Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University

Abylay Omar

Pronounciation analysis and mispronounciation detection

THESIS

Presented in Partial Fulfillment for the
Degree of Master of Science in Computer Science
(degree code: 7M06102)
Department of Computer Science
Faculty of Engineering and Natural Sciences

Supervisor: **Larissa Kiziyeva**

Kaskelen, 2022

Suleyman Demirel University

Faculty of Engineering and Natural Sciences

Department of Computer Science

Dean of Faculty

Associate Professor, PhD Zhamanov A.

«20» 06 2022

Topic of the thesis:

Pronunciation analysis and mispronunciation detection

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science", SDU, 2020-2022

Head of Department	Assoc. Prof. PhD. Cemil Turan
Academic Supervisor	Larissa Kiziyeva
Master's student	Omar Abylay

Kaskelen, 2022

Abstract

We are interested in developing software that can automatically recognize particular phone segments that a non-native student of a foreign language has pronounced incorrectly. A language training system can provide the student with feedback about individual pronunciation errors by using the information about the phone level. For this purpose, in this work, I am trying to develop a transformer model that will recognize spell errors in speech. There were two strategies that were examined: the first one on the original audio dataset, and the second one on the synthetically augmented dataset. Both experiments were compared in this work.

Аңдатпа

Бұл жұмыста ағылшын тілін үйреніп жүрген адам, ағылшын тілінде қате айтылған сегменттерді автоматты түрде тани алатын бағдарламалық

жасақтаманы әзірлеуге мүдделіміз. Тіл үйрету жүйесі буын деңгейінде ақпаратты пайдалана отырып, студентке жеке айтылу қателері туралы кері байланыс бере алады. Осы мақсатта мен бұл жұмыста сөйлеудегі емле қателерін танитын трансформен үлгісі негізінде жаңа модель жасауға тырыстым. Қаралға екі стратегия болды: біріншісі бастапқы аудио деректер жинағында, екіншісі синтетикалық түрде толықтырылған деректер жиынында. Бұл жұмыста екі тәжірибе де салыстырылды.

Аннотация

Мы заинтересованы в разработке программного обеспечения, которое может автоматически распознавать определенные звуковые сегменты,

которые неправильно произносит не-носитель иностранного языка. Система языковой подготовки может предоставить учащемуся обратную связь об

отдельных ошибках произношения, используя информацию об уровне

телефона. Для этого в данной работе я пытаюсь разработать модель преобразователя, которая будет распознавать звуковые ошибки произношения в речи. Были рассмотрены две стратегии: первая — на исходном наборе аудиоданных, а вторая — на синтетически дополненном наборе данных. В данной работе сравнивались обе модели. И практическим методом было доказано что второй эксперимент дает лучшую точность.

Contents

1 Introduction **6**

1.1 Motivation 9

1.1.1 Assumptions 10

2 Background **12**

2.0.1 Pronunciation and Language Learning 12

2.0.2 Computer-Assisted Pronunciation Training (CAPT) . . . 13

2.0.3 Individual Error Detection 13

2.0.4 Posteriorgram-based Pattern Matching in Speech 14

2.0.5 Application in Speech 15

2.0.6 Application in Keyword Spotting and Spoken Term Discovery 15

3 Literature review **17**

3.0.1 Mispronunciation detection and diagnosis 19

3.0.2 CNN-RNN-CTC models 22

3.0.3 Probability based method 24

3.0.4 DNN based method 25

3.0.5 Classifier 26

4 Dataset **29**

4.0.1 Metrics for standardized assessment of phone recognition 30

4.0.2 A summary of recent and previous research on the TIMIT phone recognition challenge 31

4.0.3 Features 34

5 Transformers **38**
5.0.1 Transformer Architectures 40
5.0.2 Multi-Headed Attention 41
5.0.3 Feed Forward Architecture 43
5.0.4 Adapting Transformer Architecture for raw waveforms . 43
5.0.5 Transformer Architectures: Pooling 44
5.0.6 Multi-scale embeddings 45

6 Experiments and Results **47**
6.1 Data 47
6.1.1 Synthetic augmentation 47
6.2 Experiment 48
6.2.1 Data preprocessing 50
6.2.2 Train and test 52
6.2.3 Evaluation 52
6.2.4 Results 52

7 Summary and Feature work **54**
7.1 Feature Work 54
7.1.1 Complete E2E model 54
7.1.2 Application to Other Languages 54
7.1.3 Implementation in web or mobile applications 55

References **56**

1. Introduction

English is the language of inter ethnic communication in a certain area of human life, so called lingo-franc - the "Frankish language" of the modern world. It is the first foreign language that is an official or working language in education, government and the media. Therefore, mastery communicative competence in English has paramount importance. Learning any foreign language, especially its pronunciation, includes the correct perception and articulation of the sounds of the language being studied. The process of learning a foreign language is subject to the influence of the established laws of the native language, both in perception and in the production of sounds. This inter lingual influence is called language transfer or interference. Negative transfer from native language in the studied is the cause of inaccuracies and errors in pronunciation and, as a result, difficulties in understanding. Studies of the process of the emergence of speech in the study foreign languages are of great interest among phoneticians, linguists, language teachers, as well as programmers who are involved in the development of computer-based learning systems pronunciation and intonation of foreign languages. The developed computer programs can complement the learning process in the classroom and provide unique opportunities on the way to mastering a non-native language: overcoming the language barriers, reduced fear of language, accessibility and an individualized learning program. Computer systems for teaching pronunciation are essential for students, as they provide them with a detailed feedback report indicating incorrect pronunciation, diagnosis and opportunities for correction, as well as improving pronunciation. A distinctive feature of computer systems for teaching pronunciation and intonation is the automatic processing of spoken speech and the corresponding adjustment for word or sentence level, which gives more accurate calculations (as opposed to human expert) to improve pronunciation student. Definition of incorrect pronunciation on phonetic level

allows you to maintain statistically important improve the pronunciation of existing sound images. The diagnostics provided in the reverse report are significantly improves pronunciation in the learning process.

Today, mobile technologies provide opportunities to improve almost any speech skills - speaking and pronunciation skills are no exception. Automated pronunciation learning systems ("Computer aided pronunciation training (CAPT)) are used by non-native English speakers to improve their pronunciation. The CAPT tool records the student's speech, identifies and diagnoses errors in it, and suggests a way to correct them. It is noted that CAPT systems can detect up to 86% of incorrect pronunciations in speech and help students reduce the likelihood of mispronounced speech by 23%

Teaching pronunciation is a complex process, both for students, as well as for computing. In this case, it is necessary to take into account the concept of interference, i.e. the influence of the native language on learning pronunciation. It is not enough for a student to simply know and be able to correctly articulate sounds - it is necessary to do this at the level of reflexes. And in order to bring these skills to automaticity, constant pronunciation practice is necessary, it is obvious that not everyone can get this practice with a native speaker or with a teacher, for this and systems for teaching pronunciation have been established. Computers have been used to teach pronunciation since the middle of the last century. The very first systems are usually classified as early systems. For the most part, they provided reading and listening material, apparently without any post-processing or feedback. Modern systems are obviously more technologically advanced and offer more options. One such possibility is to provide feedback in various forms, as well as to evaluate results of students' work. Thus, we can highlight the main requirements for such systems:

1. Detection of deviations in pronunciation;
2. Classification of these deviations;
3. Illustration of the difference between correct and incorrect pronunciation;
4. Suitable specifications;
5. Ability to act as a simulator;

6. A clear evaluation mechanism.

To date, there are a number of platforms whose tools to some extent meet the above requirements - "English Practice", "Rosetta Stone", "Lingualeo", "EF". A lot type of simulator works as follows: the student is asked to pronounce a phrase from the section just studied. If the student is not sure how to pronounce the phrase correctly, he can use the hint and listen to the correct option. After that, the student presses the record button and pronounces phrases. The program analyzes and issues result. It should also be noted that the system does not always clearly and accurately capture inaccuracies, or vice versa, it can give out the pronunciation that is correct at first glance for being incorrect. All influencing factors are presented in the pros and cons section of pronunciation improvement systems. This exercise is not the only simulator for improve pronunciation in the app. There is also a dialogue simulation exercise that also trains listening skills. The sequence of operation of such systems can be described as follows:

1. speech signal processing - identification and elimination of interference and distortion;

2. then the pure signal enters the acoustic adaptation module, which controls the module for calculating the speech parameters necessary for recognition

3. sections containing speech are distinguished in the signal, and assessment of speech parameters. There is a selection of phonetic and prosodic probabilistic characteristics for syntactic, semantic and pragmatic analysis;

4. hen the speech parameters enter the main block recognition system - the decoder, which compares the input speech signal with acoustic and language models

This is how almost any speech recognition system works, however, pronunciation improvement systems require at least one more block - a block for comparing the reference pronunciation with what the student says. This block is a software component after the output result is determined by the decoder. One of the main advantages of pronunciation learning systems is the possibility of learning the basics of pronunciation of the language without a teacher or native speaker. The system issues a detailed report and diagnoses incorrect pronunciation. The

main problem of this kind of systems is the multivariance pronunciation of the same thing, it may not necessarily be a wrong pronunciation, it may be a fast pace of speech, too quiet speech and so on. In addition, the input signal is influenced by numerous factors, such as noise, echo and channel interference. As mentioned above, the first block of the speech recognition system is the noise filter. This block is software-implemented and is hardly capable of suppressing high-level noise. Thus, there is a need for equipment (microphones, sound cards) capable of compensating for the imperfection of the digital filter. The next problem may sound paradoxical, but it exists - too good acoustic and language models. What's in could this be bad? Many models "guess" what is said word for those phonemes that were pronounced clearly or guessed the next word according to the words spoken before. Of course, this is a huge plus in smart home systems or navigators, but in systems improving pronunciation, this does a disservice to students. In conclusion, we can say about the prospects for the development of systems improve pronunciation. First of all, this is the use of neural networks ("END2END" technology). For example, for training acoustic models. Neural networks are free from many of the limitations of Gaussian mixtures and have a better generalization ability. In addition, acoustic models based on neural networks are more noise-resistant and have better performance. Therefore, close collaboration between language teachers and developers is recommended. software for the development of CAPT tools, their wide distribution and integration with curricula on school and university level, as well as further exploration of mobile and collaborative CAPT systems.

1.1 Motivation

In order to provide an accurate automated assessment of a student's speech, the student's voice must be analyzed to past approaches, which may include one maybe more native speakers of the language being studied. The conventional approach of mispronunciation detection is dependent on precise automatic speech recognition (ASR) algorithms, which will be discussed in further depth in the following chapter. However, there are a number of limitations associated with a recognizer-based technique. To begin, a sizeable amount of the training data has to be manually translated at the phone layer in the language that will ultimately

be used, which is a process that is both moment and expensive. A CAPT system often needs an acoustic model is trained on speaking speech, which necessitates well-labeled nonnative data for training. This is because native and nonnative speaking differ in a number of ways.

Because of these two constraints, recognizer-based algorithms are language-dependent. This is in addition to the fact that preparing training data requires a significant amount of human labeling labor. A new recognizer has to be constructed each whenever a CALL system is implemented for a diverse target language. Even worse, a non-native acoustic model needs to take into consideration the many different native languages spoken by the students. There are 6,909 different languages in the world, according to the list that is now considered to be the most complete of its kind. However, only about 50 of a world's most frequent languages are supported by recognizers that are available for purchase commercially. As a consequence of this, it is of the utmost need to do research on a CAPT construction technique that does not rely on recognizers and is independent of language.

