Phonological Feature-based Speech Recognition System for Pronunciation Training in Non-native Language Learning

Vipul Arora,[1] Aditi Lahiri,[1, a)] and Henning Reetz[2]

[1]*Faculty of Linguistics, Philology and Phonetics, University of Oxford, U.K.*

[2]*Goethe University, Frankfurt am Main, Germany*

(Dated: 27 October 2017)

We address the question whether phonological features can be used effectively in an automatic speech recognition (ASR) system for pronunciation training in non-native language (L2) learning. Computer-aided pronunciation training (CAPT) consists of two essential tasks - detecting mispronunciations and providing corrective feedback, usually either on the basis of full words or phonemes. Phonemes, however, can be further disassembled into phonological features, which in turn define groups of phonemes. A phonological feature-based ASR system allows us to perform a sub-phonemic analysis at feature level, providing a more effective feedback to reach the acoustic goal and perceptual constancy. Furthermore, phonological features provide a structured way for analysing the types of errors a learner makes, and can readily convey which pronunciations need improvement. This paper presents our implementation of such an ASR system using deep neural networks as acoustic model, and its use for detecting mispronunciations, analysing errors and rendering corrective feedback. Quantitative as well as qualitative evaluations are carried out for German and Italian learners of English. In addition to achieving high accuracy of mispronunciation detection, our system also provides accurate diagnosis of errors.

Keywords: Phonological features; mispronunciation detection; automatic speech recognition

[a)]aditi.lahiri@ling-phil.ox.ac.uk; Corresponding author.

I. INTRODUCTION

Learning a new language (L2) is common in the modern era of globalisation. Adults often experience difficulties in learning and even perceiving new sounds that are not present in their native language (L1). On the other hand, automatic speech recognition (ASR) technology has made tremendous progress in recent times, becoming a useful tool in assisting the L2 learners, commonly known as computer aided language learning (CALL). An essential component of CALL systems is computer-aided pronunciation training (CAPT), where the system can detect mispronunciations in the learner's utterances, and can also provide corrective feedback to the learner. These systems are all based on whole phonemes. In contrast, this work highlights the utility of phonological features (which make up individual phonemes) in CALL applications. We propose a CAPT system using features not only to detect and analyse mispronunciations in learners utterances, but also to render corrective feedback through which they can efficiently improve their articulation to reach acoustic targets. Further, phonological features can also be used to find patterns of mispronunciations of a particular speaker, that can be useful for designing his/her course based on the types of mistakes that occur. The proposed system uses an automatic speech recognition system

that consists of deep neural networks (DNNs) in the acoustic front-end and a hidden Markov model (HMM). The DNNs learn to estimate phonological features from the speech signal. These features are then mapped to phonemes for the task of speech recognition and mispronunciation detection. The estimated phonological features are then used to construct a corrective feedback for the phonemes or groups of phonemes that are mispronounced.

The main characteristics of this work are:

- A DNN based acoustic model to extract phonological features from the speech signal
- An ASR system using phonological features to recognise and analyse learners speech
- A mispronunciation detector
- Analysis of mispronunciations based on phonological features
- Rendering feedback in terms of phonological features

The paper is organized as follows: Sec. II discusses the previous relevant literature. The ASR framework used for implementing the proposed system is described in Sec. III. Secs. IV and V provide details of the proposed system for detecting mispronunciations and rendering feature-based corrective feedback, respectively, along with experimental evaluation. The conclusion in Sec. VI also discusses the future directions.

II. RELATED WORKS

A. Feature based Automatic Speech Recognition

Typically, an ASR system consists of an acoustic model and a decoder. The acoustic model analyses the input speech signal delivering probability scores of different phoneme states. It normally consists of Gaussian mixture models (GMM) or (D)NNs. These scores are further used by a decoder, which comprises of dynamic models like HMMs and finite state transducers (FSTs), to estimate the final output in the form of sequences of phonemes, words or sentences ((Mohri *et al.*, 2002)).

The idea of using distinctive features for speech recognition has a long history. (Jakobson *et al.*, 1952) devised a finite number of features which had articulatory and acoustic correlates, but pointed out that the 'perceptual and aural' levels of highest relevance were lacking. Later, Stevens and colleagues (and other related researchers) pursued a research direction to extract invariant acoustic correlates from the signal ((Blumstein and Stevens, 1979; Lahiri *et al.*, 1984; Stevens and Blumstein, 1978)). Based on this research Stevens proposed a "landmark" speech recognition model ((Stevens, 2004)) which was taken up successfully by several researchers. For instance, (Juneja and Espy-Wilson, 2008) argued in

support of a probabilistic framework to speech recognition where the landmark detection module used acoustic parameters as input to extract acoustic correlates for manner-based phonetic features.

Other ASR systems have also been proposed, where the crucial idea is to extract various types of linguistic features from the speech signal, and to use those features for ASR. (Deng and Sun, 1994) proposed an ASR system based on the articulatory features developed by (Browman and Goldstein, 1992); they estimated these features from spectral parameters (MFCCs) of speech, and used HMMs to model the trajectories of overlapping features. Additional linguistic constraints and long context dependencies were modelled by (Sun and Deng, 2000). (Hasegawa-Johnson *et al.*, 2004) used a very large number of spectral parameters to estimate the manner and place of articulation for ASR. Moving away from spectral parameters, (Niyogi and Sondhi, 2002) proposed optimal filters designed to detect stop consonants from the speech waveform and extended this work to detect various phonological objects ((Amit *et al.*, 2005)), developing a complete ASR system ((Jansen and Niyogi, 2008)) using a hierarchy of distinctive features. As an alternative to HMMs, (Jansen and Niyogi, 2009) proposed point process models that use distinctive features detected discretely in continuous speech.

Although the use of linguistic features for ASR was pursued, the rise of DNNs in ASR opened up new possibilities in exploring whether the hidden layers of a DNN correspond to certain kinds of linguistic features. (Nagamine *et al.*, 2015) found that the nodes of a phoneme recogniser DNN correlate well with phonetic features. (Tan *et al.*, 2015) constrained the DNN such that the response to each phoneme is localised in a region of each hidden layer, thereby enhancing the interpretability of the DNN in terms of linguistic knowledge. (Siniscalchi *et al.*, 2013) and (Yu *et al.*, 2012) proposed an ASR system based on phonetic features, which they estimated using DNN based automatic speech attribute transcription (ASAT) framework, and further merged them for phoneme estimation. Given the success of DNNs in ASR systems, there has been a general move to make use of them in L2 learning.

B. Mispronunciation Detection

The probability scores generated by the ASR system can be used for CAPT to assess the quality of utterances of L2 learners. L2 learners can make three kinds of errors in pronunciations – insertions (usually vowels), deletions (both vowels and consonants) and

substitutions (of phonemes). The general framework for detecting mispronunciations is based on two main components:

- Acoustic model:

  It generates probability scores for each linguistic unit (features, phoneme, etc.).

- Mispronunciation detector:

  It uses the probability scores generated by the acoustic model to estimate which linguistic units are mispronounced.

Different implementations of this general framework are found in the literature. The methods for automatic detection of substitutions and deletions, which comprise the major part of mistakes, can broadly be classified into three classes: (i) posterior probability based methods, (ii) classifier based methods, and (iii) decoding network based methods. We describe each of them below.

(i) Posterior probability based methods define some measure using the probability scores of phonemes estimated by the ASR acoustic model. These probability scores are obtained by force-aligning the learner's speech with the target phoneme sequence of the sentence, which the learner intends to utter. Since the measure is mostly scalar, a simple threshold can be

defined so as to detect a mispronunciation. On these lines, a widely used measure is the goodness of pronunciation (GOP) measure ((Witt, 1999)), which is defined as the difference between the probability of the target phoneme and that of the most likely phoneme, other than the target, while decoding. Several CAPT systems have been proposed using the GOP measure with slight variations in its definition ((Hu *et al.*, 2015; Li *et al.*, 2016; van Doremalen *et al.*, 2013)). (Franco *et al.*, 1999) defined a measure using two kinds of acoustic models – one trained for correct pronunciations of phonemes and the other trained for their incorrect pronunciations. The difference in phoneme probabilities obtained from the two models, averaged over the phoneme duration, was used as the measure of mispronunciation. This measure performed better than the mean phoneme probability obtained from only one acoustic model, i.e., the one based on correct pronunciations. In order to overcome the problem of limited training data, (Franco *et al.*, 2014) proposed to adapt the two acoustic models using Bayesian adaptation, which further improved the detection accuracy.

