Combining Automatic Speech Recognition Models To Reduce Error Propagation in Low-Resource Transfer-Learning Speech-Command Recognition

Jazeem Mohamed Isham

(209333X)

Degree of Master of Science

Department of Computer Science and Engineering

Faculty of Engineering

University of Moratuwa

Sri Lanka

March 2022

Combining Automatic Speech Recognition Models To Reduce Error Propagation in Low-Resource Transfer-Learning Speech-Command Recognition

Jazeem Mohamed Isham

(209333X)

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree

Master of Science specializing in Data Science

Department of Computer Science and Engineering

Faculty of Engineering

University of Moratuwa

Sri Lanka

March 2022

DECLARATION

I declare that this is my work and this dissertation does not incorporate without acknowledgment any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to the University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic, or another medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the Master's dissertation under my supervision. I confirm that the declaration made above by the student is true and correct

Name of the supervisor: Dr. Uthayasanker Thayasivam

Signature of the supervisor: Date:

ABSTRACT

There are several applications when comes to spoken language understanding such as topic modeling and intent detection. One of the primary underlying components used in spoken language understanding studies is automatic speech-recognition models. In recent years we have seen a major improvement in the automatic speech recognition system to recognize spoken utterances. But it is still a challenging task for low-resource languages as it requires hundreds of hours of audio input to train an automatic speech recognition model.

To overcome this issue recent studies have used transfer learning techniques. However, the errors produced by the automatic speech recognition models significantly affect the downstream natural language understanding models used for intent or topic identification. In this work, we have proposed a multi-automatic speech recognition set up to overcome this issue. We have shown that combining outputs from multiple automatic speech recognition models can significantly increase the accuracy of low-resource speech-command transfer-learning tasks than using the output from a single automatic speech recognition model.

We have come up with convolution neural network-based setups that can utilize outputs from pre-trained automatic speech recognition models such as DeepSpeech2 and Wav2Vec 2.0. The experiment result shows a 7% increase in accuracy over the current state-of-the-art low resource speech-command phoneme-based speech intent classification methodology.

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor Dr. Uthayasanker Thayasivam for being my supervisor and guiding me throughout the project. His expertise in this related area helped me a lot in setting the right direction for this project and obtaining the needed datasets.

I would not have achieved this without the immense support from my family. I would like to thank my parents for supporting me throughout my studies from the start the up until today. A special thanks to my wife and family for coping with me throughout my research program and encouraging me to complete it.

I also want to thank the University of Moratuwa for giving me an opportunity to participate in the MSc program and for providing the necessary resources to complete this research. This would not have been easy without the support from my workplace as well. I would like to thank Sysco LABS Srilanka for allowing me to do this part-time MSc and research while working with them. Also, my colleagues have been very understanding and allowed me to spend time on my research even during busy office schedules.

TABLE OF CONTENTS

DECLARATION i

ABSTRACT ii

ACKNOWLEDGMENTS iii

TABLE OF CONTENTS iv

LIST OF FIGURES vi

LIST OF TABLES vii

LIST OF ABBREVIATIONS AND ACRONYMS viii

1 INTRODUCTION 1

1.1 Background 1

1.2 Research Problem 2

1.3 Research Objectives 3

1.4 Outline 3

2 LITERATURE REVIEW 4

2.1 Low-resource Transfer Learning 4

2.2 DeepSpeech2 5

2.3 Wav2Vec 2.0 6

2.4 Dual-Input CNN Model 7

2.5 Feature Concatenation 7

3 METHODOLOGY 9

3.1 Multi-ASR Combinations 9

3.2 Method 1 - Dual-input CNN Models 10

3.2.1 Input Layer (DeepSpeech2 and Wav2Vec 2.0) 10

3.3 Method 2 - Feature Concatenation 12

3.4 Commonly Used Techniques In Both Models 15

3.4.1 Convolutional Layer 15
3.4.2 Activation Function 15
3.4.3 Max-pooling and Dropout Layers 15
4 DATA SET 16
5 EXPERIMENT 18
5.1 Hyperparameter Tuning 18
5.2 K-Fold Cross-Validation 19
5.3 Experiment Setup 20
6 RESULTS 21
7 DISCUSSION 22
8 CONCLUSION AND FUTURE WORKS 24
9 REFERENCES 25

LIST OF FIGURES

Figure 2.1 DeepSpeech2 Architecture [4] 6
Figure 2.2 Wav2Vec 2.0 Architecture [5] 7
Figure 3.1 Overview of low-resource transfer learning 9
Figure 3.2 Overview of proposed multi-ASR transfer learning architecture 10
Figure 3.3 Architecture of the dual-input CNN model 11
Figure 3.4 Overview of combined-input model architecture 12
Figure 3.5 DeepSpeech2 Feature Padding 13
Figure 3.6 Wav2Vec 2.0 Feature Padding 13
Figure 3.7 DeepSpeech2 Feature Transition 14
Figure 3.8 Combination of Wav2Vec 2.0 and DeepSpeech2 Features 14
Figure 5.1 Process of K-Fold Cross-Validation 20

LIST OF TABLES

Table 4.1 Details of the dataset 16
Table 5.1 Details of the experiments 18
Table 6.1 Details of the experiment results 21
Table 6.2 F1-Score per class 21
Table 7.1 Difference in the model transcriptions for a given utterance (DS2 - DeepSpeech2, W2V - Wav2Vec 2.0) 22

LIST OF ABBREVIATIONS AND ACRONYMS

Abbreviation	Description
CNN	Convolution Neural Network
ASR	Automatic Speech Recognition
NLU	Natural Language Understanding
SLU	Spoken Language Understanding
ML	Machine Learning
DS2	DeepSpeech2 Model
W2V	Wav2Vec 2.0 Model
Exp	Experiment
WER	Word Error Rate
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
CTC	Connectionist Temporal Classification
SMBO	Sequential Model-based Optimization

1 INTRODUCTION

1.1 Background

With the advancement of voice-enabled technologies, spoken language understanding has evolved so much now it is almost as good as a real human. There are various applications when it comes to Spoken Language Understanding(SLU) including Speech command recognition and topic identification [1] [2] [3]. The common approach to SLU is to first use the automatic speech recognition(ASR) model to transcribe the voice and use a natural language understanding(NLU) model train on the text corpus to do various classification tasks.

Recently end-to-end ASR models like DeepSpeech2 [4] and Wav2Vec 2.0 [5] have been introduced as an alternative to previously predominant solutions based on Hidden Markov Models such as Kaldi [6]. However, these deep neural network-based models required an enormous amount of audio data during the training. For example, the original DeepSpeech2 implementations were trained on more than 7000 hours of data and the new DeepSpeech2 was trained on 11,000+ hours of data. Only a few mainstream languages have these kinds of large data-set, to begin with, such as English and Mandarin. But these models won't perform that well even with popular languages such as Swiss and German which have around 100 hours of data each [7].