A comparison-based strategy is what we employ instead of a recognizer because of this. When a student's voice is "close" to a teacher, then the student should be doing well, according to this commonsense assumption. We match two statements using system's dynamic warping (DTW), which was inspired by the unconstrained pattern matching approaches that will be discussed in the following chapter. Our goal is to assess if the student's utterance is similar to the one spoken by the teacher. Therefore, we could find mispronunciations if we look for instances where there is just a partial connection. Our approach does not need language skills of either the target language or the student's mother tongue; hence, it ought to be language-independent. In addition, we suggest the use of an autonomous phone segmentor so that each word may be broken down into smaller parts for a more detailed examination. This will allow us to get around the problem of unnecessary human labeling effort.

1.1.1 Assumptions

Visualize the incorporation of our architecture into the a CALL system by seeing it as a code system, which could be something like a read game or a conversation led by an instructor. In the presence of a sentence that is written out and displayed

on the screen, the first assumption that may be made is that the student is striving to correctly pronounce words. If a student does not learn to read the sentence that has been provided, they can press the play button, which will then provide them with a read sample from a "teacher." If a student does not already learn to read a particular word, they have the option of merely listening to the word's pronunciation instead. Our second premise states that for every code in our teaching materials, there is at least one written record from a native speaker of a target language, and also that the recording includes word-level timing labels. These features are based on this premise, that also states that there is at least one written record from a native speaker of a target language for each script.

Given that we are now able to instantly access a substantial amount of audio data through the Internet, we believe that the assumptions made in this article are feasible. The content includes a wide variety of things including public speeches, television shows, and news broadcasts; all of them provide sufficient audio sources for language education. In contrast to comments made at the phonetic level, which are typically carried out by professionals, the ability to annotate word boundaries is possessed from every native speaker of a target language.

2. Background

2.0.1 Pronunciation and Language Learning

When learning a target language, non-native speakers, particularly adults, are readily influenced by their native language (L1) (L2). Pronunciation encompasses several aspects. It is recognized that a language learner's mistake patterns correspond with his or her degree of competence. As a learner begins to learn a new language, phonetic mistakes, such as the replacement, insertion, or deletion of one or more phones, are the most frequent. These mistakes are attributable to a lack of familiarity with L2 phoneme inventory and phonological norms. The learner may replace an unknown phoneme with one that appears in L1 and is comparable. Japanese learners of English at the beginner level often substitute /s/ for /th/ and /I/ for /r/ . Due to a lack of vocabulary, the learner may not know the right rules for pronouncing a word when encountering it for the first time. For instance, the sound of an English vowel varies depending on its context. As the proficiency of the learner increases, these types of faults may occur less often, and prosody may become a major concern. Lexical stress, tone, and temporal length are examples of prosodic categories. Previous research has shown that prosody has a greater effect on the intelligibility of language learners than phonetic elements . Nevertheless, prosodic elements of a language are sometimes concealed by details. Learning these specifics requires accurate perception of the target language. However, a student's Li may restrict his/her capacity to cognizant of certain prosodic aspects of L2 For instance, for many language learners whose home languages are non-tonal, it is difficult to perceive the tones in tonal languages such as Mandarin Chinese or Cantonese, much alone produce them . When creating a CALL system, individuals are more concerned with the system's accuracy than its recall, since the student would be dissuaded from learning if the system detected numerous faults that were really correct pronunciations . In

addition to accurately recognizing the aforementioned faults, the system's feedback is also crucial. Multimodal feedback, such as text messages, audio playback, and animation of the lips or vocal tract, is popular because it may increase the system's comprehensibility and hence the learning efficiency.

2.0.2 Computer-Assisted Pronunciation Training (CAPT)

CAPT systems are a kind of CALL system that are created specifically for pronunciation instruction. Individual mistake detection and pronunciation evaluation are the two forms of evaluation. The former involves identifying pronunciation problems at the word or subword level and delivering feedback, while the latter measures a student's overall fluency. As word-level mispronunciation detection is the emphasis of this thesis, we exclusively provide past work in this field.

2.0.3 Individual Error Detection

There are many different applications for the ASR technology that may be found in CAPT. The speech training system that was created by Kewley-Port and colleagues use a person speaking, isolated-word, template-based recognizer. It was designed specifically for use with youngsters. The spectra of a child's input sentence is recorded as a series of 16-bit binary vectors, and then it is compared to the templates that have been saved by calculating the fraction of matching bits in relation to the number of bits included in each template. The value of this number is the score that is used to determine the quality of the articulation. For the creation of the CAPT system, which is used for learning German, Wohlert made use of a voice recognizer that is based on templates. In addition, Dalby and Kewley-Port evaluated two distinct types of recognizers, namely, a template-based method and a neural network, both of which match patterns, and also an HMM-based system that is based on randomly generated stochastic modeling. Both of these recognizers were successful in matching the patterns. An HMM-based identification system outdoes a template-based recognizer when it comes to accuracy in sensing words, but a framework recognizer outperforms a pattern recognizer when it comes to discriminating among minimum pairs. This is the conclusion that they came to as a result of their research.

Witt and Young developed a goodness of pronunciation (GOP) score, which

may be defined as the duration normalized log of a speaker's posterior probability of speaking a phone given acoustic data. Phone-dependent criteria are established to determine whether or not each phone in the forced alignment is mispronounced. Their ten participants spoke a range of Lis, including Latin-American Spanish, Italian, Japanese, and Korean. Their experimental findings demonstrate that pronunciation scoring based on probability ratings from a recognizer may obtain acceptable results. Franco et al trained three recognizers using data of varying degrees of nativeness and utilized the proportion of the log-posterior probability-based values from each recognizer to identify mispronunciation. In recent work, Peabody advocated anchoring nonnative speakers' vowel space to that of native speakers while generating probability ratings.

Some techniques took the knowledge of the students' Li into account. They concentrated on forecasting a set of probable mistakes and improving the recognizer's ability to recognize incorrect pronunciations. These mistake patterns may be handcoded using language understanding or learnt using data. Based on the notion of language transmission, Meng et al presented an enlarged pronunciation lexicon that contains probable phonetic confusions. By thoroughly comparing phonological between the two languages, potential phonetic confusions were anticipated. They conducted a detailed examination of the most common pronunciation errors that Cantonese speakers make while learning English. Kim et al. have meticulously created phonological principles that account for Korean (Li) impact on English (L2). Harrison et al. investigated context-sensitive phonological rules as opposed to context-insensitive rules. Qian et al. used the enlarged pronunciation lexicon to develop a discriminatively-trained audio model that reduces both mispronunciation and diagnostic mistakes to improve the recognizer. Wang and Lee have extended the integration of GOP scores using error pattern detectors. Their experimental findings reveal that the holistic design outperforms employing just one of them in identifying mispronunciation in a class of students studying Mandarin Chinese from 36 different nations.

2.0.4 Posteriorgram-based Pattern Matching in Speech

Dynamic time warping (DTW) is a method that determines the best match between two sequences that differ in time or speed. It was crucial in the early days of template-based speech recognition. Rabiner and Juang's book has a compre-

hensive treatment of problems as to what the distortions measurements should be and how to establish timenormalization limits. Early studies suggested that the distortion metric be based on filter bank output or LPC characteristics . Recently, posterior characteristics have been effectively used to voice recognition, as well as to assist unsupervised spoken keyword discovery. Below, we explain what a posteriorgram is and look at some past work on pattern matching applications that use posteriorgrams. A posteriorgram is a vector with posterior probabilities across a set of preset classes that may be thought of as a small subset of speech. Given a speech frame, the phonological posteriorgram is a N x 1 vector, with each member representing the prior distribution of the relevant phonetic class given that speech frame. Posteriorgrams may be calculated directly using likelihood scores for each phonetic class, or through executing a phonetic recognizer to build a decoding lattice .

2.0.5 Application in Speech

Aradilla et al. re-investigated the issue of pattern speech recognition utilizing posterior-based features as templates and found that they performed much better than typical template-based recognition systems on a continous digit identification challenge. They employed a multi-layer number of neurons to estimate phone posterior probability based on spectral-based information, and the distance metric was KL-divergence.

2.0.6 Application in Keyword Spotting and Spoken Term Discovery

Speech phrase detection with spoken voice files seeks to scour the audio information for a given phrase in sound without a speech recognizer, whereas spoken term discovery strives to pinpoint common themes in audio data, which could be a term, a word, or a sentence. Spoken phrase sensing for spoken voice files seeks to browse the audio information for a given phrase in sound without a speech recognizer. In their initial attempt, Hazen et al. intended to develop a spoken word recognition system that was based on phonetic-posteriorgrams. Phonetic recognizers have the ability to decode a word or phrase into a collection of phonetic posteriorgrams. Following the execution of dynamic temporal warping between

both the posteriorgram representation of the phrase as well as the posteriorgram of an existing utterance, the alignment score was utilized in order to evaluate the results of the detection process.

The process of training a phonological recognizer involves the use of a set of train data that contains phone-level labels. Zhang et al. looked at an unsupervised method for recognizing spoken questions by decoding posteriorgrams generated by a GMM trained on labeled speech. Their later research shown that posteriorgrams that are processed using Deep Boltzmann Machines have the potential to boost performance of the system even more. In addition to comparing the matching scores, Muscariello et al. also investigated several image processing methods for contrasting the self-similarity matrices of two different phrases (SSMs). The combination of DTW-based with SSM-based scores has the potential to improve the quality of spoken phrase recognition.

3. Literature review

This work [9] aims to build an end-to-end voice recognition system for the Mispronunciation Detection and Diagnosis (MDD) issue by utilizing Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Connectionist Temporal Classification. Specifically, the MDD problem is related to mispronunciation detection and diagnosis (CTC). Our approach makes use of end-to-end models, and it is not essential to provide phonemic or graphemic data or to align discrete language units in a predetermined manner in order for it to be successful. Experiments are carried out in order to evaluate the effectiveness of the proposed CNN-RNNCTC method in comparison to previous mispronunciation detection and diagnostic (MDD) methods. Our approach has an F-measure that is 74.65 percent, which dominates both the Extension Recognition Network (ERN) (S-AM) and the State-level Acoustics Model (S-AM) by a significant margin of 44.75 percent and 32.28 percent, respectively. The relative advancement in F-measure is 9.57 percent for the Acoustic-Phonemic Model (APM), 5.04 percent for the Input Noise (AGM), and 2.77 percent for the Acoustic-Phonemic-Graphemic Model (APGM) when compared to the Acoustic-Graphemic Model (AGM), Acoustic-Phonemic Model (APM), and Acoustic-Graphemic-Graphemic Model (APGM), respectively.

Language transfer of characteristics from the main language can be responsible for some of the mispronunciations that occur in non-native products. Phonological rules are developed in order to represent the patterns of incorrect pronunciation that are created by learners. In classical speech recognizers, the phonetic rules are applied to the Extended Recognition Network (ERN) in order to provide the capacity of MDD [9]. In order to better assist university students, ERN collaborated with CAPT to establish an online testing platform. However, ERN is unable to provide a guarantee that all possible mispronunciations made by

language learners are handled. When the identification network gets excessively bushy and covers an excessive number of mispronunciations, its acoustic model may not even be able to provide sufficient discrimination among the various alternative pronunciations, which may result in a reduction in the detection performance. Free-phone recognition was developed as a solution to free up resources that were previously restricted by ERN. The development of the state-level acoustic model served as the basis for the MDD baseline method [10]. In addition, the Acoustic-Phonemic Model (APM), the Acoustic-Graphemic Model (AGM), and the Acoustic-Phonemic-Graphemic Model (APGM) were all developed with the purpose of improving performance in comparison to the baseline. The F-measure for MDD was raised from 51.55 percent (ERN (S-AM)) to 72.61 percent with the use of information on phonemes and graphemes (APGM). On the other hand, these methods all include some sort of forced alignment. The effectiveness of the MDD would be determined primarily by the precision with which the force-alignment was performed. This motivates us to study the strategy for provide such recognition in MDD that does not need force-alignment as a prerequisite.