(ii) The classifier-based systems use posterior probabilities as an input to a mispronunciation classifier specifically trained for this purpose. Each target phoneme is classified as correctly or incorrectly pronounced using the (phoneme) posterior probabilities obtained during decoding (forced alignment). (Franco *et al.*, 2000) analysed several kinds of linear

as well as non-linear classifiers. (Franco *et al.*, 2014) proposed a support vector machine classifier to classify supervectors obtained by adapting the GMM to each occurrence of a phoneme. (Hu *et al.*, 2015) proposed classifiers using NNs as well as support vector machines, and found that NN classifiers outperform other systems, including the GOP measure based system.

(iii) The decoding network based systems force-align the learner's utterance using a decoding FST with multiple paths. In addition to the path of the target phoneme sequence, the decoding FST also contains alternate paths for mispronunciations, i.e. paths with phonemes likely to be uttered in place of the target phonemes. These networks are known as extended recognition networks (ERNs) ((Meng *et al.*, 2007; Qian *et al.*, 2016)). They are either rule-based or are automatically derived from training data from non-native speakers ((Lo *et al.*, 2010)). Instead of using pre-trained patterns in mispronunciations, (Lee and Glass, 2015) proposed to construct them directly from the learner's speech. While insertions, in principle, can be inferred with methods (i) and (ii), method (iii) detects them more explicitly. ERNs are discussed further in Sec. II C.

C. Mispronunciation Correction

In addition to mispronunciation detection, a general CAPT system may also provide feedback. As far as we are aware, present day CAPT systems provide feedback at the phoneme level, i.e. which phoneme has been mistakenly uttered by the learner in place of the target phoneme, generally referred to as *mispronunciation diagnosis*.

ERNs are popular for mispronunciation diagnosis. The force alignment process matches the actually spoken phoneme, thereby achieving both tasks, viz., detection and feedback, simultaneously. A common problem with this kind of diagnosis, however, is that the mispronunciations that can be detected and diagnosed become limited by the decoding network. (Li *et al.*, 2017) proposed a way to circumvent this problem by using a free-phoneme recogniser. However, their system is limited as the phoneme uttered as mispronunciation might not belong to the L2 phoneme repertoire. For example, the German phoneme /ø/ does not exist in English and hence, cannot be recognised by an English ASR system. Training on a global phoneme set is one way to incorporate phonemes not in the L2 set ((Wang and Lee, 2015)), but it requires more training data in order to train for extra phonemes. The number

of phonemes can be quite large; for instance, the UPSID database[1] has 919 phonemes across 451 languages.

This problem of rendering corrective feedback can be tackled in a more efficient way with the help of linguistic features, since only a small set of features is required to define the phonemes of any given language. Instead of indicating which phoneme was replaced with what, we could point out which feature of that phoneme was incorrect and how it should be corrected. We elaborate more on this in Sec. III. Now the question is what kind of features should be considered and how should they be used as feedback. Many publications have reported that L2 pronunciation improves by using visual feedback for certain features. (Suemitsu *et al.*, 2015) used electromagnetic articulometry to measure and visualise the positions of articulators of the learner in real-time; a comparison of these with native target positions of articulators provided feedback to the learner. Similarly, (Kartushina *et al.*, 2015) reported effective learning when learners were provided with a 2 dimensional plot of the first two formants while learning the pronunciation of vowels.

There has been little research in using automatic feature detection to provide feedback. The only one that we are aware of is by (Li *et al.*, 2016), which unfortunately provides us with little detail and insufficient analysis of their method. In the present paper, we present

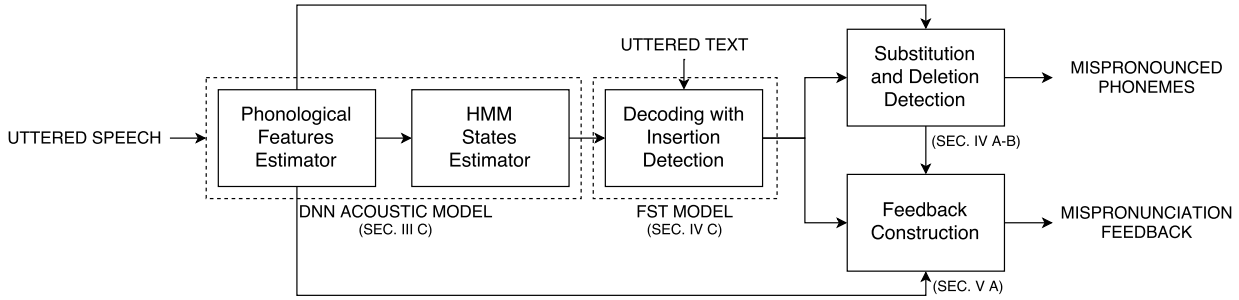

FIG. 1: Block schematic of the proposed CAPT system

our system based on phonological features for detecting mispronunciations and delivering corrective feedback. An overview of the entire proposed CAPT system is depicted in fig. 1, different parts of which are detailed in the following sections. Feedback based on phonological features provides information on how to improve the articulation, not just for a single phoneme but a group of phonemes. We define our phonological features in more detail below.

III. PROPOSED FRAMEWORK

A. Phonological Features

Why phonological features instead of phonemes? Features are, on one level, the minimal phonological unit, but in reality are common to a number of phonemes and group them into

classes. For instance, the feature VOICE is relevant for the English phonemes /b d ɡ ʤ z ʒ ð v/ since they distinguish them from their unvoiced counterparts /p t k tʃ s ʃ θ f/. Any error involving the de-voicing of, for instance, /b/ will probably extend to a de-voicing problem in all the other voiced phonemes, viz., /d g/ etc. Similarly, a mistake with aspiration for one phoneme such as /t^h/ will be carried over to all other aspirated phonemes, viz., /p^h k^h/. This implies, that instead of correcting the pronunciation of each phoneme of a group, it is much more efficient to correct one feature. The feature HIGH for vowels includes the vowels /i ɪ u ʊ/, while the feature ATR (or Advanced Tongue Root) cross-classifies these HIGH vowels into /i u/. Thus /ɪ ʊ/ are not ATR although they are HIGH. Assuming a feature system suggests that if the language learner makes an error concerning the feature HIGH and ATR he/she will most likely to make an error concerning both phonemes /i/ and /u/, and not just one of them since these are the only two phonemes which are classified by both these features. A further implication of features is that even if very similar phonemes exist in two languages, the features that distinguish them need not be identical because the presence of features are determined by the number of phonemes that exist in the language. For instance, both English and German have the LOW vowel /a/ (e.g. English: *father*, German: 'Tag' *day*). However, English has another LOW vowel with a different place of articulation /æ/

(e.g. *bag*). To distinguish these vowels, English must mark /a/ as DORSAL while in German it is redundant.

Thus, a model based on features would predict that feature errors in one word will extend to errors in the same feature in other phonemes in other words. If Germans speaking English are unable to voice /b/ in the word *cab* to distinguish it from *cap*, they will probably make the same voicing error in words like *badge, bag, love*, etc. Isolated phoneme errors cannot make this prediction or give a corrective feedback which would be an immediate effective generalisation.

The proposed ASR system uses 18 phonological features (Table I) of the FUL model (Featurally Underspecified Lexicon, ((Lahiri , 2012; Lahiri and Reetz, 2010))). Apart from these features, SIL marks silence.