But when it comes to intent detection and topic modeling tasks, we don't necessarily need to have the complete transcript, rather we only need a way to model the utterance in a way so that this can be used by the downstream NLU models. There many transfer learning studies have been conducted [8] [9] [10]. The main fundamental approach used in these studies is to use the scriptions or N-hypothesis produced by the ASR model as the feature of an NLU model. However, the performance of these NLU models greatly depends on how accurate the transcription is provided by the ASR model.

In this study, we have proposed a way to reduce the effect of the ASR model on the downstream model by introducing ways to combine multiple ASR features to a single CNN NLU model. During our experiment, we have observed significant improvement over using a single ASR model and reported state-of-the-art accuracy of 88.25% for Tamil speech to command recognition.

1.2 Research Problem

Spoken language understanding is a common research area that is been studied and improved significantly in recent years. This is the field where people focus on processing audio signals and transferring them into a meaningful form such as text or vector form which can be processed further depending on the application. When it comes to speech-to-text transformations, the current models have human-level proficiency in understanding the audio input. There are many automatic speech recognition models that are capable of transcribing major languages such as English, French, and Mandarin.

But the low resource spoken language understanding is still an evolving field. One of the main reasons is that the lack of annotated data in those languages as well as the ASR models requires an enormous amount of data to be trained. People have used many alternative methods to overcome this issue, one such method is to transfer learn from an ASR model which is trained using another popular language. The applications of this kind of transfer learned features are limited to areas like topic modeling, and intent identification. When using ASR models for transfer learning, there is often a natural language understanding model downstream to use the features learned from the transfer learning and do further classifications to predict intent or topic.

The transfer learned features are not always going to be accurate in representing the low resource audio form primarily due to the nature of the mode, it always tries to transcribe the audio input into the original language it is trained with. So the error introduced in the transfer learning process is cary forwarded to the downstream natural language understanding model and thus affecting the overall performance of the classification.

There are researches happening to handle these errors by introducing more robust natural language understanding models that can tolerate these errors introduced. On the other hand, ASR models that are trained with multiple languages can also be used to reduce the error rate in the transcription process itself. Usually, when there is a better ASR model and a better NLU model, we often see better results. Research has improved low resource spoken language understanding by using CNN and other neural network-based models as well as using better ASR models when they are introduced. Instead of relying on better ASR models, In this research, we are focusing more on

"Techniques to combine ASR models to reduce the error propagated to downstream NLU models in low resource transfer learning setup."

1.3 Research Objectives

The above research problem is to be addressed by achieving the following objectives:

- To use the existing data in the previous studies to explore new ASR models and their performances in transfer-learning setup.
- To compare the transcribed features of each ASR model and identify the complementary models.
- To explore different Dual-input CNN model architectures to train a single NLU model with features from 2 ASR models
- To explore other pre-combination techniques to create more robust features that can be used in different types of CNN models.

1.4 Outline

The rest of this report has been arranged as follows. Chapter – 2 presents a detailed literature review on the current related works done under the problem such as low-resource transfer-learning studies and the statue-of-the-art ASR models. Chapter – 3 covers the methodologies used to carry out the experiments such as the ways the multiple ASR features are combined, and the model architecture. Chapter – 4 explains the dataset in detail followed by Chapter – 5 with more details on the experiment. Chapter – 6 discusses the results. In Chapter – 7 the results are explained and interpreted, followed by the conclusion and future works in Chapter – 8

2 LITERATURE REVIEW

2.1 Low-resource Transfer Learning

When it comes to recognizing speech commands, the usual approach is to use an audio model that is directly trained using annotated field data. In this recent study [11], the authors used CTC-based(Connectionist Temporal Classification) LSTM voice models to train Google Voice Search traffic on a mobile phone. The LSTM structure of this baseline contains three hidden layers, each containing about 850 LSTM cells. The authors have used a recurrent project layer which has the size of 450 for each hidden layers. However, LSTMs are computationally expensive, and sometimes difficult to train with CTC criteria. Due to the need for large data, the authors had to use 2000 hours of data in their study. 2000 hours of data is a difficult task to collect and this volume of data cannot be found when it comes to all areas.

In such scenarios where there are not much domain-specific speech data, transcripts generated by ASR models are widely used to produce the text feature of the voice. ASR, or Automatic Speech Recognition, refers to the problem of having algorithms that transform spoken language automatically (speech-to-text conversion). Usually, the goal is to have a machine learning model that reduces the word error rate (WER) when transcribing speech input. In other words, given some audio file (such as a WAV file) that contains the speech, how can the model convert this into a matching text with as low errors as possible. This text form of the features is then used in downstream Natural Language Understanding(NLU) models to identify the intent or topic [3] [12]. But building an ASR model is not viable most of the time, the sheer volume of data required to train those models is so high, that only a few mainstream language models are so far built. This is one of the main drawbacks when it comes to low-resource languages.

To address the low-resource ASR problems, transfer learning ASR [8] [13] and multilingual transfer learning ASR [14] are experimented with through different source languages to make the performance better for low-resource languages. Recent works have focused on coming up with such low resource systems by transfer learning techniques such as leveraging the intermediate outputs of English based DeepSpeech2 ASR model as the input feature for the downstream NLU models [8] and further improved by using phoneme-based ASR model trained by the English language and CNN based model as the NLU(Spoken Language Understanding) model [15]. In this research, authors have tried to use these techniques to improve the SLU for Tamil and Sinhala languages archiving 81% and 97% accuracy respectively. A phoneme is any distinct unit of sound in a language that differentiates a word from another.

ASR models can be divided into two categories based on the output features, character-based ASR models and phoneme-based ASR models. A character-based ASR model [4] [5] is trained to produce a distribution of the probability $p_t(c)$ over characters denoted as c per every step of time denoted as t. Where in a phoneme-based ASR model [16] it is trained to produce a probability distribution sound over a vocabulary at each time. Since a phoneme is the smallest distinct sound rather than a character, This study [15] was able to get a better result than the one that used a character-based ASR [8]. As we can represent more units of sound via the phoneme-based ASR model which is not bound to only the sound of the source language in which the ASR model is trained.

2.2 DeepSpeech2

DeepSpeech2 [4] converts the input speech into Melspectrograms, then applies CNN and RNN, and finally outputs the text using Connectionist Temporal Classification (CTC). Connectionist Temporal Classification (CTC) is a method often used in character recognition and speech recognition, in combination with LSTM and RNN. This method solves the problem of variable width and time length of a phoneme by erasing the same character in succession on the decoder side.