In training the Recurrent Neural Network (RNN) for labeling unsegmented sequences directly, connectionist temporal classifying (CTC) was developed. CTC has already been utilized in end-to-end speech recognition using models that convert acoustics to letters and convert acoustics to words [2]. The Convolutional Neural Network, or CNN, is another type of neural network that is commonly employed in image recognition tasks. CNNs have also been successfully used to a variety of ASR tasks [19]. Applying CNN to a speech recognition model has the potential to lower the error rate of the phone and potentially improve the model's performance in environments that are mismatched or loud.

This work provides a description of the CNN, RNN, and CTC strategy for constructing an end-to-end speech recognition method for the position of MDD. Because the system was developed in such a manner that it does not require any explicit combination of these two and graphemic data input, there is no requirement for forced alignment. This frees users from having to worry about whether or not their input is correct. Our technique does not require the presence of the any phonemic or graphemic data, nor does it require any kind of force realignment to take place. In addition, our method does not require any kind of force alignments to take place. The results of the experiment imply that our method

is effective significantly superior to other methods, including those which end up making the use phonemic and graphemic details input. This is demonstrated by a method consisting of 9.57 percent when compared to APM, 5.04 percent when compared to AGM, and 2.77 percent when compared to APGM. The findings of this research have the potential to provide the basis for an all-encompassing solution to the MDD problem. In the future, one of our priorities will be to work on integrating the language information so that we can contribute to improving performance.

In this work [18] proposed to carry out MDD using an innovative technique based on end-to-end automated speech recognition (E2E-based ASR). In particular, we expand the original L2 phone set with their corresponding anti-phone set. This gives the E2E-based MDD approach a better capability to take in both categorical and non-categorial mispronunciations, and our ultimate goal is to provide better mispronunciation detection and diagnosis feedback. In addition, a fresh transfer-learning paradigm has been developed in order to acquire the preliminary model estimate of the E2E-based MDD system without the utilization of any phonological rules as a resource.

Many experiments also on L2-ARCTIC dataset show that our fairest model outperforms both the current E2E baseline systems and the pronouncing scoring based method (GOP) in term of the F1-score, by margins of 11.05% and 277.1%, respectively, on the dataset.

3.0.1 Mispronunciation detection and diagnosis

The approaches to MDD (Mispronunciation detection and diagnosis) that have been developed up until this point may, for the most part, be divided into two types. The first type of method is one that is based on pronunciation scoring. This type of method computes phone-level pronunciation scores based on confidence measures that are derived from ASR. Some examples of these confidence measures include phone time duration, phone posterior probability scores, and segment duration scores. The most representative approaches in this category are the ones that use the goodness of pronunciation (GOP) test, which is derived from the log-likelihood ratio test, and its several iterations. On the other hand, these technologies can often only give the functionality of mispronunciation detection, but they are unable to provide the right diagnosis of mispronunciation.

The second group of approaches is designed to analyze the specifics of incorrect pronunciations by offering diagnostic and feedback for problems such as phone substitutions, deletions, and insertions , . The extend recognition network (ERN) method is a well-known example of a method that falls into this category. This method extends the decoding network of ASR with phonological rules, and as a result, it is able to readily provide diagnosis feedback based on comparison between an ASR output and the relevant text prompt. Nevertheless, on the one hand, it is difficult to enumerate and add adequate phonological rules into the decode network for all L1-L2 language combinations. This is due to the fact that enumerating phonological rules is a complex process. On the other side, the addition of an excessive number of phonological rules would result in a reduction in ASR accuracy, which would then lead to subpar performance in MDD. Compared to the GOP-based method that is built on the hybrid deep neural network-hidden Markov model (DNN-HMM) based acoustic model, the end-to-end (E2E) based ASR paradigm instantiated with gain insight classification (CTC) has also been introduced to MDD with promising results. This was done more recently. There has also been some follow-up work done utilizing other E2E-based strategies to solve the MDD problem , . This study is only one example among many others. However, the majority of the methods mentioned have concentrated solely on identifying categorical pronunciation errors (such as phoneme substitutions, insertions, or deletions), and they have paid less attention to identifying mispronunciations that belong to non-categorical or distortion errors , . Figure 1 is an illustration showing the MDD findings of a mispronounced utterance made by a speaker of English as a second language (L2 English), where the yellow blocks correspond to mispronounced elements of the speech. The word "The" should be pronounced with the phonetic sequence [dh iy], but a second-language speaker pronounced it with the sequence [d ah]. The phonetic sequences [dh][d] and [iy][ah] are both examples of categorical mistakes (viz., substitutions). In addition, the consonant in the word "He" should be sounded as [hh], but it is instead spoken as [hh*], which is a non-categorical speech mistake. The correct spelling of the consonant in the word "He" is [hh].

In light of this, we suggest attacking MDD using a highly specialized E2E-based ASR structural model, wherein the included E2E-based model is integrated with such a hybrid CTCAttention model. In addition, we believe that this ap-

proach will be the most effective. Because of the great degree of personal variability that is associated with MDD, it would be done taking into consideration that fact. Even though the attention-based model offers flexible gentle between the output label sequence and the input acoustic vector sequence without any Markov assumptions as CTC does, it is anticipated that the resulting composite model will be able to use CTC to aid the attention-based model in compensating for the imbalance problem and improving the speed of the decoding process. This is despite the fact that the attention-based model will produce the same results as CTC in terms of speeding up the decoding The CTC as well as the attention-based approach both have their merits, and by combining them, we can make this a reality.

In addition to this, we enlarge the initial L2 phone set by include the anti-phone set that corresponds to each phone. This makes it possible for the suggested E2E-based MDD strategy to have the capability to take in both category and non-categorial mispronunciations, which in turn enables us to deliver enhanced mispronunciation detection and diagnosis feedback.

In addition, a completely new domain adaptation paradigm has been designed in order to acquire the initial model estimation of the E2E-based ASR system without making use of any phonological rules as a resource. This was done in order to ensure that the system is as accurate as possible.

There has been a significant amount of study done on the detection of mispronunciation [3-9], and the approaches that have been developed may be divided into two types. The first of them is a scoring system for pronounciaton that is based on confidence metrics that were initially presented for automatic speech recognition (ASR). The approach and its major variations are now the most popular and provide a promising performance on mispronunciation identification. The goodness of pronunciation (GOP) scores are derived via log-posterior probability based on force alignment. However, a system of this sort is not only unable to cope with the insertion problems in pronunciation, but it also fails to provide learners a more complete diagnostic. The second category is designed to determine the nature of the incorrect pronunciation and offer instructive commentary on particular pronunciation mistakes. The extend recognition network (ERN) is a well-known approach that falls into this category. It incorporates predicted pronunciation mistake patterns into the lexicon in order to confine the recognition routes to the

canonical pronunciation and the likely phonetic mispronunciations. These ERNs that have been generated by handmade or data driven rules have the benefit that both the errors and the error kinds are recognized simultaneously, and as a result, they may be used by the system to offer diagnostic feedback. However, it is difficult to construct ERNs that integrate as many different mispronunciation routes as are conceivable, which means that the gain in recall performance is restricted .

3.0.2 CNN-RNN-CTC models

In recent times, the end-to-end structure has demonstrated a strong aptitude for ASR tasks , and it has gotten encouraging results in MDD . First, a CNN-RNN-CTC free phone recognize model was suggested , and it demonstrated an improved level of performance when compared with methods that had been used in the past . For the purpose of identifying categorial and non-categorial mistakes, a hybrid CTC-Attention architecture was utilized, and the initial L2 phone set was expanded. These CTC-based approaches do not require any sort of forced alignment and integrate the whole training pipeline. However, in the case of the previously read material, the aforementioned study does not make use of the information that was already included in the earlier text. A recent work presented a text-dependent end to end model that employs attention mechanism to merge the high-level hidden characteristics of acoustics and the sentence-to-character encoder feature. This was done in order to make use of the past linguistic knowledge. To make use of the previous text information, the model that is suggested in this study is comparable to sed-mdd , although there are still some discrepancies between the two. Feng et al. transform the original text into a string of characters that may then be embedded into sentences. Converting text into phoneme sequences and then feeding those sequences into a sentence encoder makes perfect sense given that the objective of the MDD work is to identify mistakes at the phoneme level. In addition, we found that the sed-mdd employed manually labeled phoneme boundaries in order to compute the frame-level cross-entropy loss function. This was something else that caught our attention. The connectionist temporal classification (CTC) loss function was utilized in this work; however, there was no labeled time information utilised.

This work [11] suggest two alternative approaches, one of which is based on convolutional neural network features (CNN Features), and the other on transfer

learning. Both of these approaches use different strategies. The first approach makes use of deep CNN characteristics in order to identify instances of incorrect pronunciation. Also, in order to train k-nearest neighbor (KNN), support vector machine (SVM), and neural network (NN) classifiers, we collect features from different layers of CNN (layer4 to layer7). Via the strategy that is based on transfer learning, we taught the CNN to recognize incorrect pronunciations using transfer learning. We compare the outcomes of these approaches with a baseline handmade features-based method for the 28 Arabic phonemes in order to evaluate the effectiveness of the system. In the technique that serves as a baseline, we employ the same classifiers—namely, KNN, SVM, and NN—to identify instances of incorrect pronunciation. According to the findings of the experiments, the handcrafted features approach, CNN features, and the transfer learning-based method each obtain an accuracy of 82%, 91.75%, and 92.25%, respectively. The performance study reveals that the accuracy achieved by the transfer learning-based approach is 92.2 percent, making it superior to the handcrafted features and transfer CNN features-based methods in terms of performance. In terms of accuracy, the suggested method that is based on transfer learning performs significantly better than the state-of-the-art techniques.

In this [5] work, we study the impacts of sentence-to-phoneme mapping and provide a framework for the end-to-end detection of mispronunciation that is based on the previous text attention mechanism. In addition, because there are a significant number of correctly pronounced phonemes in the training set in comparison to incorrectly pronounced phonemes, there will be a problem with an asymmetry between positive and negative samples in the phoneme sequence that we send to the attention model. As a result, the attention model will have a tendency to output the feedback phoneme sequences. In light of this, we will now provide three simple strategies for augmenting data in order to investigate the impact that altering the source phoneme sequences has. Lastly, in this project, all of the datasets, measurements, and baseline systems are available to the public as open source.

Mispronunciation detection has traditionally relied on an acoustic-phonetic feature set in conjunction with a support vector machine (SVM) classification technique. There has been a significant amount of research conducted on the identification of mispronounced words in a variety of languages, including En-

glish, Dutch, Chinese, Mandarin, and Japanese; nevertheless, there has been very little research conducted on the detection of mispronounced Arabic phonemes. Studies of the relevant literature demonstrate that several strategies and methodologies are employed for mispronunciation detection, including methods based on posterior probability, methods based on classifiers, and methods based on deep learning. The currently available methods only function on some of the many phonemes that make up language, which can be quite perplexing. The huge number of people who speak the Arabic language has led to an increase in the prevalence of incorrect pronunciation. The identification of the characteristics that are most suited for the detection of mispronunciation is still a topic of active study. In this research, we created I an approach that is based on CNN features and (ii) a method that is based on transfer learning. The CNN Features technique uses a deep convolutional neural network to automatically identify differentiating characteristics from the spectrograms of audio recordings. We search through the CNN's different levels (convolutional layers 4 and 5, fully connected layers 6, 7, and 8) to discover the features that are most suited for mispronunciation, and then we extract those features. The handmade features-based technique involves the process of extracting features from audio recordings. These characteristics include the Mel-frequency cepstral coefficients (MFCC), Chroma, Mel spectrogram, spectral contrast, and Tonnetz. In order to assess the efficacy of the HandCrafted Features and CNN Features approaches, we make use of three distinct classifiers: KNN, SVM, and NN.

