Phoneme Recognition and its Applications in Computer-Aided Pronunciation Training

Priyank Avijeet
Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
pavijeet@uwaterloo.ca

Ishrat Ishtiaq Patel
Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
iipatel@uwaterloo.ca

Abstract

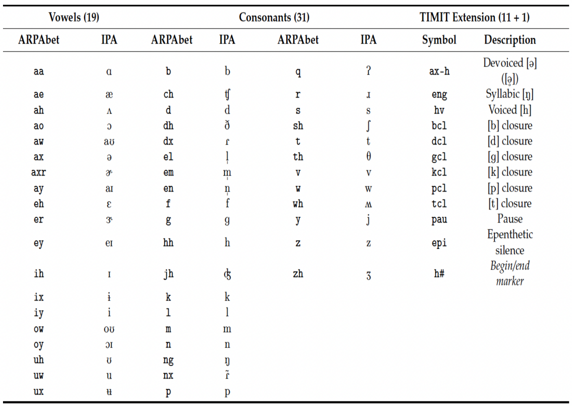
Phoneme recognition is a fundamental task in automatic speech recognition (ASR) systems, which aims to identify and classify individual speech sounds known as phonemes. In this report, we provide an overview of the phoneme recognition task, highlighting the challenges involved and discussing the approaches used to address them. We explore various models, including DNN, RNN, CNN+RNN, and Wav2Vec2 architecture-based models, to investigate their effectiveness in phoneme recognition. The performance of these models is evaluated by comparing the Phoneme Error Rate (PER) and Accuracy metrics. Additionally, we investigate the application of these models in computer-aided pronunciation training (CAPT). We propose an approach for detecting mispronounced words using the Wav2Vec2-based model. To assist learners in practicing correct pronunciation, we utilize the Double Metaphone algorithm to generate phonetically similar words.

Github link: Phoneme Recognition and its Applications in Computer-Aided Pronunciation Training

1 Introduction

The task of identifying phonemes, or the smallest distinct sound units that can differentiate words within a specific language, is known as phoneme recognition or automatic phonetic transcription. This is a critical aspect of computational linguistics and speech processing that entails the automatic spotting and categorization of phonemes. We can use these phonemes to verify if a person is pronouncing the words correctly. Computer-assisted Pronunciation Training (CAPT) provides innovative possibilities for language learning, as the automated system is readily accessible around the clock for immersive learning, and also addresses the issue of a shortage of teachers. One of the central technologies in CAPT is mispronunciation detection and diagnosis (MDD) at the phonetic level. MDD presents more of a challenge than automatic speech recognition (ASR) as ASR employs a language model that can compensate for inaccurate acoustic signals to produce the correct character sequence, whereas MDD cannot (Leung et al., 2019a).

Phonemes are units that signify the unique sounds found in a spoken language, with most languages comprising approximately 20 to 60 of these units. In our study, we utilize ARPAbet (Klautau, 2001), developed by the Advanced Research Projects Agency (ARPA), to represent IPA (International Phonetic Alphabet) symbols (Association, 2005) using standard English alphabetic symbols. The transition from ARPAbet to IPA symbols is illustrated in Figure 1, which includes an extra 11 symbols (Gold et al., 2011) employed in the TIMIT dataset (Consortium, 1990), (Garofolo et al., 1993) for a more in-depth phoneme examination. The symbol [h#], although not a phoneme, serves as a specific marker for the start and conclusion of a sentence, so the total number of supplementary TIMIT phonemes amounts to 11.



Vowels (19)		Consonants (31)				TIMIT Extension (11 + 1)	
ARPAbet	IPA	ARPAbet	IPA	ARPAbet	IPA	Symbol	Description
aa	ɑ	b	b	q	ʔ	ax-h	Devoiced [ə] ([ə̥])
ae	æ	ch	ʧ	r	ɹ	eng	Syllabic [ŋ]
ah	ʌ	d	d	s	s	hv	Voiced [h]
ao	ɔ	dh	ð	sh	ʃ	bcl	[b] closure
aw	aʊ	dx	ɾ	t	t	dcl	[d] closure
ax	ə	el	l̩	th	θ	gcl	[g] closure
axr	ɚ	em	m̩	v	v	kcl	[k] closure
ay	aɪ	en	n̩	w	w	pcl	[p] closure
eh	ɛ	f	f	wh	ʍ	tcl	[t] closure
er	ɝ	g	g	y	j	pau	Pause
ey	eɪ	hh	h	z	z	epi	Epenthetic silence
ih	ɪ	jh	ʤ	zh	ʒ	h#	*Begin/end marker*
ix	ɨ	k	k				
iy	i	l	l				
ow	oʊ	m	m				
oy	ɔɪ	n	n				
uh	ʊ	ng	ŋ				
uw	u	nx	ɾ̃				
ux	ʉ	p	p				