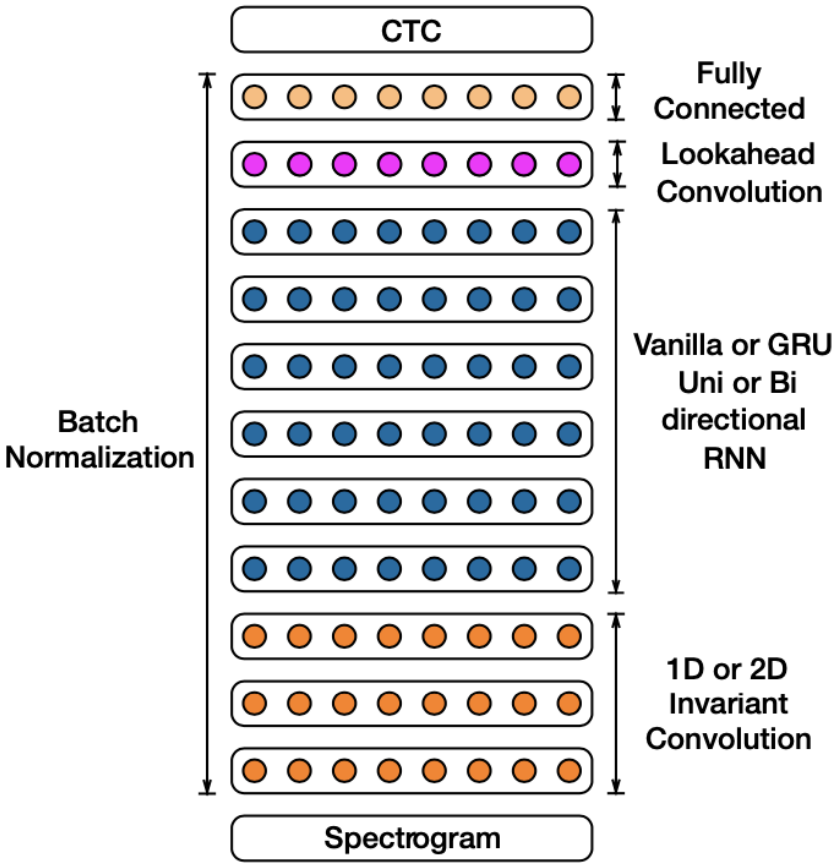


Figure 2.1 DeepSpeech2 Architecture [4]

2.3 Wav2Vec 2.0

Wav2Vec 2.0 [5] is one of the state-of-the-art algorithms that is used to build models for automatic speech recognition. The model leverages a new concept when it comes to this field, which is self-supervised learning. The Wav2Vec 2.0 model is trained in two steps. The first step tries to determine the best speech representation as much as possible using unlabled data by utilizing self supervise techniques. The model only used the labeled data in the second phase where it leverages supervised fine-tuning to teach itself to predict the particular word or phonemes.

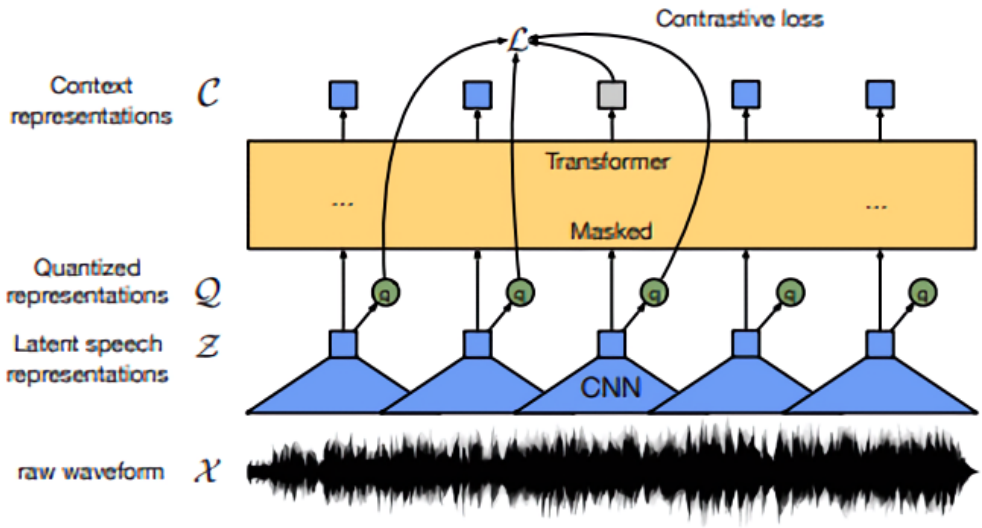


Figure 2.2 Wav2Vec 2.0 Architecture [5]

2.4 Dual-Input CNN Model

When it comes to combining features from multiple sources, it is often common to use dual input models [17] [18]. In these studies, the authors have used 2 different CNN models to learn the features from 2 different inputs and then combine the features into a single CNN model which is used for classification purposes. These types of CNN models are used in scenarios where there need to be multiple sources of inputs need to be combined. For example, combining voltage trend and current flow trend, or combining pictures taken from multiple angles.

2.5 Feature Concatenation

In one similar study [19] the authors have improved Speech Emotion Recognition (SER) by combining two spectrograms in a novelty manner, one is the original audio and the second one is obtained after injecting synthetic noise into the training data. Although we have not used spectrogram in this study to improve the SLU, the idea of combining multiple inputs from a different source as a means to improve the overall performance was an inspiration for this study.

The main approaches previous studies have taken to address low-resource speech intent identifications are to build a low-resource ASR model or retrain some of the layers of the pre-trained ASR models via transfer learning or leverage a pre-trained ASR model trained in with a more rich domain or language to generate n-hypothesis and use the outcome of it to feed the downstream NLU model to achieve classification tasks. Although studies have primarily used a single ASR model at a time, we have

seen in similar other domains such as SRE, that authors have tried to combine features from multiple upstream to train a single downstream SRE model. This is not yet been tried when it comes to combining the features from multiple ASR models to train a single NLU model. In the next section, we propose 2 main approaches to do so.

3 METHODOLOGY

All the previous researches mainly focus on using a single ASR model to predict either a word sequence or phonemic sequence which is then to be used in an NLU model.

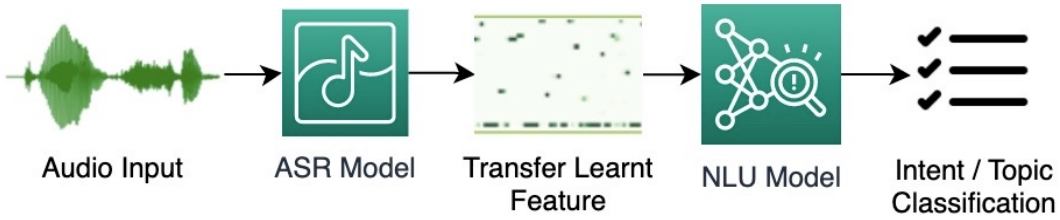


Figure 3.1 Overview of low-resource transfer learning

This would mean that any error produced by the ASR model will be propagated down to the NLU model thus affecting its performance of it. In this study, we focused more on improving the performance of NLU models not just by using a better ASR model to transfer learning from, but to reducing the error propagation by combining different ASR models.