3.0.3 Probability based method

In the past, a significant amount of emphasis was placed on the use of ratings determined by log-likelihood and scores determined by posterior probability in the procedure of mispronunciation detection. After taking into account the data ratings that were standardized by the duration of each phone section, Witt and Young [18] came up with the concept of the strength of pronunciation (GOP) score. [GOP] stands for "generally acceptable pronunciation." They devised a benchmark by which they could evaluate the manner in which each phone was used and decide whether or not it was suitable. The authors Franco et al. [19] produced three recognizers by collecting data from various levels of fluency and taking into consideration the proportion of log-likelihood-based ratings. The primary

emphasis of the work of other individuals has been to isolate relevant information from the background noise of the speech. Strik and colleagues [7] investigated the use of acoustic-phonetic characteristics, such as logs root mean squared, power, and 0 rate, with the intention of differentiating between velar sounds and velar plosives. Minematsu et al. [20] proposed the concept of a well-known auditory structure in speech that avoids non-semantic content. This pattern would be used to identify words. In addition, Wang and Lee [8] combined GOP score with error pattern identifiers in order to enhance the performance of a bunch of students from 36 various nations who were learning Mandarin Chinese. The goal was to increase the performance of the group's ability to recognize faults. Zhang et al. [9] developed the scaling file probability (SLPP) and the weighted phone SLPP in order to give a more precise estimation of the quality of the pronunciation.

3.0.4 DNN based method

Researchers Li et al. [24] combined neural networks with time data in order to identify instances of mispronunciation. In order to improve the mispronunciation detection, they suggested a Multi-layered perceptron technique that uses TRAPs features (TempoRAl Patterns) rather than standard features (i.e. spectral features). This method is referred to as the (TRAPs-MLP). In the study by Mohamed et al. [25], the researchers demonstrated that DBN work much more effectively when they gathered data from speaker adaption and discriminative characteristics. DNN-HMM is a framework that was introduced by Gao et al. [14] for error detection. They employed three different types of acoustic features, including MFCC, PLP, and filter band. They compared the outcome to the baseline GMM-HMM approach, and their findings demonstrate that three characteristics behave in various ways. Transfer learning, along with a logistic regression classifier and a deep neural network, was utilized by Hu et al. [17] in order to improve the mispronunciation detection procedure. A deep neural network-based method was presented by Hu et al. [15] not only for identifying incorrect pronunciations but also for providing diagnostic assistance to second language learners. They increase the quality of pronunciation while also extending the GOP estimation to use DNN-HMM rather than GMM-HMM. The study of Li et al. [26] proposed a deep belief network-based method for detecting lexical stress in speakers of English. They improved the stress categorization of words that included

three or more syllables by using syllable-based prosodic cues in conjunction with the words' lexical stress. A neural network based audio model training was proposed by Joshi et al. [16] in order to detect incorrect vowel pronunciation. They learned a acoustic model with a small amount of data from the target language and initialized it with data from other languages. An audio phonological model is proposed by Li et al. [27] that uses multidistribution bayesian belief network with speaker recognition and equivalent canon pronunciations for the purpose of mispronunciation detection for the English corpus. The research conducted by Lee et al. [28] concentrates on the identification of word-level mispronunciation using a deep belief network. The authors employ posteriorgrams as the input characteristics. The non-native speech is aligned with at minimum one native utterance, and characteristics are extracted. After that, they calculated the misalignment and distance matrix in order to display the amount of divergence from the path that had been aligned. Mispronunciation detection is accomplished by certain authors through the utilization of recent ideas such as fuzzy estimate [29], adaptive output [30], and enhanced stability criteria [31].

3.0.5 Classifier

Although probability-based techniques are able to detect the quality of the pronunciation, the scoring computations used in these strategies are not sufficient to differentiate the type of error and the precise position of that fault. As a result, classification-based mispronunciation detection approaches are utilized for the aim of fulfilling this need. The clusters of pronouncing rules were produced by Ito et al. [10], who also established a threshold for every cluster. In order to improve the reliability of error detection, a method of clustering that is decision-based has been presented. A wavelet transform was utilized in the process of feature extraction by Georgoulas et al. [11]. Wavelet analysis is a one-of-a-kind signal processing approach that makes use of a support vector machine (SVM) classification algorithm to identify instances of incorrect pronunciation. Mispronunciation detection was examined from several angles in Strik et al[7] .'s comparative research of four different techniques (GOP, logistic regression, LDA APF, and LDA MFCC). This research has shown that the LDA-APF as well as the LDA-MFCC techniques give superior outcomes to GOP score and the logistic regression when it comes to the accurate categorization of velar fricatives (x) and velar plosives

(k) respectively. Amdal et al. [21] distinguished between short vowels and long vowels of speech in their research. They developed a classifier using a technique known as linear discriminant analysis (LDA), using acoustic-phonetic characteristics. An efficient detection method that is based on GLDS-SVM (General Linear Linear discriminate Sequence- Support Vector Machine) was presented by Li et al. [12]. In order to improve the execution, they merged the GLDS-SVM method with the universal background model-GMM (universal background model) framework. The log-likelihood ratios between all of the acoustic models were employed as features for the SVM classifier by Wei et al. [13]. For each phone, many parallel acoustic models were created in order to determine the several ways in which the phone may be pronounced given its varying degrees of capabilities. The framework was able to deliver superior performance as a result of this approach compared to more typical likelihood-based tactics. The acoustic-phonetic feature-based CAPT method was developed by Maqsood et al. for the most perplexing Arab phoneme pairs (/ / vs / /) and (/ / vs / / or / /). They used four different classifiers, including Random Forest, Naive Bayes, Ada-boost, and K-NN, and the results indicated that Random Forest was the classifier that performed the best in comparison to the others [22]. In addition, the racist and discriminatory features for Arabic misspelling detection were chosen by Maqsood et al. [23] using the serial floating forward selection approach. In addition to this, they suggested using a grouping-based approach to extract discriminative characteristics of Arabic phonemes. However, in order to extract discriminative auditory characteristics, the majority of these algorithms require the involvement of a person.

In this work [11] in order to identify instances of incorrect pronunciation, we constructed three models, which are presented in figure 1. In the HandCrafted Features model, we start with an audio dataset, then manually extract features from audio files, and then feed those features to classifiers such as KNN, SVM, and NN in order to identify instances of incorrect pronunciation. The technique of extracting features from audio data involves converting the audio data set into a spectrograms dataset and then passing it on to a convolutional neural network. This is done in the CNN Features model. Convolution layer (Conv4 and Conv5) as well as convolution layers (FCL6, FCL7, and FCL8) of a pre-trained models were utilized in the feature extraction process. After that, we used KNN, SVM, or NN to analyze the retrieved characteristics in order to identify

instances of incorrect pronunciation. In the transfers learning model, we sent the spectrogram data to a pertained convolutional neural network called AlexNet. The neural network automatically performed feature extraction and classification, and it found instances of mispronunciation. In the subsections that follow, all of the specifics of each approach, in their entirety, will be discussed.

Whenever we use the handmade feature model, the first thing that we do is extract the characteristics that are necessary to differentiate the phonemes. Literature makes use of an impressively large number of qualities, each of which represents a property of auditory material. In order to identify instances of incorrect pronunciation, we made use of spectral parameters such as Mfcc, Chroma, Mel spectral analysis, spectral intensity, Tonnetz, pitch, root - mean - square square energy, and zero-crossing rate, in addition to statistical information such as mean, standard deviation, and slope. Second, the data that was acquired is scant and is presented in its raw form. Because of this, we normalized the data in order to deal with the issue of sparsity. Third, we narrowed down the feature space in order to find the characteristics that were most useful for categorization. In the end, we take the characteristics and run them through classification algorithms in order to identify mispronunciations. In the next subsections, the specifics of each stage are broken down in further depth.

4. Dataset

This experiment based on the TIMIT corpora. Corpus contains the recordings of the English speech. It was recorded by using close talking microphone at 16kHz rate with 16 bit resolution. There are 630 speakers from the 8 major dialects of the United States. Each speaker record the audio of the 10 sentences. Overall 6300 audios of sentences was recorded. Approximately it takes 6 hours.

Set	# speakers	#sentences	#hours
Training	462	3696	3.14
Core test	24	192	0.16
Complete test set	168	1344	0.81

Figure 4.1: TIMIT Corpus training and test sets

TIMIT corpus has already divided into the train and test parts. Documentation suggest to use this division. The corpa divided into train and test sets in 70% and 30% proportion. The train set contains 4620 utterances. This is 3696 sentences that recorded from the 462 native speakers. The test set consists of 1344 utterances from the 168 speakers recordings. There is also a core test set, which is the short version of the main test set. The core test set contains 192 utterances, that recorded by 24 speakers that pronounce 8 sentences.

For many decades, this voice corpus served as a standard resource for the speech recognition field and is still frequently used today for speech and word recognition research. Is not only because each utterance is phonetically labeled and assigned codes for speaker number, gender, and dialect area, but also it is considered small enough to ensure a relatively quick turnaround time for complete experiments while being large enough to demonstrate the capabilities of systems.

	Phone Label	Example		Phone Label	Example		Phone Label	Example
1	iy	beet	22	ch	choke	43	en	button
2	ih	bit	23	b	bee	44	eng	Washington
3	eh	bet	24	d	day	45	l	lay
4	ey	bait	25	g	gay	46	r	ray
5	ae	bat	26	p	pea	47	w	way
6	aa	bob	27	t	tea	48	y	yacht
7	aw	bout	28	k	key	49	hh	hay
8	ay	bite	29	dx	muddy	50	hv	ahead
9	ah	but	30	s	sea	51	el	bottle
10	ao	bought	31	sh	she	52	bcl	b closure
11	oy	boy	32	z	zone	53	dcl	d closure
12	ow	boat	33	zh	azure	54	gcl	g closure
13	uh	book	34	f	fin	55	pcl	p closure
14	uw	boot	35	th	thin	56	tcl	t closure
15	ux	toot	36	v	van	57	kcl	k closure
16	er	bird	37	dh	then	58	q	glotal stop
17	ax	about	38	m	mom	59	pau	pause
18	ix	debit	39	n	noon	60	epi	epenthetic silence
19	axr	butter	40	ng	sing			
20	ax-h	suspect	41	em	bottom	61	h#	begin/end marker
21	jh	joke	42	nx	winner			

Figure 4.2: 61 TIMIT original phone set.

4.0.1 Metrics for standardized assessment of phone recognition

The most often used metrics for evaluating ASR systems are the phone error rate (PER) or the associated performance statistic, phone accuracy rate. The latter is defined as follows:

$$Accuracy = \sum_{i=0}^{N/2} m_j c_n j = 0, 1, ..., P$$

where NT is the total number of label in the reference utterance and S, D, and I denote the corresponding substitution, deletion, and insertion errors. PER=100% Accuracy. Another metric is accuracy, which is comparable to accuracy but ex-

cludes insertion mistakes. The amount of insertion, deletion, and substitution mistakes is calculated by comparing the best alignment of two token sequences: the one aligned manually (reference) and the one recognized (test). Typically, an alignment resulting from dynamic programming-based search algorithms is employed effectively for a vast variety of voice recognition tasks [12]. Using this dynamic programming approach, speech recognition toolkits such as HTK contain tools for calculating accuracy and associated metrics on the basis of transcribed input and recognition outputs.

4.0.2 A summary of recent and previous research on the TIMIT phone recognition challenge

Despite recent advancements, the TIMIT phone identification test remains a complex and hard assignment. Numerous efforts have been made to enhance the performance of phone recognizers, including the use of more accurate features or various feature sets, increased prediction tests, training criteria, pronunciation modeling, acoustic modeling, noise management, and language modeling.