Figure 1: The extended 2-letter ARPAbet in the TIMIT dataset maps English phonetic symbols to IPA equivalents, comprising 19 vowels, 31 consonants, and 11 unique symbols, totaling 61 phonemes. The begin/end sentence marker isn't included in phoneme recognition. (Oh et al., 2021)

By analyzing the limitations of various models, we aim to identify the best-performing Phoneme Recognition model based on the available resources. Throughout this study, we will also explore the challenges faced in real-world applications of Phoneme Recognition models when used for detecting mispronounced words.Upon comparing the phoneme transcription of the given text with the phonemes uttered by the user, we can accurately identify instances of mispronunciation. When a mispronunciation is detected, we will provide the user with a list of phonetically similar words using the Double Metaphone algorithm (Philips, 2000) that can be used for practice and improvement.

2 Related Work

In order to comprehend the field of phoneme recognition, we traced the progression of methodologies, starting from simpler models and gradually moving toward more intricate ones.

The study (Schwarz et al., 2004) focuses on optimizing the TempoRAl Patterns (TRAP) based phoneme recognizer for the TIMIT database, incorporating more states per phoneme and bi-gram language models, and addressing the issue of insufficient training data, resulting in a 23.6% relative improvement in phoneme error rate. The phoneme insertion penalty in the decoder was also tuned to minimize PER, further reducing the error rate to a final value of 24.50%. The proposed novel speech recognition system in (Ying, 2019) integrates deep neural networks and hidden Markov models (HMM-DNN) to leverage multi-task learning, thereby enhancing accuracy through shared layers and transforming multi-phoneme error detection into a collective task. High-dimensional phonetics serve as primary input features, with each phoneme's binary classifier built into the same network, leading to improved results over single-task learning and a significant reduction in word error rate when the outputs of both systems are combined.

(Graves et al., 2013) explores the effectiveness of deep recurrent neural networks that blend the advantages of multiple levels of representation from deep networks with the long-range context utilization of RNNs. By incorporating the CTC (Connectionist Temporal Classification) (Graves et al., 2006a) approach and training with appropriate regularization, the researchers achieved a notable 17.7% test set error on the TIMIT phoneme recognition benchmark using deep Long Short-term Memory RNNs. Application of CTC with CNN-RNN networks is explored by (Leung et al., 2019b). This work presents an end-to-end speech recognition system for Mispronunciation Detection and Diagnosis (MDD) using Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Connectionist Temporal Classification (CTC) techniques. The results demonstrate a substantial improvement in the F-measure (74.65%) compared to alternative MDD approaches, surpassing them by 44.75% (Extended Recognition Network - S-AM) and 32.28% (State-level Acoustic Model - S-AM) respectively.

In recent times, there have been remarkable advancements in Natural Language Processing across all domains, thanks to transformers. Meta AI's Wav2Vec and Wav2Vec2 architectures (Schneider et al., 2019; Baevski et al., 2020b) have played a significant role in these breakthroughs. Leveraging these architectures, researchers have developed numerous Phoneme Recognition models through fine-tuning variations of Wav2Vec. Notably, an implementation (Phy, 2022) achieved an impressive phoneme error rate of just 7.996% on the challenging DARPA TIMIT dataset. We will be analyzing this model in our study. Wav2Vec2Phoneme was built on the research of (Xu et al., 2021), this builds upon prior research in zero-shot cross-lingual transfer learning by refining a multilingual pre-trained wav2vec 2.0 model to transcribe languages not previously encountered. This is achieved by aligning phonemes from the training languages with the target language through the use of articulatory attributes. Experimental results suggest that this straightforward approach markedly surpasses previous attempts that implemented task-oriented architectures and utilized merely a portion of a unilingually pre-trained model.

Similar sounding word generation for practicing the sounds people struggle uttering would be our approach towards our CAPT solution. Phoneme Embeddings (Silfverberg et al., 2018) and Double Metaphone (Philips, 2000) algorithms can be used to generate a list of similar-sounding words. It is possible to attribute pronunciation errors in non-native speech to the primary language's influence, which results in the transmission of specific linguistic features (Lo et al., 2010). Phonological rules have been created to model these mispronunciation patterns made by language learners (Lo et al., 2010). The Extended Recognition Network (ERN) of a conventional speech recognizer is then given these rules, enabling the capacity of Mispronunciation Detection and Diagnosis (MDD) (Harrison et al., 2009). For the benefit of university students, an online Computer-Assisted Pronunciation Training (CAPT) system has been created (Yuen et al., 2011), (Liu et al., 2012) employing ERN. It's crucial to understand that not all potential pronunciation errors made by learners of different languages may be covered by ERN. The acoustic model may find it difficult to successfully distinguish between a large number of mispronunciation possibilities when the recognition network becomes too sophisticated.

