

smallest in Table 6.4.

The bias bar plot for the ABSLoss model in Figure 6.5 also shows output classes with a large negative μ_{L_b} for which the MAXLoss model did not. Said phonemes are listed in Table 6.5 along with their μ_{L_A} and μ_{L_b} for both correctness labels. Coefficient I, κ and N are also listed. Similar to the phonemes with a positive μ_{L_b}, Table 6.5 shows low counts. The high I and small κ could indicate a proportion of agreement to be caused by chance. The split of the output classes into *correct* and *incorrect* realizations offer a different visualization of agreement. For /ch/ and /p/ in Table 6.5, $|\mu_{\mathbf{L_b}^{(0)}}| > |\mu_{\mathbf{L_A}^{(0)}}|$ and $|\mu_{\mathbf{L_A}^{(1)}}| > |\mu_{\mathbf{L_b}^{(1)}}|$ in a similar proportion. It seems like the model considers correct pronunciations to be more assessor-independent than mispronunciations.

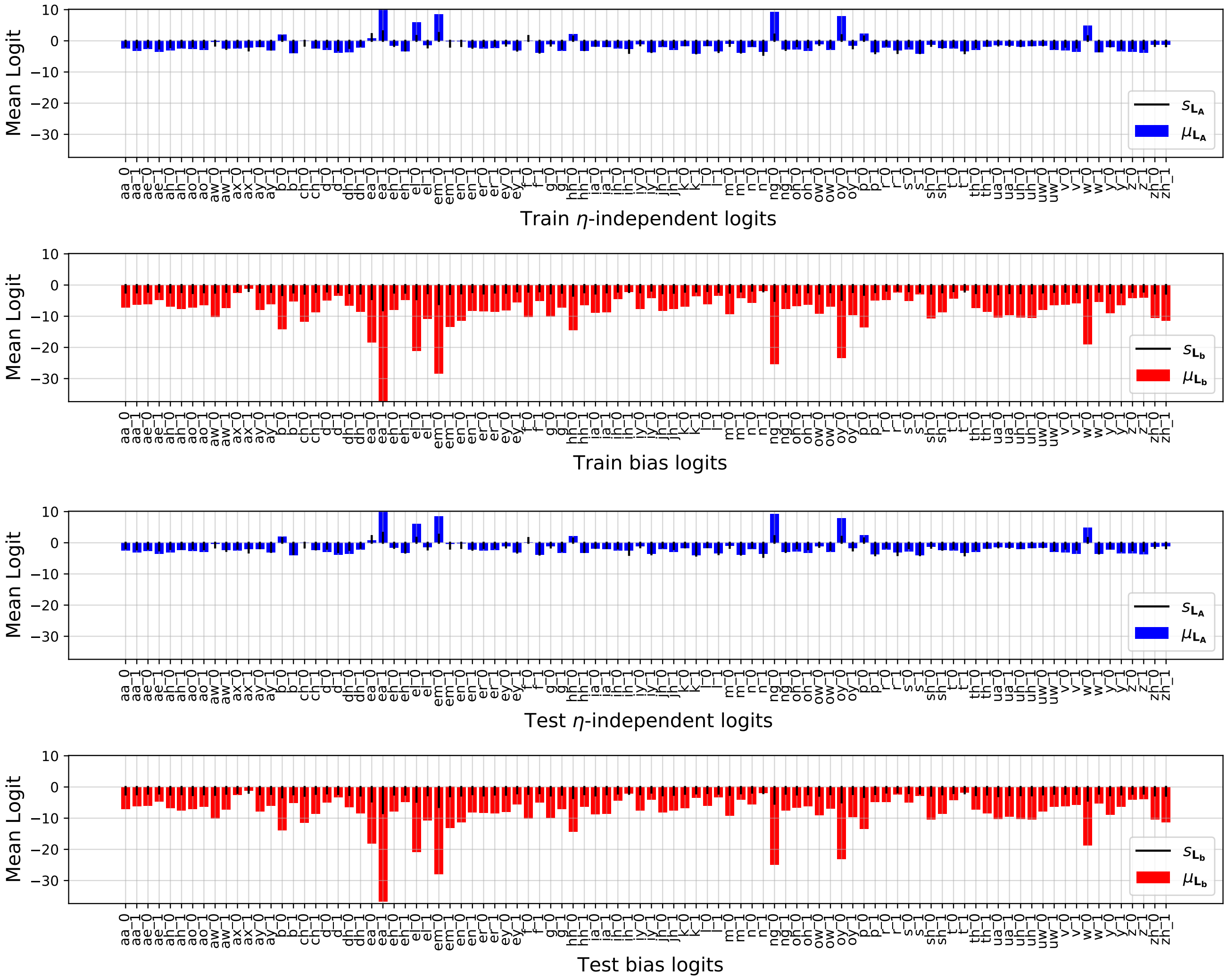


Figure 6.3: The η-independent mean logits $\mu_{\mathbf{L_A}}$ and standard deviation $s_{\mathbf{L_A}}$ of the model trained using $\alpha = 0.001$ and ABSLoss. The mean bias logits $\mu_{\mathbf{L_b}}$ with their standard deviation $s_{\mathbf{L_b}}$ are also plotted.

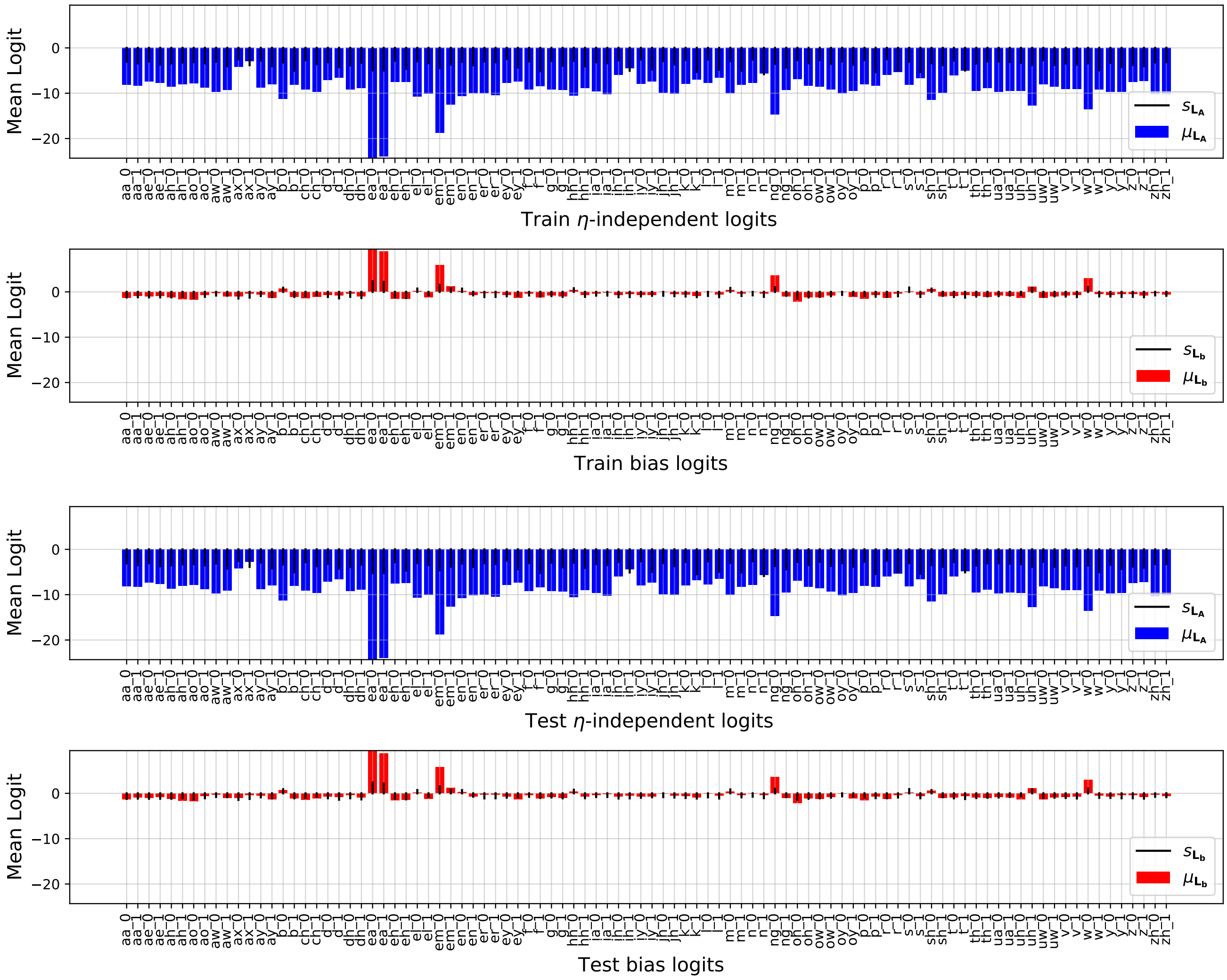


Figure 6.4: The η-independent mean logits $\mu_{\mathbf{L_A}}$ and standard deviation $s_{\mathbf{L_A}}$ of the model trained using $\alpha = 0.1$ and MAXLoss. The mean bias logits $\mu_{\mathbf{L_b}}$ with their standard deviation $s_{\mathbf{L_b}}$ are also plotted.

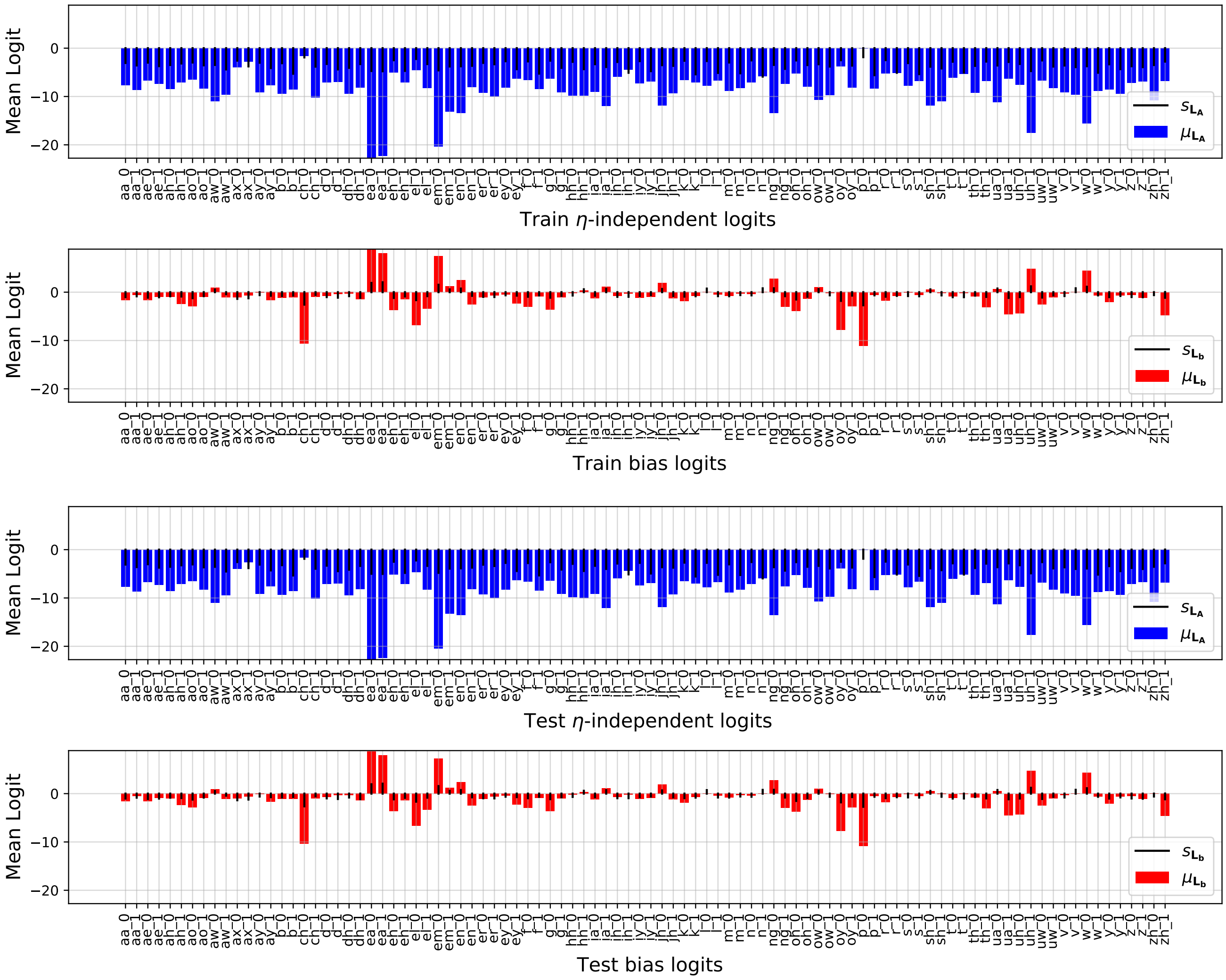


Figure 6.5: The η-independent mean logits $\mu_{\mathbf{L_A}}$ and standard deviation $s_{\mathbf{L_A}}$ of the model trained using $\alpha = 0.1$ and ABSLoss. The mean bias logits $\mu_{\mathbf{L_b}}$ with their standard deviation $s_{\mathbf{L_b}}$ are also plotted.

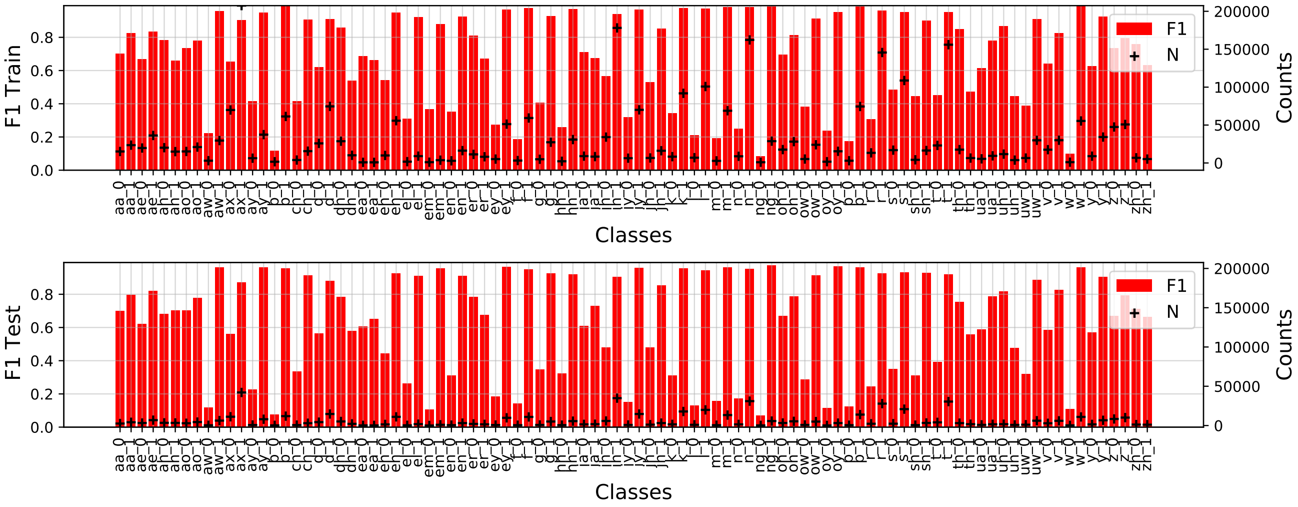


Figure 6.6: Bar plots corresponding to the F1 scores of the ABSLoss model with $\alpha = 0.1$ for detecting each output class in the x-axis. The results on Train are shown on top while the results on Test are at the bottom. The counts (N) of each class are marked with a + for the y-axis on the right side.