Statistical models may be classified into two major types based on their approaches: generative and discriminative. Phone recognition is the process of determining the optimal sequence of phones (Ph) that corresponds to a given input voice X. This is a search issue in which the objective is to determine the ideal phone sequence Ph provided by.

$$Ph = argmaxP(Ph|X)$$

Bayes rule is used in generative techniques. arrives at

$$Ph = argmaxP(X|Ph)P(Ph)$$

In order to arrive at this formula, a trained model of a conditional probability distribution of the observed acoustic features X given the phone classifier of the linked phone class was utilized. Because the model "generates" the input data to ensure that it is consistent with the model's Ph value, the word "generative" was coined to describe this phenomenon. A few examples of generative approaches include HMMs, segmented HMMs, hidden path models, Gaussian mixture model

(GMMs), stochastic segmentation designs, gradient boosting, or Markov random fields. Since a long time ago, the application of maximum likelihood maximum entropy models has been the most widespread in ASR. The most important advantage of influencing the academic performance is that it makes it really easy to exploit the inherent dependencies or various linkages in data by imposing a variety of structural constraints on it [7].

In comparison, discriminative approaches, like those based on the highest entropy models, logistic regression, artificial neural networks (multi-layer perceptron (MLP), time-delay neural networks (TDNN), or Boltzmann machines), support vector machines (SVMs), and conditional random fields (CRFs), aim to model the posterior class distributions, trying to maximize the discrimination between acoustically similar targes.

By attempting to cover this broad variety of technologies, the pertinent research on TIMIT phone recognition during the last several years will be covered.

Shortly after the first release of TIMIT in December 1988, Lee and Hon published one of the earliest concepts[8] employing phone identifying on the TIMIT database. They cited lee1989speaker as the source of their information. Their strategy utilizes discrete Hidden Markov Models as its foundation. When phones were simulated by employing a bigram language model with 1450 diphones (right–context), the findings were the most successful. Three codebooks, each of which had 256 prototype vector for linear prediction mel frequency cepstrum coefficients, were used as features in this study. With only one set of test questions and a total of 160 responses, they achieved a correctness rate of 73.80 percent and an accuracy rate of 66.08 percent (TID7). They suggest that these data be used as a standard for phone recognition at TIMIT. In point of fact, their work has set a benchmark not only for the performance of the phone, as well as for the folding method for the phone that they introduced. These writers consolidated the 61 TIMIT labels into a total of 48 phones for the sake of training. For the sake of evaluation, they whittled the 61 TIMIT designations down to 39 phones, which is now the norm throughout the industry. Figure ?? provides an explanation of the folding process, along with the 39 phone set that was produced as a consequence. The mobile devices listed in the left column are concealed within the corresponding labels of the items listed in the right column. The list is updated with the addition of the label "sil" and the removal of 23 phone labels. The original set,

which consisted of a total of 61 phones, has been maintained in its entirety.

aa, ao	aa
ah, ax, ax-h	ah
er, axr	er
hh, hv	hh
ih, ix	ih
l, el	l
m, em	m
n, en, nx	n
ng, eng	ng
sh, zh	sh
uw, ux	uw
pcl, tcl, kcl, bcl, dcl, gcl, h#, pau, epi	sil
q	-

Figure 4.3: Mapping from 61 classes to 39 classes

In the same year, 1989, Steve Young also presented the first version of the HTK (hidden Markov model toolkit). The development of this software program at Cambridge University made it possible to construct and manipulate hidden Markov model (hmm, which led to substantial improvements in the field of speech recognition. The idea of HMM state tying achieved by the application of triphone models (left and right context). The creation of a condensed collection of context-dependent HMMs that illustrates how state tying may significantly cut down on the number of physical triphone models utilized in training is the goal of this project. They take a phone set that has 48 elements and turn it into a triphone. The experimental conditions are analogous to those that were supplied by citation [8], with the exception that these researchers used Mel-frequency cepstral coefficient (MFCCs) features and log energy, in addition to their first order coefficient of determination (deltas -). When using the 39 phone sets that were proposed and 160 phrases that were randomly picked from the test set, the best results were achieved, which were 73.7 percent correct and 59.9 percent accurate.

In 1991, Robinson and Fallside [14] created a phone identification system based on a recurrent error propagation network that obtained an astounding 76.4 percent correctness and 68.9 percent accuracy utilizing the identical Lee and Hon evaluation set [8]. When the whole test set is used, these values increase to 76.5

percent and 69.8 percent for correctness and accuracy, respectively. The authors report even better rates (71.2 percent for accuracy), however they utilized a group of 50 phones rather than the typical 39. Robinson et al. connected the recurrent network with an HMM decoder in 1993 [14], where the network is utilized to estimate the HMM state posterior probability. The Wall Street Journal database was used to test this system. The TIMIT findings were obtained in 1994 using a hybrid RNN/HMM. The neural network's inputs are features taken from a lengthy left context. The network is trained by comparing its output to a softmax value using the cross-entropy criteria. The outputs of the network were trained using the 61 original TIMIT labels. The results for the 39 classical phone sets were 78.6 percent correct and 75% accurate. This finding continues to outperform previous publications! Additionally, the article includes an intriguing comparison of numerous previous research on the phone detection job.

Some works was published on speaker-independent phone identification utilizing continuous density HMMs (CDHMMs) for context-dependent phone models trained using probability distribution and maximum a posteriori (MAP) estimation approaches. Cepstral coefficients created from linear prediction coefficients (LPC) are included in the feature set, as are and cepstrum (second order regression coefficients). The findings were 77.5 percent /72.9 percent (correctness / accuracy) when the whole test set was used.

Halberstadt and Glass introduced a system in 1998 as a consequence of their PhD study that combines numerous classifiers. Through several heterogeneous acoustic measurements, training was conducted to optimize the acoustic modeling. Each classifier is tasked with the task of recognizing a subset of the TIMIT labels. Six classifiers independently train 60 TIMIT phone labels (they ignore the glottal stop /q/). Three further classifiers combine the data from the previous three classifiers.

4.0.3 Features

Study of the robustness of systems for automatic recognition (ACS) of phonemes is to search for features that are invariant to a possible significant difference "exemplary" speech used in teaching CAP, and the conditions of real speech on output of telephone communication channels. It is known that the main requirement for signal parameterization (features, extracted from the speech signal) in

speaker-independent speech recognition is to smooth out the individual characteristics of voices speakers . It is assumed that the speech signal is stationary over time intervals on the order of a few milliseconds. During the analysis, the speech signal is divided into blocks data (windows). Based on data obtained by weighting the speech signal window, feature vectors are calculated. This paper explores two widely used methods parametrization of the speech signal, namely, small-particular cepstral coefficients - MFCC (Mel-frequency cepstral coefficients) and perceptual linear prediction coefficients - PLP (perceptual linear predictive) The MFCC and PLP coefficients are some kind of cepstrum, which allows us to speak about their effectiveness when working under conditions multiplicative noise.

Mel Frequency Cepstral Coefficents, or MFCCs for short, are a characteristic that is utilized extensively in computerized speech analysis and speaker recognition. Linear Prediction Coefficients (LPCs) and Linear Prediction Cepstral Coefficients (LPCCs) were the primary feature type for automated speech recognition (ASR), particularly when using HMM classifiers, until the advent of MFCCs. Click here for a lesson on cepstrum and LPCCs. On this page, we will discuss the most important characteristics of MFCCs, as well as why and how to implement them for automatic speech recognition (ASR). In the final portion of this discussion, we will delve into a more in-depth explanation of how to compute MFCCs.

The procedure for obtaining MFCC coefficients in practice consists in the following: the sample of cepstrum values is calculated through the sample of values:

$$M_j = \sum_{i=0}^{N/2} m_j c_n j = 0, 1, ..., P$$

obtained by averaging the nonparametric estimate of the spectrum by triangular weight functions:

$$c_n = \sum_{i=0}^{N/2} m_j c_n j = 0, 1, ..., P$$

The width of the weight functions is constant on the non-linear chalk frequency scale. Due using the small scale it is possible to take into account the nonlinear dependence of auditory perception on the frequency of the speech signal. Thus, a formal smoothing of the spectrum of the speech signal is carried out, which,

in turn, greatly simplifies speech modeling by reducing dimensions of the feature vector. The need to use spectrum estimation, obtained using the fast Fourier transform (FFT), leads to the fact that the process of obtaining MFCC coefficients in a computational sense is more costly. Therefore, in practice, a different approach is used to calculate MFCC coefficients: a sample of cepstrum values n c is calculated through coefficients , 0 k k K , parametric (autoregressive) estimation of the speech signal spectrum, using the recursive relation. One of the advantages of MFCC coefficients is their statistical independence, which in turn allows us to model the probability density functions using diagonal covariance matrix . An alternative to using MFCC coefficients are coefficients perceptual linear predictive PLP (perceptual linear predictive) . Technics using PLP parameterization is based on psychoacoustic concepts when spectrum estimation: spectral analysis in critical frequency bands; curves of equal loudness; non-linear relationship between intensity and perceived loudness of a sound.

Extraction of PLP coefficients based on standard low-frequency analysis Fourier spectrum using a filter comb. Fourier spectrum precalculated from N – signal samples $1, 2, ..., N$. Output coefficients the filter bank formula is weighted by the equal loudness curve. where j is the frequency of the j-th triangular window and then are compressed by cube root. Further, by calculating the inverse Fourier transforms based on the values M_j calculate the coefficients of the linear LP (linear predictive) predictions. In this work, the basic phoneme recognition system was modeled with using the NTK software toolkit (Hidden Markov Model (HMM) Toolkit - tools based on Hidden Markov Models) and taking into account the recommendations works. Left-right HMM models were used, consisting of 3 states without gaps with continuous Gaussian mixtures. The analysis of the speech signal was carried out with using a 25 ms Hamming window, with an analysis step of 10 ms. This window applied to each frame of speech before further processing. Voice signal passed through a high-pass filter with a transfer characteristic. The number of triangular windows for analysis on a nonlinear chalk frequency scale is 26. 12 cepstral coefficients were calculated, supplemented logarithm of energy. In order to take into account the change in parameters over time, the coefficients cepstra, and the logarithm of energy have been completed with the first (D prefix) and the second derivatives (prefix A) . To PLP odds, instead of adding energy logarithm, a zero cepstral value was added to the pa-

rameter vector coefficient (prefix 0). By adding the prefix Z, normalization was carried out cepstral mean normalization (CMN) aimed at suppressing effects due to the difference in the frequency characteristics of the recording and transmission channels speech signals . Training of acoustic models began with a flat start (flat start) , with This created a universal unimodal model (Gaussian). Prototypes generated models contained a single Gaussian mixture with a single stream. For further training cycles, the number of Gaussian mixtures was gradually increased to maximum value for monophones and triphones, equal to 20. The number of direct and number of reverse moves with an increase in the number of mixtures was 4. The monophones obtained at the training stage with a single Gaussian mixture, were used for cloning context-dependent phonemes - tryphons. When building a decision tree based on training data, after one training cycle, trifons were connected by means of clustering algorithm, taking into account the rules of the English language.

5. Transformers

The Transformer (Vaswani et al., 2017) has quickly become the dominant design for natural language processing, outperforming competing neural models such as convolutionary neural neural networks in natural language interpretation and generation tasks. The architecture grows with the training data and model size, allowing for fast parallel training and capturing long-range sequence characteristics. Model pretraining (McCann et al., 2017; Howard and Ruder, 2018; Peters et al., 2018; Devlin et al., 2018) enables models to be trained on generic corpora and then quickly tailored to particular tasks while maintaining high performance. The Transformer architecture is particularly well-suited to pretraining on large text corpora, resulting in significant improvements in accuracy on downstream tasks such as text classification (Yang et al., 2019), language recognizing (Liu et al., 2019b; Wang et al., 2018, 2019), language processing (Lample and Conneau, 2019a), work perfectly (Joshi et al., 2019), sensible inference (Bosselut et al., This advancement raises a number of practical issues that must be solved before these models can be extensively used. The Transformer's widespread usage necessitates the development of systems to train, evaluate, scale, and supplement the model on a number of platforms. The architecture serves as a foundation for more complex expansions and precise experimentation. The widespread usage of pretraining techniques has necessitated the need to disseminate, fine-tune, deploy, and condense the community's core pretrained models. Transformers is a library devoted to enabling Transformer-based architectures and making pretrained models more widely available. The Transformer implementation, built for both development and technology, lies at the heart of the library. The idea is to provide industrial-strength implementations of widely used model variations that are simple to understand, extend, and deploy.

