Towards End-to-End Speech Recognition System for Pashto Language Using Transformer Model

Munaza Sher[1,2], Nasir Ahmad[1], Madiha Sher[1]
[1]University of Engineering and Technology.
[2]Bahria University Lahore Campus.
* **Correspondence**: Munaza Sher munazasher@gmail.com

Citation | Sher. M, Ahmad. N, Sher. M, "Towards end-to-end Speech Recognition System for Pashto Language using Transformer Model", IJIST, Vol. 6 Issue. 1 pp 115-131, Feb 2024
Received | Jan 18, 2024, **Revised |** Feb 15, 2024, **Accepted |** Feb 18, 2024, **Published |** Feb 25, 2024.

The conventional use of Hidden Markov Models (HMMs), and Gaussian Mixture Models (GMMs) for speech recognition posed setup challenges and inefficiency. This paper adopts the Transformer model for Pashto continuous speech recognition, offering an End-to-End (E2E) system that directly represents acoustic signals in the label sequence, simplifying implementation. This study introduces a Transformer model leveraging its state-of-the-art capabilities, including parallelization and self-attention mechanisms. With limited data for Pashto, the Transformer is chosen for its proficiency in handling constraints. The objective is to develop an accurate Pashto speech recognition system. Through 200 hours of conversational data, the study achieves a Word Error Rate (WER) of up to 51% and a Character Error Rate (CER) of up to 29%. The model's parameters are fine-tuned, and the dataset size increased, leading to significant improvements. Results demonstrate the Transformer's effectiveness, showcasing its prowess in limited data scenarios. The study attains notable WER and CER metrics, affirming the model's ability to recognize Pashto speech accurately. In conclusion, the study establishes the Transformer as a robust choice for Pashto speech recognition, emphasizing its adaptability to limited data conditions. It fills a gap in ASR research for the Pashto language, contributing to the advancement of speech recognition technology in under-resourced languages. The study highlights the potential for further improvement with increased training data. The findings underscore the importance of fine-tuning and dataset augmentation in enhancing model performance and reducing error rates.

Research Keys
Hidden Markov Models (HMMs)
Gaussian Mixture Models (GMMs)
End-to-End (E2E)
Word Error Rate (WER)
Character Error Rate (CER)
Automatic Speech Recognition (ASR)
Deep Neural Networks (DNNs)
Connectionist Temporal Classification (CTC)
Natural Language Processing (NLP)
Position Embedding (PE)
Mel-Frequency Cepstral Coefficients (MFCC)
Convolutional Neural Networks (CNN)
Intelligence Advanced Research Projects Activity (IARPA)
Pulse-Code Modulation (PCM)
Short-Time Fourier Transform (STFT)

Keywords: Hidden Markov Models (HMMs), Gaussian Mixture Models (GMMs), End-to-End (E2E), Character Error Rate (CER)

Introduction:

There are several ways to interact with computers, including visual, gesture-based, and speech-based interactions. Among these, using speech is the easiest and most natural. Speech recognition involves converting speech data into text, and there are different approaches to doing this automatically. Some systems are designed as speaker-dependent systems, while others can be speaker-independent. Some require a small vocabulary, while others can recognize a large set of words. Further characterization of Automatic Speech Recognition (ASR) system is into isolated words ASR system and continuous speech recognition system [1][2][3].

English is the most used language in the world, so most speech recognition research has focused on developing systems for that language. However, in recent years, attention has turned to low-resource languages [4][5][6][7]. Pashto is one of the most frequently spoken languages in the country, but many people in the Khyber Pakhtunkhwa (KPK) province where it is spoken as their mother tongue are not literate in English. This makes it difficult for them to use applications with English language interfaces. This research aims to improve human-machine interaction for Pashto speakers.