A small set of phoneme classes with the smallest bias observed in the ABSLoss model are listed in Table 6.6. It was noted that for this particular case, the phonemes showed higher counts than the ones with either a positive or a large negative μ_{L_b}. Table 6.6 also shows a high I while κ ranges from close to zero to approximately 0.5. However, μ_{L_b} remained constantly smaller than μ_{L_A}. This behaviour is still preferred to a bias FFN_b with an overall expected response larger than the one of FFN_A.

Part of the unexpected behaviour observed in μ_{L_b} could be related to a class imbalance. The use of ABSLoss showed that μ_{L_b} could have a positive output, diverging from the previous interpretation of the bias acting as a gating function. The network could also just not be able to learn classes with relatively low occurrences. Figure 6.6 shows bar plots for the ABSLoss model with $\alpha = 0.1$ on detecting each output class. The bar plots correspond to the F1 score per class on the Train set (top) and Test set (bottom). The counts for each acoustic segment in INA containing a given class are marked using (+) for the scale shown on the right y-axis. Recall that the acoustic segments were obtained using a sliding window (see Section 6.4.2); hence the numbers in Figure 6.6 are larger than the N values shown in Tables 6.4, 6.5 and 6.6.

The bar plots in Figure 6.6 indicate the model was better at detecting correct pronunciations. The lack of mispronunciation examples in the Train set is noticeable. For most of the phonemes with $\mu_{L_b} > 0$ listed in Table 6.4, the F1 scores in Figure 6.6 were the lowest. Even the mispronounced /ng/, with a perfect inter-assessor agreement, showed a low $F1 = 0.08$ on Train and $F1 = 0.07$ on Test. Class imbalance remains one of the main problems in Machine Learning (ML) and in this experiment.

6.4.5 Summary

The ASIM3N showed a similar behaviour between the assessor-independent logits $\mathbf{L_A}$ and the bias logits $\mathbf{L_b}$. To reduce the similarity and redundancy between the components of the assessor model, cosine similarity was used as a training criterion. The $CS(\mathbf{L_A}, \mathbf{L_b})$ was added to the BCE in Equation (6.4) to train a deep model for detecting mispronounced segments. The ASIM3N was trained using multiple α weights for the $CS(\mathbf{L_A}, \mathbf{L_b})$ penalty. Two loss functions were used: MAXLoss in Equation (6.5) and ABSLoss in Equation (6.5). MAXLoss only penalizes cases where $CS(\mathbf{L_A}, \mathbf{L_b}) > 0$ while ABSLoss aims for $CS(\mathbf{L_A}, \mathbf{L_b}) \approx 0$. Both loss functions reduced $CS(\mathbf{L_A}, \mathbf{L_b})$ considerably and showed a slight improvement in the performance of the ASIM3N on detecting mispronunciations. The models trained using ABSLoss showed that certain phonemes would show a $\mu_{L_b} > 0$, changing the previous interpretation of the bias logits acting as a gating function observed in Section 5.7.5. A deeper observation in the classification metrics for the individual output classes confirmed the ASIM architecture difficulties from the lack of mispronunciation examples.

6.5 Mutual Information Minimization

Section 2.1.2, it is explained the importance of the personal experience of the listener on perception bias. Therefore, it is preferred for the bias output $\mathbf{L_b}$ to be more dependent on the assessor tag η than the assessor-independent $\mathbf{L_A}$. In the previous Section 6.4, the reduction of $CS(\mathbf{L_A}, \mathbf{L_b})$ made the logits on average less correlated and practically orthogonal with respect to each other. The loss function could decrease $corr(\mathbf{L_A}, \mathbf{L_b})$ further or even reach to be zero, yet this does not mean the logits have reduced their dependency at all (Kotz and Drouet, 2001).

It is true, the bias function in the assessor model in Equation (5.10) was made explicitly dependant on the bias-free assessment function. However, similar to the case of $CS(\mathbf{L_A}, \mathbf{L_b})$, the dependence $P(\mathbf{L_b}|\mathbf{L_A})$ could be reduced without decreasing the performance of the assessment model. Any decrease in dependency means an increase in the uncertainty of $\mathbf{L_b}$ given that $\mathbf{L_A}$ is known. The model would have to rely more on η to determine $\mathbf{L_b}$.

A measure for dependence between two random variables is called mutual information (MI). The MI for two random variables x and y is:

$$\mathrm{MI}(x; y) = \mathbb{E}_{p(x,y)} \left[\log \frac{p(x, y)}{p(x)p(y)} \right] \tag{6.7}$$

, where $\mathbb{E}_{p(x,y)}$ is the expected value over the joint probability $p(x, y)$.

The MI has been used in ML to both increase and reduce the dependency between variables. Some of the most successful uses of MI in ML have been representation learning (Chen et al., 2016, Zhu et al., 2020), feature selection (Vergara and Estévez, 2014) and disentangled representation (Sanchez et al., 2020). The minimization of MI is also useful for reducing the bias in a model caused by the data set used for training. The reduction of the model bias is often called *unbiased representation* and has been explored for domain adaptation and for what is known as *algorithm fairness* (Khan and Heisterkamp, 2016, Ragonesi et al., 2021). In a nutshell, unbiased representation aims for better generalization by learning a representation which is not subject to particular data attributes (Ragonesi et al., 2021).

It is worth exploring strategies for algorithm fairness in the context of L2 PA for future work. Meanwhile, this section focuses on the reduction of $\mathrm{MI}(\mathbf{L_b}; \mathbf{L_A})$. Similar to the penalization of $CS(\mathbf{L_A}, \mathbf{L_b})$ as part of the ASIM loss in Equation (6.4), $\mathrm{MI}(\mathbf{L_b}; \mathbf{L_A})$ is included in the loss to be reduced along the BCE. A scalar β is used as a weight for the term $\mathrm{MI}(\mathbf{L_b}; \mathbf{L_A})$, hence the ASIM loss function becomes:

$$\mathrm{Loss}(\mathbf{l}, \hat{\mathbf{l}}, \mathbf{L_A}, \mathbf{L_b}) = -\frac{1}{N} \sum_{i=1}^{N} \left[\mathbf{l}_i \cdot \log \hat{\mathbf{l}}_i + (1 - \mathbf{l}_i) \cdot \log(1 - \hat{\mathbf{l}}_i) - \beta \cdot \mathrm{MI}(\mathbf{L_b}_i; \mathbf{L_A}_i) \right] \tag{6.8}$$

The use of MI presents the problem of estimating its actual value when the true distribution $p(x)$ is not known. However, a sample $x \sim p(x)$ is usually available. The estimation of

$\mathrm{MI}(x;y)$ is intractable due to $p(y)$, yet a parametric distribution $p_\theta(y|x))$ can be learned as a variational bound (Poole et al., 2018).

6.6 Contrastive Log-ratio Upper Bound of MI

The goal of using the parametric $p_\theta(y|x))$ is to obtain a bound for $\mathrm{MI}(x;y)$ which is differentiable and scalable. Since $\mathrm{MI}(\mathbf{L_b};\mathbf{L_A})$ needs to be reduced, an upper bound is required. The Contrastive Log-ratio Upper Bound of Mutual Information (CLUB) (Cheng et al., 2020) fulfils the role. CLUB defines an MI upper bound as the likelihood ratio in Equation (6.9).

$$\mathrm{MI}_C(x;y) = \mathbb{E}_{p(x,y)}[\log p(y|x)] - \mathbb{E}_{p(x)}\mathbb{E}_{p(y)}[\log p(y|x)] \tag{6.9}$$

In (Cheng et al., 2020), $\mathrm{MI}_C(x;y)$ is proven to be an upper bound of $\mathrm{MI}(x;y)$ by calculating the gap Δ between them:

$$\Delta = \mathrm{MI}_C(x;y) - \mathrm{MI}(x;y) \tag{6.10}$$
$$\Delta = \mathbb{E}_{p(x,y)}[\log p(y|x)] - \mathbb{E}_{p(x)}\mathbb{E}_{p(y)}[\log p(y|x)] - \mathbb{E}_{p(x,y)}[\log p(y|x) - \log p(y)] \tag{6.11}$$
$$\Delta = \mathbb{E}_{p(x,y)}[\log p(y)] - \mathbb{E}_{p(x)}\mathbb{E}_{p(y)}[\log p(y|x)] \tag{6.12}$$
$$\Delta = \mathbb{E}_{p(y)}[\log p(y)] - \mathbb{E}_{p(x)}\mathbb{E}_{p(y)}[\log p(y|x)] \tag{6.13}$$
$$\Delta = \mathbb{E}_{p(y)}\Big[[\log p(y)] - \mathbb{E}_{p(x)}[\log p(y|x)]\Big] \tag{6.14}$$
$$\tag{6.15}$$

The definition of the marginal distribution states that:

$$p(y) = \int p(y|x)p(x)dx = \mathbb{E}_{p(x)}[p(y|x)] \tag{6.16}$$

Therefore:

$$\Delta = \mathbb{E}_{p(y)}\Big[\log[\mathbb{E}_{p(x)}[p(y|x)]] - \mathbb{E}_{p(x)}[\log p(y|x)]\Big] \tag{6.17}$$

Jensen's Inequality states that $log[\mathbb{E}_{p(x)}[p(y|x)]] > \mathbb{E}_{p(x)}[\log p(y|x)]$, meaning $\Delta > 0$. To make $\mathrm{MI}_C(x;y) \approx \mathrm{MI}(x;y)$, the likelihood $p(y|x)$ needs to show the same value for any x. In other words, Δ is minimized when $p(x)p(y) = p(x,y)$.

Since $p(y|x)$ is not known, a variational distribution $q_\theta(y|x)$ with parameters θ is learned to approximate $p(y|x)$. An estimate $\mathrm{MI}_{C\theta}(x;y)$ can be obtained from a sample of pairs $\{(x_i,y_i)\}_{i=1}^N$.

$$\mathrm{MI}_{C\theta}(x;y) = \frac{1}{N}\sum_{i=1}^{N}\left[\log q_\theta(y_i|x_i) - \frac{1}{N}\sum_{j=1}^{N}\log q_\theta(y_j|x_i)\right] \tag{6.18}$$

There is no guarantee that $q_\theta(y|x) = p(y|x)$. Therefore, the estimate $\mathrm{MI}_{C\theta}(x;y)$ is now

Algorithm 6.1 MI minimization using $\mathrm{MI}_{C\theta}(x;y)$

```
for each training iteration do:
    Sample $\{(x_i, y_i)\}_{i=1}^{N}$ from $p_\sigma(x,y)$
    Log-likelihood $\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\log q_\theta(y_i|x_i)$
    Update $q_\theta(y_i|x_i)$ by maximizing $\mathcal{L}(\theta)$
    for $i = 1$to$N$ do
        $U_i = \log q_\theta(y_i|x_i) - \frac{1}{N}\sum_{j=1}^{N}\log q_\theta(y_j|x_i)$
    end for
    Update $p_\sigma(x,y)$ by minimizing $\mathrm{MI}_{C\theta}(x;y) = \frac{1}{N}\sum_{i=1}^{N} U_i$
end for
```

bounded by the Kullback–Leibler divergence (KL):

$$\mathrm{KL}(p(x,y)||q_\theta(x,y)) = \sum_{x,y} p(x,y)\log\frac{p(x,y)}{q_\theta(x,y)} \tag{6.19}$$

The variational approximation $q_\theta(y|x)$ is learned along the MI minimization of a joint variational distribution of $p_\sigma(x,y)$. The latter one corresponds, for example, to $p(\mathbf{L_b}, \mathbf{L_A})$ from the ASIM. The process consists in sampling from $p_\sigma(x,y)$, update $q_\theta(y|x)$ by maximizing the sample log-likelihood $\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\log q_\theta(y_i|x_i)$, then calculate $\mathrm{MI}_{C\theta}(x;y)$ as in Equation (6.18) and propagate it to the parameters of $p_\sigma(x,y)$ for a further update. The alternate optimization of both $q_\theta(y|x)$ and $p_\sigma(x,y)$ is shown in Algorithm 6.1.

6.7 MI Minimization of the Assessor Model Components

The implementation of CLUB for MI reduction requires more details about the variational distributions $q_\theta(y|x)$ and $p_\sigma(x,y)$. It was mentioned earlier that $p_\sigma(x,y)$ corresponds to the ASIM's own $p(\mathbf{L_A}, \mathbf{L_B})$. Said joint probability is also dependent on the acoustic segment $\mathbf{O}^{(w)}$ associated with the prompt w and the assessor tag η. Since the objective is to reduce the dependency $\mathbf{L_B}$ has on $\mathbf{L_A}$, other dependencies are ignored for now. It is expected that by decreasing $\mathrm{MI}(\mathbf{L_b}; \mathbf{L_A})$, the bias classifier FFN_b will increase the dependence of $\mathbf{L_B}$ on η.

In terms of Algorithm 6.1, the pairs sample $\{(x_i, y_i)\}_{i=1}^{N}$ corresponds to the $\mathbf{L_A}$ and $\mathbf{L_b}$ of a mini-batch of size N. The parametric $q_\theta(\mathbf{L_b}|\mathbf{L_A})$ is not known. For the sake of simplicity and to satisfy the log-concavity condition of Jensen's Inequality, $q_\theta(\mathbf{L_b}|\mathbf{L_A})$ is assumed to follow a multivariate log-normal distribution. In (Cheng et al., 2020), $q_\theta(y_i|x_i)$ was computed as a log-normal probability using the mean μ_θ and variance σ_θ^2 obtained from two FFNs respectively. The FFNs used x as input and were updated at the beginning of each training step for $p_\sigma(x,y)$. The same approach is taken for $q_\theta(L_{bi}|L_{Ai})$. Finally, for each forward-pass, $\mathrm{MI}_{C\theta}(\mathbf{L_A}; \mathbf{L_b})$ is propagated as part of the training loss.