B. Dataset

For the evaluation of CAPT systems, various non-native speech datasets have been collected by different research groups ((Raab *et al.*, 2007)). In our work, we have used the Interactive Spoken Language Education (ISLE) corpus ((Menzel *et al.*, 2000)), which contains noise-free utterances by learners of English, who read sentences in English. The database

TABLE I: Phonological features used in this work, with the corresponding phonemes in the ISLE dataset.

VOC	(vowel) aː æ ʌ ɔː aʊ ə aɪ ɛ ɚ eɪ ɪ iː ɔ oʊ ɔɪ ʊ uː
CONS	(consonant) b tʃ d ð f g h ʤ k p s ʃ t θ v z ʒ l m n ŋ r
CONT	(continuant fricative consonant) ð f h l s ʃ θ v z ʒ
OBSTR	(obstruant) b tʃ d ð f g ʤ k p h s ʃ t θ v z ʒ
STR	(strident) tʃ s ʃ θ z ʒ
VOICE	(voiced consonant) b d ð g ʤ v z ʒ
SON	(sonorant) aː æ ʌ ɔː aʊ ə aɪ ɛ ɚ eɪ ɪ iː l m n ŋ ɔ oʊ ɔɪ r ʊ uː w ɹ
STOP	(stop consonant) b tʃ d g ʤ k p t
LOW	(low) aː æ aʊ aɪ
HIGH	(high) tʃ ɪ iː ʤ ʃ ʊ uː w ɹ ʒ
LAB	(labial) ɔː b f m ɔ oʊ ɔɪ p ʊ uː v w
COR	(coronal) æ tʃ d ð ɛ eɪ ɪ iː ʤ l n r s ʃ t θ ɹ z ʒ
DOR	(dorsal) aː ɔː aʊ aɪ g k ŋ ɔ oʊ ɔɪ ʊ uː w
RTR	(retracted tongue root vowel) ʌ ə ɛ ɚ ɪ ʊ w
NAS	(nasal) m n ŋ
LAT	(lateral) l
RHO	(rhotic) ɚ r
RAD	(radical) h

contains data from German and Italian native speakers, 23 of each language. The data is divided into two non-overlapping sets – one for training and the other for testing. The training data consists of 19 speakers from each language, with total audio duration of 8 hours

25 minutes with 6378 utterances. The test data consists of 4 speakers from each language and a total audio duration of 1 hour 34 minutes with 1327 utterances. The speakers for the training and test data do not overlap. In total, the dataset has 13878 substitution, 1516 deletion and 3539 insertion errors.

Each audio file consists of a single sentence and is annotated at word and phoneme levels, where the phonemes are from the UK English phoneme set. Several phonemes that do not match any phoneme (due to mispronunciations), are denoted in this paper as 'unmap'. Phoneme level transcriptions comprise of two levels, viz., intended or target phonemes and actually uttered phonemes. The target phonemes correspond to the canonical transcription of the words in the sentence, while the actually uttered phonemes are manually transcribed by expert language teachers, who listened to these utterances. For example, for the word *said*, the target utterance is /s ɛ d/, and if the learner pronounced an incorrect vowel, it can be transcribed as, e.g., /s eɪ d/. The two phoneme transcriptions are time-synchronised, and hence, it is easy to label the target transcriptions with binary mispronunciation markers (i.e., correct or not). In addition, each speaker is also rated for pronunciation proficiency, on a 5-level scale.

C. Acoustic Model with Phonological Features

In this section, we present our acoustic model that is based on phonological feature based representation of phonemes ((Arora *et al.*, 2016)).

With the development of deep learning techniques and their effective use in ASR, we endeavored to use DNNs to extract phonological features from the acoustic parameters of the speech signals. We extract short-time power spectra from the speech signal, using a Hamming window of 25ms that shifts with a hop size of 10ms. Each power spectrum is then binned with 23 Mel-scaled filters, whose log scaled outputs form the acoustic parameters for that time frame. These acoustic parameters are fed as input into the DNN, after applying mean and variance normalisation and appending each time frame with a context of ± 5 time frames. The purpose of this DNN is to map the input into phonological feature probabilities.

An important step is the training of the DNNs. The data is annotated for phoneme boundaries, which do not necessarily coincide with the boundaries of phonological features corresponding to the phoneme. Several solutions to this alignment issue have been proposed. (Livescu *et al.*, 2015; Livescu and Glass, 2004) used graphical models allowing asynchronous and interdependent occurrence of features within phonemes. (King and Taylor,

2000), however, have shown that neural networks are intrinsically capable of taking care of the asynchronous behaviour of features. For our system, we have followed the latter approach.

Each phoneme is represented with a binary vector of features, with each element as 1 or 0, denoting the presence or absence, respectively, of that feature. The DNN output is a vector of phonological features, with each element representing the phonological feature estimated from the input, whose value is a real number between 0 and 1. This mapping is learned from the training data. The NN consists of three hidden layers of 500 neurons each, with the rectified linear unit as the non-linearity. The output layer has sigmoid function as the non-linearity at each neuron so as to limit the output between 0 and 1.

The estimated phonological features are further mapped onto phonemes. For this purpose, we use an HMM framework, representing each phoneme with 3 temporal states. The mapping from features to the probabilities of phoneme states is done with another NN, which consists of one hidden layer with 500 neurons, each having a rectified linear unit non-linearity, and the output layer with soft-max non-linearity.

To train both the above NNs, we align the speech waveforms in the training data with the corresponding actually uttered phonemes by force-alignment, using a GMM-HMM based

system ((Povey *et al.*, 2011)). The first DNN extracting probabilities of phonological features is trained to minimise squared-error objective function with stochastic gradient descent algorithm. The second NN for estimating HMM state probabilities is learned by minimising categorical cross-entropy objective function with stochastic gradient descent.

1. Experimental Evaluation

The proposed feature extraction system is trained over the training data and is evaluated over the testing data. The system extracts phonological features from each frame in each utterance in the testing data. To quantitatively assess its performance, we analyse its precision, recall and F-measure over the features detected at each frame using different threshold values (learned from the training data). Precision is defined as the number of true detections divided by the total number of detections, and recall is defined as the number of true detections divided by the number of actual occurrences. F-measure is the harmonic mean of precision and recall. The performance of the phonological feature extraction system is shown in Table II. Here, presence of a feature denotes the number of frames having that feature divided by the total number of frames in the test set. We can see that certain features are detected more easily than others, as indicated by their respective F values. Features

TABLE II: Evaluation of phonological feature estimation. Presence denotes the frequency of occurrence of the feature in the test set. Precision, recall and F-measure are defined in the text. All values are in %ge.

Feature	Presence	Precision	Recall	F
VOC	20.9	88.4	80.4	84.2
CONS	30.0	87.1	86.4	86.8
CONT	11.7	71.9	83.6	77.3
OBSTR	21.8	86.9	88.1	87.5
STR	7.4	84.7	87.5	86.0
VOICE	6.3	51.4	58.8	54.9
SON	31.3	92.8	91.5	92.1
STOP	11.8	79.5	79.9	79.7
LOW	2.3	56.1	44.9	49.9
HIGH	8.4	68.0	74.0	70.9
LAB	10.9	67.3	68.4	67.8
COR	29.6	87.7	77.2	82.1
DOR	12.1	75.9	67.1	71.2
RTR	8.0	60.2	54.4	57.1
NAS	4.7	76.0	71.7	73.8
LAT	1.7	48.2	38.0	42.5
RHO	3.3	56.5	44.9	50.1
RAD	0.6	38.5	60.0	46.9
SIL	47.0	97.0	97.3	97.1

like LAT, RAD, LOW, RHO have low F values showing that the system finds them difficult to extract.

IV. MISPRONUNCIATION DETECTION

In a typical CALL application, a learner is given a sentence to read, where the target sequence of phonemes is known from the dictionary. The task of the program is to detect the mistakes in pronunciations made by the learner. The mistakes can be substitutions, deletions and/or insertions of phonemes.