On this basis, the library facilitates the distribution and use of a brbad range of

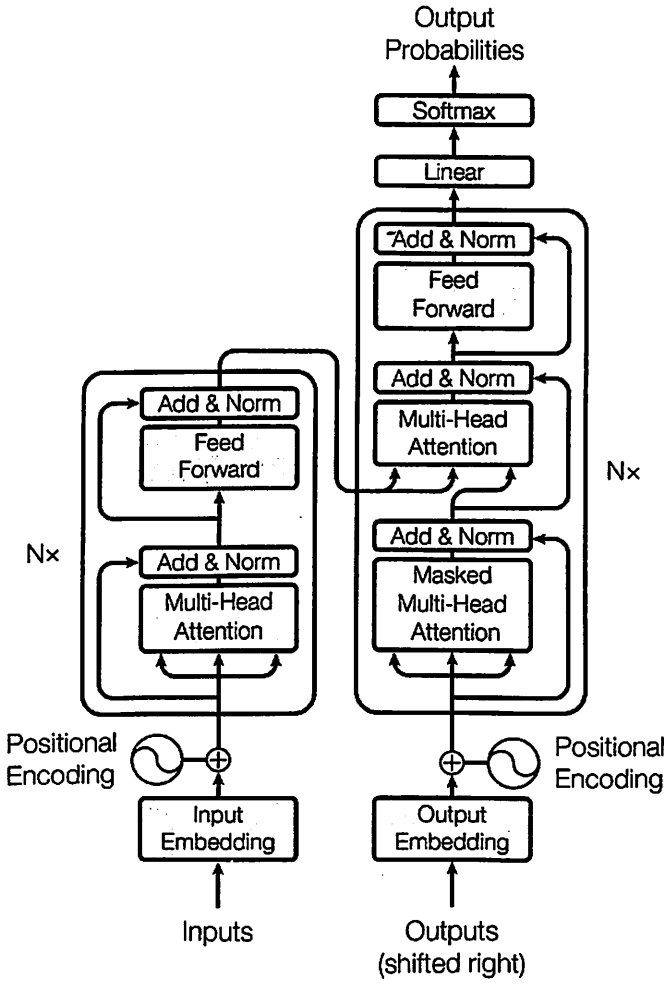

Figure 5.1: Transformer architecture

pretrained models via a centralized model hub. This hub allows users to evaluate multiple models using the same simple API and experiment with shared models on a wide range of tasks. Transformers is a continuing project managed by Hugging Face's team of engineers and academics, with help from a thriving community of over 400 external collaborators. The library is distributed under the Apache 2.0 license and may be found on GitHub1. Hugging Face's website2 has extensive documentation and lessons. Audio scene analysis is a traditional signal processing and artificial intelligence issue in which the aim is to anticipate the contents of an input signal in a short period of time, usually one second. Aside from modeling perception, simulation of hearing along with models of other sensory inputs will aid in the bridge between people and computers. CNNs have established the de-facto architecture for learning mappings from fixed dimensions inputs to fixed dimensional outputs during the last decade . CNN architectures inspired by vision and applied for acoustic scene comprehension provide comparable au-

dio performance benefits. Transformer architecture, which has recently generated state-of-the-art results in a range of areas, such as protein , text , conceptual music , video, and picture comprehension , serves as the foundation of this study. They were able to produce convincing results in music creation and style transfer by learning transformers on latent representations and conditioning a wave net generator, which would have been unachievable without the supervision of meta-data and neural architectures . They've also been used to learn latent audio representations like for performing pseudo-tasks like infilling to train time-dependent models . In contrast to training latent representations, the shortened time-scales of Transformers may model input representations more effectively. The fixed filter over the whole input is a significant disadvantage of convolution design. Furthermore, Transformers use an attention system, with the output at one point reliant on the input at another. The central concept of our study is to replace standard convolutions architectures , hybrid convolution and Transformer designs , and recurrent architectures with a solely Transformer architecture. Our work differs from the technique provided in , it was not an end-to-end approach and needed a two-step procedure (learning a dictionary of latent codes and utilizing the discrete latent codes as an input to transformer designs). Similar techniques have been used effectively to simulate BERT in domains such as voice recognition . All of these cutting-edge results were made feasible by the designs' capacity to represent long-term dependence inputs as well as the attention mechanism built into them.

5.0.1 Transformer Architectures

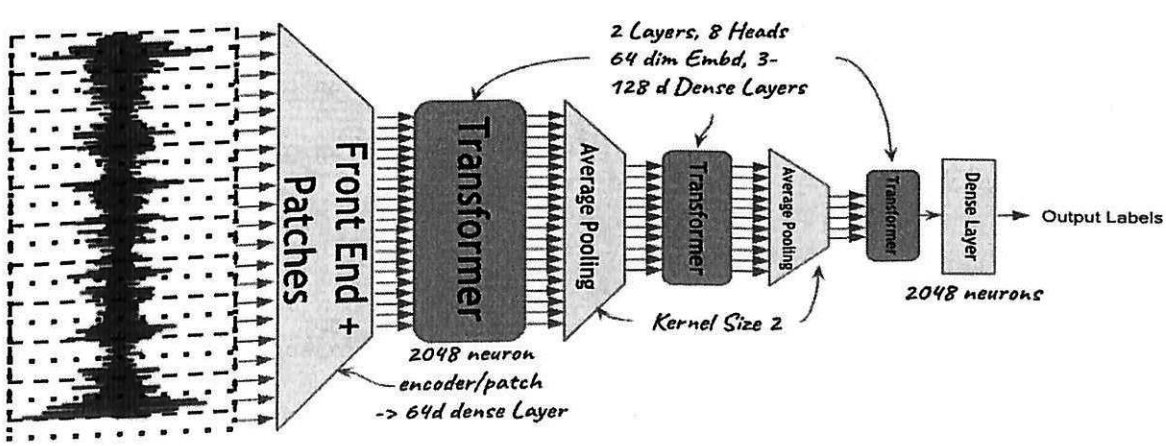

Figure 5.2: Audio Transformer architecture

This section discusses the Transformer design described in , which was utilized

to train the system shown in Figure 5.2. provides a lengthy explanation, but for clarity and completeness, we present it here. As a black-box, which we will discuss in more depth later in this section, it accepts a sequence of fixed length T as input and outputs the same length but with a selected dimension, which we call E, which specifies the size of the latent space. It transfers a sequence

$$x = (x_1, x_2, ..., x_T)$$

to a sequence of the same length T, namely

$$z : (z_1, z_2, ..., z_T)$$

where each dimension of

$$(z_1, z_2, ..., z_T)$$

is the specified hyper parameter E, which in our example is 64, the size of the embedding. For the purpose of simplicity, we will only discuss one Transformer Encoder, so for a design with L layers, each stack is overlaid on the other. Each Transformer module is made up of an attention and a feed-forward block. Each one's output is routed via a layer standard and a residual level. If it the inputs to a sub channel (attention F_a or feed-forward F_{ff} block) is indeed a sequence x_b, instead of delivering the output immediately to the next module/sub-block, we pass along the block layer norms and the residual output x_{bo} as

$$x_{bo} = LayerNorm(x_b + F_{a/ff}(x_b))$$

This is consistent with the idea that layer-norm/skip connections aid in greater convergence/performance. We will now go over each of the two sub-blocks that comprise the transformer block: I multi-headed causal attention ii) architecture based on feed-forward architecture.

5.0.2 Multi-Headed Attention

A weighting function that chooses how to receive the outcome of each step is an example of a multi-headed deterministic attention function. This kind of function may be characterized using the word "attention." During the process of output

prediction, it acquires a probabilistic score that indicates the relative significance of each of the embeddings. The first step in developing a multi-headed attention is to get familiar with a probabilistic score. After that, it is multiplied with each of the inputs in order to assess how significant each of the inputs is for the prediction of a embed for a position pos that belongs to $1, 2, 3, ...T$. The sort of attention mechanism that we deploy is referred to as scaled-dot product attention. It is learnt for each of the positions in each of the inputs a query, a key, and a value vector for that position. This is accomplished by implicitly learning matrices, which provide a query vector q, key vector k, and value vectors v each of the input for just a given attention head. W_Q stands for query vector, W_K stands for key vector, and W_V stands for value vector. We compute the linear combination of the query vector and the key vector, often known as the a normalization factor, which is the inverse of square root, is multiplied with the result. Root of the size of the vector, as was done in , prior to performing a soft-max calculation across the whole of the inputs. This factor is multiplied by each element of the value vector. Score in order to get the results of the attentiveness module. In terms of mathematics, for a query matrices Q, key knowledge areas K, and a response matrix V , it is defined as

$$Attention(Q, K, V) = softmax(Q * K^T/sqrt(d_k))$$

Additionally, we may also Acquire many attention maps of this kind for your attention heads, which are specified as

$$MutliHeadAttention(Q, K, V) = Concat(h_1, h_2, ..., h_h)W_o$$

formulas are used to define each one of the attention headings h_i and defined as

$$Attention(Q_i, K_i, V_i) = softmax(Q_i * K_i^T/sqrt(d_k))$$

and W_o is a matrices that is taught during the training process. The primary emphasis of this work is on on the causal attention map, which may be generated by the process of multiplying with a mask in the form of a triangle matrix, designed to ensure that each of the attention is focused solely on the head provides consideration to the sample that was taken at position *pos* as well as to all of the

future additions are reduced to 0. [4]

5.0.3 Feed Forward Architecture

We use a multi-headed attention method to determine the relative importance from each of the incoming signal in order to pass at a point *pos*. Every one of the inputs located at location *pos* is routed via an architecture known as feed-forward. We have the output of a feed-forward layers, which is denoted by x_{bo}, for an input denoted by x_b, and the dimension of the feed-forward layers, denoted by d_{ff}, for a network with two layers is as follows:

$$FF(x_b) = max(0, x_b W_1 + b_1)W_2 + b_2$$

At each of the inputs, this function is applied in exactly the same way. In accordance with what is stated in [16], positional encoding is applied to each of the inputs. Because the input is sent to the model in the form of a list, the models does not take the relative position into account; hence, positional encoding is required. We utilize a sinusoidal function for each point *pos* along the I dimension of the latent space. This means that for each position *pos*, along with the embedding of the i dimension in E, we add

$$PE_pos, 2i/2i + 1 = sin/cos(pos/10000^(2_i/E))$$

Before moving through the self-attention layers, this adds information on the position of each input with dimension E at each moment in time.