An experiment on using CLUB to minimize $\mathrm{MI}_{C\theta}(\mathbf{L_A}; \mathbf{L_b})$ was carried out. The architecture used was the ASIM3N, as shown in Figure 5.6 and defined in detail in Section 6.7.1. The architecture, hereby called Mutual Information driven Attention-Based Segmental Incorrectness

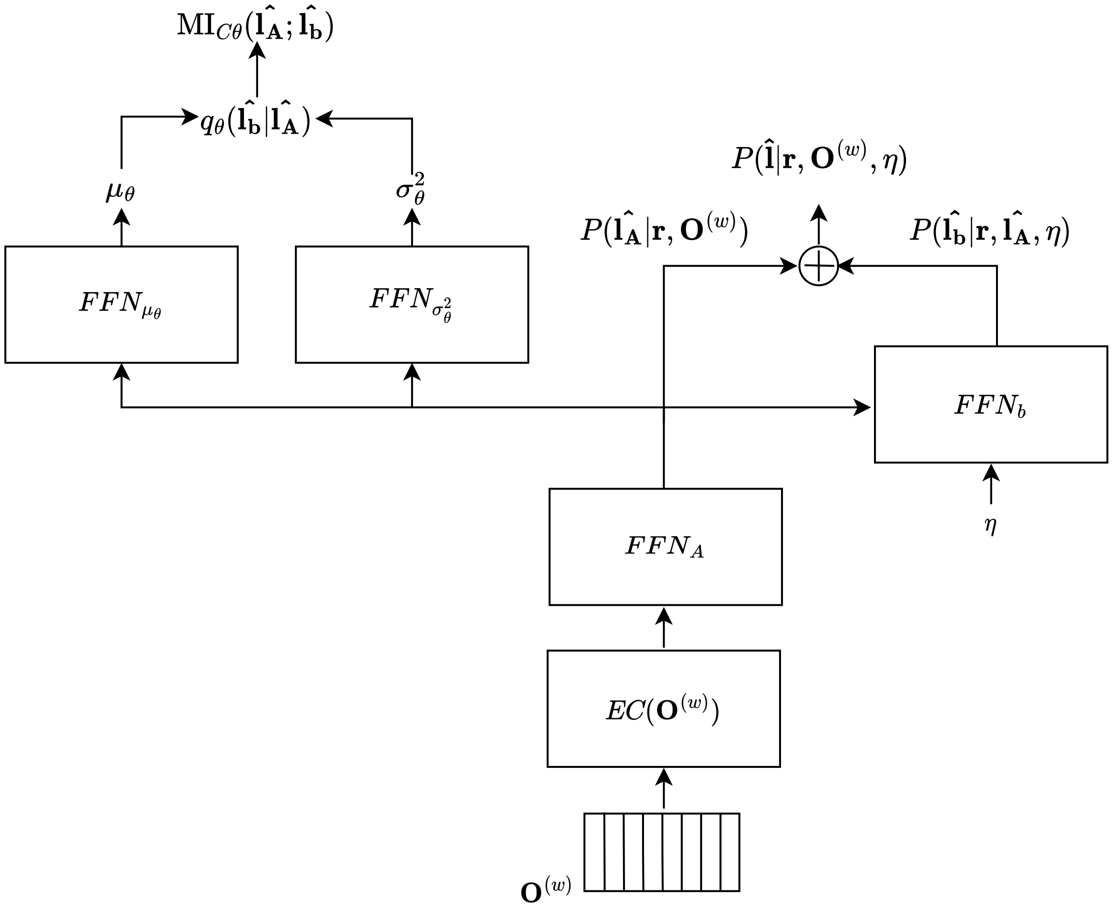

$\mathrm{MI}_{C\theta}(\hat{\mathbf{l}_\mathbf{A}};\hat{\mathbf{l}_\mathbf{b}})$
$q_\theta(\hat{\mathbf{l}_\mathbf{b}}|\hat{\mathbf{l}_\mathbf{A}})$
μ_θ
σ^2_θ
FFN_{μ_θ}
$FFN_{\sigma^2_\theta}$
$P(\hat{\mathbf{l}}|\mathbf{r},\mathbf{O}^{(w)},\eta)$
$P(\hat{\mathbf{l}_\mathbf{A}}|\mathbf{r},\mathbf{O}^{(w)})$
$P(\hat{\mathbf{l}_\mathbf{b}}|\mathbf{r},\hat{\mathbf{l}_\mathbf{A}},\eta)$
FFN_b
η
FFN_A
$EC(\mathbf{O}^{(w)})$
$\mathbf{O}^{(w)}$

Figure 6.7: IASIM architecture. Two FFNs are trained to infer the mean μ_θ and variance σ^2_θ to estimate the parametric $q_\theta(\hat{\mathbf{l}_\mathbf{b}}|\hat{\mathbf{l}_\mathbf{A}})$.

Model (IASIM), was trained to learn the joint annotation reference from various trained assessors of L2 pronunciation. The IASIM was trained using the loss defined in Equation (6.8) using different values for the weight β. The F1 score and Cohen's κ were used to score the IASIM on detecting mispronounced segments given the assessor η. The final bound $\mathrm{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$ was also observed for effect on the behaviour of the network components. The $\mathrm{KL}(\mathbf{L_b}||\mathbf{L_A})$ was used to compare the information gain from $\mathbf{L_b}$ over $\mathbf{L_A}$ from using CLUB. The KL served as a more consistent criterion due to the expected error $\mathrm{KL}(p(\mathbf{L_A},\mathbf{L_b})||q_\theta(\mathbf{L_A},\mathbf{L_b}))$. It is expected that $p(\mathbf{L_A},\mathbf{L_b})$ gets closer to $p(\mathbf{L_A})p(\mathbf{L_b})$. A reduction in $corr(\mathbf{L_A},\mathbf{L_b})$ could also occur, yet is not a necessary condition for $p(\mathbf{L_A},\mathbf{L_b}) = p(\mathbf{L_A})p(\mathbf{L_b})$. Since the true $p(\mathbf{L_b}|\mathbf{L_A})$ is not known, the only metric available for measuring the effect of CLUB are both the upper bound $\mathrm{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$ and the log-likelihood of $q_\theta(\mathbf{L_b}|\mathbf{L_A})$. A variational $q_\vartheta(\mathbf{L_b}|\eta)$ was learned simultaneously for the assessment of the IASIM. It is also expected to observe an increase in $\mathrm{MI}_{C\theta}(\eta;\mathbf{L_b})$, an upper bound similar to $\mathrm{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$.

6.7.1 Model Training Setup

The IASIM is an ASIM3N with an additional pair of FFNs to learn mean μ_θ and variance σ^2_θ for the variational $q_\theta(\mathbf{L_b}|\mathbf{L_A})$. Figure 6.7 shows a diagram for the IASIM architecture. The IASIM network used in this experiment consisted of a 6-layer BDLSTM of size 64 and an additive self-attention module with 128 linear weights for the encoding of the acoustic segment $\mathbf{O}^{(w)}$. The assessor-independent FFN_A and bias FFN_b were 6 layers deep, each with a size of 1024 linear units. The output layers of both FFN_A and FFN_b have two units for each phoneme class, corresponding either to a correct or incorrect realization given the annotation. The variational parameters μ_θ and σ^2_θ were estimated, each with a 5 layer deep FFN with a size of 512 linear units. The dimensionality of μ_θ and σ^2_θ correspond to each output class in IASIM, 94.

The first 13 PLP coefficients with their first and second order time differentials were used as the input for the BDLSTM. The assessor identity η is a one-hot encoding concatenated to $\mathbf{L_A}$ before passing it to FFN_b. After obtaining both $\mathbf{L_A}$ and $\mathbf{L_b}$ from the current mini-batch, these are used to update the networks for μ_θ and σ^2_θ. Different IASIM were trained using the weight β set to $1E-6$, $1E-5$, $1E-4$, 0.001, 0.01, 0.1 and 1. An IASIM trained with $\beta = 0$ was used as a baseline. As in previous experiments, the models were trained using the Adam optimizer (Kingma and Ba, 2014) until reaching 6 epochs without any improvement in the loss function for the Test set. The final model assessed corresponds to the one reaching the lowest loss on the Test set.

6.7.2 Experiment Dataset

The IASIM was trained on the INA set introduced originally in Section 3.6.1. The data set was split as defined in Section 3.6.1. The INA data was split at 85% of recordings for Train, with 15% left for Test. The split was balanced for sex, age, and L2 proficiency and had no speaker overlap between Train and Test. All the recordings were marked for mispronunciation at phoneme label by three phoneticians *a1*, *a2* and *a3*.

Short acoustic segments were created using a moving window of $0.5s$ with a $0.05s$ stride. The alignment for the correctness labels $\mathbf{l}$ was done via a triphone-based DNN-HMM AM used in (Nicolao et al., 2015) (see Section 3.6.1).

6.7.3 Effect of CLUB on Detecting Mispronounced Segments

The glsiasim models trained using CLUB mainly showed gradients in the order of $1E10$. The plots in Figure 6.8 show the BCE component for the range of β factors used in the IASIM loss (Equation 6.8). The BCE for models using $\beta > 1E-4$ grew during the first 10 training epochs and then stayed the same. Table 6.7 shows the variational bounds $\text{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$, the log-likelihood $\mathcal{L}(\theta)$ for the distribution $q_\theta(\mathbf{L_b}|\mathbf{L_A})$ and the resulting divergence of the output logits $\text{KL}(\mathbf{L_b}||\mathbf{L_A})$ in bits. The divergence was computed directly from the observed logits. The $\mathcal{L}(\theta)$ for the models using $\beta > 1E-4$ were relatively low, hence their CLUB ratios cannot be considered a valid upper bound for $\text{MI}_(\mathbf{L_A};\mathbf{L_b})$. Figure 6.9 shows the evolution of $\mathcal{L}(\theta)$ for all

Table 6.7: The upper bound $\mathrm{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$ and the divergence $\mathrm{KL}(\mathbf{L_b}||\mathbf{L_A})$ in bits for both IASIM and the baseline. The final log-likelihood $\mathcal{L}(\theta)$ for the parametric distribution is also shown.

		Train			Test	
Model	$\mathrm{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$	$\mathcal{L}(\theta)$	$\mathrm{KL}(\mathbf{L_b}\|\|\mathbf{L_A})$	$\mathrm{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$	$\mathcal{L}(\theta)$	$\mathrm{KL}(\mathbf{L_b}\|\|\mathbf{L_A})$
1	-4.63E24	-2.75E25	9.5E7	5.8E14	-1.51E25	9.5E7
Baseline	308.24	-121.52	0.0186	306.10	-98.33	0.0189
1E-1	-5.83E24	-3.8E25	1.26E8	7.39E16	-4.10E25	1.26E8
1E-2	-1.29E25	-2.76E26	8.97E8	8.07E15	-3.74E25	8.97E8
1E-3	-1.76E26	-1.14E27	2.98E8	2.27E17	-4.18E27	2.98E8
1E-4	9.11	-102.28	0.0635	9.7748	-98.44	0.0638
1E-5	90.52	-31.57	0.0306	93.28	-5.00	0.0308
1E-6	214.13	-82.79	0.0285	219.64	-56.16	0.0291

Table 6.8: F1 score and Cohen's Kappa (κ) for the baseline and the IASIM on detecting mispronounced segments across all assessors in the INA set.

	Train		Test	
β	F1	κ	F1	κ
1	0.3832	-0.0390	0.3626	-0.0453
Baseline	0.8105	0.6538	0.7404	0.5573
1E-1	0.2739	0.0403	0.2653	0.0330
1E-2	0.4329	0.0347	0.4213	0.0348
1E-3	0.4562	0.0363	0.4403	0.0417
1E-4	0.8083	0.6498	0.7400	0.5554
1E-5	0.8092	0.6514	0.7405	0.5566
1E-6	0.8007	0.6357	0.7380	0.5520

the various IASIM. Meanwhile, Figure 6.10 shows the variational bound $\mathrm{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$ for the IASIM models during training. The IASIM with $\beta = 1E-3$ shows the most negative $\mathcal{L}(\theta) = -1.14E27$. The poor parametric $q_\theta(\mathbf{L_b}|\mathbf{L_A})$ for $\beta = 1E-3$ is reflected in the inconsistent and large curve for the Mutual Information (MI) upper bound in Figure 6.10.

The training behaviour for $\beta \leq 1E-4$ is appreciated better without the models that failed to learn a valid $q_\theta(\mathbf{L_b}|\mathbf{L_A})$. Figure 6.11 shows the BCE component for $\beta \leq 1E-4$ was not too different from the baseline. Hence, when IASIM manages to approximate $p(\mathbf{L_b}|\mathbf{L_A})$, the BCE is not heavily affected. The $\mathcal{L}(\theta)$ in Figure 6.12 for $\beta \leq 1E-4$ seemed analogue to Figure 6.13 of $\mathrm{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$ for the same models. The gap between the $\mathcal{L}(\theta)$ curves between the $\beta \leq 1E-4$ and the baseline was reflected in the MI bound. As the $\mathcal{L}(\theta)$ decreases, $\mathrm{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$ grows. The exception was $\beta = 1E-4$, for which $\mathcal{L}(\theta)$ did not vary during training in the same way the rest of the models did.

The models were scored for detecting mispronounced segments according to the three annotators in INA. The use of CLUB decreased the baseline results overall. Table 6.8 shows the overall F1 score and Cohen's Kappa (κ) for the IASIMs trained in this experiment. The decrease in performance was dependent on the ability of the FFNs for approximating $p(\mathbf{L_b}|\mathbf{L_A})$. As expected, the $\beta > 1E-4$ models with an invalid $q_\theta(\mathbf{L_b}|\mathbf{L_A})$ also performed poorly when detecting mispronunciations. The rest of the models performed slightly under the baseline. All the different β tested in Table 6.7 increased the $\mathrm{KL}(\mathbf{L_b}||\mathbf{L_A})$; however, only $\mathrm{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b}) = 0$ were considered valid upper bounds.

The IASIM with $\beta = 1E-4$ showed the largest $\mathrm{KL}(\mathbf{L_b}||\mathbf{L_A})$ from the models with a valid $q_\theta(\mathbf{L_b}|\mathbf{L_A})$ (Table 6.7). However, $\beta = 1E-5$ performed the closest to the baseline with $\kappa = 0.6514$ for Train and $\kappa = 0.5566$ for Test. Similar to the models trained using the CS penalty listed in Table 6.1 and 6.2, a small decrease in performance might be the price for more independent network components. Therefore, the name IASIM is hereby reserved for the model trained with $\beta = 1E-5$ as it managed to increase $\mathrm{KL}(\mathbf{L_b}||\mathbf{L_A})$ with the least reduction in performance. The baseline remains the same, the ASIM3N.