For detecting substitutions and deletions, we implement two methods, one of which is posterior based and uses the GOP measure, while the other is classifier based and uses an NN classifier. For detecting insertions, we modify the decoding FSTs to allow insertions by introducing filler models.

Although it has been observed that classifier based methods perform much better than posterior based methods ((Hu *et al.*, 2015), (van Doremalen *et al.*, 2013)), the advantage of using the GOP measure is that it does not require explicit training for mispronunciation detection. Instead, the posterior probabilities obtained from the ASR acoustic model can directly be used to detect mispronunciations.

A. GOP measure

Given the acoustic parameters o_t at each time frame t, the acoustic model returns probability of states at each time frame. For each phoneme p, $P(p|o_t)$ is estimated as the maximum state probability amongst the states belonging to the phoneme p. Further, the average value of phoneme posterior probability for the phoneme p over segment i is obtained as

$$\log P(p|i) = \frac{1}{T_i} \sum_{t=t_i^0}^{t_i^0+T_i} \log P(p|o_t) \tag{1}$$

Here, t_i^0 and T_i denote the start time and duration of the segment i, respectively, and are obtained from the forced alignments. Then, the GOP measure for the target phoneme p_i is defined as

$$\mathrm{GOP}(p_i|i) = \log \frac{P(p_i|i)}{\max_{q \neq p_i} P(q|i)} \tag{2}$$

A low value of the GOP measure entails a low probability of the target phoneme p_i as compared to other phonemes, as judged by the acoustic model, and hence, a low quality

of pronunciation. A threshold can be set on the GOP measure to detect mispronunciation. One could have a common threshold for all phonemes, but we use different threshold values for different phonemes. These thresholds are learned statistically using the training data, using simple logistic regression.

B. Classifier Based Measure

Given the target sequence of phonemes to be spoken and the speech signal uttered by the learner, the mispronunciation detector has to decide which phonemes are correctly pronounced and which ones are incorrectly pronounced by the learner. Hence, the mispronunciation detector produces a scalar output corresponding to the score that a phoneme is correctly pronounced. First of all, the learner utterance is force-aligned to the target phoneme sequence with the help of the DNN based acoustic model. The probability scores obtained from the acoustic model are then processed to form the input to the mispronunciation detector.

The mispronunciation detector comprises of an NN with one hidden layer of 500 neurons with ReLU non-linearity[2]. The reason for avoiding many hidden layers is primarily the unavailability of very large data for training them. Furthermore, several researchers (such

as (Hu *et al.*, 2015; Li *et al.*, 2016)) have obtained their best performance with a single hidden layer.

The input to the detector NN is prepared as follows. The average probability of each phonological feature f is obtained for each phoneme segment i as

$$P(f|i) = \frac{1}{T_i} \sum_{t=t_i^0}^{t_i^0+T_i} P(f|o_t) \tag{3}$$

Here, t_i^0 and T_i denote the start time and duration of the phoneme state s, respectively, and are obtained from the forced alignments. Since, each phoneme is modelled with 3 states, the input to the NN consists of three such vectors of Eq. 3 concatenated together.

The output of the NN is denoted as $P(p|t_p^0)$, where p is the target phoneme starting at time t_p^0. The number of neurons in the output layer is equal to the total number of phonemes in English (as defined in the dataset for transcriptions). Each neuron has sigmoid non-linearity to restrict the output between 0 and 1. Output 0 denotes mispronunciation and 1 signifies correct pronunciation. The phoneme p in the target transcription is detected to be mispronounced if $P(p|t_p^0) < \epsilon$, with ϵ being a scalar threshold, which is set the same for

all phonemes. Note that only the pth output of the NN is used, while the rest are not. We can also use different threshold values for different phonemes. In this architecture, all the phonemes have a shared hidden layer. This allows for improvement in the performance in the face of scanty training data as different phonemes benefit by mutual sharing of statistical properties. The benefit of shared representations has also been empirically observed by (Hu *et al.*, 2015).

1. Training

For the input corresponding to target phoneme p_i, the desired output of the NN at the p_ith neuron is set to 0 or 1, for incorrect or correct pronunciation, respectively. In order to estimate the ground truth for training, the speech utterance is force-aligned with the actually uttered phoneme sequence. If the aligned segment of the target phoneme overlaps (in time) more than 50% with the same phoneme in the actually uttered alignments, it is marked as correctly pronounced; if the temporal overlap is less than 10%, it is marked as incorrect pronunciation; while the rest of the segments are not used for training. Stochastic gradient descent is used to update the weights of all the layers by minimising the squared error objective function. While computing the weight update for the phoneme p_i, an important

issue for the shared classifier is to determine the desired output for other phonemes. We set the error feedback from output neurons corresponding to other phonemes to 0, implying that those errors do not contribute to the weight update.

Another concern here is the problem of unbalanced data. In order to deal with the uneven distribution of data over phonemes, apart from using a shared NN, we use a two-stage training scheme for the NN classifier. In the first stage, all the layers are trained together. In the second stage, only the output layer is trained, while keeping the hidden layer as fixed. As we have little data, which is unevenly distributed over different phonemes, the first step provides shared learning of layers, while the second step tunes the output for individual phonemes.

But even for the same phoneme, the number of correct and incorrect pronunciations are heavily unbalanced. We adopt sample weighting to deal with this, i.e., the output error value used for updating the NN is weighted using the data size for each class. The output error for each class (0 or 1) of phoneme p_i is weighted with the inverse ratio of the number of samples available for training each class.

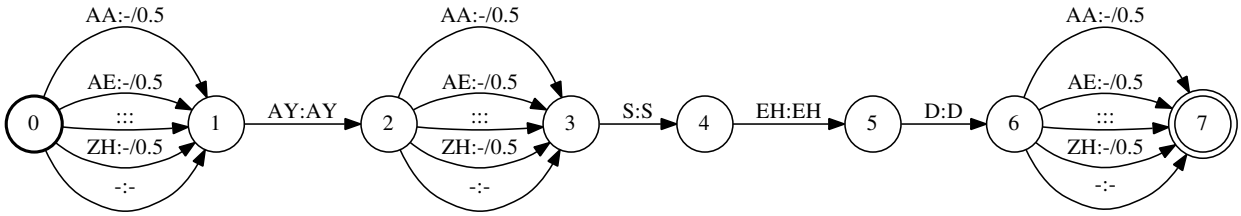

FIG. 2: FST used for decoding the target phoneme sequence of the sentence *I said*. There is a filler at each word boundary to allow insertions of single phonemes; '-' denotes no insertion. The bottom-most arc of each filler makes the insertion optional. P_{ins} is set to 0.5 in this case.

C. Filler Model for Insertions

Insertions can be handled during decoding of the phoneme sequence from the acoustic scores. ERN, which is popularly used, needs expert knowledge or extra training. Moreover, it limits the types of possible insertions. We tackle this problem with the help of a filler FST. The decoder FST is meant to find the sequence of phonemes in the learner's utterance that maps onto the target phoneme sequence. The filler FST takes any phoneme as the input and maps it on to no output, thereby allowing an extra phoneme in the input inserted by the learner. Thus, the filler FST is able to detect the insertion of a single phoneme. Since

too many fillers make the decoding computationally expensive and most insertions occur at word boundaries, we introduce fillers only at word boundaries.

For instance, Italians tend to insert a /ə/ after word-final consonants, while Spanish speakers tend to insert a /ə/ before word-initial /s/+CONS clusters. Since, only single phonemes are usually inserted, the filler FST allows insertion of only one phoneme at any given position. Fig. 2 shows an example FST used for decoding, with fillers at word boundaries.

The insertion of a phoneme by the filler model is controlled by introducing a penalty term, which is the probability of phoneme insertion P_{ins}. There is no penalty when there is no insertion, as transition probability is 1; but transition probability on the arc through any inserted phoneme is $P_{\text{ins}} \leq 1$.