There have been few efforts to develop Pashto speech processing capabilities. The development of isolated speech recognition systems was one of the first steps in this field. There are two approaches to speech recognition: traditional ASR and E2E speech recognition. Traditional ASR involves three steps: an acoustic model, a pronunciation model, and a language model [8]. The acoustic model takes a representation of the audio signal, which is usually in the form of a waveform or spectrogram, and attempts to determine a probability distribution of phonemes over time windows of 10-80 ms throughout the entire recording. The output is not just a phonemic transcription, but a large number of possible phonemes over time. Next, the pronunciation model takes the phoneme lattice as input and tries to determine a probability distribution of words over time windows. This step also produces a large lattice of possible words as a function of time. Finally, a language model is used with a beam search to reduce the number of possible transcriptions. The language model takes the word lattice as input and eliminates possibilities that are deemed less likely until it reaches the final transcription. The beam search discards all possibilities below a certain cut-off at each time step, ensuring that they are not considered again. Each model is trained separately, and the process is time-consuming. E2E systems directly map input acoustic features into words or graphemes, and they are more efficient than traditional ASR systems. However, Developing a pronunciation lexicon and establishing phoneme sets for a particular language demands expert knowledge of that language and is a time-intensive process [9]. Acoustic model before the advent of deep learning, Hidden Markov Models (HMMs), and Gaussian Mixture Models (GMMs) were essential technologies used in dealing with speech signals. The Expectation Maximization (EM) algorithm was initially used as a crucial technique for training HMMs. HMMs are essentially Markov Chains that connect output symbols or probability functions to either states or transitions between states and are commonly used as acoustic models to determine the probability of different classes generating observation sequences or sentences. These models consist of hidden variables (states) and observables, with a Markov chain indicating the likelihood of transitioning from one state to another. HMM, modeling plays a critical role in transitioning between phonemes and the corresponding observable. Additionally, GMMs are utilized to model the distribution of features for phonemes, allowing for features to be modeled with a few possible values, and providing flexibility in speech variations [10].

Deep Neural Networks (DNNs) have established more effective outcomes as compared to GMMs in speech recognition because GMMs are not suitable for modeling nonlinear data or data that is close to it. Current ASR systems rely on a series of processing steps, like extracting noise robust input features, acoustic models, and HMM. However, deep learning can replace

these stages and achieve better performance in recognizing difficult speech. A complete end-to-end deep learning technique can substitute the whole pipeline of manually designed components with neural networks, making it possible to handle various speech types, including those in noisy environments, different languages, and accents. By utilizing deep neural networks, speech recognition accuracy can improve by 10 to 20 percent relative to that of GMMs, representing a significant improvement noticeable to speakers. Obtaining such good recognition rates in speech recognition tasks using neural networks can be considered the first generation of using deep learning approaches in speech recognition [11].

End-to-end automatic Speech Recognition simplifies traditional speech recognition by removing the requirement for manual labeling, as the neural network can automatically acquire language and vocabulary information. The Connectionist Temporal Classification (CTC) based method initially generates all feasible hard alignments and then uses them to obtain soft alignment. During the enumeration of hard alignments, CTC assumes that output labels are not dependent on each other [12]. On the other hand, the attention-based method utilizes the mechanism of attention to calculate the soft alignment information between the input data and the output label, bypassing the need to enumerate all possible hard alignments. The Transformer model is a neural network architecture initially introduced for machine translation tasks. The Transformer model has replaced Recurrent Neural Networks (RNNs) in various Natural Language Processing (NLP) tasks. Unlike RNNs, the Transformer model eliminates recurrence entirely and instead employs an internal attention mechanism, known as the self-attention mechanism, to determine the relevance of other sequences to a given input statement. It utilizes the mechanism of multi-head attention, which enables it to simultaneously attend to different sections of the input sequence. This generates features and an input statement that results from linear transformations of sequence features deemed significant based on the attention weights. This architecture can handle sequences of varying lengths and parallelize computations. The attention mechanism is now getting widely used component in neural networks and has been employed in various tasks, including generating captions for images [13][14], classifying text [15][16], translating languages [17][18][19][20], recognizing actions [21], analyzing images [22][23], recognizing speech, making recommendations [24][25], and working with graph data [26][27][28].