The IASIM $\mathrm{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$ was indeed lower than the one of the baseline. The IASIM $\mathcal{L}(\theta) = -31.57$ for Train is considerably higher than the baseline $\mathcal{L}(\theta) = -121.52$ also for Train. The difference in $\mathcal{L}(\theta)$ indicates the MI bound for the baseline was itself bounded by a wider $\mathrm{KL}(p(\mathbf{L_A},\mathbf{L_b})||q_\theta(\mathbf{L_A},\mathbf{L_b}))$. Since the baseline did not propagate the bound, it kept growing until stabilizing.

6.7.4 Effect of CLUB on Logit Similarity

The $CS(\mathbf{L_A},\mathbf{L_b})$ on the IASIM was observed for changes caused by CLUB. The baseline model showed $CS(\mathbf{L_A},\mathbf{L_b}) = 0.8460$ on Train and $CS(\mathbf{L_A},\mathbf{L_b}) = 0.8479$ on the Test set. Meanwhile, IASIM showed a small decrease in the similarity of the logits with $CS(\mathbf{L_A},\mathbf{L_b}) = 0.8369$ on Train and $CS(\mathbf{L_A},\mathbf{L_b}) = 0.8406$ on Test. A more significant de-correlation will not necessarily occur from using CLUB. Therefore, the CS penalty was included along with the MI upper bound in the loss function.

The new loss function in Equation (6.20) uses the scalar α to weight the CS penalty. Equation (6.20) is referred to as Mutual Information driven Attention-Based Segmental Incorrectness Model with Cosine Similarity Penalty (IASIMC). Three additional networks were trained using the same IASIM architecture training setup, except for using IASIMC as the loss function. Three IASIMCs were trained using 0.1, $1E-3$ and $1E-5$ for α. For all the models, $\beta = 1E-5$.

$$\begin{aligned}\mathrm{IASIMC}(\mathbf{l},\hat{\mathbf{l}},\mathbf{L_A},\mathbf{L_b}) = -\frac{1}{N}\sum_{i=1}^{N}\Bigg[&\mathbf{l}_i\cdot\log\hat{\mathbf{l}}_i + (1-\mathbf{l}_i)\cdot\log(1-\hat{\mathbf{l}}_i) - \alpha\cdot\left|\frac{\mathbf{L_A}_i\cdot\mathbf{L_b}_i}{||\mathbf{L_A}_i||\,||\mathbf{L_b}_i||}\right| \\ &-\beta\cdot[\log q_\theta(\mathbf{L_b}_i|\mathbf{L_A}_i) - \frac{1}{N}\sum_{j=1}^{N}\log q_\theta(\mathbf{L_b}_j|\mathbf{L_A}_i)]\Bigg]\end{aligned} \quad (6.20)$$

The IASIMCs were scored for detecting mispronounced segments given the annotation reference. Table 6.9 shows the F1 and κ for each α used. The IASIMC with $\alpha = 1E-3$ outperformed both the baseline and the IASIM results shown in Table 6.8. The IASIMC with $\alpha = 0.1$ did show the best $\kappa = 0.5621$ on Test; yet, its $\mathrm{KL}(\mathbf{L_b}||\mathbf{L_A}) \approx 0$. Table 6.10 shows $\mathrm{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$, the corresponding variational $\mathcal{L}(\theta)$ and $\mathrm{KL}(\mathbf{L_b}||\mathbf{L_A})$ for the three IASIMC. The positive $\mathcal{L}(\theta)^{(\alpha=0.1)}$ indicates the FFNs failed to model the variational parameters μ_θ and σ_θ^2.

Table 6.9: F1 score and Cohen's Kappa (κ) for the baseline and the IASIMC on detecting mispronounced segments given the assessors in the INA set.

	Train		Test	
α	F1	κ	F1	κ
Baseline	0.8105	0.6538	0.7404	0.5573
0.1	0.8118	0.6562	0.7433	0.5621
$1E-3$	0.8152	0.6626	0.7435	0.5612
$1E-5$	0.8102	0.6534	0.7413	0.5583

Table 6.10: The upper bound $\text{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$ and the divergence $\text{KL}(\mathbf{L_b}||\mathbf{L_A})$ for the IASIMC. The final log-likelihood $\mathcal{L}(\theta)$ is included.

		Train			Test	
α	$\text{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$	$\mathcal{L}(\theta)$	$\text{KL}(\mathbf{L_b}\|\|\mathbf{L_A})$	$\text{MI}_{C\theta}(\mathbf{L_A};\mathbf{L_b})$	$\mathcal{L}(\theta)$	$\text{KL}(\mathbf{L_b}\|\|\mathbf{L_A})$
0.1	25.0499	26.2665	0	26.4028	35.4644	0
$1E-3$	89.3971	-43.4641	0.2112	90.2600	-12.4450	0.2119
$1E-5$	89.5556	-32.1807	0.0444	90.5953	-4.0927	0.0448

Table 6.11: The $CS(\mathbf{L_A},\mathbf{L_b})$ for networks trained with MI and CS criteria. The IASIMC combines CLUB with CS penalty.

Model	α	Train	Test
Baseline		0.8460	0.8479
IASIM		0.8369	0.8406
IASIMC	0.1	-7.69E-6	0.0025
IASIMC	$1E-3$	0.0116	0.01471
IASIMC	$1E-5$	0.8479	0.8509

The bound $\text{MI}_{C\theta}^{(\alpha=0.1)}(\mathbf{L_A};\mathbf{L_b}) = 25.0499$ on Train is not a valid one. The CS penalty for $\alpha = 0.1$ dominated over the MI bound. For α values $1E-3$ and $1E-5$, $\mathcal{L}(\theta)$ lies within an acceptable range for log-probabilities. The CS penalty did increase $\text{KL}(\mathbf{L_b}||\mathbf{L_A})$ compared to the values reported in Table 6.7 when CLUB learned a valid $q_\theta(\mathbf{L_b}|\mathbf{L_A})$. Particularly, $\alpha = 1E-3$ achieved the largest $\text{KL}(\mathbf{L_b}||\mathbf{L_A}) = 0.2112$ in the Train set across all the models trained.

The $CS(\mathbf{L_A},\mathbf{L_b})$ for all the networks trained in this section is shown in Table 6.11. The IASIMCs when $\alpha = 0.1$ reached the smallest $CS(\mathbf{L_A},\mathbf{L_b})$, yet it did not manage to make $\mathbf{L_b}$ less dependent on $\mathbf{L_A}$. The IASIMC for $\alpha = 1E-3$ also decreased $CS(\mathbf{L_A},\mathbf{L_b})$ considerable, while showing the largest $\text{KL}(\mathbf{L_b}||\mathbf{L_A})$. IASIMC with $\alpha = 1E-3$ has produced the model with the most independent and least correlated components in the assessor model so far. Finally, $\alpha = 1E-5$ was too small for minimizing $CS(\mathbf{L_A},\mathbf{L_b})$. The fact that $\alpha = 1E-5$ still showed a larger $\text{KL}(\mathbf{L_b}||\mathbf{L_A})$ than the baseline, indicates the MI upper bound dominated over the CS penalty.

6.7.5 Effect of CLUB on the Bias Model

As CLUB makes $\mathbf{L_A}$ and $\mathbf{L_b}$, it is expected that the assessor tag η becomes more relevant for $\mathbf{L_b}$. An additional IASIM was trained for observations of $\text{MI}(\eta;l_{bi})$. The new model

Table 6.12: The upper bound $\mathrm{MI}_{C\vartheta}(\eta; \mathbf{L_b})$ and the log-likelihood $\mathcal{L}(\vartheta)$.

	Train		Test	
Model	$\mathrm{MI}_{C\vartheta}(\eta; \mathbf{L_b})$	$\mathcal{L}(\vartheta)$	$\mathrm{MI}_{C\vartheta}(\eta; \mathbf{L_b})$	$\mathcal{L}(\vartheta)$
Baseline	6.9559	-222.75	0.0059	-236.37
IASIM	9.1273	-88.65	0.0081	-82.38
IASIMSP	8.0498	-97.0531	0.0066	-86.9798

was called Mutual Information driven Attention-Based Segmental Incorrectness Model with Speaker Factors (IASIMSP); it was trained using speaker factors as part of the input for FFN_b. The inclusion of IASIMSP in this section comes from previous findings on speaker metadata having an effect on the performance of the ASIM (see Section 3.3 and 4.3.4). Two speaker factors were chosen from the INA metadata: birthplace (BP) and whether the speaker inhabits a multilingual household (MLH). The combination BP.MLH showed the largest effect for the MaxVote consolidated annotation of INA (see Table 4.5). The BP.MLH was concatenated as a one-hot encoding to the normalized $\mathbf{L_A}$ and the one-hot vector for η. The IASIMSP was also trained using CLUB with $\beta = 1E - 5$. The IASIMSP managed to increase $\mathrm{KL}(\mathbf{L_b}||\mathbf{L_A})$ up to 0.0458, surpassing the logit divergence in IASIM of 0.0306. The performance of the IASIMSP for detecting mispronounced segments showed $\mathrm{F1} = 0.8155$ and $\kappa = 0.6631$ for Train, and $\mathrm{F1} = 0.7383$ and $\kappa = 0.5536$ for Test. Compared to the baseline and IASIM results in Table 6.8, IASIMSP performed similarly to the IASIM.

The MI upper bound of $\mathbf{L_b}$ and η, $\mathrm{MI}_{C\vartheta}(\eta; \mathbf{L_b})$, with its corresponding $\mathcal{L}(\vartheta) = \mathbb{E}_i[\log q_\vartheta(L_{b_i}|\eta_i)]$ are shown in Table 6.12. Both IASIM and IASIMSP showed a $\mathrm{MI}_{C\vartheta}(\eta; \mathbf{L_b})$ larger than the baseline bound of 6.9559 in the Train set. However, the difference in $\mathcal{L}(\vartheta)$ between IASIM and the baseline was considerably large. The $\mathrm{MI}_{C\vartheta}(\eta; \mathbf{L_b})$ for the Test set was also close to zero for all models. The large difference in $\mathrm{MI}_{C\vartheta}(\eta; \mathbf{L_b})$ between Train and Test deemed the values in Table 6.7 inconclusive for the three models.

The different dimensionality of $\mathbf{L_b}$ and the uniformity of $p(\eta)$ made it difficult for CLUB to find an adequate $q_\vartheta(\mathbf{L_b}|\eta)$. However, it is possible to compute MI between η and individual phoneme bias outputs l_{bi}. A small set of English phonemes were selected to observe changes in $\mathrm{MI}(\eta; l_{bi})$. Said phonemes are known to be prone to mispronunciations by Native Language (L1) Dutch speakers. The phonemes chosen were /ʊ/ and /g/ for not having a near equivalent in Dutch, /ɛ/ often confused with the Dutch /æ/, /ɪ/ as it varies depending on the native dialect of the speaker, /w/ usually confused with /v/, and /ʌ/ which is often pronounced as /ə/ (Tops et al., 2001).

The $\mathrm{MI}(\eta; l_{bi})$ in bits for the selected phonemes is shown in Table 6.13 for all three models. For each phoneme class, recall, the underscore 0 stands for the mispronunciation label ($p0$) and the underscore 1 stands for a correct pronunciation ($p1$). Overall, IASIM did increase $\mathrm{MI}(\eta; l_{bi})$ only for $p0$. IASIM reduced $\mathrm{MI}(\eta; l_{bi})$ only for $p1$. The speaker metadata in IASIMSP increased $\mathrm{MI}(\eta; l_{bi})$ for all $p0$ more than IASIM did. Although IASIMSP also reduced $\mathrm{MI}(\eta; l_{bi})$ for all $p1$, the reduction was mostly less than what IASIM did. CLUB reduced the gap of $\mathrm{MI}(\eta; l_{bi})$ between a given $p1$ class and its $p0$ counterpart. The speaker factors helped reduce

Table 6.13: MI($\eta; l_{bi}$) for the selected phonemes on Baseline, IASIM and IASIMSP.

Phoneme	BASE	IASIM	IASIMSP
ʊ$_0$	0.6016	0.6684	0.6732
ʊ$_1$	0.7087	0.6358	0.6425
g_0	0.2228	0.3319	0.4460
g_1	0.6665	0.4971	0.5055
ɛ$_0$	0.1395	0.1634	0.2123
ɛ$_1$	0.5618	0.2887	0.2953
ɪ$_0$	0.0703	0.0777	0.1019
ɪ$_1$	0.3239	0.1394	0.1713
w_0	0.0668	0.1703	0.2295
w_1	0.6159	0.4400	0.4342
ʌ$_0$	0.5455	0.5586	0.5732
ʌ$_1$	0.6244	0.5758	0.5731

Table 6.14: MI($l_{Ai}; l_{bi}$) for the selected phonemes on Baseline, IASIM and IASIMSP.

Phoneme	BASE	IASIM	IASIMSP
ʊ$_0$	3.2032	3.0729	2.2517
ʊ$_1$	5.7693	5.6761	5.6498
g_0	0.2611	0.2730	0.2011
g_1	6.1589	5.8484	5.8750
ɛ$_0$	0.8531	0.8558	0.7162
ɛ$_1$	6.4404	5.8346	5.8240
ɪ$_0$	0.4896	0.4754	0.5903
ɪ$_1$	6.4360	5.7485	5.9283
w_0	0.0889	0.1144	0.0969
w_1	6.4444	6.0541	6.0662
ʌ$_0$	1.7482	1.9635	1.1712
ʌ$_1$	6.0117	5.8425	5.8342

the MI gap even further by increasing the relevance of η for the bias.

Changes in MI($l_{bi}; l_{Ai}$) across the different models are listed in 6.14 for the same phoneme set. The MI($l_{bi}; l_{Ai}$) is greater for $p1$ than for $p0$ across all the models. The use of CLUB decreased MI($l_{bi}; l_{Ai}$) with a few exceptions: g_0 and ʌ$_0$ only for IASIM, ɪ$_0$ for IASIMSP, and w_0 for both models.

The trained models can offer more information about the learned annotation reference. For example, MI($\eta; l_{bi}$) can serve as an indicator for which assessor is more relevant to the bias component. Table 6.15 shows MI($\eta; l_{bi}$) for the selected phonemes prone to mispronunciation and each INA assessor *a1*,*a2* and *a3*. Table 6.15 was built using IASIMSP. As a reference, Table 6.15 shows the inter-assessor agreement coefficient (I) for the phoneme set used in this section. Phonemes with a high coefficient I coincidentally showed smaller values for MI($\eta; l_{bi}$). This was the case for /ʊ/, /ɛ/ and /w/. Phonemes /ʊ/ and /ʌ/ had the lowest agreement I of 0.20 and 0.25 respectively. The MI($\eta; l_{bi}$) for both /ʊ/ and /ʌ/ in Table 6.15 were the largest observed for each assessor.