3 Approach

We preprocessed the dataset to map speech to its phoneme transcriptions. The performance of the models is evaluated based on accuracy and phoneme error rate.

3.1 Preprocessing for Neural Networks

In the preprocessing stage of the project, the initial steps were focused on standardizing the phonetic representation and setting the necessary parameters for feature extraction. The phoneme transcriptions were stripped of numerical stress indicators to ensure a consistent phonetic representation across all data points. Furthermore, to reduce complexity, the phoneme set was simplified from 61 to a more manageable 39 (Lee and Hon, 1989). To complement these steps, window length and step size parameters were meticulously defined to align with the sample rate of the audio files.

The following phase of preprocessing centered around data cleansing and preparation. Descriptions of the training and testing datasets were imported from the corresponding CSV files. A rigorous cleaning process was implemented, removing rows with NaN entries and filtering the data to include only specific dialect regions of interest. Meanwhile, the audio data was read and pre-emphasized, converting it into a format ready for feature extraction. In parallel, corresponding phoneme files were also processed to prepare the labels.

The concluding steps of preprocessing involved feature extraction and label preparation. Mel-frequency cepstral coefficients (MFCCs) (Qamar et al., 2013) also shown in Figure 2 were computed from the audio data, offering an insightful representation of its spectral characteristics. The option to include the delta and delta-delta of these MFCCs was also incorporated to provide a more comprehensive representation, if required. Phoneme labels were derived for each time step in the audio files and were subsequently converted into their 39-phoneme equivalent. These labels were then transformed into a one-hot encoded format, ensuring their compatibility with machine learning algorithms. A provision was also made to reverse this encoding if required. Ultimately, these preprocessing steps succeeded in transforming raw audio files into a set of feature vectors and corresponding labels, creating a robust foundation for the subsequent training phase of the project. This extraction of audio features was done for DNN, RNN, and CNN-RNN models. Wav2Vec2.0 (Baevski et al., 2020b), a model developed by Facebook AI that is trained to learn useful representations directly from raw audio data.

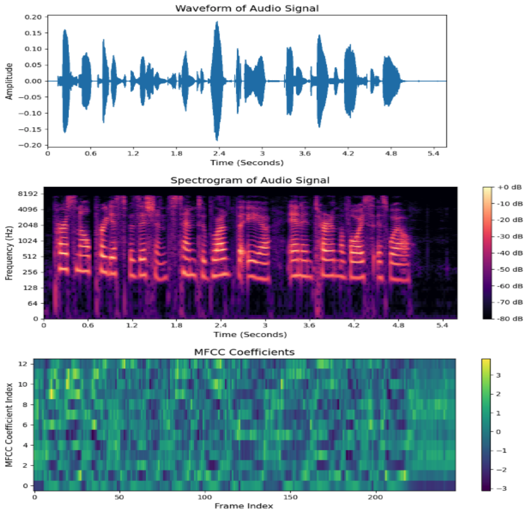


Figure 2: Audio signal, spectrogram and the MFCC features for a sample audio

3.2 Deep Neural Networks based architecture

We implemented a straightforward, fully-connected neural network model designed for the task of phoneme recognition, built using TensorFlow's Keras API. This model was selected as a baseline due to its simplicity. The model architecture consists of four layers in total. The input data is expected to be a 1D array of features with a size equal to three times the number of Mel frequency cepstral coefficients (192, since n_mels = 64), as we are also sending delta (differential) and delta-

delta (acceleration) coefficients is that in order to recognize speech better. The first three layers of the model are densely connected (also known as fully connected), each comprising of 1024 neurons and using the ReLU (Rectified Linear Unit) activation function, which helps the network learn complex patterns by introducing non-linearity. The fourth and final layer consists of 39 neurons, corresponding to the total number of phonemes the model is expected to recognize. This layer employs the softmax activation function, which is particularly useful in multi-class classification tasks because it generates a probability distribution across the output classes. The model uses the Adam optimizer and the categorical cross-entropy loss function, which is standard for multi-class classification tasks.

3.3 Recurrent Neural Networks (LSTM) based architecture

This model for phoneme recognition is designed around using the pattern-learning ability of recurrent neural networks, specifically Long Short-Term Memory (LSTM) units. In simple terms, it works by taking in a sequence of audio features (think of these as the 'DNA' of the sound), where each feature is represented as a set of Mel frequency cepstral coefficients. Each LSTM layer (and we have three of them, each with 1000 'neurons' or decision units) then analyses these sequences and tries to find patterns or 'rules' that help it predict the phoneme being spoken. LSTMs are great for this because they have a kind of memory, allowing them to understand longer sequences and the relationship between different parts of the sequence. Finally, the model uses a Dense layer that makes the final decision on which phoneme is being spoken. This layer provides a kind of 'vote' on which phoneme is the most likely, based on the patterns detected by the LSTMs. The model is trained using a method called Adam, which is a way of gradually improving the model's predictions, and its performance is measured by how accurately it can predict the correct phonemes.