Pashto is an Indo-European language primarily spoken in northwestern Pakistan and Afghanistan, and it is recognized as one of Afghanistan's official national languages. Unlike other Arabic scripts, the Pashto alphabet comprises 46 letters, including some that are not found in any other Arabic script. Similar to Arabic dialects, most standard Pashto writing does not include many diacritics [29]. Pashto speech has a large number of dialects and accents, making it challenging to gather sufficient data for all cases. The Most common language on which a lot of work in the said domain is done on the English language because it is the most commendable language all over the world. In recent years, there has been a focus on languages with limited resources. A significant amount of work is required to be done on different languages specifically for languages that are spoken in our homeland Pakistan. Pashto is one of the most frequently spoken languages in the country. The literacy rate in Province KPK where the Pasho language is their mother tongue is just 50%, this depicts that almost half of the population is not able to use applications with an English language interface. Therefore, this research in the future will be helpful for Pashto speakers to have better human-machine interaction. In the past few years, very little effort has been made for the development of Pashto speech processing. In the Speech-processing domain for Pashto, language development started with the recognition of an isolated speech-recognition system [30][31][32]. The Transformer model comprises a single large block, which is made up of encoder and decoder blocks (as depicted in Figure 1).

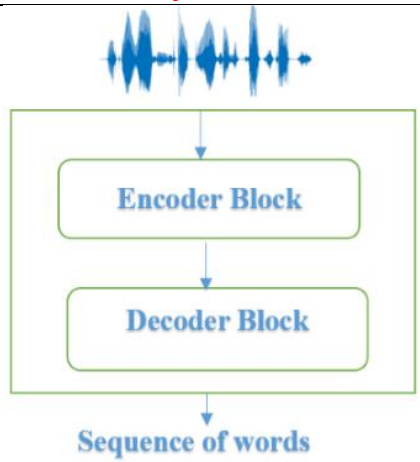

Figure 1: General Block Diagram of the model.

The encoder block receives feature vectors from the audio signal as input and generates an intermediate representation sequence based on these inputs. Subsequently, the decoder block uses these representations to produce the output sequence. Since the model predicts autoregressively, each stage employs the previous symbols to generate the next symbol. The Transformer architecture leverages multiple layers of self-attention in the encoder as well as decoder blocks, which are interconnected with each other. Whereas each block is considered separately.

In the Transformer model, attention is used to focus on relevant information based on the current processing. The attention mechanism is based on the decoder state (Q) and encoder hidden states (K and V), which calculate the attention weights that represent the relevance of the encoder hidden states (V) in processing the decoder state. The attention function, represented by Eq. 1, uses a query (Q), a set of key-value pairs (K, V), and calculates attention weights through a distribution function (F-distribution) and an alignment function (F-alignment). The most used distribution functions are the logistic, sigmoid, and SoftMax functions. The Transformer architecture also includes a feed-forward neural network to learn the attention weights as a function of the inputs from the decoder and encoder states. The alignment function assesses the significance of the encoder's hidden state about the decoder's hidden state, creating energy scores that are subsequently employed to compute attention weights via the distribution function [27].

$$\text{Attention}\,(Q,K,V) = F_{\text{distribution}}\left(F_{\text{alignment}}\,(K_i,Q)\right) * V_i \qquad (1)$$

Attention-based models allow for out-of-order treatment of encoded words, potentially producing a randomizing effect [33]. Unlike RNNs, the multi-head attention network can't naturally take advantage of the word order in the input sequence. To address this, the Transformer includes Position Embedding (PE) which encodes the positions of each word in the sequence [34]. PE is calculated using the sine and cosine functions based on the position of the word and the embedding dimension. As shown in Eq. 2(a) and 2(b). The PE is then added to the word embeddings to show the input position. Another form of position encoding is relative positional encoding, used by the Transformer-XL model, which provides relative positioning information for the keys and values in the attention.