The most biased assessors can be identified, as well as the examples most affected by

Table 6.15: MI$(\eta; l_{bi})$ for assessors *a1*,*a2* and *a3* learned by IASIMSP.

Phoneme	*a1*	*a2*	*a3*
ʊ$_0$	0.3650	0.3642	0.4185
ʊ$_1$	0.3631	0.3718	0.3546
g_0	0.1798	0.1892	0.3553
g_1	0.2560	0.2742	0.3001
ɛ$_0$	0.1144	0.1572	0.1188
ɛ$_1$	0.1507	0.1622	0.1507
ɪ$_0$	0.0358	0.0551	0.0680
ɪ$_1$	0.0692	0.0974	0.0967
w_0	0.1098	0.1134	0.1432
w_1	0.2119	0.2410	0.2495
ʌ$_0$	0.2775	0.3340	0.3409
ʌ$_1$	0.3055	0.3317	0.3189

Table 6.16: Inter-assessor agreement coefficients (I).

Phoneme	I
/ʊ/	0.20
/g/	0.68
/ɛ/	0.91
/ɪ/	0.81
/w/	0.93
/ʌ/	0.25

the bias. The MI$(\eta; l_{bi})$ illustrates how much an assessor deviates from a scoring function assumed to be assessor-independent. Consider the row for /g/ in Table 6.15. MI$(a3; l_b^{(/g/)})$ was the largest over all assessors. In a real-world scenario, *a3* could reduce the effect of their own bias through further training and consulting with their peers. As mentioned earlier in Section 2.2, if a bias-free PA is not possible, it should at least be consistent across the assessor's sample.

6.7.6 Summary

The reduction of CS$(\mathbf{L_A}, \mathbf{L_b})$ for ASIM3N observed in Section 6.4.3 did not guarantee a smaller dependence of $\mathbf{L_A}$ on $\mathbf{L_b}$. Therefore, MI$(\mathbf{L_A}; \mathbf{L_b})$ in the IASIM architecture was minimized using the CLUB algorithm. A variational upper bound MI$_{C\theta}(\mathbf{L_A}; \mathbf{L_b})$ was approximated using a variational distribution $q_\theta(\mathbf{L_b}|\mathbf{L_A})$ assumed Gaussian. The variational parameters μ_θ and σ_θ^2 were learned with FFNs. CLUB did increase KL$(\mathbf{L_b}||\mathbf{L_A})$ as long as $q_\theta(\mathbf{L_b}|\mathbf{L_A}) \approx p(\mathbf{L_b}|\mathbf{L_A})$. The combination of CLUB and CS$(\mathbf{L_A}, \mathbf{L_b})$ penalty managed to both reduce CS$(\mathbf{L_A}, \mathbf{L_b})$ and increase KL$(\mathbf{L_b}||\mathbf{L_A})$ while slightly improving the performance of the IASIM for detecting mispronounced segments. It was shown that CLUB increased MI$(\eta; l_{bi})$ simultaneously. The use of speaker factors increased MI$(\eta; l_{bi})$ even further. IASIMSP was useful for detecting the most biased annotators, along with the phonemes most affected by the bias.

6.8 Conclusion

Assumptions on the relationship between the assessor-independent and the bias component of the assessor model were tested in this section. The objective was to obtain a less redundant and more independent model for assessor bias. The assumed behaviour of the assessor model was enforced via the requirements of CS and MI. The penalization of $\text{CS}(\mathbf{L_A}, \mathbf{L_b})$ and the reduction of the upper variational bound $\text{MI}_{C\theta}(\mathbf{L_A}; \mathbf{L_b})$ were included along the BCE loss function.

The CS penalty in both MAXLoss and ABSLoss did make $\angle(\mathbf{L_A}, \mathbf{L_b}) \approx 90^\circ$ with a slight improvement on detecting pronunciation errors with $F1 = 0.8127$ for Train and $F1 = 0.7434$ for Test. It was also noted that the CS penalty made $\mathbf{L_A}$ on average larger than $\mathbf{L_b}$. The models trained with CS penalty changed the former interpretation of the bias logits'role as a gating function over $\mathbf{L_A}$. Phoneme classes with a low count in the data showed $\mu_{L_b} > 0$. Phoneme classes with counts closer to the average in the data and with a high coefficient I showed that $|\mu_{L_b}|$ remained constantly smaller than $|\mu_{L_A}|$. It was clear that class imbalance remained an important problem for learning the assessor model. The models were better at detecting correct pronunciations, as the lack of mispronounced examples in the data was noticeable even for phoneme classes with $I = 1.0$.

The reduction of $\text{CS}(\mathbf{L_A}, \mathbf{L_b})$ did not necessarily make $\mathbf{L_b}$ more independent of $\mathbf{L_A}$. Therefore, the reduction of $\text{MI}(\mathbf{L_A}; \mathbf{L_b})$ was tested to make the components of the assessor model more independent of each other. The CLUB algorithm was used to estimate a MI upper bound obtained using a variational distribution $q_\theta(\mathbf{L_b}|\mathbf{L_A})$ assumed Gaussian with parameters θ. Two FFNs were used to learn the mean and standard deviation of $q_\theta(\mathbf{L_b}|\mathbf{L_A})$ respectively along the training of the IASIM for detecting mispronunciations. As long as $q_\theta(\mathbf{L_b}|\mathbf{L_A})$ approximated the real $p(\mathbf{L_b}|\mathbf{L_A})$, the $\text{KL}(\mathbf{L_b}||\mathbf{L_A})$ increased. The metrics of the IASIM on detecting mispronounced segments were slightly below the baseline, yet $\text{KL}(\mathbf{L_b}||\mathbf{L_A})$ grew from 0.0186 to 0.0306 on the Train set. The combination of both CLUB and CS penalty outperformed both the baseline and the original IASIM with $F1 = 0.8152$ for Train and $F1 = 0.7435$ for Test. The CLUB algorithm increased $\text{MI}(\eta; \mathbf{L_b})$ for mispronounced phoneme classes and decreased it for correctly pronounced phoneme classes. The use of speaker metadata as part of the input for the bias FFN_b, increased both $\text{KL}(\mathbf{L_b}||\mathbf{L_A})$ and $\text{MI}(\eta; \mathbf{L_b})$ further. A low coefficient I for a given phoneme class would be reflected as a high $\text{MI}(\eta; \mathbf{L_b})$. The IASIM was useful for identifying how relevant assessor η is for the bias output of a given phoneme class. IASIM was useful for detecting the most biased annotators and phoneme classes prone to be affected by this. From this point forward, further actions could be taken for the sake of an impartial assessment or the increase of inter-annotator agreement.

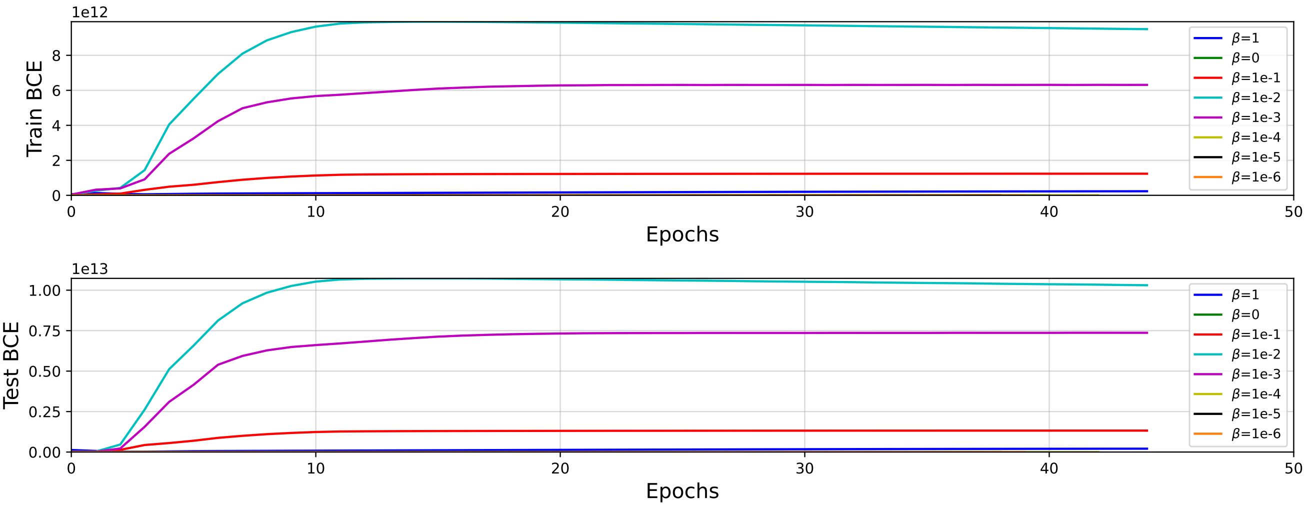


Figure 6.8: BCE curve for all β used for the IASIM. The curves for Train (top) and Test (bottom) show loss values in the order of 1E13.

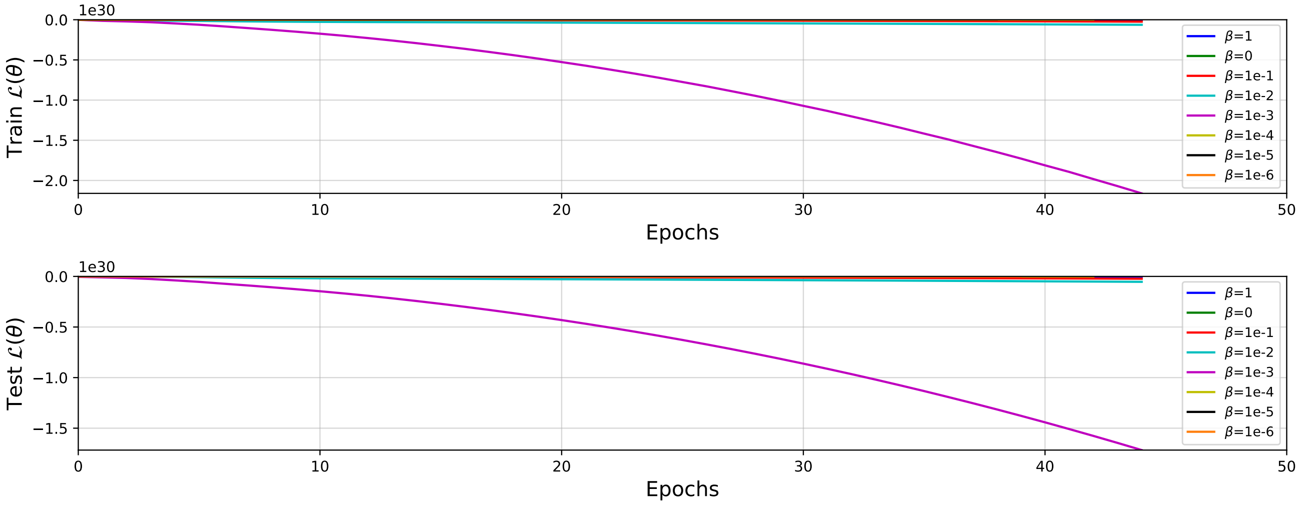


Figure 6.9: Log-likelihood curve of $q_\theta(\mathbf{L_b}|\mathbf{L_A})$ for Train (top) and Test (bottom) for all β used for the IASIM..

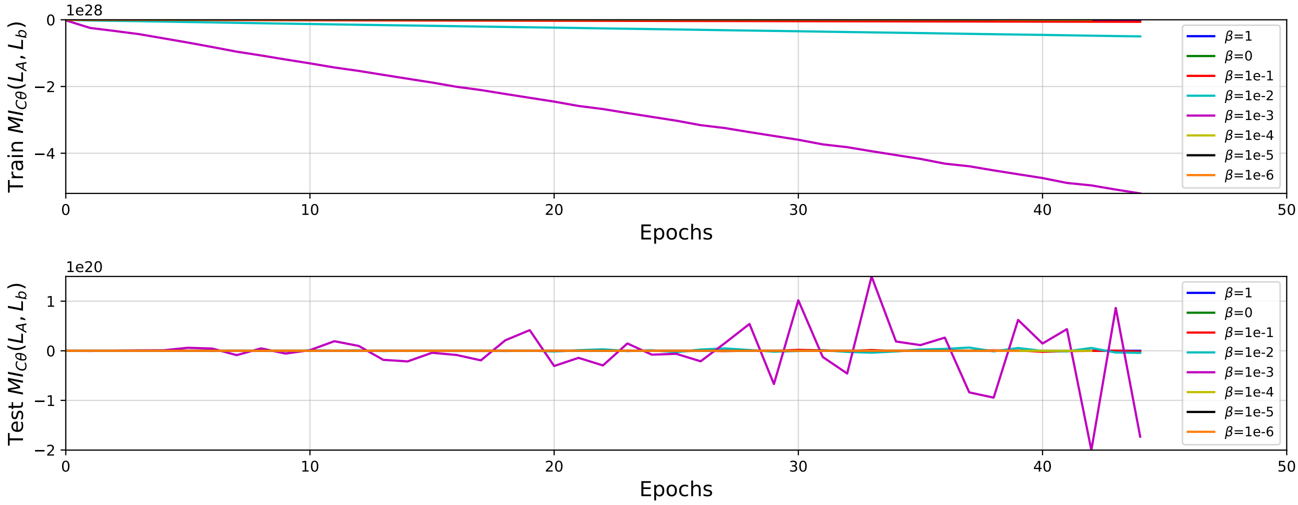


Figure 6.10: Variational MI upper bound curve for Train (top) and Test (bottom) for all β used for the IASIM.