3.4 Convolutional Neural Networks and Recurrent Neural Networks based architecture

The model is designed for phoneme recognition and starts with an input layer ready to process Mel frequency cepstral coefficients - a representation of the audio. It then utilizes two 1-dimensional convolutional layers, equipped with 32 filters and an 11-point kernel. These layers meticulously scrutinize the audio information, highlighting important patterns. Each convolutional layer is followed by a batch normalization process and a ReLU function, which standardize the activations and ensures the data remains positive, creating a stable environment for the calculations.

Following the initial processing, the model employs two LSTM layers with 128 and 64 neurons, serving as the model's memory. They maintain track of how the audio patterns change over time, a crucial aspect of understanding spoken language. The model then uses a TimeDistributed Dense layer to interpret the processed audio patterns and assigns phoneme probabilities, making informed predictions about the most likely phonemes for each sound segment. Training of this model involves the Adam optimizer with a learning rate of 0.0001, and the performance is evaluated based on accuracy and the reduction of categorical cross-entropy loss.

3.5 XLSR-Wav2Vec2 finetuned with CTC loss based architecture

Wav2Vec2, a pre-trained model developed for transforming speech into written text (Automatic Speech Recognition or ASR), was launched in September 2020 by Alexei Baevski, Michael Auli, and Alex Conneau (Baevski et al., 2020a). Following its impressive performance on the English language dataset LibriSpeech, Facebook AI introduced an upgraded version called XLSR-Wav2Vec2 (Baevski et al., 2020b). XLSR stands for cross-lingual speech representations, meaning XLSR-Wav2Vec2 can learn and understand speech across various languages.

Like its predecessor Wav2Vec2, XLSR-Wav2Vec2 gains its understanding of speech by learning from a massive amount of data, specifically hundreds of thousands of hours of speech from more than 50 different languages. It uses a learning approach similar to BERT's masked language modelling. The model randomly hides parts of the speech data and then attempts to predict the hidden parts, allowing it to understand the context of the speech, all before passing the data through a transformer network to generate the final output.

XLSR-Wav2Vec2 is finetuned and improved using a method known as Connectionist Temporal Classification (CTC) (Graves et al., 2006b). This

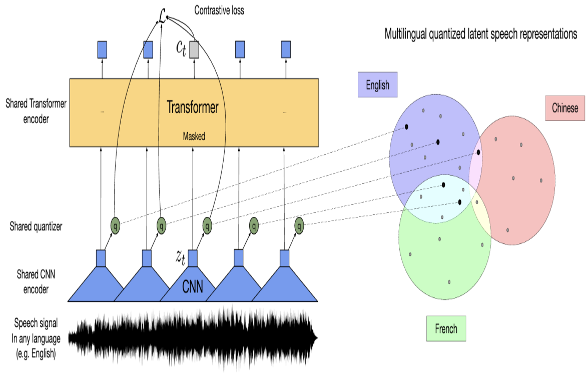

Contrastive loss
$\mathcal{L}$
Multilingual quantized latent speech representations
c_t
Shared Transformer encoder
Transformer
Masked
English
Chinese
Shared quantizer
q
z_t
Shared CNN encoder
CNN
Speech signal In any language (e.g. English)
French

Figure 3: The XLSR approach utilizes shared quantization and contrastive learning to create multilingual speech units, enabling the model to understand shared tokens across multiple languages. (Baevski et al., 2020b)

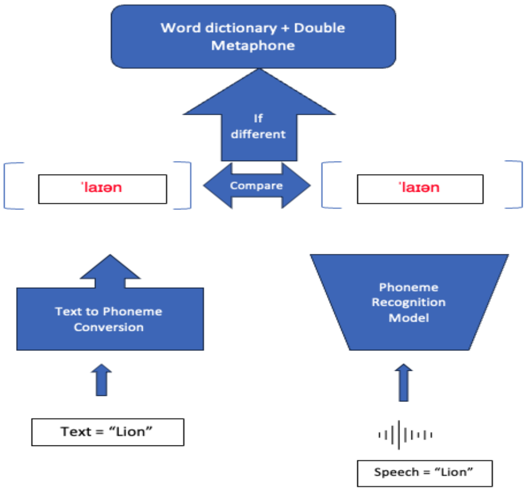

Word dictionary + Double Metaphone
If different
ˈlaɪən
Compare
ˈlaɪən
Text to Phoneme Conversion
Phoneme Recognition Model
Text = "Lion"
Speech = "Lion"