5.0.4 Adapting Transformer Architecture for raw waveforms

We adjust the design of the Transformer by using concepts from conventional signal processing. We have decided to stick with the tried-and-true method of preemptive multitasking the signal since the complexity of the Transformer is $O(n^2)$ in terms of the amount of memory and compute it requires. As was previously mentioned, we conduct all of the studies using one second of audio input captured at sixteen thousand hertz, which results in sixteen thousand samples. A

nonoverlapping square window with a length of 25 milliseconds has been selected as the window type. The network receives an ideal windowing function thanks to the rectangular window, which, as we shall see in a few of the learnt filters, adjusts itself to a form that is similar to that of a hanning/hamming window. We decide to have the front end consist of a thick layer of size 2048 neurons, followed by another layer of size 64 neurons. The primary reason for this decision is to accommodate ourselves to the dimensions of the hidden layers of Transformer. According to the results presented in [17], a single dense layer with a size of 2048 effectively learnt a filter-back in order to learn a cellular frequency representation. Because it delivered state-of-the-art results for an equally challenging issue of pitch prediction in polyphonic audio [17], using feed forward layers, this architecture was selected as the superior option. We are able to accomplish an end-to-end architecture that does not include any convolutional operators because to the fact that Transformer layers only have attention and feed-forward blocks. 1 This results in a front side representation of forty time steps, each of which has sixty-four dimensions (64 being a hyper-parameter). To convert to the appropriate feature space, we choose 6 levels of the Transformer module, with the size of the latent code being 64 for each of the layers, and 8 attention heads, with 128 dim 3-layer dense layers. In order to facilitate a comparison with a more compact model, we opted for 3 levels of Transformers with an identical arrangement. The most recent layer of the Transformer is shrunk down to a more manageable size utilizing calculation of an overall average throughout time. The output of the most recent dense layer, which has a size of 200 and is selected to have the same value as the set of output labels.

5.0.5 Transformer Architectures: Pooling

I investigate further performance improvements that may be made to the underlying Transformer design that was suggested in the preceding part. In order to do this, we use ideas from the convolutional architectures that have been employed over the course of the last ten years to comprehend pictures and audio [6]. The conventional models, such as Resnet-50 are comprised of a mixture of convolutional layers, which is then followed by pooling. The utilization of pooling layers comes with two distinct benefits. By decreasing the amount of the inputs in the upper layers, it brings the total number of calculations under control. Even

more critically, it enables the greater neurons to have considerably greater receptive field widths. Additionally, it enables the network to acquire hierarchically ordered characteristics, which is essential for a specific challenge. While maintaining a same number of parameters as the baseline Transformer design, the performance of pooled transformers is superior to that of the baseline. Due to the fact that it stores more information about the signal, the average pooling method fared noticeably better than the maximum pooling method in our tests. As shown in Figure 1, we lower the complexity of a input by the a factor of two by using pooling over time after every 2 layers or Transformers with stride 1. This results in a large performance boost when compared to the initial Transformer design, which did not employ pooling.

5.0.6 Multi-scale embeddings

The wavelet decomposition and the triumph of pooling layers have both served as sources of motivation for this modification. We investigated if it is possible for us to breakdown the intermediate embeddings coming out of the Transformer on several scales, in a manner that is analogous to the concept of wavelet decomposition. In order to do that, we have to adjust our kernel so that it performs similarly across all of the windows that are selected at a certain level. Take note that we use windows of varying widths for the numerous aspects of embedding all along time axis. Again, the method in which the implementation is carried out is a design option, and there are a number of fascinating concepts that may be used in the future, one of which is the selection of the kernel. In the work [3] by following a geometric development, we make adjustments to the size of the window using variables such as 1, 2, 4, 8, and so on. Instead of decreasing the size, as is done in pooling, the value is allocated to each and every one of the elements, which results in the size remaining the same. This operation can be learned in end-to-end architectures since it can be totally differentiated from other operations. We operate first with variable scale factor as compared to fixed windows, and secondly, we do not accept explicitly hand-crafted bands of filters, which distinguishes this from previous work done on spectrum filtering [15]. This is because we chose to operate in this manner. In addition, we decided to describe the domain of embeddings-time in a hierarchical fashion by using just a few big windows and a big number of smaller windows, with the majority of the smaller

windows having a value of 1 in order to keep the word embedding at the same size as they were initially. This keeps the original transformer word embedding while modifying the other half of the embeddings in some way. However, this keeps the original transformer embeddings. Wavelet transforms have relied heavily on this combination throughout their development.

6. Experiments and Results

6.1 Data

In this experiment, I used the TIMIT dataset. Which I wrote about in the previous chapter 4. In the first experiment, I took the original audio files as well as the original division into training and test parts. After analyzing the data, it was decided to reduce the number of phonemes. In order to improve the accuracy of the model location. I removed those phonemes that are similar to each other, such as *a* and *aa*. In the end, I had 40 phonemes left. You can see them in the Table 6.1

6.1.1 Synthetic augmentation

In the first experiments, I got low results. Also, in the process of training the model for 10 epochs, the model began to retrain. After that, it was decided to synthetically increase the size of the dataset. To achieve this, the following steps were taken

Phonemes			
dh	eh	ao	ae
b	n	r	th
ah	iy	k	f
ng	s	y	ax-h
g	ih	er	v
l	ch	t	uh
ow	uw	m	oy
w	ey	jh	aw
z	dx	ay	hh
p	sh	d	sil

Table 6.1: 40 phonemes

Filename	Phoneme labels
SA2_719	d ow n ae s sil m iy dx ih sil k ih r iy eh n ao l iy r ae sil.
SA2_719_speed_08	d ow n ae s sil m iy dx ih sil k ih r iy eh n ao l iy r ae sil.
SA2_719_speed_125	d ow n ae s sil m iy dx ih sil k ih r iy eh n ao l iy r ae sil.
SI1263	b ow th hh ae v eh sil k s l eh n ih n ih sil g r ey sh n ah.
SI1263_speed_08	b ow th hh ae v eh sil k s l eh n ih n ih sil g r ey sh n ah.
SX330_1673	k aw n sil dh ah n ah m b er ih v sil t iy s sil p uw n z ah
SX185_3059	d ih sil jh uw sil b ay ih n iy sil k ao r dx er r oy...

Table 6.2: Phoneme labels

1. Change playback speed. The speed of a randomly selected record from the dataset was either increased or decreased.

2. Add white noise to the background. I added white noise by randomly choosing the degree of sound.

3. Add other noises and various sounds such as the sounds of the city, nature to the background. This item is similar to the previous one, the only difference is that here we add really possible natural sounds.

All these operations were done only on the training dataset. After that, we managed to double the training set. Labels of original audio tracks and generated ones remained the same. You can see some examples of labels in Table 6.2

6.2 Experiment

As an architecture I use this [13] end to end model. The sequence-to-sequence (S2S) method in voice recognition (ASR) has received a considerable amount of interest due to its capability to collectively train all components towards a common goal, which reduces complexity and error propagation in comparison to traditional hybrid systems. This is one of the reasons why the S2S approach has received so much attention. Traditional systems separate global features (such as stream and speech characteristics) and feature points (on the phoneme level) from one another by dividing the representation into multiple levels within the acoustic model. In particular, global features are separated from phoneme-level features. After being trained separately using a variety of loss functions, the language model and the audio model are brought together for the decoding process. In comparison, neural

S2S models are capable of performing a mapping function from acoustic input to text sequences. This is accomplished through the dynamic interactions between the two primary model components, an encoder and a decoder. During training, these components work together to maximize the likelihood of the sequence that is generated as an output. The audio characteristics are read by the neural encoder and converted into high-level representations. These representations are then fed into such an auto-regressive decoder, which constructs the output sequences in a careful manner. And I use transformer model [16] to create E2E model that will find the incorrect pronounced parts in an English speech. On this 6.1 6.2 figures you can see the spectograms of an original audio and the synthetically augmented.

After passing it through a variety of band-pass filters, the spectrum is next subjected to a computation that determines the strength of each frequency band. The filter bank is made up of a collection of filters that have triangle frequency responses, and the Mel scale is used to evenly distribute their center frequencies over the scale. This particular kind of filter bank layout is utilized for automated speech recognition (ASR) as well as speaker verification. The human ear system has a non-linear sense of pitch, hence the Mel scaled is a non-linear scaling that adapts to this non-linearity. One way to define the Mel scale is as follows:

$$MEL_{scale} = 2595log10(1 + f/700)$$

In order to get the Mel spectrogram, the spectrogram is first processed using the Mel-scale filter.

The difference in decibel levels between the peak and the valley of a spectra is referred to as spectral contrast. A distinct spectrum, denoted by a vector, is associated with each phoneme. The angle of zero degrees demonstrates that two spectra are identical, but the angle difference of ninety degrees demonstrates the greatest possible distinction between spectra. Before attempting to compute the spectral contraction SCk, we must first determine the peak pk and valley vk values for each subband. The letter K in this case denotes the kth subband. The spectral contrast may be determined using the formula:

$$SC_k = p_k v_k$$

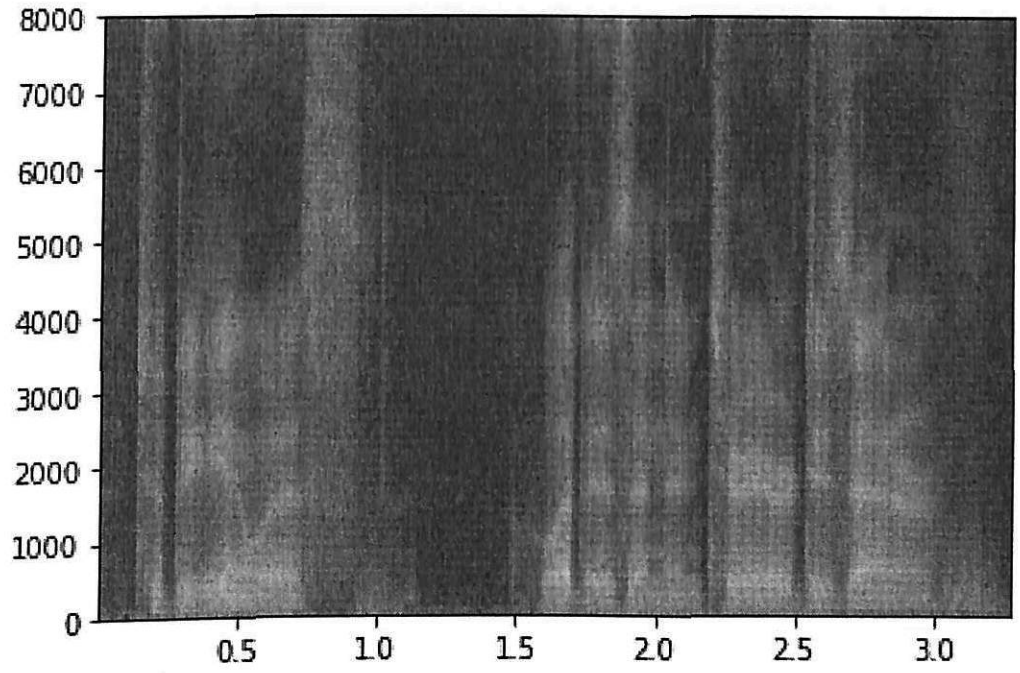

Figure 6.1: Original audio files spectogram

6.2.1 Data preprocessing

The application of machine learning algorithms to current culmination includes a fundamental step known as data preparation. Its goal is to carry out an in-depth study of the data, with the end goal being the achievement of optimum results. The following are the stages that are involved in the data preparation in our approach.

Cleaning

The act of smoothing out noisy information, identifying and removing abnormalities, and settling inconsistencies is referred to as data cleaning. This technique is used to "clean" the data. During the process of data cleansing, repetitive entries, spelling mistakes, and information that does not make sense are identified and eliminated. In addition to that, we cleaned out the noise in the audio signal.