We propose two different ways where we can combine the character probability sequence provided by two different character-based ASR models, DeepSpeech2 [4] and Wav2Vec 2.0 [5].

3.1 Multi-ASR Combinations

using the above two ASR models, we generated features independently. One of the main differences we observe apart from the difference in the character space of those two algorithms is the length of the mapping. where DeepSpeech2 outputs a longer character probability sequence of max of 555, Wav2Vec 2.0 only produced up to 256 character sequences. We have proposed 2 ways of combining features learned from multiple ASR models.

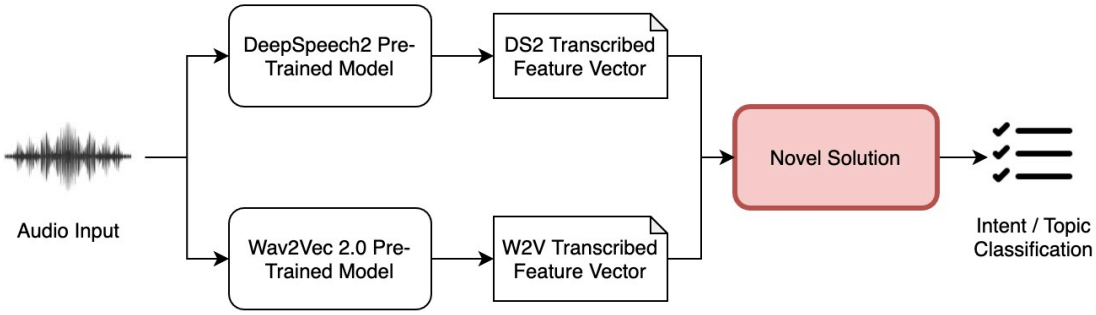


Figure 3.2 Overview of proposed multi-ASR transfer learning architecture

3.2 Method 1 - Dual-input CNN Models

There are previous researches [15] [13] [8] done to identify a fixed set of intents using the features extracted from audio inputs. These mainly used a single ASR model to extract the feature from the audio input and used classification models such as Support Vector Machine (SVM), Feed-forward Neural Networks(FFN), and Convolution Neural Networks (CNN). Out of all the studies, [15] has shown a state-of-the-art accuracy using 1D and 2D CNN models.

In this study, we mainly focus on combining the ASR features from DeepSpeech2 and Wav2Vec 2.0 ASR models. The first method we tried is to train 2 2D CNN models with each feature and train the last layers by combining the models and applying a softmax layer to identify the classification. Figure *3.3* explains the overall architecture of the model.

3.2.1 Input Layer (DeepSpeech2 and Wav2Vec 2.0)

There are 2 CNN model streams merged into a single downstream model constructing the overall architecture, one steam is connected with the feature transcribed from the

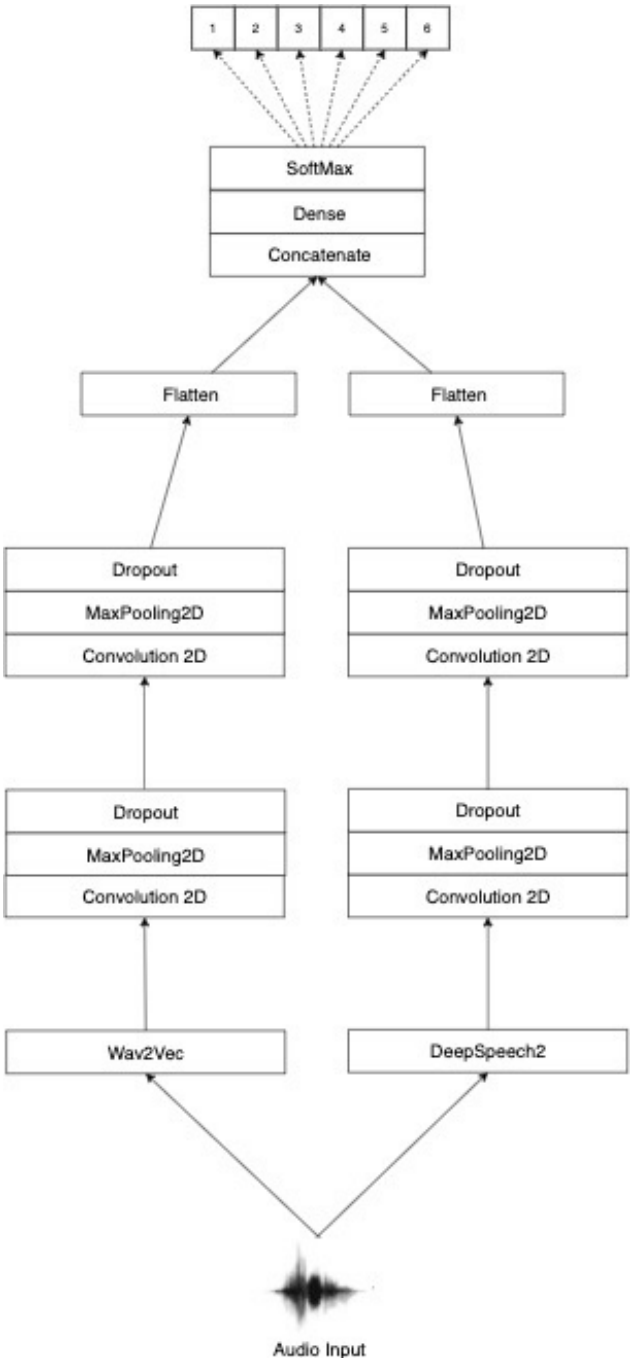


Figure 3.3 Architecture of the dual-input CNN model

DeepSpeech2 ASR model and the other stream is connected with the feature transcribed from the Wav2Vec 2.0 model for the same audio input as the deepSpeech model.