Figure 4: CAPT Architecture

method is particularly useful for training neural networks to solve problems where we want to transform one sequence into another sequence, which often happens in tasks like turning spoken words into written text (Automatic Speech Recognition) or interpreting handwritten text. We have used a pretrained XLSR-Wav2Vec2 model that is finetuned on the TIMIT dataset using CTC loss. (Phy, 2022)

3.6 Computer-Aided Pronunciation Training

The developed CAPT module is a comprehensive tool for improving pronunciation accuracy, combining Speech-to-Phoneme and Text-to-Phoneme conversion models. Users are presented with a word to pronounce, and their spoken and text inputs are converted into phonetic representations for comparison. Discrepancies are detected, and the Double Metaphone algorithm and NLTK word dictionary are utilized to generate lists of similar-sounding words for practice. This real-time feedback and personalized practice approach empower language learners to refine their pronunciation skills effectively and enhance their overall communication abilities. The CAPT module holds promise as a valuable resource in language learning and proficiency assessment contexts, contributing to more confident and proficient language speakers.

4 Experiments

We performed the task of Phoneme Recognition on Darpa TIMIT Dataset (Consortium, 1990). We explored simple DNN, RNN, CNN+RNN, pretrained XLSR-Wav2Vec2 model improved using CTC and finetuned on TIMIT dataset.

4.1 TIMIT Dataset

The TIMIT corpus, comprising read speech samples, was designed to facilitate the study of acoustic-phonetic properties and aid in developing and evaluating automatic speech recognition systems. This project is the result of collaborative efforts from the Massachusetts Institute of Technology (MIT), Stanford Research Institute (SRI), and Texas Instruments (TI), sponsored by the Defense Advanced Research Projects Agency - Information Science and Technology Office (DARPA-ISTO). The TIMIT corpus includes a total of 6300 sentences, each of the 630 speakers contributing ten sentences, representing eight primary dialect regions across the United States. The speaker's dialect region is identified by where they spent their childhood, aligning with recognized U.S. dialect regions, except for the Western region and a special category for those who moved frequently in their early years.

4.2 Evaluation Metrics

A phoneme recognition system tries to predict the phonemes occurred in an audio signal, hence all the evaluation metrics mentioned would focus on measuring the performance on the basis of phonemes. Here are the evaluation metrics that we had focused on:

1. **Phoneme Error Rate (PER)**: This is the most widely applied measurement in the field of phoneme recognition. PER is computed by

adding up all the replacement, removal, and addition mistakes, dividing this sum by the total phonemes in the reference, and then multiplying the result by 100. While it shares similarities with the Word Error Rate (WER), PER operates specifically at the level of phonemes.

2. **Accuracy**: This represents the proportion of correctly predicted phonemes out of all phoneme predictions made. But in situations where data is unevenly distributed, as is often the case in speech recognition tasks where certain phonemes occur more than others, accuracy may not serve as the most effective assessment measure.

3. **Phoneme Recognition Accuracy (PR-Accuracy)**: Since the generic accuracy metric does not work well with speech-recognition related tasks. We used a variation of accuracy as used by (Leung et al., 2019b).

$$\text{Accuracy} = \frac{N - S - D - I}{N} \qquad (1)$$

In this equation, N stands for the total count of labelled phonemes, while S, D, and I are the counts of substitution, deletion, and insertion errors, respectively.

4.3 Results and Analysis

We trained our DNN, RNN and CNN+RNN models from scratch for the task of Phoneme Recognition and then compared it against a pre-trained XLSR-Wav2Vec2 model improved using CTC and fine-tuned on the TIMIT dataset. We will assess both qualitative and quantitative results. For the task of Computer-Aided Pronunciation Training (CAPT), we will share our qualitative feedback about the uses and challenges of Phoneme Recognition models for mispronunciation detection.

4.3.1 Phoneme Recognition

Quantitative Analysis: The average training times including PER calculations for the DNN, RNN and CNN+RNN networks per epoch were 64.52 seconds, 972.27 seconds, and 1086.83 seconds and the generic accuracies of the models were 58.28%, 58.56%, 51.15% respectively. This increase in training times for more complex models is expected. The XLSR-Wav2Vec2 model had the generic accuracy of 62.56%. Since, the generic accuracy is not ideal for the category of speech recognition tasks we will focus more on PER and PR-Accuracy defined in the previous section.