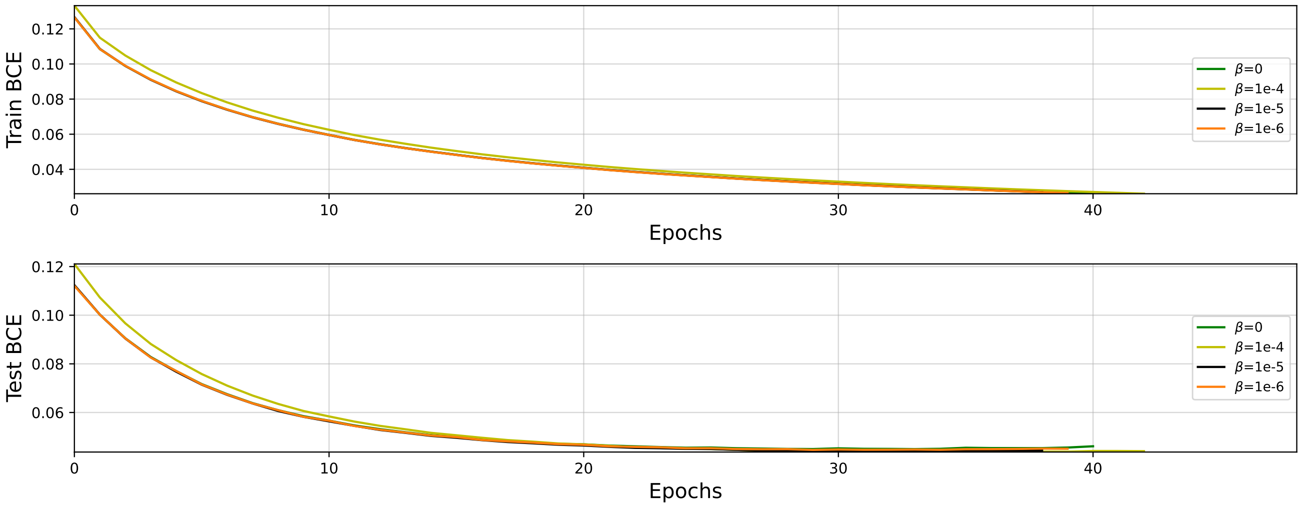


Figure 6.11: BCE curve of the IASIM for Train (top) and Test (bottom) using different β coefficients. The curves are barely affected by β.

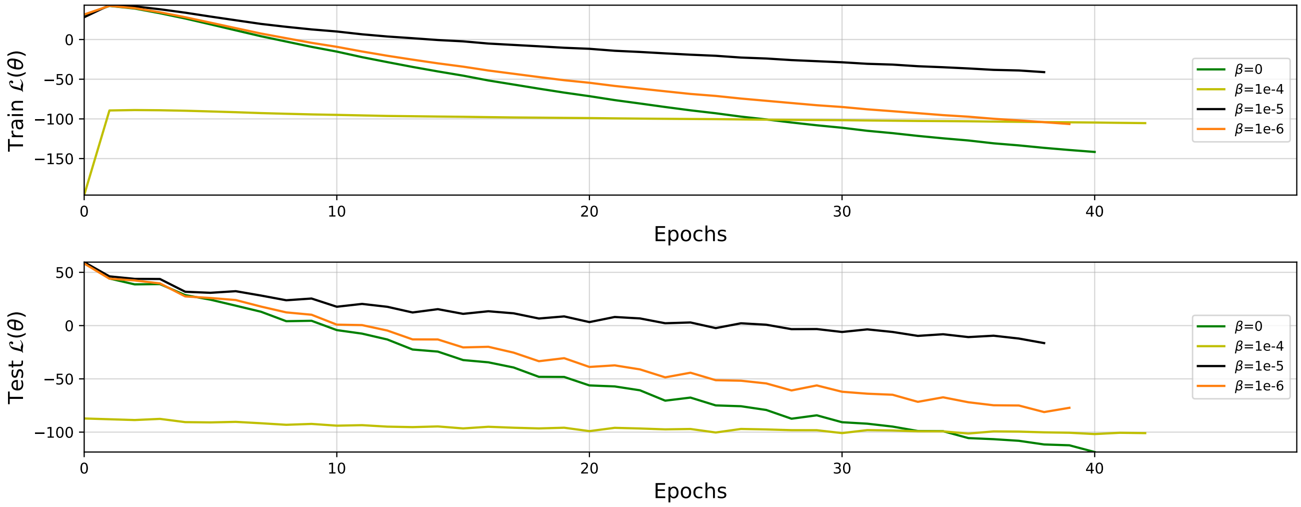


Figure 6.12: Log-likelihood curve of $q_\theta(\mathbf{L_b}|\mathbf{L_A})$ for Train (top) and Test (bottom) using different β coefficients.

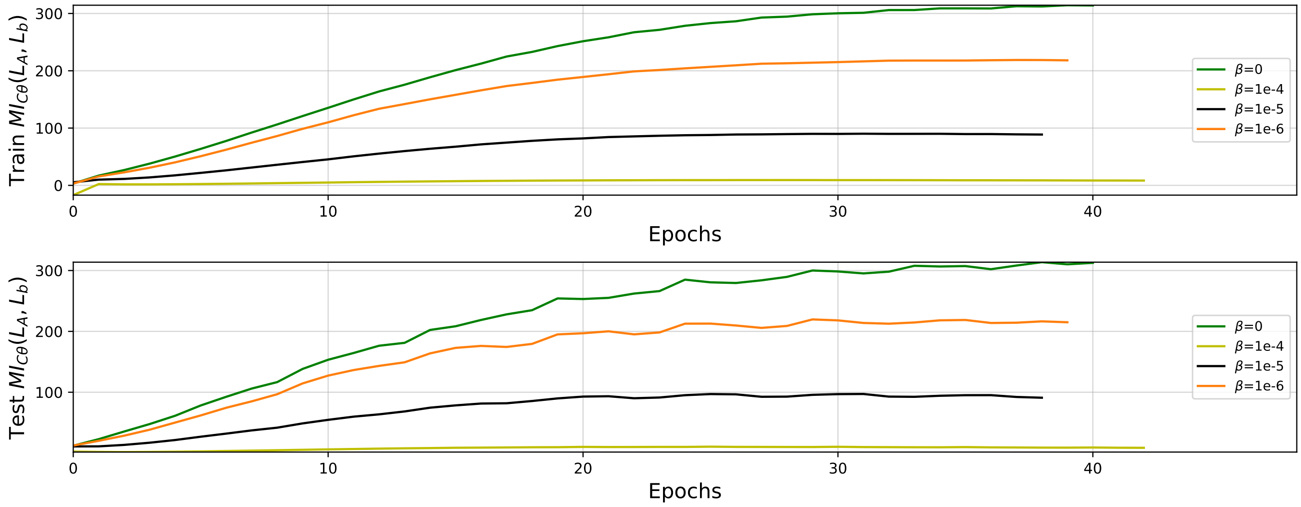


Figure 6.13: Variational MI upper bound curve of the IASIM for Train (top) and Test (bottom) using different β coefficients.

Chapter 7

Conclusion and Future Work

7.1 Thesis Summary

The goal of this thesis was to find a model for assessor bias in Pronunciation Assessment (PA). The findings resulted in four main contributions for Computer Assisted Pronunciation Assessment (CAPA) and the model of assessor bias. Background information about PA and the current take on CAPA at phoneme level was presented in Chapter 2. Chapter 3 introduces the first contribution, the segment-based approach for detecting mispronunciations. The Attention-Based Segmental Incorrectness Model (ASIM) is introduced in Chapter 3 as well. In Chapter 4, speaker metadata was tested for augmenting the performance of the ASIM. It is known that assessor bias can be affected by the perceived identity of the speaker. Therefore, speaker metadata was used as an alternative to using additional speech examples not labelled by the assessor to improve the performance of the model. The second contribution in the thesis was the confirmation of different sensitivity from the assessor to information about the linguistic background of the speakers. Chapter 5 proposes a model for the pronunciation assessor as an assessor-independent scoring function offset by a bias function specific to the individual assessor. The third contribution was the implementation of the assessor model as the Dual Attention-Based Segmental Incorrectness Model (DASIM). In this chapter, the assessor identity was used to adjust a subnetwork in charge of modelling disagreement across assessors. In Chapter 6, the similarity and co-dependence between the two functions of the assessor bias were reduced for the sake of a less redundant model. The cosine similarity (CS) and mutual information (MI) between the assessor-independent scoring function and the bias function were penalized during the training of the model. The final contribution of this work was the interpretation of MI values for detecting the annotators and phonemes most affected by the bias.

An overview of the contributions of this thesis is presented next.

7.1.1 Chapter 3: Attention-Based Method for Automatic Pronunciation Assessment

The first contribution of the thesis comes from realizing that a pronunciation assessor does not care about the precise location in time of phonemes. Instead, they focus on both the identity and the sequence of the uttered phonemes. A novel approach for CAPA was introduced in this chapter: to detect phonemes defined in a pronunciation reference directly from a speech segment. Instead of relying on phoneme alignments, the ASIM would estimate the presence of phoneme labels from an acoustic encoding. The ASIM consists of a Bidirectional Long Short-Term Memory (BDLSTM) with a self-attention module and an FFN trained to learn the corresponding phoneme labels as a multi-label classification problem. The normalized network posteriors were used to determine if all the expected phonemes in the reference were detected as correctly pronounced. The ASIM was tested on real Second Language (L2) speech from learners of English in the Netherlands. The experimental task consisted of declaring a segment mispronounced given the annotation reference. The ASIM outperformed a Goodness of Pronunciation (GOP) baseline (see Section 2.4.1) for all the assessors and consolidated references used. It was also found that the normalized attention weights would show spikes aligning with the phoneme boundaries defined in the baseline, although the ASIM uses no alignment information at all.

7.1.2 Chapter 4: Speaker Metadata for Improving Automatic Pronunciation Assessment

It is known that the perceived identity of the speaker affects the perception of their speech. It is also the case that L2 speech data annotated for mispronunciation is often scarce, hence Native Language (L1) speech is used to model the pronunciation reference. The contribution from this chapter was the use of speaker metadata to augment the performance CAPA without the need for any data which was not labelled by the assessors available. The ASIM introduced in Chapter 3 was trained on the same L2 data of learners of English with the addition of speaker metadata. Information related to the linguistic background of the speakers was encoded as one hot vector and concatenated as a constant dimension to the acoustic feature vectors. The ASIM was trained using multiple speaker factor combinations to learn each of the three assessors in the data set. The individual speaker factors did not cause an improvement in the performance of the ASIM. Specific combinations of speaker factors could improve the performance of the ASIM, particularly the ones with a more balanced class distribution. It was noted that not all assessors responded similarly to all combinations of speaker factors, confirming the effect of the speaker identity on the assessor bias.

7.1.3 Chapter 5: A Model for the Assessor Bias

The third contribution of the thesis is the assessor model introduced in this chapter. The model is based on the idea of an ideal bias-free assessment function, which is offset by a bias function specific to the assessor. The assessor-independent function can be found by learning the bias model for each assessor and then averaging across all assessors. Each of the assessment functions corresponds to an ASIM subnetwork which combines their respective output logits as an arithmetic sum. Both ASIM subnetworks observed the same acoustic features input. The bias subnetwork was made sensitive to assessor identity by concatenating it to the acoustic input as a constant dimension. The dual-ASIM was tested on L2 speech from young students of English in the Netherlands. The model was trained to learn all three assessors from the data set simultaneously and use the speaker identity to adjust the bias subnetwork. The bias subnetwork was proven sensitive to assessor identity. The assessor-independent subnetwork was used to score a MaxVote consolidated reference, yet it was found that the subnetwork was better at scoring one of the three assessors. The self-attention mechanism of the bias model could indicate the acoustic frames in which different assessors would disagree in their judgement. The normalized attention weights also showed the elements in DASIM were redundant. The dual-ASIM was re-designed along with a re-interpretation of the assessor model by making the bias dependent on the assessor-independent score and the identity of the assessor. The resulting architecture (ASIM3) consisted of a single BDLSTM with self-attention; the sequential encoding was passed to an FFN which outputs the logits for the assessor-independent scores. A second FFN receives both the assessor-independent logits and the assessor tag to estimate the bias output. ASIM3 outperformed the DASIM and reduced the number of parameters by 30%. ASIM3 also allowed the interpretation of the bias as a gating function controlling the assessor-independent scoring function.

7.1.4 Chapter 6: Methods for Encouraging Bias Specialization in ASIM

Since the goal of this thesis was to find a model for the bias, any kind of redundancy between the components of the assessor model must be kept at a minimum. The final contribution from this thesis was an exploration of methods for reducing both the correlation and dependency between the assessor-independent and bias functions. The CS and MI between the logits of the two FFNs in ASIM3 were minimized during training. The Contrastive Log-ratio Upper Bound of MI (CLUB) was used to estimate the MI between the logits. Experiments for the penalization of both CS and CLUB were carried out on ASIM3. The model was trained on to learn the annotation reference of three assessors scoring L2 speech from young learners of English in the Netherlands. The CS penalty managed to make the logits on average perpendicular with respect to each other. The use of CLUB increased the Kullback–Leibler divergence (KL) between the logits, hence reducing the dependence of the bias on the assessor-independent output. Speaker metadata was also tested as part of the input for the bias FFN. Similar to the findings in Chapter 4, speaker metadata improved the performance of ASIM3 and increased the KL between logits even further. A consequence of the increase in KL was a rise in MI between the bias output and the assessor identity. The MI between a phoneme bias logit and the assessor identity was found useful for identifying which phonemes and assessors were the most prone to be affected by the bias.

7.2 Future Work

The limitations of time and resources faced during this thesis made multiple plans and experiments unfeasible. A further take on this research can improve CAPA for L2 speech and consequently the model of the assessor bias. A list of ideas for future work is presented in this section.

7.2.1 Self-Training for L2 CAPA

The main limitation of this thesis was the reduced L2 learner data available, marked for mispronunciation. Section 2.5.1 already mentions the problems with not counting with a corpus that serves as a baseline for L2 CAPA. The corpus used in this thesis was the only one available with joint annotation with multiple assessors. However, there are more recordings in the corpus which were not annotated by the assessors. A regime of self-training (Scudder, 1965) for the ASIM could take advantage of the unlabelled recordings and improve the model's performance. A trained ASIM can be used to label new data of the same domain. The pseudo-labels for each assessor will be used to train the ASIM on additional data strongly related to the real pronunciation reference. An unsupervised method for data selection based on the contrastive loss ratios of models trained on target and training data (Park et al., 2022) is proposed for this task for the sake of keeping the assessor reference the most like the annotation available.

7.2.2 Speaker-Invariant Representations for the Assessor-Independent Pronunciation Scoring Function

The size of the assessor sample in the L2 learners corpus is often not reported and is assumed to be small. Therefore, it is likely the assessor-independent components of the ASIM learn a bias disguised as a relatively large agreement in a small sample. An alternative is to make the assessor-independent output to be speaker-invariant. Adversarial multitask learning (Shinohara, 2016) will be used for making the sequential encoding of the ASIM robust to the speaker identity. It is expected the bias FFN of the ASIM will increase the MI between the bias output, the assessor identity, and the speaker factors. Simultaneously, multitask learning can be used to make the bias outputs more specific to a speaker representation.

7.2.3 Unsupervised Representations for L2 CAPA

Another alternative to deal with the lack of L2 learning speech corpus is to take advantage of pre-trained networks for unsupervised feature extraction. WaveNet encoders (Chorowski et al., 2019) and Vector Quantized Variational encoders (Van Den Oord et al., 2017) are known for separating speaker information from the acoustic content. Both these encoders replace the BDLSTM with self-attention in the ASIM. The bias FFN could be augmented by also receiving an additional speaker representation input.