3.3 Method 2 - Feature Concatenation

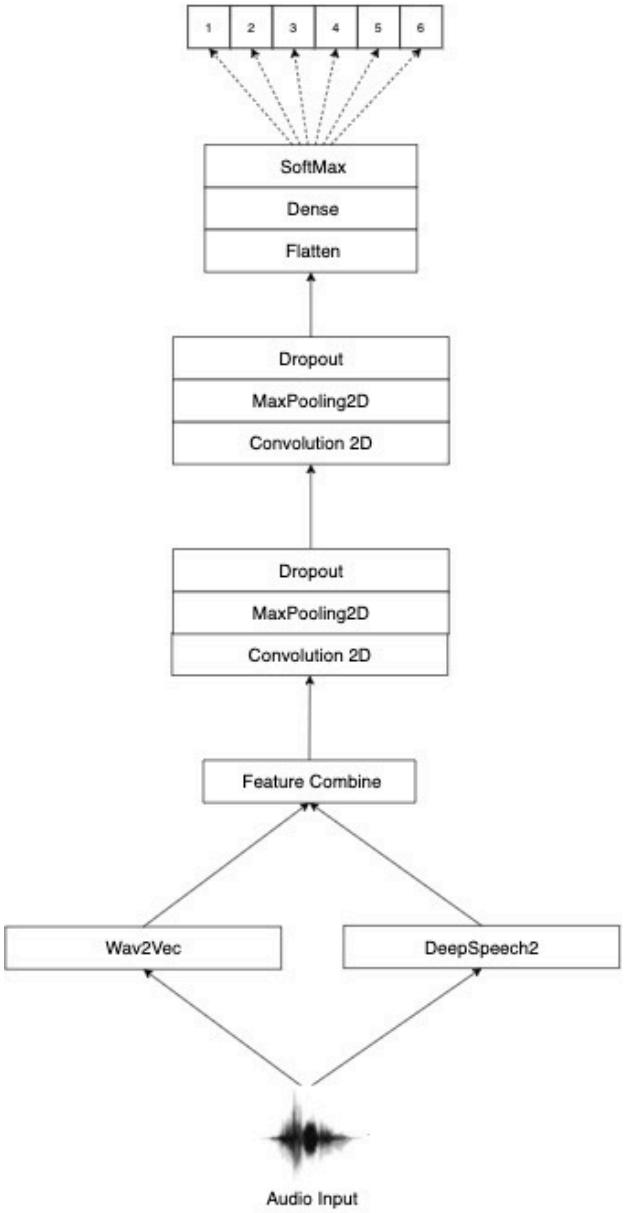


Figure 3.4 Overview of combined-input model architecture

First, we tried the obvious, which is a concatenation of the 2 features into a single feature. When it comes to CNN it is important to have the features uniform in dimensions, to tackle this issue we initially tried adding paddings of zeros to make every probability sequence has the same dimension. Then we used a 2D-CNN neural network explained in Figure *3.4* to treat the concatenated feature as a single feature do the classification.

The feature combine module of the combined-input model is a simple concatenation of NumPy arrays. For DeepSpeech2, if a transcribed probability sequence of an audio

The clip is $DS2_{act\_length}$ in length, we add $DS2_{max\_length}$ - $DS2_{act\_length}$ zeros to each character probability sequence such that the length of a given character sequence has the same dimension for each audio clip, ($DS2_{char\_size}$, $DS2_{max\_length}$).

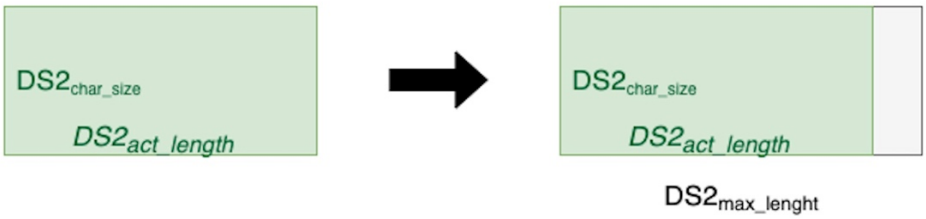


Figure 3.5 DeepSpeech2 Feature Padding

Similarly, for Wav2Vec 2.0, if a transcribed probability sequence of an audio clip is $W2V_{act\_length}$ in length, we add $W2V_{max\_length}$ - $W2V_{act\_length}$ zeros to each character probability sequence such that the length of a given character sequence has the same dimension for each audio clip,($W2V_{char\_size}$, $W2V_{max\_length}$).

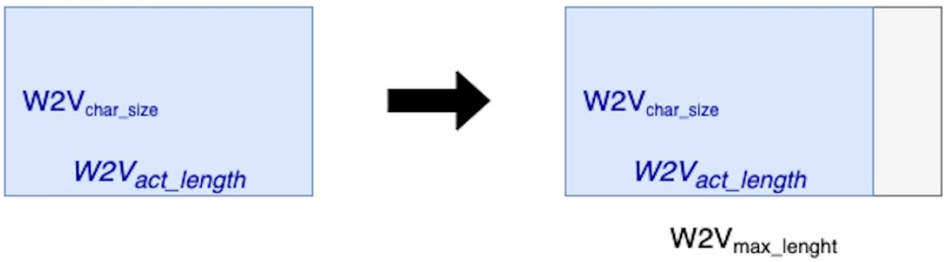


Figure 3.6 Wav2Vec 2.0 Feature Padding

The Wav2Vec 2.0 model we used have 32 characters and the DeepSpeech2 model we used has 29 character which creates a difference in number of rows in each feature output from Wav2Vec 2.0 and DeepSpeech2. To overcome the difference in the row count for each sets of features, we again used zero padding to the DeepSpeech2 features such as a feature would have a dimension of ($W2V_{char\_size}$, $DS2_{max\_length}$)

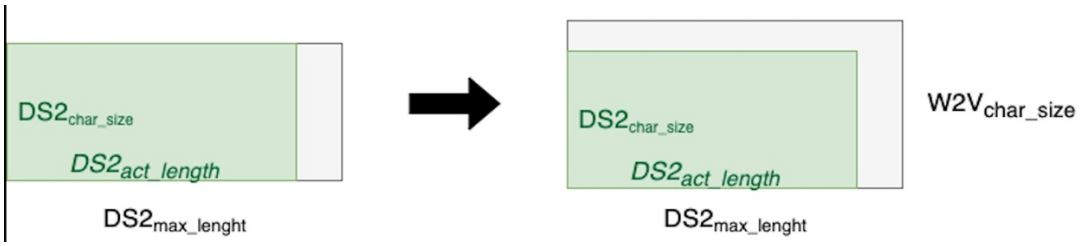


Figure 3.7 DeepSpeech2 Feature Transition

so that it matches the character size of Wav2Vec 2.0 models. Once the DeepSpeech2 character size is adjusted, we combined both feature so that a combined feature will be $\$(W2V_{char\_size}, DS2_{max\_length} + W2V_{max\_length})$ in dimension. The Figure *3.8* illustrates the dimension of the final combined feature of a given audio clip.