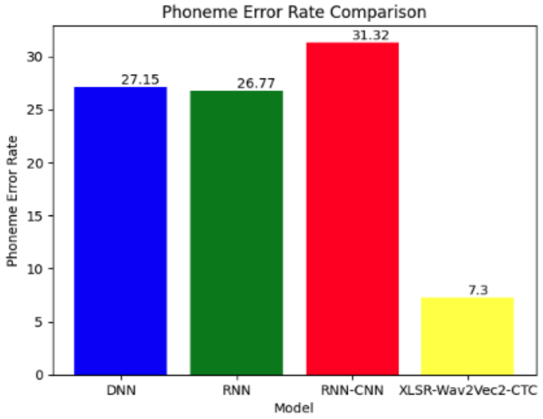


Figure 5: Phoneme Error Rate comparison for all networks

Phoneme recognition is an essential task in speech processing, and the models were evaluated based on Phoneme Recognition Accuracy (PR-Accuracy) and Phoneme Error Rate. The results indicate that the XLSR-Wav2Vec2-CTC model outperformed all other models with a Phoneme Recognition Accuracy of 92.62% and a remarkably low Phoneme Error Rate of 7.3%. The XLSR-Wav2Vec2-CTC model's success can be attributed to its pre-training on a large-scale speech corpus, CTC loss optimization, and advanced transformer-based architecture.

Among the other models, the RNN and DNN models showed comparable performance, both achieving a Phoneme Recognition Accuracy of around 60% and a Phoneme Error Rate of approximately 28%. These models demonstrated better results than the initial CNN-RNN model, which had a Phoneme Recognition Accuracy of 52.50% and a Phoneme Error Rate of 33.08%. The CNN-RNN model's relatively poor performance in phoneme recognition can be attributed to its limitations in capturing long-range dependencies and handling complex phonetic patterns. As a model combining Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), CNN-RNN struggles to understand the broader context and temporal dependencies present in sequential data, impacting its ability to correctly identify phonemes that rely on contextual information over longer time spans. Additionally, the absence of the Connectionist Temporal Classification (CTC) loss, used in the superior XLSR-Wav2Vec2-CTC model, hindered

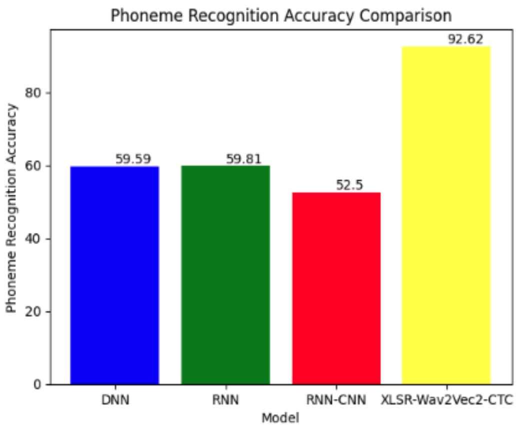


Figure 6: PR-Accuracy comparison for all networks

the handling of variable-length phoneme sequences, leading to higher Phoneme Error Rates in CNN-RNN, RNN, and DNN models.

On the other hand, the DNN with simple dense layers outperformed the CNN+RNN architecture in phoneme recognition possibly due to its better suitability for sequential data processing. Phoneme recognition involves sequential data, where the order of phonemes matters. While CNNs are designed for image data and excel at extracting local features, they may struggle with temporal dependencies, making the simple dense layers of DNNs more appropriate for phoneme recognition. The CNN+RNN architecture's complexity, potential overfitting, and difficulties in learning phoneme representations due to the mismatch between CNN and sequential data contributed to its inferior performance.

Qualitative Analysis: We focus on the best performing model, XLSR-Wav2Vec2-CTC as we will be using this for the CAPT task. We can see in Figure 7 that most of the errors occurred because of replacement of one phoneme with another. As evident in Figure 8 the model's frequent confusion between phonemes with similar acoustic properties, such as 'ih' and 'ah', 'iy' and 'ih', and 'ah' and 'ih', suggests that it might struggle to distinguish subtle variations in vowel sounds. This highlights a common challenge in phoneme recognition, where small differences in pronunciation can lead to misclassifications. Additionally, the model's difficulties with rhotic sounds, evident in the replacements of 'er' with 'ah', and its occasional confusion between nasal consonants like 'n' and 'ng', reflect its limitations in capturing specific phonetic characteristics that vary across different regional accents.