$$PE\left(\text{pos}_{\text{word}}, 2 * \text{pos}_{\text{emb}}\right) = \sin\left(\frac{pos_{word}}{\frac{10{,}000^{2*pos_{emb}}}{d_{model}}}\right) \qquad 2(a)$$

$$PE\left(\text{pos}_{\text{word}}, 2 * \text{pos}_{emb} + 1\right) = \cos\left(\frac{pos_{word}}{\frac{10{,}000^{2*pos_{emb}}}{d_{model}}}\right) \qquad 2(b)$$

In speech recognition tasks, in traditional E2E the encoder converts acoustic feature vectors into a different form, and the decoder makes it possible to predict the label sequence from the data provided by the encoder using attention to focus on important parts of the frame. The Transformer model can incorporate multiple encoders and decoders, with each block having its respective internal attention mechanism.

An encoder block usually consists of 6 encoders stacked on top of each other, but the number of encoders can vary and be experimented with. All encoders in the Transformer model share the same structure but have different weights. The input to the encoder block is the feature vector, which is extracted from the speech signal using either Mel-Frequency Cepstral Coefficients (MFCC) or Convolutional Neural Networks (CNN). The first encoder employs self-attention to convert this data into a set of vectors, which are then passed to the next encoder through a feed-forward neural network. The final encoder block processes these vectors and sends the encoded data to the decoder block. In the decoder block of the Transformer model, there are multiple decoders, usually equal to the number of encoders. Among the two sublayers of the decoder, the initial sublayer is designed as a multi-head attention layer, which assists the decoder in considering the other words present in the input sequence while decoding a particular word. The output of this sublayer is then passed to a feed-forward neural network, which is applied independently to each word in the sequence. The second sublayer is also an attention layer that helps the decoder focus on the relevant parts of the input sequence, similar to the attention mechanism used in sequence-to-sequence models. This component considers previous words and gives the probability of the next word as output based on that information.

Material and Methods:

Data Set:

The Pashto speech used for experimentation represents that Spoken in four distinct dialect regions across Afghanistan and Pakistan. Intelligence Advanced Research Projects Activity (IARPA) Babel Pashto Language Pack is used for experimentation [35]. The dataset contains around 200 hours of Pashto speech collected through telephone conversations, both scripted and spontaneous, during the years 2011 and 2012. The phone calls were made from various environments, including public places, homes or offices, vehicles, and streets, using different types of phones such as mobile and landline phones. All the audio data is in sphere format, encoded as 8-bit a-law at 8kHz. Two versions of transcripts are available, one in an extended Arabic script, and the other in a modified Buckwalter transliteration scheme, both transcriptions are UTF-8 encoded. The data set is divided into 97% for training and 3% for testing. The creation of a phoneme dictionary is not needed for experimentation, only speech files along with the corresponding transcription are required.

Preprocessing of Audio Data:

To train a neural network for speech recognition, the training data must consist of pairs of corresponding sequences - an input audio signal and the corresponding correct text output [36]. Figure 2a. illustrates the steps involved in preprocessing the audio files. Since the original dataset was in sph audio format, all files were converted to the .wav format. During this process, the framework ensured that the sample rate was changed to 16000 and the Pulse-Code Modulation (PCM) was changed to 16-bit. Additionally, all silence was removed from the audio files. In this work, Short-Time Fourier Transform (STFT) is used. STFT is a commonly used mechanism of extraction of the speech signal in an audio signal analysis. In STFT, a signal is divided into overlapping frames and each frame is transformed into the frequency domain using the Fourier Transform [37]. This results in a representation of the signal in terms of its frequency components over time, which can be used for various purposes such as denoising, feature extraction, and classification.

Preprocessing of Transcript:

In Pashto language, the text is written from right to left, so the sentences were reversed to match the transcript file's requirements. Additionally, any numerical values in the dataset were converted into their verbal form, while special characters such as punctuation were eliminated. As a result, the final text contains only words. All words in sentences were separated by whitespaces. All the silence ie, labeled with <no speech> is removed from the transcription. Similarly, other labels like <laugh>,<cough>, <click>, etc are removed.