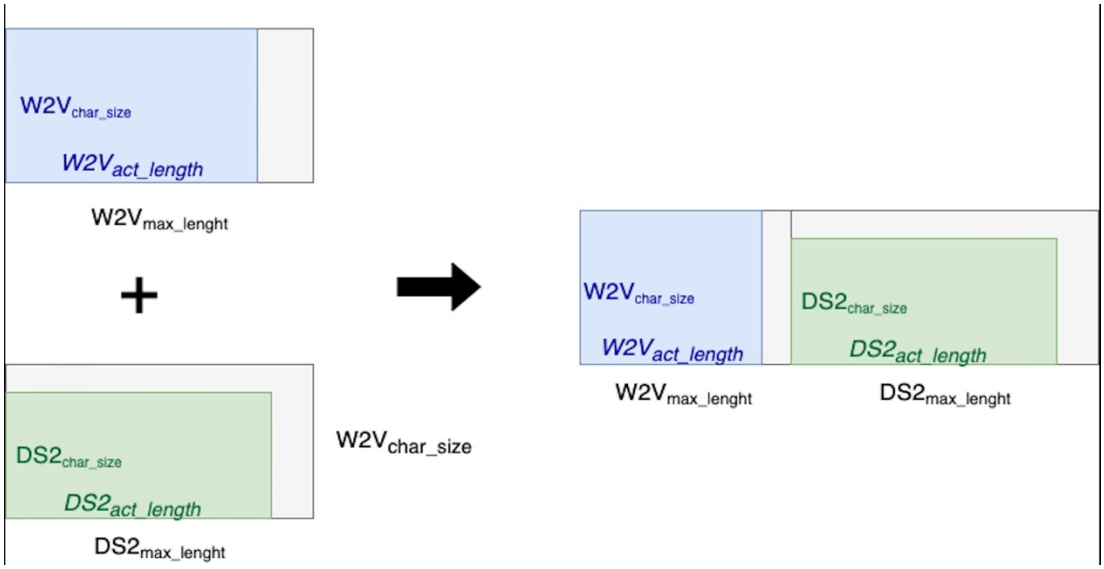
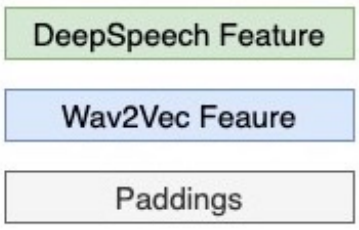


Figure 3.8 Combination of Wav2Vec 2.0 and DeepSpeech2 Features



3.4 Commonly Used Techniques In Both Models

As the dual-input CNN model(section 3.2) and feature-combined CNN model(section 3.3) are both CNN based models, there are common techniques we have applied in both methodologies

3.4.1 Convolutional Layer

It is the responsibility of the convolution layer to extract the critical local features from the inputs connected via the convoluted kernel implementation. The Equation *3.1* shows the process of convolution of the convolutional layer is expressed, where the activation function is denoted by f, the convolution process is denoted by*, w_i represents the weight tensor at $i-th$ neuron, x_i represents the input vector at $i-th$ neuron and, b denotes the bias value.

$$y = f(\sum w_i * x_i + b) \qquad \text{Equation 3.1}$$

3.4.2 Activation Function

In this study, all outputs of the convolution layer are subject to an activation function which is the rectified linear unit (ReLU). The linear behavior and scattered representation of ReLU showed computational efficiency in CNN models favoring LReLU. The multi-definition function is expressed in Equation *3.2* where a denotes the permissible negative slope coefficient and x and y are the inputs and outputs respectively. On the other hand, the Softmax activation function that computes the probability distribution of each type-AB over the total number of classes is applied to the expected result restricted within 0 and 1 on the output of the CNN model.

$$y = \begin{cases} x, & if\ x > 0 \\ ax, & if\ x \leq 0 \end{cases} \qquad \text{Equation 3.2}$$

3.4.3 Max-pooling and Dropout Layers

Pooling layers serve the purpose of gradually reducing feature dimensions to control computational complexity and over-fitting. Expressed as max-pooling, the output is the maximum element of non-overlapping partitioned subregions. As well as The dropout layer improves the generalization ability of the CNN model by randomly deleting a small percentage of neurons during training to prevent over-fitting.

4 DATA SET

The data set is used in previous experiments [8], and improved in [15]. This data set is based on the banking domain. In respect of the banking domain, six different queries are been selected for the previous studies, which are often used by customers in the event of interacting with the bank.

1. Checking the balance of an account.
2. Money deposit to an account.
3. Money withdraw from an account.
4. Paying bills.
5. Money transfer between 2 accounts.
6. Payment for credit card.

These are the 6 different intents which we are tring to predict based on the audio. This data set not only contains 6 different intents but also contains varieties in the way these queries are requested in speech form for each intent, which is referred as "inflections".

e.g.

- Domain Bank Intent - Checking the balance of an account
- Inflections – May I know the balance?, What is the my account balance?

Even though the original dataset contains both Tamil and Sinhala lingual data. We have only considered data in Tamil although it is code-mixed with some English terms. Table *4.1* briefly explains the data set. This data set contains only 400 entries added up to only 0.5 hours of training data. The audio clips are collected via mobile phone microphones, which allows us to simulate real-world scenarios.

Table 4.1 Details of the dataset

Intent	Inflections	Samples
Request Acc. Balance	7	101
Money Deposit	7	75

Money Withdraw	5	62
Bill Payments	4	46
Money Transfer	4	49
Credit Card Payments	4	67
Total	**31**	**400**
Unique Words	**46**	
Size in Hours	**0.5**	
Number of Speakers	**40**	

5 EXPERIMENT

For this experiment, we used pre-trained DeepSpeech2 model [4] with 11% WER on LibriSpeech corpus and Wav2Vec 2.0 model [5] with 7.4% WER on the same LibriSpeech corpus. We also used the phoneme-based model used in [15] for bench-marking our works.

These ASR models are used to extract the character probability distribution of each audio clip. Then this character probability distribution is used as the input for the CNN models which will classify the intent of the original audio clip.

We first experiment with only using one single ASR model each time with the CNN model benchmark the performance of the NLU model when each of these ASR models is used alone. Afterwards, we combined features extracted DeepSpeech2 and Wav2Vec 2.0 models with the proposed architecture in the methodology section 3. The Table *5.1* shows the details of the experiments.

Table 5.1 Details of the experiments

Experiment No.	**ASR Models Used**	**NLU Models Used**
Exp. 1	Phoneme	1D CNN
Exp. 2	DeepSpeech2	2D CNN
Exp. 3	Wav2Vec 2.0	1D CNN
Exp. 4	Wav2Vec 2.0	2D CNN
Exp. 5	DeepSpeech2 + Wav2Vec 2.0 (Combined Feature)	2D CNN
Exp. 6	DeepSpeech2 + Wav2Vec 2.0	Dual-Input CNN

5.1 Hyperparameter Tuning

We optimized only a selected number of parameters of the CNN models such as the number of filters, kernel size, and the dropout rate for each layer. We used the Bayesian search algorithm for our hyper-parameter tuning and smoothen uniform distribution to generate each value also known as *quniform* functions. A *quniform* function will return a value v as explained in the below Equation *5.1*

$$v = round\left(\frac{uniform(high,low)}{q}\right) * q \qquad \text{Equation 5.1}$$

This type of function is more suitable for a discrete variable with respect to which the objective is still smooth, but which should be bounded both above and below. We used this function to generate values for some of the model hyperparameters such as the following.