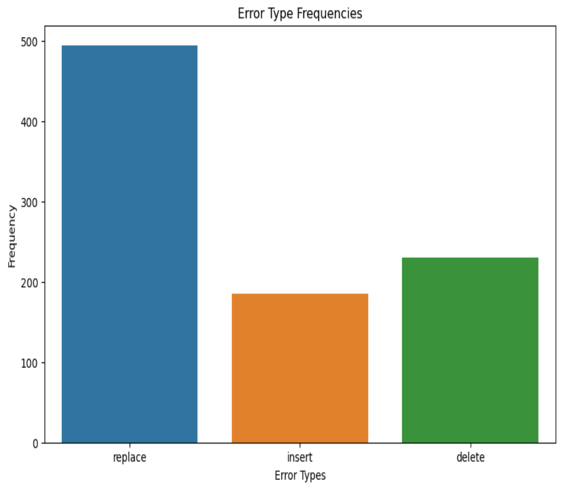


Figure 7: Classification of error types of XLSR-Wav2Vec2-CTC model

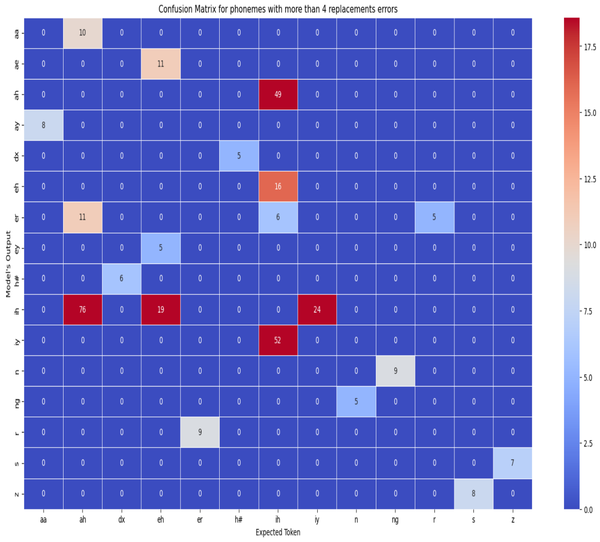


Figure 8: Confusion Matrix for phonemes with more than 4 replacements errors

The model's tendency to replace 'z' with 's' indicates challenges in recognizing voiced and voiceless fricatives, which can pose difficulties in distinguishing between certain consonant sounds. Overall, the qualitative analysis underscores the need for further refinement in the model's representation learning and its ability to handle intricate acoustic features of phonemes. Addressing these challenges could lead to improved phoneme recognition accuracy, and additional training data or techniques to reduce biases towards specific phonemes might further enhance the model's performance.

Additionally, the presence of the "h#" symbol as a sentence boundary marker is leading to most of the insertions and deletions type of errors dur-

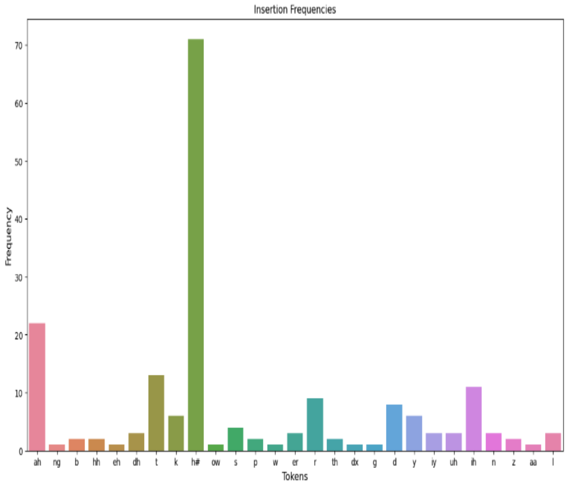
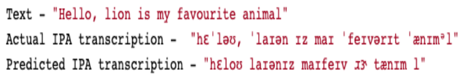


Figure 9: Insertion error frequencies across phonemes



```
Text - "Hello, lion is my favourite animal"
Actual IPA transcription -  "hɛˈləʊ, ˈlaɪən ɪz maɪ ˈfeɪvərɪt ˈænɪməl"
Predicted IPA transcription - "hɛloʊ laɪənɪz maɪfeɪv ɪɝ tænɪm l"
```

Figure 10: Example use of XLSR-Wav2Vec2-CTC model for CAPT task

ing phoneme recognition. We can see this in Figure 9 for insertion error frequencies, and we observe a similar trend for deletion errors. The model may mistakenly predict additional phonemes after the "h#" symbol or skip certain phonemes following it, resulting in misalignments and affecting the accuracy of the entire sentence. To address this issue, preprocessing the input data to remove or mask "h#" symbols, modifying the decoding process to handle sentence boundaries explicitly, and fine-tuning the model on datasets with sentence boundary markers can help mitigate these errors and improve the overall performance of the phoneme recognition system.

4.3.2 Computer-Aided Pronunciation Training

We used the IPA format of the model's phoneme outputs for easier character comparisons during experiments with XLSR-Wav2Vec2-CTC for phoneme recognition. The model performed well overall, accurately detecting most phonemes within sentences. However, a major challenge arose from the "h#" issue discussed earlier. The model struggles to identify breaks between sentences or words, often merging phonemes from different words. Figure 10 illustrates this, with examples like "lion is" and "my favourite animal" showing merged phonemes and unexpected word breaks, despite mostly correct phoneme detection.