7.2.4 Search for a True Assessor-Independent Reference

The motivation for finding a model for assessor bias is to infer the corresponding bias-free assessment. The assessor-independent logits of the IASIM have only been compared to individual assessor references and consolidated annotations. It is complicated to use unbiased scores for L2 CAPA as there is not a real reference to compare it to. However, there should be a trend in the bias-free scores observed in proficient speakers with a high level of agreement across assessors. A ratio between the normalized assessor-independent outputs for correct pronunciations and the amount of bias predicted by the model could be found for cases in which all assessors agree on correct pronunciation.

Acronyms

CAPA Computer Assisted Pronunciation Assessment. v, 3–5, 18, 19, 28, 29, 33–36, 38, 39, 44–46, 48, 51, 53, 54, 59, 66, 76, 77, 79, 146, 147, 149, 150

CEFR Common European Framework of Reference for Languages: Learning, teaching, assessment. 15, 16, 43, 45

CLUB Contrastive Log-ratio Upper Bound of Mutual Information. 133–136, 138–143

CS Cosine Similarity. 118–122, 124, 138, 139, 143

DASIM Dual Attention-Based Segmental Incorrectness Model. ix, xiii, 7, 8, 83–89, 91, 92, 94–97, 101, 103, 107, 114–117, 146, 148

DNN Deep Neural Network. 19, 61, 64, 79, 83

EER Equal Error Rate. 55, 69

EM Expectation-Maximization. 21

ERN Extended Recognition Network. 33, 35

ETS Educational Testing Service. 14

FC Fully Connected. 80, 81

FFN Feed-Forward Network. x, 6, 22, 24, 35, 48, 49, 51, 54, 55, 65, 68, 83, 85, 91, 94, 114–116, 135–137

G2P Grapheme to Phoneme. 54

GMM Gaussian Mixture Model. 20, 21, 29, 34, 53

GOP Goodness of Pronunciation. viii, xii, 6, 29, 30, 33–35, 45–47, 52–61, 64, 65, 68, 76, 77, 85, 94, 147

HMM Hidden Markov Model. 19–22, 24, 29, 30, 35, 36, 53, 54, 121, 136

IASIM Mutual Information driven Attention-Based Segmental Incorrectness Model. xiv, 134–143

IASIMC Mutual Information driven Attention-Based Segmental Incorrectness Model with Cosine Similarity Penalty. xiv, 138, 139

IASIMSP Mutual Information driven Attention-Based Segmental Incorrectness Model with Speaker Factors. 140–142

IELTS International English Language Testing System. xii, 14, 16, 17

ITSL ITSLanguage. 52–54, 57, 67, 68

L1 Native Language. 2, 3, 7, 11–15, 17, 19, 28–30, 38, 40–43, 46, 48, 52, 59, 66, 67, 70, 76, 115, 140, 147

L2 Second Language. 1–4, 6–8, 12–19, 27–30, 33–36, 38–46, 48, 52, 53, 56, 57, 60, 64–68, 76, 77, 79, 81, 84, 85, 88, 89, 92, 96, 114–117, 119–121, 132, 135, 136, 147–150

LD Listener Dependant. 80, 81

LSTM Long Short-Term Memory. 25–27, 48, 49

MF Most Frequent Label. 55

MI Mutual Information. 137

ML Machine Learning. 3, 4, 19, 130, 132

MOS Mean Opinion Score. 80, 81

PA Pronunciation Assessment. 1–8, 12–16, 18, 28, 29, 38, 39, 45, 48, 56, 66, 67, 77, 81, 82, 92, 115–120, 132, 142, 146

PDF Probability Density Function. 19, 20

PLP Perceptual Linear Prediction. 85, 121

RELU Rectifier Linear Unit. 23, 25, 102, 107, 108

RNN Recursive Neural Network. 22, 24–26, 35–37

STR Stratified. 55

TOEFL Test of English as a Foreign Language. 14

Bibliography

G. Albaum. The Likert Scale Revisited. *Market Research Society. Journal.*, 39(2):1–21, mar 1997.

V. Arora, A. Lahiri, and H. Reetz. Phonological feature-based speech recognition system for pronunciation training in non-native language learning. *The Journal of the Acoustical Society of America*, 143(1):98–108, Jan 2018.

J.L. Ba, J.R. Kiros, and G.E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

D. Bahdanau, K.H. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, 2015.

B.A. Baker. Individual differences in rater decision-making style: An exploratory mixed-methods study. *Language Assessment Quarterly*, 9(3):225–248, 2012.

C. Baur, J. Gerlach, E. Rayner, M. Russell, and H. Strik. A shared task for spoken call? 2016.

C.M. Bishop and N.M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

P. Bonaventura, P. Howarth, and W. Menzel. 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, 2000.

A. Brown. Minimal pairs: minimal importance? *ELT Journal*, 49(2):169–175, apr 1995.

M.D. Carey, R.H. Mannell, and P.K. Dunn. Does a rater's familiarity with a candidate's pronunciation affect the rating in oral proficiency interviews? *Language Testing*, 28(2):201–219, 2011.

L. Chambers and K. Ingham. The bulats online speaking test. *Research Notes*, 43(1):21–25, 2011.

C.A. Chapelle and Y.R. Chung. The promise of nlp and speech processing technologies in language assessment. *Language Testing*, 27(3):301–315, 2010.

C.A. Chapelle and E. Voss. 20 years of technology and language assessment in language learning & technology. 2016.

L. Chen, J. Tao, S. Ghaffarzadegan, and Y. Qian. 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, 2018.

L. Chen, Q. Gao, Q. Liang, J. Yuan, Y. Liu, and L.I.S. China. Automatic scoring minimal-pair pronunciation drills by using recognition likelihood scores and phonological features. In *SLaTE*, pages 25–29, 2019.

N.F. Chen and H. Li. Computer-assisted pronunciation training: From pronunciation scoring towards spoken language learning. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–7. IEEE, 2016.

X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*, pages 1779–1788. PMLR, 2020.

J. Chorowski, R. Weiss, S. Bengio, and A. van den Oord. Unsupervised speech representation learning using wavenet autoencoders. *IEEE Transactions on Audio, Speech, and Language Processing*, 2019.

J.K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. *Advances in neural information processing systems*, 28, 2015.

W. Chu, Y. Liu, and J. Zhou. Recognize Mispronunciations to Improve Non-Native Acoustic Modeling Through a Phone Decoder Built from One Edit Distance Finite State Automaton. In *Proc. Interspeech 2020*, pages 3062–3066, 2020.

Council of Europe. Council for Cultural Co-operation. Education Committee. Modern Languages Division (Strasbourg). *Common European framework of reference for languages : learning, teaching, assessment*. Press Syndicate of the University of Cambridge, 2001.

C. Cucchiarini, F.d. Wet, H. Strik, and L. Boves. Assessment of dutch pronunciation by means of automatic speech recognition technology. 1998.

C. Cucchiarini, H.V. Hamme, O.v. Herwijnen, and F. Smits. Jasmin-cgn: Extension of the spoken dutch corpus with speech of elderly people, children and non-natives in the human-machine interaction modality. 2006.

N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4): 788–798, 2010.

M. Dodigovic. Speech processing technology in second language testing. In *Proceedings of the Conference on Language & Technology*, pages 113–120, 2009.

J.v. Doremalen, C. Cucchiarini, and H. Strik. Automatic pronunciation error detection in non-native speech: The case of vowel errors in dutch. *The Journal of the Acoustical Society of America*, 134(2):1336–1347, 2013.

R. Duan and N.F. Chen. Unsupervised feature adaptation using adversarial multi-task training for automatic evaluation of children's speech. In *INTERSPEECH*, pages 3037–3041, 2020.

S. Dudy, S. Bedrick, M. Asgari, and A. Kain. Automatic analysis of pronunciations for children with speech sound disorders. *Computer Speech and Language*, 50:62–84, 2018.

D.M. Eberhard, G.F. Simons, and C.D.e. Fennig. Ethnologue: Languages of the World., 2022. URL `http://www.ethnologue.com`.

A. Elizaincín. Situación actual de las Academias de la lengua en el mundo hispánico. *Revista de la Academia Nacional de Letras*, (12):111–117, 2016.

M. Eskenazi. Detection of foreign speakers' pronunciation errors for second language training-preliminary results. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, volume 3, pages 1465–1468 vol.3, 1996.

M. Eskenazi. An overview of spoken language technology for education. *Speech Communication*, 51(10):832–844, 2009.

ETS. TOEFL Speaking Rubrics, 2014. URL `https://www.ets.org/s/toefl/pdf/toefl{_}speaking{_}rubrics.pdf`.

Y. Feng, G. Fu, Q. Chen, and K. Chen. SED-MDD: Towards Sentence Dependent End-To-End Mispronunciation Detection and Diagnosis. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2020-May:3492–3496, 2020.

E. Finegan. *Language: Its structure and use*. Cengage Learning, 2014.

J.E. Flege. Second language speech learning: Theory, findings, and problems. *Speech perception and linguistic experience: Issues in cross-language research*, 92:233–277, 1995.

M. Formentelli. *Taking stance in English as a lingua franca : managing interpersonal relations in academic lectures.*

G.D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

H. Franco, H. Bratt, R. Rossier, V. Rao Gadde, E. Shriberg, V. Abrash, and K. Precoda. Eduspeak®: A speech recognition and pronunciation scoring toolkit for computer-aided language learning applications. *Language Testing*, 27(3):401–418, 2010.

K. Fu, J. Lin, D. Ke, Y. Xie, J. Zhang, and B. Lin. A full text-dependent end to end mispronunciation detection and diagnosis with easy data augmentation techniques. *arXiv e-prints*, pages arXiv–2104, 2021.

E.D. Galaczi, B. Post, A. Li, and C. Graham. Measuring L2 English Phonological Proficiency: Implications for Language Assessment. In *Proceedings of the 44th Annual Meeting of the British Association for Applied Linguistics, The impact of Applied Linguistics*, pages 67–72, London, 2011.

H.J. Giegerich. *English phonology : an introduction*. Cambridge University Press, 1992.

E.M. Golonka, A.R. Bowles, V.M. Frank, D.L. Richardson, and S. Freynik. Technologies for foreign language learning: a review of technology types and their effectiveness. *Computer Assisted Language Learning*, 27(1):70–105, 2014.

I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

A. Graves and J. Schmidhuber. 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, 2005.

A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

L. Harding. What Do Raters Need in a Pronunciation Scale? The User's View. In T. Isaacs and P. Trofimovich, editors, *Second Language Pronunciation Assessment: Interdisciplinary Perspectives*, chapter 2, pages 12–34. Multilingual Matters / Channel View Publications, 2017.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:770–778, 2016.

H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29 (6):82–97, 2012.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

W. Holmes. *Speech synthesis and recognition*. CRC press, 2001.

G. Huang, J. Ye, Y. Shen, and Y. Zhou. A evaluating model of english pronunciation for Chinese students. In *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, pages 1062–1065. May 2017a.

G. Huang, J. Ye, Z. Sun, Y. Zhou, Y. Shen, and R. Mo. English mispronunciation detection based on improved gop methods for chinese students. In *2017 International Conference on Progress in Informatics and Computing (PIC)*, pages 425–429, 2017b.

W.C. Huang, E. Cooper, J. Yamagishi, and T. Toda. Ldnet: Unified listener dependent modeling in mos prediction for synthetic speech. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 896–900, 2022.

IELTS. SPEAKING: Band Descriptors (public version) , 2019. URL `https://www.ielts.org/-/media/pdfs/speaking-band-descriptors.ashx`.

IELTS Australia. How to improve your pronunciation for your IELTS – Part 2, 2019. URL `https://www.ieltsessentials.com/global/blog/2018/01/08/how-to-improve-your-pronunciation-for-your-ielts-part-2`.

C. Inoue, N. Khabbazbashi, D. Lam, and F. Nakatsuhara. Towards new avenues for the ielts speaking test: insights from examiners' voices. *IELTS Research Reports Online Series*, 2:1–70, 2021.

International Phonetic Association. *Handbook of the International Phonetic Association : a guide to the use of the International Phonetic Alphabet.* Cambridge University Press, 1999.

T. Isaacs and L. Harding. Pronunciation assessment. *Language Teaching*, 50(3):347–366, 2017.

T. Isaacs, P. Trofimovich, G. Yu, B.M. Chereau, et al. Examining the linguistic aspects of speech that most efficiently discriminate between upper levels of the revised ielts pronunciation scale. *IELTS research reports online series*, page 48, 2015.

E. Jamison and I. Gurevych. Noise or additional information? leveraging crowdsource annotation item agreement for natural language tasks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 291–297, 2015.

D. Jones. *The phoneme: Its nature and use.* Cambridge Univ. Press, 1976.

O.m. Kang and A. Ginther. *Assessment in second language pronunciation.*

S. Kanters, C. Cucchiarini, and H. Strik. The goodness of pronunciation algorithm: a detailed performance study. 2009.

N. Kartushina and U.H. Frauenfelder. On the effects of l2 perception and of individual differences in l1 production on l2 pronunciation. *Frontiers in Psychology*, 5, 2014.

R.D. Kent, G. Weismer, J.F. Kent, and J.C. Rosenbek. Toward phonetic intelligibility testing in dysarthria. *Journal of Speech and Hearing Disorders*, 54(4):482–499, 1989.

M.N.A. Khan and D.R. Heisterkamp. Adapting instance weights for unsupervised domain adaptation using quadratic mutual information and subspace learning. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 1560–1565. IEEE, 2016.

Y. Kim, H. Franco, and L. Neumeyer. Automatic pronunciation scoring of specific phone segments for language instruction. In *Fifth European Conference on Speech Communication and Technology*, 1997.

D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

W. Klein. *Second Language Acquisition*. 01 1986.

J. Kominek and A.W. Black. The cmu arctic speech databases. In *Fifth ISCA workshop on speech synthesis*, 2004.

F.H. Kortland. *Modelling the Phoneme : New Trends in East European Phonemic Theory.* Walter de Gruyter GmbH, 2017.

S. Kotz and D. Drouet. *Correlation and dependence*. World Scientific, 2001.

F. Kuiken and I. Vedder. Raters' decisions, rating procedures and rating scales, 2014.

K. Kyriakopoulos, K. Knill, and M. Gales. Automatic detection of accent and lexical pronunciation errors in spontaneous non-native english speech. ISCA, 2020.

V. Laborde, T. Pellegrini, L. Fontan, J. Mauclair, H. Sahraoui, and J. Farinas. Pronunciation assessment of japanese learners of french with gop scores and phonetic information. In *Annual conference Interspeech (INTERSPEECH 2016)*, pages 2686–2690, 2016.