D. Implementation and Experimental Evaluation

The test utterances of the ISLE database are analysed with the methods proposed above. The task of the system is to find the substitution, deletion and insertion errors, hence, a ground truth for substitution, deletion and insertion errors is needed for training and evaluating the system. This is prepared using the target phoneme sequences and the actually

uttered phoneme sequences. The ground truth substitution, deletion and insertion errors are found by aligning these two sequences using the Levenshtein distance. See Sec. III B for their counts in the ISLE dataset.

We analyse errors in terms of two classes – one for substitution and deletion errors, and another for insertion errors. Since these two classes are handled by different methods, we evaluate them independently.

1. Substitution and Deletion

To evaluate the performances of GOP-based and NN-based schemes over the test data, each phoneme uttered by the speaker is classified into correct or incorrect pronunciation, and the performance is measured in terms of false rejection rate (FRR) and false acceptance rate (FAR). Since they vary in opposite directions with a threshold, the performance measure is generally set as the equal error rate (EER), at which FRR and FAR are equal. Fig. 3 shows the receiver operating characteristic (ROC) curves for substitution and deletion errors in pronunciation. We can see that the NN classifier based system performs better than the GOP based system, achieving a lower EER.

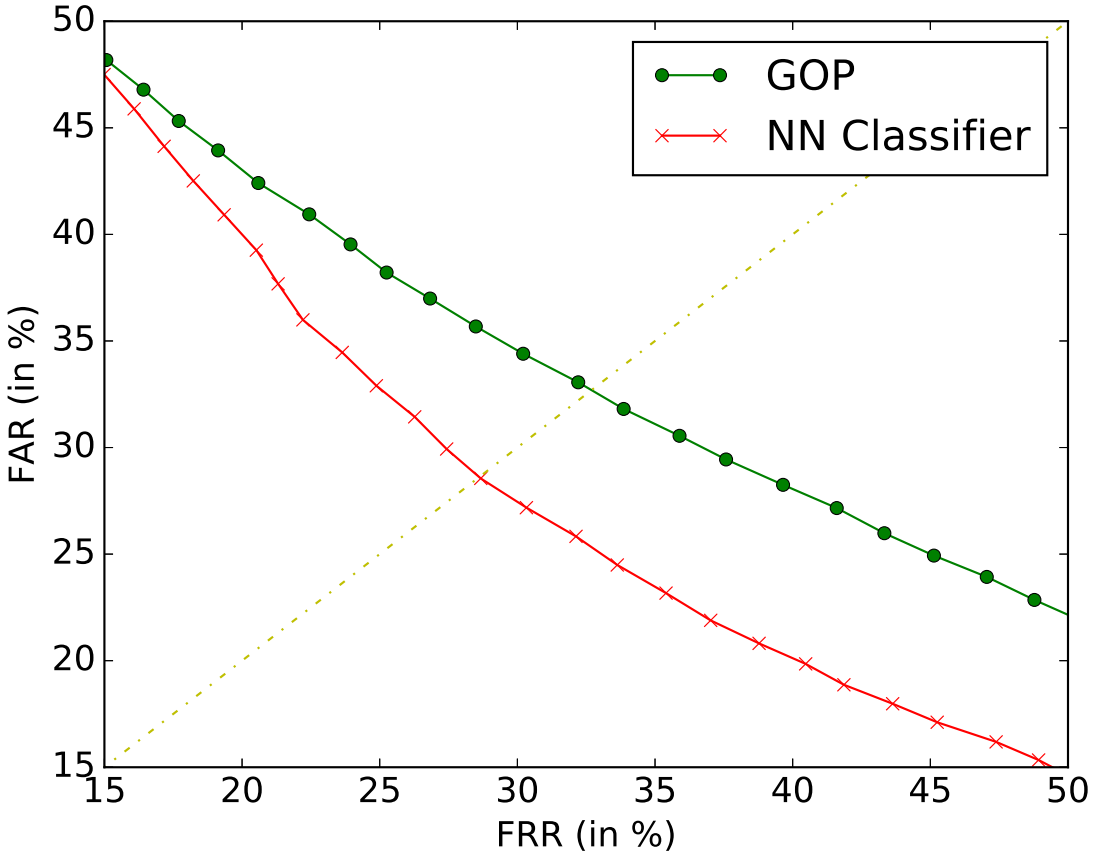

FIG. 3: ROC for substitution and deletion errors over the test set. EER for the GOP based scheme is 32.7%, while that for the NN classifier is 28.6%.

Table III provides further analyses of NN classifier system performance for the top 5 substituted/deleted phonemes. Among all phoneme substitutions/deletions (total 2661), /ə/ has the top share (12.9% or 343 times). Out of all these mispronunciations of /ə/, 8.2% of times (28 times) it is substituted with /ʊ/ and is caught by the system, while 5.5% of

TABLE III: Phoneme-wise analyses of substitution and deletion detection over the test set with the NN classifier. Presence denotes the substitution or deletion frequency of the target phoneme in the test set w.r.t. that of all target phonemes. FAR and FRR have been defined in the text. Caught and Missed, respectively, denote the best caught and missed phoneme substitutions, in the decreasing order of their share (given in parenthesis) among total mispronunciations of the target phoneme; '-' denotes deletion. All values are in %ge.

Phoneme	Presence	FAR	FRR	Caught			Missed		
ə	12.9	42.6	35.0	ɔ (18.4)	- (9.0)	ʊ (8.2)	ɛ (8.4)	ʊ (5.5)	ʌ (3.7)
ɪ	9.4	46.4	20.7	iː (56.6)	ə (6.3)	- (5.5)	iː (12.5)	ə (2.7)	- (2.3)
t	7.6	20.6	27.2	- (53.5)	d (8.4)	ʌ (3.0)	- (17.0)	ʌ (2.9)	d (2.4)
ð	6.6	36.8	31.2	d (51.5)	- (8.5)	θ (4.0)	d (19.8)	- (5.6)	v (3.3)
r	6.2	36.2	29.5	unmap (55.0)	- (10.1)	ə (1.8)	unmap (26.5)	- (1.8)	ɔ (0.6)

times (19 times) it is substituted with /ʊ/ and escapes uncaught from the system. When /r/ is mispronounced in a non-British fashion, it is labelled as 'unmap'.

Note that GOP-based system is essentially a phoneme-based error detection system, while the NN-based system is a feature-based system. We can implement an NN-based system as

TABLE IV: Evaluation of insertion detection over the test set, as a function of P_{ins}. Precision, recall and F-measure have been defined in the text, and are in %ge.

P_{ins}	Precision	Recall	F
1.0	25.6	70.8	37.6
0.9	26.3	67.2	37.8
0.8	28.1	65.0	39.2
0.7	29.5	60.9	39.7
0.6	31.4	56.8	40.4
0.5	34.9	54.5	42.5
0.4	37.3	49.9	**42.7**
0.3	40.5	42.3	41.4
0.2	44.5	35.0	39.2
0.1	50.0	26.1	34.3
0.01	61.5	12.2	20.4

a phoneme-based system as well. However, we have found the performance of the proposed feature-based NN classifier to be at par with the conventional phoneme-based NN classifier in another work ((Arora *et al.*, 2017)).

2. Insertion

Insertions are dealt with while decoding, with the help of a filler model. Hence, the performance of insertion detection is evaluated separately from that of substitution and deletion detection. Since the filler can insert phonemes at any word boundary, we measure its performance in terms of precision and recall. The two can be varied by changing the filler probability P_{ins}. Table IV shows the performance of insertion detection (precision, recall and F) as the value of P_{ins} is varied. We can observe that a lower value of P_{ins} inhibits insertions, thereby, lowering the recall and increasing precision. Changing P_{ins} affects the phoneme boundaries obtained during the forced alignment, and consequently, the performance of substitution and deletion detector as well; but, we found this effect to be negligibly small ($< 0.2\%$). The results of all other analyses have been reported with $P_{\text{ins}} = 0.4$.