When applied to multiple words or sentences, the inference time per word is approximately 0.38 seconds, ensuring efficient processing and quick predictions. However, for one-word recordings as in our case, the inference time per word increases to around 1 second, likely due to the model's architecture and the computational intensity of attention mechanisms when handling smaller input sizes. Acceptability of a tool is highly dependent on the time is takes to deliver results.

5 Conclusion

The progress in Phoneme Recognition, with the help of transformer models, brings hope for better Computer-Assisted Pronunciation Training (CAPT) tools. This report discussed the challenges of predicting phonemes in sentences, considering certain limitations. One significant challenge was distinguishing phonemes between different words. Considering this for now we developed a tool that focuses on training pronunciation for one word at a time. In future, to improve our model's performance, we plan to tackle issues like merging phonemes from different words and placing word breaks correctly. Additionally, we aim to use Generative AI techniques for suggesting practice phrases that include words with similar sounds, going beyond simpler methods like Double Metaphone. Including phrases or poems in pronunciation training will enhance speech flow and offer a more engaging learning experience for users. These advancements hold the promise of empowering learners to improve their pronunciation and fluency through CAPT tools.

References

Association. 2005. T.i.p. reproduction of the international phonetic alphabet.

Alexei Baevski, Michael Auli, and Alex Conneau. 2020a. Wav2vec 2.0: Learning the structure of speech from raw audio.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020b. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Linguistic Data Consortium. 1990. Table of all the phonemic and phonetic symbols used in the timit lexicon.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. 1993. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, 93:27403.

Ben Gold, Nelson Morgan, and Dan Ellis. 2011. *Speech and audio signal processing: processing and perception of speech and music*. John Wiley & Sons.

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

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

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee.

Alissa M Harrison, Wai-Kit Lo, Xiao-jun Qian, and Helen Meng. 2009. Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training. In *International Workshop on Speech and Language Technology in Education*.

Aldebaro Klautau. 2001. Arpabet and the timit alphabet. *an archived file. https://web. archive. org/web/20160603180727/http://www. laps. ufpa. br/aldebaro/papers/ak_arpabet01. pdf (Accessed Mar. 12, 2020)*.

K.-F. Lee and H.-W. Hon. 1989. Speaker-independent phone recognition using hidden markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(11):1641–1648.

Wai-Kim Leung, Xunying Liu, and Helen Meng. 2019a. Cnn-rnn-ctc based end-to-end mispronunciation detection and diagnosis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8132–8136. IEEE.

Wai-Kim Leung, Xunying Liu, and Helen Meng. 2019b. Cnn-rnn-ctc based end-to-end mispronunciation detection and diagnosis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8132–8136. IEEE.

Pengfei Liu, Ka-Wa Yuen, Wai-Kim Leung, and Helen Meng. 2012. menunciate: Development of a computer-aided pronunciation training system on a cross-platform framework for mobile, speech-enabled application development. In *2012 8th International Symposium on Chinese Spoken Language Processing*, pages 170–173. IEEE.

Wai-Kit Lo, Shuang Zhang, and Helen Meng. 2010. 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*.

Donghoon Oh, Jeong-Sik Park, Ji-Hwan Kim, and Gil-Jin Jang. 2021. Hierarchical phoneme classification for improved speech recognition. *Applied Sciences*, 11(1):428.

Lawrence Philips. 2000. The double metaphone search algorithm. *C/C++ users journal*, 18(6):38–43.

Vitou Phy. 2022. Automatic Phoneme Recognition on TIMIT Dataset with Wav2Vec 2.0. If you use this model, please cite it using these metadata.

S Qamar, Harsh Dev, Nidhi Srivastava, et al. 2013. Speech recognition using mfcc and neural networks. *Corpus ID: 17131143*.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Petr Schwarz, Pavel Matějka, and Jan Černockỳ. 2004. Towards lower error rates in phoneme recognition. In *International Conference on Text, Speech and Dialogue*, pages 465–472. Springer.

Miikka Silfverberg, Lingshuang Jack Mao, and Mans Hulden. 2018. Sound analogies with phoneme embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 136–144.

Qiantong Xu, Alexei Baevski, and Michael Auli. 2021. Simple and effective zero-shot cross-lingual phoneme recognition. *arXiv preprint arXiv:2109.11680*.

Wu Ying. 2019. English pronunciation recognition and detection based on hmm-dnn. In *2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pages 648–652.

Ka-Wa Yuen, Wai-Kim Leung, Peng-fei Liu, Ka-Ho Wong, Xiao-jun Qian, Wai-Kit Lo, and Helen Meng. 2011. Enunciate: An internet-accessible computer-aided pronunciation training system and related user evaluations. In *2011 International Conference on Speech Database and Assessments (Oriental COCOSDA)*, pages 85–90. IEEE.