- Kernel size
- Pool size
- Stride
- Number of hidden units
- Dropout rate
- Batch size

Bayesian optimization [20] is a sequential model-based optimization algorithm (SMBO) that uses results from the previous iteration to select candidates for the next hyperparameter value. So instead of blindly searching in hyperparameter space as in grid search or random search, this method advocates using intelligence to choose the next set of hyperparameters that will improve model performance. this process is repeated until the algorithm is converged to an optimum value.

5.2 K-Fold Cross-Validation

Due to the low amount of training data, we had to be cautious not to overfit the model. In order to avoid overfitting, we used a k-fold cross-validation mechanism while measuring the performance of the model in the training phase. We used 5-fold in particular. K-fold cross-validation divides the entire data into k subgroups. while the model is only trained by k-1 sets of data, the k'th set is used to evaluate the performance of the model in that iteration. the iterations also continuous k times where the last k'th validation set is changed each time, allowing the model to be validated against all the

entries when all the k iterations are considered. when considering the overall performance, we average the accuracy of each fold.

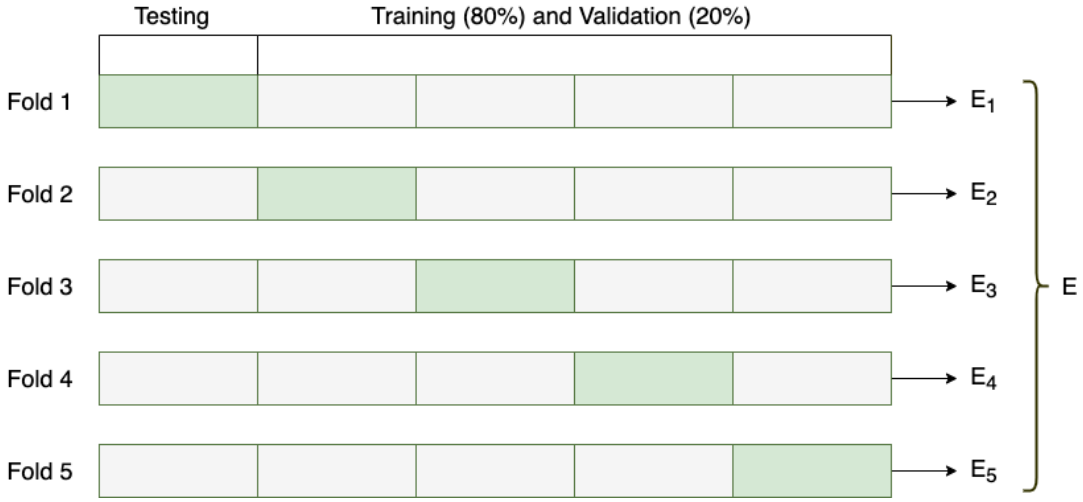


Figure 5.1 Process of K-Fold Cross-Validation

5.3 Experiment Setup

We primarily used python and Keras library for this study for both development and training purposes. We did not try to use any cloud-based infrastructure since the sheer volume of data is not that high and the local computer was able to handle the load for both the development and training of the CNN models. The local computer used has 32GB memory and has the 11th generation intel core i5.

6 RESULTS

Table 6.1 Details of the experiment results

Experiment No.	ASR Models Used	NLU Models Used	Accuracy
Exp. 1	Phoneme	1D CNN	81.35%
Exp. 2	DeepSpeech2	2D CNN	76.30%
Exp. 3	Wav2Vec 2.0	1D CNN	43.20%
Exp. 4	Wav2Vec 2.0	2D CNN	71.62%
Exp. 5	DeepSpeech2 + Wav2Vec 2.0 (Combined Feature)	2D CNN	**88.25%**
Exp. 6	DeepSpeech2 + Wav2Vec 2.0	Dual-Input CNN	83.50%

Table *6.1* shows the results obtained from the above sets of experiments. Experiment - 1 shows the benchmark accuracy we were able to reproduce from the previous works that reported the state-of-the-art accuracy [15]. The best performance we observed is from the 2D-CNN model fed with combined features from DeepSpeec2 and Wav2Vec 2.0. We were able to achieve 88.25% accuracy with this model which is higher than the current state-of-the-art performance 81.35% [15].

The proposed Dual-Model setup was also able to provide higher accuracy than the current state-of-the-art solution even though it is not a significant improvement. If either DeepSpeech2 or Wav2Vec 2.0 model is used to transfer learning, the NLU model is performing poorly than the phoneme-based solution. This leads us to believe the improvement we observed in the setup is not just because a better ASR model is been used to transfer learn from, but the technique we used combines the features generated from multiple ASR models.

Table *6.2* shows the F1 scores we obtained from each class for Experiment – 5 which had the highest accuracy.

Table 6.2 F1-Score per class

Class	F1 – Score	Number of Samples	Code mix rate
1	0.961190	101	3/7
2	0.866164	75	3/7
3	0.849077	62	1/5
4	0.768964	42	3/4
5	0.747006	49	2/4
6	0.964950	67	4/4

7 DISCUSSION

In this set of experiments, we were able to replicate the work and the accuracy of the current state-of-the-art method found in the previous study [15]. Apart from that, we were also able to produce similar performance from DeepSpeech2 character-based ASR models mentioned in the same study. When we tried to evaluate the performance of Wav2Vec 2.0, which is similar to the DeepSpeech2, we got almost similar results as DeepSpeech2. This validates that the improvement in the accuracy we got in experiments 5 and 6 is not just because of a better ASR model that is been used.

When we look into experiments 5 and 6, we can see an improvement over experiments 2 - 3 which only used a single ASR model, whereas experiments 5 and 6 used 2 ASR models. This indicates that the CNN model could learn additional features by combining the output from DeepSpeech2 and Wav2Vec 2.0 models.

The Table *7.1* explains the encoded character sequence produced by DeepSpeech2 and Wav2Vec 2.0 ASR models for a code-mix utterance. As we can see the predicted character sequence is completely different from each other. One common difference we observed is that DeepSpeech2 was able to capture similar-sounding English terms at the start of the sentence, but towards the end of the sentence, the predicted terms do not sound similar to the original audio at all. Meanwhile, the Wav2Vec 2.0 was able to predict the terms sounding similar to the audio towards the end of the sentence but at the start of the sentence, it does not output similar sounding as accurate as DeepSpeech2. This is one of the main reasons why these features output from the 2 models are complementary to each other and the model was able to learn more rich features when both features were provided.