Table V analyses the performance of insertion detection for the most frequently inserted phonemes. The phoneme that is inserted most often is /ə/. These come predominantly ($> 99\%$ times) from Italian speakers, who, as we have mentioned earlier, tend to insert a schwa after word-final consonants. Further, /g/ is inserted by Italian speakers after /ŋ/, /ɛ/ is inserted around vocalic or rhotic phonemes, and /h/ around silence.

TABLE V: Phoneme-wise analysis of insertion detection over the test set for $P_{\text{ins}} = 0.4$. Presence denotes the frequency of insertion of the phoneme in the test set. Precision, recall and F-measure have been defined in the text. All values are in %ge.

Phoneme	Presence	Precision	Recall	F
ə	54.4	98.1	56.6	71.8
unmap	6.6	100.0	44.8	61.9
g	4.6	50.0	29.0	36.7
ɛ	4.4	22.2	26.7	24.2
h	3.7	57.5	92.0	70.8

V. MISPRONUNCIATION FEEDBACK

For mispronunciation feedback, we use the estimated values of phonological features and compare them with that of the target phoneme. This gives us the extent of deviation of each feature from its target value. Based on the FUL model ((Lahiri , 2012; Lahiri and Reetz, 2010)), we can predict which features are not crucial for a particular target phoneme, due to *underspecification*, i.e., if a feature is underspecified its mispronunciation is tolerated. For example, if *rainbow* (/r eɪ n b oʊ/) is pronounced as [*r eɪ m b oʊ], it is not considered a mispronunciation, since the feature COR is underspecified. That is, /n/ and /m/ are both

457 specified as NAS, but only /m/ has the place of articulation feature LAB specified while /n/
458 remains underspecified for place. Thus, the LAB extracted from [m] in [*r eɪ m b oʊ] does
459 not mismatch with the represented underspecified /n/.

460 Among the rest of the features, the one with the strongest deviation is identified as the
461 one that needs to be corrected first. This can then be used to construct a feedback in terms
462 of articulation for easy comprehension of the learner. It is to be noted that the feedback is
463 given only for mispronounced phonemes.

464 A. Implementation

465 From $P(f|o_t)$ given by the acoustic model, the average value of a feature f over the
466 segment i is computed as

$$P(f|i) = \frac{1}{T_i} \sum_{t=t_i^0}^{t_i^0+T_i} P(f|o_t) \tag{4}$$

467 Let the target value of feature f for target phoneme p_i be given by $g(f|p_i)$. Based on the
468 FUL model, $g(f|p_i)$ can take one of the three values $\{+1, 0, -1\}$, where $+1$ signifies that

the feature f must be present, i.e., $P(f|i)$ should be close to 1; -1 implies that the feature f should be absent, i.e., $P(f|i)$ should be close to 0; and 0 entails unspecified value of f, i.e., the value of $P(f|i)$ is not important for correct pronunciation of target phoneme p_i. The extent of deviation D of each feature f is measured for the target phoneme p_i in the segment i as

$$D(f|i) = \begin{cases} 1 - P(f|i) & \text{if } g(f|p_i) = +1 \\ P(f|i) & \text{if } g(f|p_i) = -1 \\ 0 & \text{if } g(f|p_i) = 0 \end{cases} \tag{5}$$

For constructing the corrective feedback, we choose the feature with the largest deviation. We denote this feature as f^*. Mathematically, $f^* = \arg\max_f |D(f|i)|$. The sign of $D(f^*|i)$ determines the direction of change, namely, a positive sign implies that feature f^* needs to be increased, and a negative sign implies that it should be decreased. To improve the feedback using certain simple phonological observations, before determining f^*, the estimated $D(f|i)$ is processed as follows: (i) since the features VOC, SON, VOICE have very similar properties

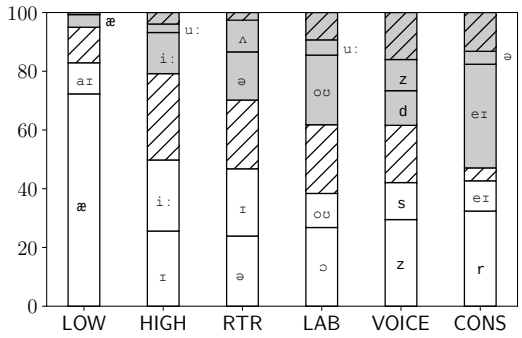

FIG. 4: Feature-wise analyses of feedback accuracy over the test set. Y-axis shows the distribution (in %) of mispronounced target phonemes in determining the accuracy for the features shown on X-axis. (Prominent phonemes are shown; the hatched region contains the share of others). White and grey regions show correct and incorrect feedback, respectively.

and they all correspond to some form of resonance, they can be merged for constructing feedback; (ii) $D(f|i)$ for SIL is made 0, since it is not relevant for feedback.

B. Experimental Evaluation

The ground truth for the test utterances is prepared using the target and actually uttered sequences, which have been aligned using minimum Levenshtein distance. The actually uttered phoneme tells which features are incorrectly pronounced, when compared with the

TABLE VI: Feature-wise evaluation of feedback accuracy over the test set. Attempts denote the number of times that feature was given as feedback. Accuracy has been defined in the text and is in %ge.

Feature	Attempts	Accuracy
SON	12	100.0
NAS	25	100.0
LAT	5	100.0
RAD	19	100.0
RHO	166	95.2
LOW	141	95.0
CONT	154	94.8
STOP	189	93.7
STR	41	85.4
HIGH	207	79.2
DOR	101	78.2
COR	249	72.3
RTR	833	70.2
OBSTR	41	68.3
LAB	173	61.8
VOICE	237	61.6
CONS	68	47.1
All	2661	76.4

TABLE VII: Phoneme-wise evaluation of feedback accuracy over the test set. The number in bracket after each phoneme is the number of times that target phoneme has been mispronounced. Accuracy has been defined in the text and is in %ge.

Phoneme	Accuracy	Phoneme	Accuracy
aː (27)	100.0	l (6)	100.0
æ (156)	96.2	m (2)	50.0
ʌ (165)	42.4	n (25)	100.0
ɔː (25)	52.0	ŋ (7)	100.0
aʊ (7)	100.0	ɔ (82)	89.0
ə (343)	58.3	oʊ (81)	35.8
aɪ (20)	85.0	ɔɪ (4)	100.0
b (10)	60.0	p (6)	83.3
tʃ (10)	80.0	r (166)	99.4
d (77)	63.6	s (56)	87.5
ð (176)	85.8	ʃ (5)	60.0
ɛ (107)	80.4	t (202)	91.6
ɚ (60)	85.0	θ (16)	100.0
eɪ (132)	43.9	ʊ (42)	100.0
f (4)	25.0	uː (44)	59.1
g (8)	87.5	v (38)	55.3
h (26)	100.0	w (8)	100.0
ɪ (251)	97.2	ɹ (25)	100.0
iː (87)	59.8	z (103)	73.8
ʤ (15)	46.7	ʒ (1)	100.0
k (36)	97.2	All (2661)	76.4

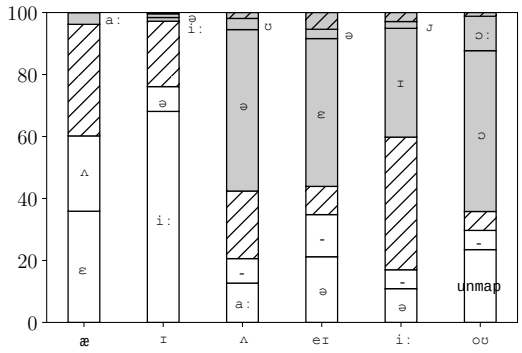

FIG. 5: Phoneme-wise analyses of feedback accuracy over the test set. Y-axis shows the distribution (in %) of actually uttered phonemes in determining the accuracy of each mispronounced target phoneme on the X-axis (Prominent ones are shown in bars; the hatched region contains the share of others). White and grey regions show correct and incorrect feedback, respectively; '-' denotes deletion.

target phoneme. The system to be evaluated should feedback an incorrectly pronounced feature with the correct sign of deviation.