Y. Leng, X. Tan, S. Zhao, F. Soong, X.Y. Li, and T. Qin. Mbnet: Mos prediction for synthesized speech with mean-bias network. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 391–395. IEEE, 2021.

J. Levis. Assessing speech intelligibility: Experts listen to two students. In *J. Levis & K. LeVelle (Eds.). Proceedings of the 2nd Pronunciation in Second Language Learning and Teaching Conference*, pages 56–69, Ames, IA, 2010.

S.Y. Li, S.J. Huang, and S. Chen. Crowdsourcing aggregation with deep bayesian learning. *Science China Information Sciences*, 64(3):1–11, 2021.

B. Lin and L. Wang. Attention-based multi-encoder automatic pronunciation assessment. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2021-June:7743–7747, 2021.

B. Lin, L. Wang, X. Feng, and J. Zhang. Automatic scoring at multi-granularity for L2 pronunciation. 2020.

S. Lindemann. Variation or 'Error'? Perception of Pronunciation Variation and Implications for Assessment. In T. Isaacs and P. Trofimovich, editors, *Second Language Pronunciation Assessment: Interdisciplinary Perspectives*, volume 107, chapter 11, pages 193–209. Multilingual Matters / Channel View Publications, 2017.

D. Litman, H. Strik, and G.S. Lim. Speech technologies and the assessment of second language speaking: Approaches, challenges, and opportunities. *Language Assessment Quarterly*, pages 1–16, 2018.

C.C. Lo, S.W. Fu, W.C. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H.M. Wang. Mosnet: Deep learning based objective assessment for voice conversion. *arXiv preprint arXiv:1904.08352*, 2019.

T.H. Lo, S.Y. Weng, H.J. Chang, and B. Chen. An effective end-to-end modeling approach for mispronunciation detection. *arXiv preprint arXiv:2005.08440*, 2020.

W.K. Lo, S. Zhang, and H. Meng. Automatic derivation of phonological rules for mispronunciation detection in a computer-assisted pronunciation training system. In *Eleventh annual conference of the international speech communication association*, 2010.

A. Loukina, M. Lopez, K. Evanini, D. Suendermann-Oeft, and K. Zechner. Expert and crowdsourced annotation of pronunciation errors for automatic scoring systems. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

B. Luo. Evaluating a computer-assisted pronunciation training (capt) technique for efficient classroom instruction. *Computer Assisted Language Learning*, 29(3):451–476, 2016.

T. McArthur and J. Lam-McArthur. *Oxford companion to the English language*. Oxford University Press, 2018.

M.L. McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282, 2012.

D. McNeish. On using bayesian methods to address small sample problems. *Structural Equation Modeling: A Multidisciplinary Journal*, 23(5):750–773, 2016.

Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

R. Milner, M.A. Jalal, R.W. Ng, and T. Hain. A cross-corpus study on speech emotion recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 304–311. IEEE, 2019.

J. Milroy. Language ideologies and the consequences of standardization. *Journal of Sociolinguistics*, 5(4):530–555, nov 2001.

R. Moore and L. Skidmore. On the use/misuse of the term'phoneme'. In *Proceedings, Interspeech 2019*, pages 2340–2344. International Speech Communication Association (ISCA), 2019.

L. Muflikhah and B. Baharudin. Document clustering using concept space and cosine similarity measurement. In *2009 International Conference on Computer Technology and Development*, volume 1, pages 58–62, 2009.

M.J. Munro. How Well Can We Predict Second Language Learners' Pronunciation Difficulties? *CATESOL Journal*, 30(1):167–281, 2018.

J. Myers. Which languages are most widely spoken? | World Economic Forum, 2015. URL `https://www.weforum.org/agenda/2015/10/which-languages-are-most-widely-spoken/`.

A. Neri, C. Cucchiarini, H. Strik, and L. Boves. The pedagogy-technology interface in computer assisted pronunciation training. *Computer assisted language learning*, 15(5):441–467, 2002.

H.V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *Asian conference on computer vision*, pages 709–720. Springer, 2010.

M. Nicolao, A.V. Beeston, and T. Hain. Automatic assessment of English learner pronunciation using discriminative classifiers. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5351–5355. apr 2015.

Z. Niu, G. Zhong, and H. Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.

J.R. Novak, N. Minematsu, and K. Hirose. Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the wfst framework. *Natural Language Engineering*, 22(6):907–938, 2016.

G.J. Ockey and R. French. From One to Multiple Accents on a Test of L2 Listening Comprehension. *Applied Linguistics*, 37(5):693–715, oct 2016.

N. Oostdijk. The spoken dutch corpus project. *The ELRA newsletter*, 5(2):4–8, 2000.

M.G. O'Brien, T.M. Derwing, C. Cucchiarini, D.M. Hardison, H. Mixdorff, R.I. Thomson, H. Strik, J.M. Levis, M.J. Munro, J.A. Foote, et al. Directions for the future of technology in pronunciation research and teaching. *Journal of Second Language Pronunciation*, 4(2):182–207, 2018.

J. Paradis. Grammatical morphology in children learning english as a second language. 2005.

C. Park, R. Ahmad, and T. Hain. Unsupervised data selection for speech recognition with contrastive loss ratios. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8587–8591, 2022.

T. Pellegrini and L. Cances. Cosine-similarity penalty to discriminate sound classes in weakly-supervised sound event detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

B. Poole, S. Ozair, A. van den Oord, A.A. Alemi, and G. Tucker. On variational lower bounds of mutual information. In *NeurIPS Workshop on Bayesian Deep Learning*, 2018.

M. Qian, X. Wei, P. Jancovic, and M.J. Russell. The university of birmingham 2017 slate call shared task systems. In *SLaTE*, pages 91–96, 2017.

R. Ragonesi, R. Volpi, J. Cavazza, and V. Murino. Learning unbiased representations via mutual information backpropagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2729–2738, June 2021.

A.K. Ramakrishna, R. Gupta, and S. Narayanan. Joint multi-dimensional model for global and time-series annotations. *IEEE Transactions on Affective Computing*, 2020.

E. Rayner, P. Bouillon, N. Tsourakis, J. Gerlach, Y. Nakao, and C. Baur. A multilingual call game based on speech translation. In *Proceedings of LREC*, 2010.

L. Rice. Hardware and software for speech synthesis. *Dr. Dobbs J. of Computer Calisthenics & Orthodontia*, 1(4), 1976.

K. Richmond, R.A. Clark, and S. Fitt. Robust lts rules with the combilex speech technology lexicon. 2009.

T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals. Wsjcamo: a british english speech corpus for large vocabulary continuous speech recognition. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 81–84. IEEE, 1995.

D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, oct 1986a.

D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986b.

Y. Saito, S. Takamichi, and H. Saruwatari. Perceptual-similarity-aware deep speaker representation learning for multi-speaker generative modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1033–1048, 2021.

H. Sak, A. Senior, K. Rao, and F. Beaufays. Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*, 2015.

E.H. Sanchez, M. Serrurier, and M. Ortner. Learning disentangled representations via mutual information estimation. In *European Conference on Computer Vision*, pages 205–221. Springer, 2020.

H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

P. Seedhouse and M. Satar. Which specific features of candidate talk do examiners orient to when taking scoring decisions? 2021.

M. Senoussaoui, P. Kenny, P. Dumouchel, and N. Dehak. New cosine similarity scorings to implement gender-independent speaker verification. In *Interspeech*, pages 2773–2777, 2013.

V. Shashidhar, N. Pandey, and V. Aggarwal. Automatic spontaneous speech grading: A novel feature derivation technique using the crowd. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1085–1094, 2015.

J. Shi, N. Huo, and Q. Jin. Context-aware goodness of pronunciation for computer-assisted pronunciation training. *arXiv preprint arXiv:2008.08647*, 2020.

Y. Shinohara. Adversarial multi-task learning of deep neural networks for robust speech recognition. In *Interspeech*, pages 2369–2372. San Francisco, CA, USA, 2016.

Y. Song, W. Liang, and R. Liu. Lattice-based GOP in automatic pronunciation evaluation. *2010 The 2nd International Conference on Computer and Automation Engineering, ICCAE 2010*, 3(4): 598–602, 2010.

Z. Soproni. Common european framework of reference for languages: Learning, teaching, assessment, companion volume with new descriptors. *Modern Nyelvoktatás*, 26(1-2):168–170, 2020.

R.C. Streijl, S. Winkler, and D.S. Hands. Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22:213–227, 2014.

S. Sudhakara, M.K. Ramanathi, C. Yarra, A. Das, and P.K. Ghosh. Noise robust goodness of pronunciation measures using teacher's utterance. In *SLaTE*, pages 69–73, 2019a.

S. Sudhakara, M.K. Ramanathi, C. Yarra, and P.K. Ghosh. An improved goodness of pronunciation (gop) measure for pronunciation evaluation with dnn-hmm system considering hmm transition probabilities. In *INTERSPEECH*, pages 954–958, 2019b.

Y. Suzukida and K. Saito. What is second language pronunciation proficiency? an empirical study. *System*, 106:102754, 2022.

D. Sztahó, G. Szaszák, and A. Beke. Deep learning methods in speaker recognition: a review. *arXiv preprint arXiv:1911.06615*, 2019.

K. Takai, P. Heracleous, K. Yasuda, and A. Yoneyama. Deep learning-based automatic pronunciation assessment for second language learners. In *International Conference on Human-Computer Interaction*, pages 338–342. Springer, 2020.

The British Council. IELTS Speaking 4 - Pronunciation.pdf. `https://takeielts.britishcouncil.org/sites/default/files/IELTS%20Speaking%204%20-%20Pronunciation.pdf`, Jan 2018.

G.A. Tops, X. Dekeyser, B. Devriendt, and S. Geukens. Dutch speakers. *Learner English: A teacher's guide to interference and other problems*, pages 1–20, 2001.

E. Trentin and M. Gori. A survey of hybrid ann/hmm models for automatic speech recognition. *Neurocomputing*, 37(1):91–126, 2001.

P. Trofimovich and T. Isaacs. Second language pronunciation assessment: A look at the present and the future. *Second Language Pronunciation Assessment*, page 259, 2017.

K. Truong, A. Neri, C. Cucchiarini, and H. Strik. Automatic pronunciation error detection: an acoustic-phonetic approach. 2004.

R.C. Van Dalen, K.M. Knill, P. Tsiakoulis, and M.J. Gales. Improving multiple-crowd-sourced transcriptions using a speech recogniser. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4709–4713. IEEE, 2015.

A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

G. Van Houdt, C. Mosquera, and G. Nápoles. A review on the long short-term memory model. *Artificial Intelligence Review*, 53, 12 2020.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-December, pages 5999–6009. jun 2017.

J.R. Vergara and P.A. Estévez. A review of feature selection methods based on mutual information. *Neural computing and applications*, 24(1):175–186, 2014.

P. Verma and P.K. Das. i-vectors in speech processing applications: a survey. *International Journal of Speech Technology*, 18(4):529–546, 2015.

K. Veselỳ, L. Burget, and F. Grézl. Parallel training of neural networks for speech recognition. In *TSD*, pages 439–446. Springer, 2010.

H. Wang, X. Qian, and H. Meng. Predicting gradation of l2 english mispronunciations using crowdsourced ratings and phonological rules. In *Speech and Language Technology in Education*. Citeseer, 2013.

Y. Wang, M. Gales, K.M. Knill, K. Kyriakopoulos, A. Malinin, R.C. van Dalen, and M. Rashid. Towards automatic assessment of spontaneous spoken english. *Speech Communication*, 104: 47–56, 2018a.

Z. Wang, J. Zhang, and Y. Xie. L2 mispronunciation verification based on acoustic phone embedding and siamese networks. In *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 444–448, 2018b.

J. Wei and L. Llosa. Investigating Differences Between American and Indian Raters in Assessing TOEFL iBT Speaking Tasks. *Language Assessment Quarterly*, 12(3):283–304, jul 2015.

M. Weinreich. *The YIVO faces the post-war world*. Yiddish Scientific Institute-Tivo, 1945.

P. Winke, S. Gass, and C. Myford. Raters' L2 background as a potential source of bias in rating oral performance:. *http://dx.doi.org/10.1177/0265532212456968*, 30(2):231–252, nov 2012.

S. Witt and S. Young. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech Communication*, 30(2-3):95–108, Feb 2000.

M.J. Witteman, A. Weber, and J.M. McQueen. Tolerance for inconsistency in foreign-accented speech. *Psychonomic Bulletin & Review*, 21(2):512–519, apr 2014.

C.H. Wu, H.Y. Su, . Chao, and H. Liu. Computer Assisted Language Learning Efficient personalized mispronunciation detection of Taiwanese-accented English speech based on unsupervised model adaptation and dynamic sentence selection. 2012.

W. Xue, R. van Hout, C. Cucchiarini, and H. Strik. Assessing speech intelligibility of pathological speech: test types, ratings and transcription measures. *Clinical Linguistics & Phonetics*, pages 1–25, 2021.

X. Yang. *Machine Learning Approaches to Improving Mispronunciation Detection on an Imbalanced Corpus*. PhD thesis, University of Illinois, dec 2015.

L. Yates, B. Zielinski, E. Pryor, et al. The assessment of pronunciation and the new ielts pronunciation scale. *IELTS Research Reports Volume 12, 2011*, page 1, 2011.

S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, et al. The htk book. *Cambridge university engineering department*, 3(175):12, 2002.

D. Yu and L. Deng. *Automatic speech recognition*, volume 1. Springer, 2016.

J. Zhang, Z. Zhang, Y. Wang, Z. Yan, Q. Song, Y. Huang, K. Li, D. Povey, and Y. Wang. speechocean762: An open-source non-native english speech corpus for pronunciation assessment. *arXiv preprint arXiv:2104.01378*, 2021.

L. Zhang, Z. Zhao, C. Ma, L. Shan, H. Sun, L. Jiang, S. Deng, and C. Gao. End-to-end automatic pronunciation error detection based on improved hybrid ctc/attention architecture. *Sensors*, 20(7):1809, 2020.

G. Zhao, S. Sonsaat, A. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna. L2-ARCTIC: A Non-Native English Speech Corpus. *Perception Sensing Instrumentation Lab*, jan 2018.

N. Zheng, L. Deng, W. Huang, Y.T. Yeung, B. Xu, Y. Guo, Y. Wang, X. Jiang, and Q. Liu. Cca-mdd: A coupled cross-attention based framework for streaming mispronunciation detection and diagnosis. *arXiv preprint arXiv:2111.08191*, 2021.

S. Zhu, X. Zhang, and D. Evans. Learning adversarially robust representations via worst-case mutual information maximization. In *International Conference on Machine Learning*, pages 11609–11618. PMLR, 2020.