Table 7.1 Difference in the model transcriptions for a given utterance (DS2 - DeepSpeech2, W2V - Wav2Vec 2.0)

Utterance	**DS2 Transcription**	**W2V Transcription**
Kācu innoru account ku mātta vēṇum (Money Transfer to another account)	causi nodid gon o coman theworom	cause he not i goin to gome out o un em
	care nnowi con ocomat_thean	caze in nore egon do cuma ta vonu
	casly an moda cound the mot thhe ra	cassly innude countemate wen
	cary nnodicont ccommatevernnam	carsi noricont comatavurno
	casi nnoo goncamatowerrn	casi norecon do gomant da verno

One other important outcome of we observed was that experiment 5 had higher accuracy than experiment 6. Experiment 5 was Method 2 - Feature and experiment 6 was Method 1 - Dual-input CNN Models explained in section 2. We observed that combining features before feeding to the CNN model increased the accuracy of the CNN model significantly than using a Dual-input CNN model. This can be due to the lower number of training, Since the dual-input CNN model has separate branching for individual features, it introduces more layers than the original CNN model in experiment 5. We need more data, at least more than 1000 entries to train these layers based on the study [15].

Overall, models which learned from the combined feature produced 2 ASR models, outperformed the state-of-the-art phoneme-based single ASR setup. We observed lower accuracy when any of the ASR models were used in a single ASR setup.

8 CONCLUSION AND FUTURE WORKS

This report presented a state-of-the-art setup for low-resource SLU tasks such as speech command recognition and topic identification. We were able to report 88.25% accuracy with less than 0.5 hours of Tamil audio clips. We identified that combining the features from pre-trained ASR models in the situation where the learned features are complementary, could significantly increase the accuracy of NLU models in low resource-transfer learning setups. Further, we have proposed 2 state-of-the-art architectures to combine the features from multiple ASR models.

In the future, we are hoping to apply this setup to different domains as well as different low-resource languages. We are also expanding this multi-ASR model to combine more than 2 ASR models and phoneme-based ASR models as well.

9 REFERENCES

[1] C. Liu, J. Trmal, M. Wiesner, C. Harman, S. Khudanpur, "Topic identification for speech without asr.," *in Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH,* Vols. 2017-August, 2017, pp. 2501–2505.

[2] A. Ram, R. Prasad, C. Khatri, A. Venkatesh, R. Gabriel, Q. Liu, J. Nunn, B. Hedayatnia, M. Cheng, A. Nagar et al, "Conversational ai: The science behind the alexa prize," *arXiv preprint arXiv:1801.03604, 2018.*

[3] Y.-P. Chen, R. Price, and S. Bangalore, "Spoken language understanding without speech recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE,* 2018, pp. 6189–6193..

[4] Amodei, Dario, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper et al, "Deep speech 2: End-to-end speech recognition in english and mandarin.," *In International conference on machine learning, pp. 173-182. PMLR, 2016..*

[5] Yi, Cheng, Jianzhong Wang, Ning Cheng, Shiyu Zhou, and Bo Xu, "Applying wav2vec2. 0 to speech recognition in various low-resource languages.," *arXiv preprint arXiv:2012.12121 (2020).*

[6] Povey, Daniel, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann et al, "The Kaldi speech recognition toolkit.," *In IEEE 2011 workshop on automatic speech recognition and understanding, no. CONF. IEEE Signal Processing Society, 2011.*

[7] Agarwal, Aashish, and Torsten Zesch, "LTL-UDE at Low-Resource Speech-to-Text Shared Task: Investigating Mozilla DeepSpeech in a low-resource setting.," *In SwissText/KONVENS. 2020.*

[8] Karunanayake, Yohan, Uthayasanker Thayasivam, and Surangika Ranathunga, "Transfer learning based free-form speech command classification for low-resource languages.," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pp. 288-294.,* 2019.

[9] Kunze, Julius, Louis Kirsch, Ilia Kurenkov, Andreas Krug, Jens Johannsmeier, and Sebastian Stober, "Transfer learning for speech recognition on a budget.," *arXiv preprint arXiv:1706.00290,* 2017.

[10] Misbullah, Alim, Kurnia Saputra, and Fauzy Nisa., "Customized Acoustic Model using Low-Resource Indonesian Speech Dataset for Short Command Speech Recognition System.," *2021 International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE), pp. 176-180. IEEE,* 2021.

[11] McGraw, Ian, Rohit Prabhavalkar, Raziel Alvarez, Montse Gonzalez Arenas, Kanishka Rao, David Rybach, Ouais Alsharif et al, ""Personalized speech recognition on mobile devices.," *In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5955-5959. IEEE,* 2016.

[12] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for endto-end spoken language understanding,," *arXiv preprint arXiv:1904.03670,* 2019.

[13] Kunze, Julius, Louis Kirsch, Ilia Kurenkov, Andreas Krug, Jens Johannsmeier, and Sebastian Stober, "Transfer learning for speech recognition on a budget.," *arXiv preprint arXiv:1706.00290,* 2017.

[14] Wang, Changhan, Juan Pino, and Jiatao Gu., "Improving cross-lingual transfer learning for end-to-end speech recognition with speech translation.," *arXiv preprint arXiv:2006.05474,* 2020.

[15] Karunanayake, Yohan, Uthayasanker Thayasivam, and Surangika Ranathunga, "Sinhala and tamil speech intent identification from english phoneme based asr.," *2019 International Conference on Asian Language Processing (IALP), pp. 234-239. IEEE,* 2019.

[16] Lugosch, Loren, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio, "Speech model pre-training for end-to-end spoken language understanding.," *arXiv preprint arXiv:1904.03670,* 2019.

[17] Chong, Thern Chang, Nien Loong Loo, Yeong Shiong Chiew, Mohd Basri Mat-Nor, and Azrina Md Ralib., "Classification Patient-Ventilator Asynchrony with Dual-Input Convolutional Neural Network.," *IFAC-PapersOnLine 54, no. 15 (2021),* pp. 322-327.

[18] Sun, Sukkyu, Ahnul Ha, Young Kook Kim, Byeong Wook Yoo, Hee Chan Kim, and Ki Ho Park., "Dual-input convolutional neural network for glaucoma diagnosis using spectral-domain optical coherence tomography.," *British Journal of Ophthalmology 105, no. 11 (2021): 1555-1560.*

[19] Wijayasingha, Lahiru, and John A. Stankovic., "Robustness to noise for speech emotion classification using CNNs and attention mechanisms.," *Smart Health 19 (2021): 100165.*

[20] Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems 25 (2012).*