The value of a feature f for the target phoneme p_i is given by $g(f|p_i)$ and that for the actually uttered phoneme q_i is given by $g(f|q_i)$. The ground truth for feedback for the feature f is simply $g(f|p_i) - g(f|q_i)$. If the sign ($+, -$, or 0) of estimated $D(f^*|i)$ matches with that of $g(f^*|p_i) - g(f^*|q_i)$, it is considered to be a correct feedback. The accuracy of feedback is defined as the number of times the feedback is correct divided by the total

number of times a feedback is rendered, and is evaluated feature-wise (Table VI) as well as phoneme-wise (Table VII). Feature-wise accuracy is measured over the total number of times a particular feature was given as feedback, while phoneme-wise accuracy is measured over the total number of times a particular target phoneme was mispronounced. All these evaluations are performed over actual mispronunciations, and not the estimated ones, so as to evaluate the feedback system alone.

Table VI shows that certain features are diagnosed better than others. Notably, though the features LAT, RAD, LOW, RHO are detected poorly (see Table II), still they are diagnosed quite well. Fig. 4 further analyses the feature-wise accuracy for some of the features with low accuracies and large number of attempts, by showing each feature's distribution over different target phonemes. For instance, among all the cases of CONS given as feedback, 10.3% times it is rendered correctly for the phoneme /eɪ/ and 35.3% of times rendered incorrectly for the same. As another example, the feature RTR is selected as feedback 833 times; out of this, 70.2% times (585 times) it is correct (see Table VI). Out of all the RTR feedbacks (see Fig. 4), 23.9% times (199 times) it is for the phoneme /ə/ and is correctly rendered (this adds to the correct feedbacks), while 16.4% times (137 times) it is for the same phoneme /ə/ but is not relevant for correcting those mispronunciations (hence, it adds

to the incorrect feedbacks). This figure shows that the system is able to recognise which features are important for which phonemes. E.g., the feature LOW is given as feedback mostly for the phoneme /æ/.

Similarly, Table VII shows which phonemes are diagnosed better than others. The reason for poor accuracies of diphthongs like /oʊ eɪ/ is that we assign single set of features to each diphthong whereas a diphthong involves change of features over time. The phoneme-wise accuracy is also broken-down in Fig. 5 for certain target phonemes with low accuracies. For instance, out of all the mispronunciations of /eɪ/ (132 times) as the target phoneme, 24.2% times (32 times) it is substituted with /ə/, where 21.2% times (28 times) the system renders correct feedback and 3.0% times (4 times) it fails to identify the feature which needs to be corrected. This figure shows which phoneme-pairs are, and which ones are not, distinguished well by the system. The system is able to efficiently recognise when (for example) the target phoneme /æ/ is mispronounced as /ɛ/ or /ʌ/, because of effective recognition of features that distinguish these phonemes. While the system needs improvement in identifying when (for example) /oʊ/ is mispronounced as /ɔ/ because of its not being able to recognise well the distinguishing features for them.

The above analysis shows the usefulness of feature based analysis. By using features, we find that many phonemes are affected in the same way. On the other hand, phoneme based analysis only provides information about a single phoneme, and without even telling what exactly is incorrect in it. Moreover, with these 18 features, it is possible to construct not only the 41 English phonemes used in this paper, but also the phonemes of many other languages. For example, a set of nasal vowels can simply be modelled by adding a NAS feature to the oral vowels. Furthermore, the features that are well diagnosed by the system predict good diagnosis of the corresponding phonemes. For instance, The features NAS, RAD, RHO, LAT have high scores (Table VI) and we find that the phonemes corresponding to these features also perform very well (see Table VII). Consequently, improving the performance of one feature will have an immediate effect on all the related phonemes. At the same time, the system gains from the underspecification of features for many phonemes: a decrease of Type I errors does not necessarily lead to an increase of Type II errors, and vice versa.

While these results very well support the high accuracy of the present system, there are several avenues for improvement. Table VII shows that diphthongs /ɔɪ, eɪ, oʊ/ perform poorly. In this work, we assigned each diphthong with the features of the first phoneme for simplification. This simplification in our model leads to some errors in diagnosis. Modelling

a diphthong as a sequence of two phonemes might help to alleviate these errors. Further research in this direction is currently underway. We are also working towards developing more effective acoustic methods for improving feature extraction from speech, using methods like cross language model transfer and multi-task learning.

VI. CONCLUSION

In this paper, we have presented a computer-aided pronunciation training system, employing an ASR system using phonological features. Our ASR system detects mispronunciation errors and renders corrective feedback by diagnosing those errors. There have been several attempts to use phonological features in speech perception rather than full phonemes. In this instance, we used phonological features in all the stages of the system, from acoustic modelling to mispronunciation detection and diagnosis. Our ASR system is successful in extracting phonological features when analysing the speech of learners and we believe that these features provide better insights for correcting mispronunciations than phoneme or word level feedback to learners, which in turn helps in more efficient learning. A feature based CAPT system can also predict the L1 phonological system. If, for instance, the learners make consistent errors with differentiating strident fricatives from others, then one could

conclude that their L1 does not have this contrast. Conversely, if the learners are good at distinguishing voicing contrast in all word positions, their L1 must allow for voiced and voiceless consonants everywhere. We can use this information to customise our system to specific L1 learners. Since our system is largely feature based, it could be extended to other languages, even if their phoneme system is not identical. Features also allow the system to capture mistakes without requiring to model the phonemes that are not present in L2, as the features can themselves cover a broad range of possible phonemes. In many instances, learners cut across several phonemes in making errors. For instance, a frequent error in voicing involves incorrectly producing /s/ instead of /z/ in English words such as *grades, bags* etc. An useful feedback for the learner is to point out that words ending with a voiced consonant will always take a voiced plural ending. Thus, a feature based CAPT system which provides learners with feedback based on feature errors, has many positive consequences.

ACKNOWLEDGMENTS

This research was supported by the ERC Proof of Concept FLEX-SR award no. 632226 and the ERC Advanced Research Grant no. 695481 for the project "MORPHON: Re-

solving Morpho-Phonological Alternation: Historical, Neurolinguistic, and Computational approaches".

[1]http://web.phonetik.uni-frankfurt.de/upsid.html

[2]$\text{ReLU}(x) = x, \text{if } x > 0; \text{and} = 0, \text{otherwise}$

Amit, Y., Koloydenko, A., and Niyogi, P. (2005). "Robust acoustic object detection," *J. Acoust. Soc. Am.*, 118, 2634-2648.

Arora, V., Lahiri, A., and Reetz, H. (2016). "Attribute Based Shared Hidden Layers for Cross-Language Knowledge Transfer," in *IEEE Workshop on Spoken Language Technology*, 617–623.

Arora, V., Lahiri, A., and Reetz, H. (2017). "Phonological Feature Based Mispronunciation Detection and Diagnosis using Multi-Task DNNs and Active Learning," in *INTERSPEECH*, 1432-1436.

Blumstein, S. E., and Stevens, K. N. (1979). "Acoustic invariance in speech production: Evidence from measurements of the spectral characteristics of stop consonants," *J. Acoust.*

589 *Soc. Am.*, 66(4), 1001-1017.

590 Browman, C. P., and Goldstein, L. (1992). "Articulatory phonology: An overview," *Pho-*

591 *netica*, 49(3-4), 155-180.

592 Deng, L., and Sun, D. X. (1994). "A statistical approach to automatic speech recognition us-

593 ing the atomic speech units constructed from overlapping articulatory features," *J. Acoust.*

594 *Soc. Am.*, 95, 2702–2719.