Sparsity

Features data acquired from audio recordings have missing values. In order to solve the problem of data sparsity, we first indicate the null values using a numerical cleaner filter, and then we fill in the values that are absent. The numeric cleaner filer is responsible for cleaning the numeric data and replacing the value

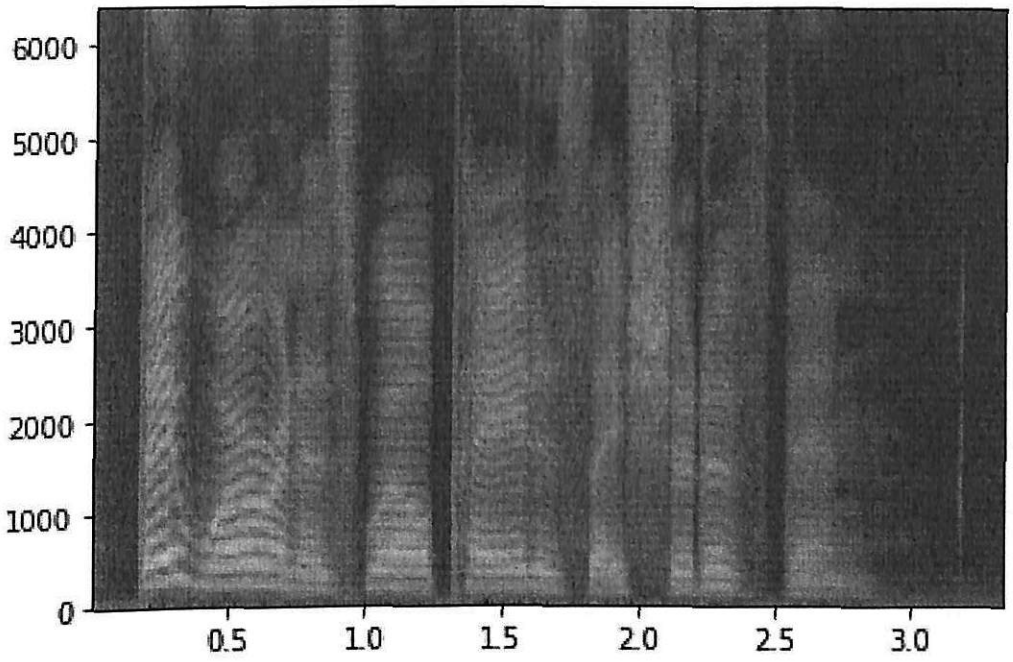

Figure 6.2: Synthetically augmented files spectogram

that are either too large or excessively little with the value that is set as the default. After the missing values have been located and distinguished, they are validated by a filter, which then replaces them with a mean guess of the distribution of data.

Audio to spectogram

Following the conversion of all stereo sounds to monotype signals, the audio files that are part of the dataset are transformed into spectrograms. In order to do extraction of features, the convolutional layer makes use of the spectrograms.

Resampling

Because the model only accepts an input of size 40*40*3, we had to adjust all of the spectrograms to fit inside the parameters of an input layer of AlexNet in order to extract features. In order to extract features from the spectrogram data, we first preprocessed the data and then passed it to a convolutional neural network known as AlexNet.

6.2.2 Train and test

I train two models on the to types of data sets. First model was trained on the original TIMIT data set without phoneme decreasing and synthetic augmentation. The second experiment was on the data set with decreased number of vocabulary(phonemes) and synthetically increased train set.

6.2.3 Evaluation

To evaluate correctness of the model I use word error rate between the source labels and predicted ones. When evaluating the effectiveness of a large vocabulary continuous speech recognition (LVCSR) system, the Word Error Rate (WER) is the method that is most often used. The word sequence that was hypothesized by the ASR system is matched with a reference transcript, and the total number of mistakes is determined as the sum of the insertions I, deletions D, and substitutions S that occurred during the alignment. If the reference transcription has a total of N words, then the word error rate, abbreviated WER, may be calculated as follows:

$$WER = (I + D + S)/N * 100$$

It is necessary to collect at least two hours' worth of test data from a standard LVCSR system in order to arrive at an accurate estimate of the WER. It is necessary to hand transcribe the test results down to the word level in order to carry out the alignment, which is a procedure that is both time demanding and costly. It is for this reason that the development of methods that can measure the quality of an autonomously produced transcription without the need for a gold-standard reference is of importance.[1]

6.2.4 Results

Two experiments were carried out. You can see the results of training the first model in the 6.3. In the first experiment, the model trained well for the first 10 epochs. After the tenth epoch, the model stopped training due to overfitting. The results of testing the first model gave us a result of 15% accuracy. In the second experiment, thanks to a synthetic increase in the dataset and a decrease in the number of labels, I managed to achieve little progress in training and in the

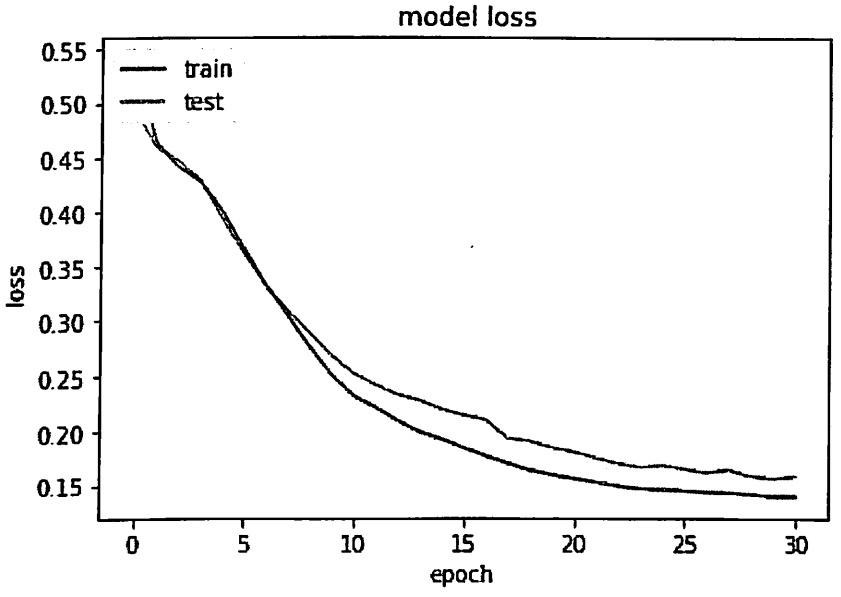

Figure 6.3: Experiment 1. On the original data set

results. You can see the process of training the second model in the 6.4. Testing the second model gave us a result of 29% accuracy.

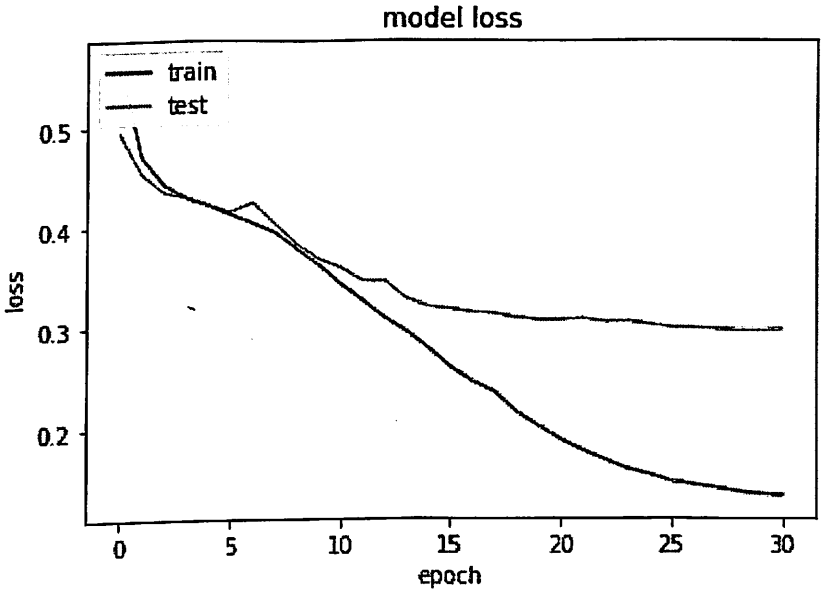

Figure 6.4: Experiment 2. On the augmented datas set

7. Summary and Feature work

In this thesis, I created a methodology for detecting mispronunciations that works by assessing the alignment between synthetically modified data and the original data. The objective is to circumvent the issue of needing well-labeled training data, which is required for traditional speech recognizer-based approaches.

7.1 Feature Work

There are several possible applications that may be built on the model described in this thesis. We've listed several probable paths below.

7.1.1 Complete E2E model

The present model is a first effort at identifying pronunciation problems like as insertion, deletion, and replacement errors. One typical mistake that students make is using the incorrect lexical stress pattern. For the time being, the model cannot identify this kind of inaccuracy using MFCC-based or GP-based alignments since those features were not intended to extract pitch information from speech. Finally, to promote student learning, an effective CAPT system not only should indicate that mistakes were made, but also indicate the kind of errors produced. This would need the system's ability to differentiate between various sorts of mistakes. One conceivable way for achieving such a goal would be to train separate detectors for different sorts of pronunciation problems.

7.1.2 Application to Other Languages

The model did not need any prior understanding of the target language. The model received no information on the "language" itself. This comparison-based

paradigm has a significant benefit in that it may be used to any language. It would be great to do trials on other languages in the future to see if the performance holds up.

7.1.3 Implementation in web or mobile applications

This model can be help full to develop the user friendly mobile application or the website. Where users can learn the correct pronunciation. Or a lot of language learn platforms that currently popular can upgrade their systems by adding this model.

References

[1] Ahmed Ali and Steve Renals. "Word error rate estimation for speech recognition: e-WER". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2018, pp. 20–24.

[2] Kartik Audhkhasi et al. "Building competitive direct acoustics-to-word models for english conversational speech recognition". In: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2018, pp. 4759–4763.

[3] Jonathan Berger, Ronald R Coifman, and Maxim J Goldberg. "Removing noise from music using local trigonometric bases and wavelet packets". In: *Journal of the Audio Engineering Society* 42.10 (1994), pp. 808–818.

[4] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[5] Kaiqi Fu et al. "A full text-dependent end to end mispronunciation detection and diagnosis with easy data augmentation techniques". In: *arXiv preprint arXiv:2104.08428* (2021).

[6] Jort F Gemmeke et al. "Audio set: An ontology and human-labeled dataset for audio events". In: *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2017, pp. 776–780.

[7] Hui Jiang. "Discriminative training of HMMs for automatic speech recognition: A survey". In: *Computer Speech & Language* 24.4 (2010), pp. 589–608.

[8] K-F Lee and H-W Hon. "Speaker-independent phone recognition using hidden Markov models". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.11 (1989), pp. 1641–1648.

[9] Wai-Kim Leung, Xunying Liu, and Helen Meng. "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)*. IEEE. 2019, pp. 8132–8136.

[10] Kun Li et al. "Automatic lexical stress and pitch accent detection for L2 English speech using multi-distribution deep neural networks". In: *Speech Communication* 96 (2018), pp. 28–36.

[11] Faria Nazir et al. "Mispronunciation detection using deep convolutional neural network features and transfer learning-based model for Arabic phonemes". In: *IEEE Access* 7 (2019), pp. 52589–52608.

[12] Hermann Ney and Stefan Ortmanns. "Progress in dynamic programming search for LVCSR". In: *Proceedings of the IEEE* 88.8 (2000), pp. 1224–1240.

[13] Ngoc-Quan Pham et al. "Very deep self-attention networks for end-to-end speech recognition". In: *arXiv preprint arXiv:1904.13377* (2019).

[14] Tony Robinson and Frank Fallside. "A recurrent error propagation network speech recognition system". In: *Computer Speech & Language* 5.3 (1991), pp. 259–274.

[15] Alex Tamkin, Dan Jurafsky, and Noah Goodman. "Language through a prism: A spectral approach for multiscale language representations". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5492–5504.

[16] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[17] Prateek Verma and Ronald W Schafer. "Frequency Estimation from Waveforms Using Multi-Layered Neural Networks." In: *INTERSPEECH*. 2016, pp. 2165–2169.

[18] Bi-Cheng Yan et al. "An end-to-end mispronunciation detection system for L2 English speech leveraging novel anti-phone modeling". In: *arXiv preprint arXiv:2005.11950* (2020).

[19] Wei Zhang et al. "Towards end-to-end speech recognition with deep multi-path convolutional neural networks". In: *International Conference on Intelligent Robotics and Applications*. Springer. 2019, pp. 332–341.