595 Franco, H., Neumeyer, L., and Bratt, H. (1999). "Automatic Detection of Phone-Level

596 Mispronunciation for Language Learning," in *EUROSPEECH*, 851–854.

597 Franco, H., Neumeyer, L., Digalakis, V., and Ronen, O. (2000). "Combination of machine

598 scores for automatic grading of pronunciation quality," *Speech Commun.*, 30, 121–130.

599 Franco, H., Ferrer, L., and Bratt, H. (2014). "Adaptive and discriminative modeling for

600 improved mispronunciation detection," in *IEEE ICASSP*, 7709–7713.

601 Hasegawa-Johnson, M., Baker, J., Borys, S., Chen, K., Coogan, E., Greenberg, S., Juneja,

602 A., Kirchoff, K., Livescu, K., Mohan, S., Muller, J., Sonmez, K., and Wang, T., (2004),

603 "Landmark-Based Speech Recognition: Report of the 2004 Johns Hopkins Summer Work-

604 shop," in *IEEE ICASSP*, 213–216.

605 Hu, W., Qian, Y., Soong, F. K., and Wang, Y. (2015). "Improved mispronunciation detection
606 with deep neural network trained acoustic models and transfer learning based logistic
607 regression classifiers," *Speech Commun.*, 67, 154–166.

608 Jakobson, R., Fant, G., and Halle, M. (1952). "Preliminaries to Speech Analysis: The
609 Distinctive Features and Their Correlates," *MIT Press*, Cambridge, MA, 1–64.

610 Jansen, A., and Niyogi, P. (2008). "Modeling the temporal dynamics of distinctive feature
611 landmark detectors for speech recognition," *J. Acoust. Soc. Am.*, 124, 1739–1758.

612 Jansen, A., and Niyogi, P. (2009). "Point process models for spotting keywords in continuous
613 speech," *IEEE Trans. Audio, Speech, Lang. Process.*, 17, 1457–1470.

614 Juneja, A., and Espy-Wilson, C. (2008). "A probabilistic framework for landmark detection
615 based on phonetic features for automatic speech recognition," *J. Acoust. Soc. Am.*, 123(2),
616 1154–1168.

617 Kartushina, N., Hervais-Adelman, A., Frauenfelder, U. H., and Golestani, N. (2015). "The
618 effect of phonetic production training with visual feedback on the perception and produc-
619 tion of foreign speech sounds," *J. Acoust. Soc. Am.*, 138, 817-832.

620 King, S., and Taylor, P. (2000). "Detection of phonological features in continuous speech
621 using neural networks," *Computer Speech and Language*, 14, 333–353.

Lahiri, Aditi (2012). "Asymmetric phonological representations of words in the mental lexicon," In Cohn, A. C., Fougeron, C., and Huffman, M. K. (eds) *The Oxford Handbook of Laboratory Phonology*, Oxford University Press, Oxford, 146–161.

Lahiri, A., Gewirth, L., and Blumstein, S. E. (1984). "A reconsideration of acoustic invariance for place of articulation in diffuse stop consonants: Evidence from a crosslanguage study," *J. Acoust. Soc. Am.*, 76(2), 391–404.

Lahiri, A., and Reetz, H. (2010). "Distinctive features: Phonological underspecification in representation and processing," *Journal of Phonetics*, 38(1), 44-59.

Lee, A., and Glass, J. (2015). "Mispronunciation detection without nonnative training data," in *INTERSPEECH*, 643–647.

Li, K., Meng, H., Chinese, T., Kong, H., and Sar, H. K. (2017). "Mispronunciation Detection and Diagnosis in L2 English Speech Using Multi-Distribution Deep Neural Networks," *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, 25, 193-207.

Li, W., Siniscalchi, S. M., Chen, N. F., and Lee, C.-H. (2016). "Improving non-native mispronunciation detection and enriching diagnostic feedback with DNN-based speech attribute modeling," in *IEEE ICASSP*, 6135–6139.

638 Livescu, K., Jyothi, P., and Fosler-Lussier, E. (2015). "Articulatory feature-based pronunci-
639 ation modeling," *Computer Speech and Language*, 36, 212–232.

640 Livescu, K., and Glass, J. (2004). "Feature-based pronunciation modeling for automatic
641 speech recognition," in *HLT-NAACL, Association for Computational Linguistics*, 81–84.

642 Lo, W., Zhang, S., and Meng, H. (2010). "Automatic Derivation of Phonological Rules
643 for Mispronunciation Detection in a Computer-Assisted Pronunciation Training System
644 Effects with Phonological Rules," in *INTERSPEECH*, 765–768.

645 Meng, H., Lo, Y. Y., Wang, L., and Lau, W. Y. (2007). "Deriving salient learners' mispronun-
646 ciations from cross-language phonological comparisons," in *IEEE Workshop on Automatic*
647 *Speech Recognition & Understanding (ASRU)*, 437–442.

648 Menzel, W., Atwell, E., Bonaventura, P., Herron, D., Howarth, P., Morton, R., and Souter,
649 C. (2000). "The ISLE corpus of non-native spoken English," in *Proceedings of LREC*,
650 957–963.

651 Mohri, M., Pereira, F., and Riley, M. (2002). "Weighted finite-state transducers in speech
652 recognition," *Computer Speech and Language*, 16(1), 69-88.

653 Nagamine, T., Seltzer, M. L., and Mesgarani, N. (2015). "Exploring how deep neural net-
654 works form phonemic categories," in *INTERSPEECH*, 1912–1916.

Niyogi, P., and Sondhi, M. M. (2002). "Detecting stop consonants in continuous speech," *J. Acoust. Soc. Am.*, 111, 1063–1076.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). "The kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*.

Qian, X., Meng, H., and Soong, F. (2016). "A Two-pass Framework of Mispronunciation Detection & Diagnosis for Computer-aided Pronunciation Training," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 24, 1020–1028.

Raab, M., Gruhn, R., and Noeth, E. (2007). "Non-native speech databases," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 413–418.

Siniscalchi, S. M., Svendsen, T., and Lee, C.-H. (2013). "A Bottom-Up Modular Search Approach to Large Vocabulary Continuous Speech Recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, 21, 786–797.

Stevens, K. N. (2004). "Fifty years of progress in acoustic phonetics," *J. Acoust. Soc. Am.*, 116, 2496.

671 Stevens, K. N., and Blumstein, S. E. (1978). "Invariant cues for place of articulation in stop
672 consonants," *J. Acoust. Soc. Am.*, 64, 1358–1368.

673 Suemitsu, A., Dang, J., Ito, T., and Tiede, M. (2015). "A real-time articulatory visual
674 feedback approach with target presentation for second language pronunciation learning,"
675 *J. Acoust. Soc. Am.*, 138, EL382–EL387.

676 Sun, J., and Deng, L. (2002). "An overlapping-feature-based phonological model incorpo-
677 rating linguistic constraints: applications to speech recognition," *J. Acoust. Soc. Am.*, 111,
678 1086–1101.

679 Tan, S., Sim, K. C., and Gales, M. (2015). "Improving the interpretability of deep neural
680 networks with stimulated learning," in *IEEE Workshop on Automatic Speech Recognition
681 and Understanding (ASRU)*, 617–623.

682 van Doremalen, J., Cucchiarini, C., and Strik, H. (2013). "Automatic pronunciation error
683 detection in non-native speech: the case of vowel errors in Dutch," *J. Acoust. Soc. Am.*,
684 134, 1336–1347.

685 Wang, Y. B., and Lee, L. S. (2015). "Supervised detection and unsupervised discovery of
686 pronunciation error patterns for computer-assisted language learning," *IEEE Trans. Audio,
687 Speech, Lang. Process.*, 23, 564–579.

688 Witt, S. (1999). "Use of speech recognition in computer assisted language learning," Ph.D.
689 thesis, University of Cambridge, Cambridge, UK, 1–139.
690 Yu, D., Siniscalchi, S. M., Deng, L., and Lee, C. H. (2012). "Boosting attribute and phone
691 estimation accuracies with deep neural networks for detection-based speech recognition,"
692 in *IEEE ICASSP*, 4169–4172.