When building an ASR system using DNN, a major challenge arises from the fact that the audio and letter sequences are of unequal length. This means that the label sequence, i.e. the text, does not consider the duration of the corresponding audio clip. Consequently, an additional training step is required to stretch the sequence and fill it with blank spaces to align each letter in time with the sound it represents. Figure 2b illustrates some essential preprocessing steps that must be carried out on the data before inputting it into the model.

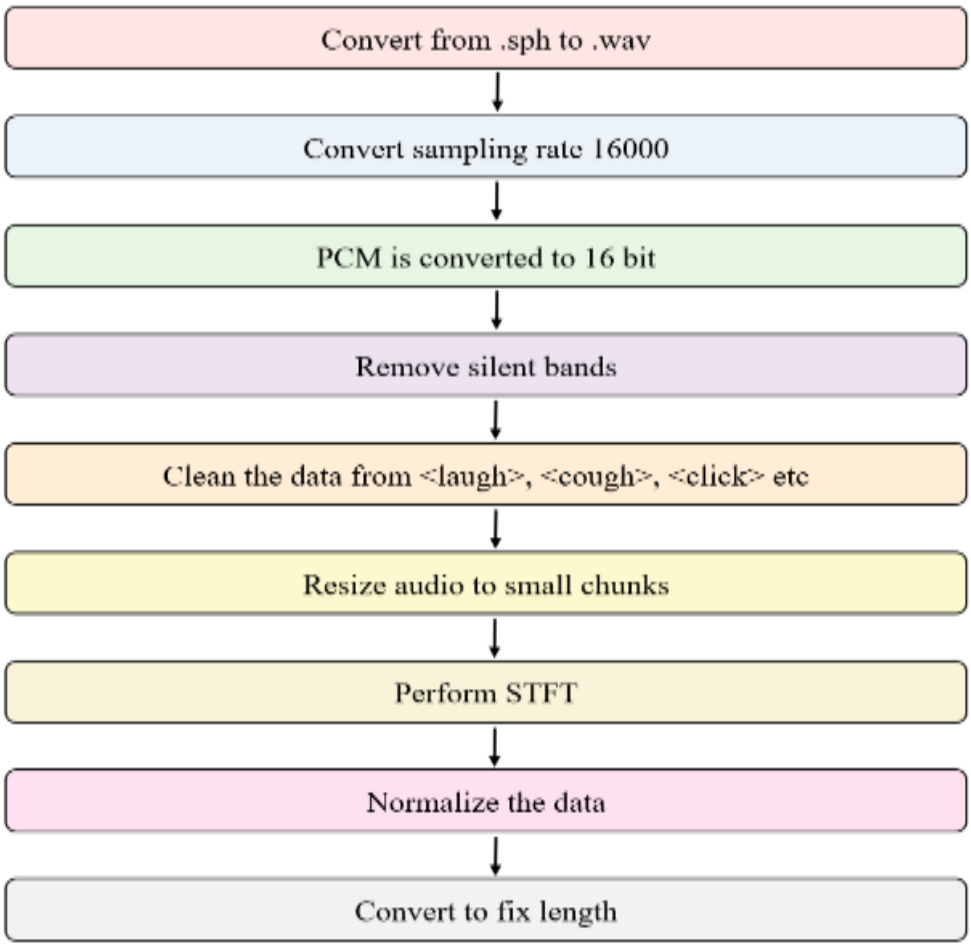

Figure 2a. Preprocessing of Audio signal

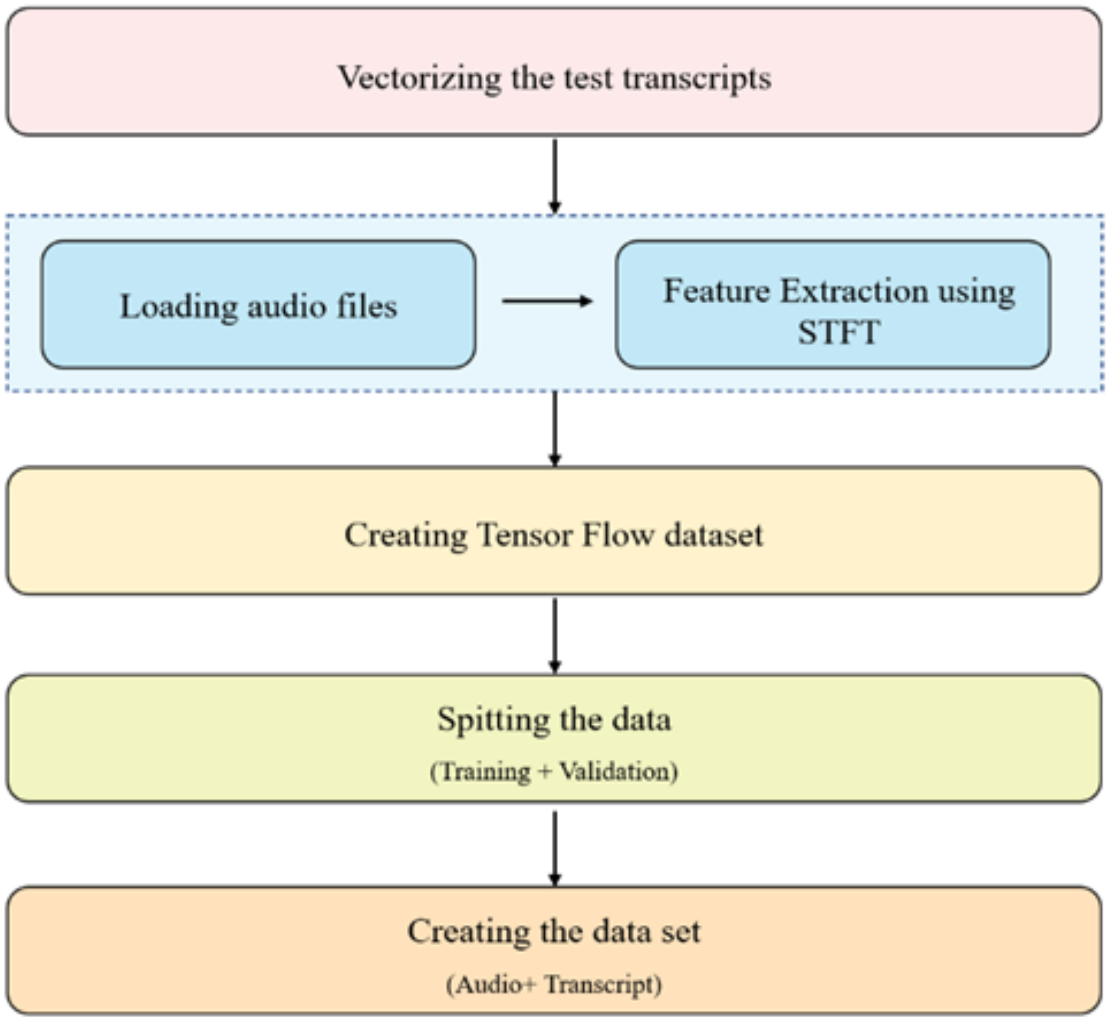

Figure 2b. Data preparation steps for training

Model Description:

The encoder of the transformer model is responsible for taking the input speech signal and converting it into a sequence of features that can be processed further. This is achieved using a series of convolutional layers that extract relevant acoustic features from the input signal. Figure 3. Shows that 3 convolutional layers are used, and its layer parameters are shown in Table 1. The input to the Transformer Encoder goes through a Multi-Head Attention layer, followed by a Layer Normalization and a Dropout layer. The output of this block is then fed into a feed-forward neural network consisting of two dense layers, which is then followed by another Layer Normalization and Dropout layer. Then self-attention mechanism is capable of directing its attention to various segments of the input sequence during distinct time intervals. It allows the network to dynamically adjust its attention to different parts of the input signal, depending on the current context. The decoder is responsible for taking the output of the encoder and converting it into a sequence of text tokens. Further details of parameters while performing experimentations are depicted in Table 2. After hyperparameters tuning the values are shown in Table 3. Were used for training the model.

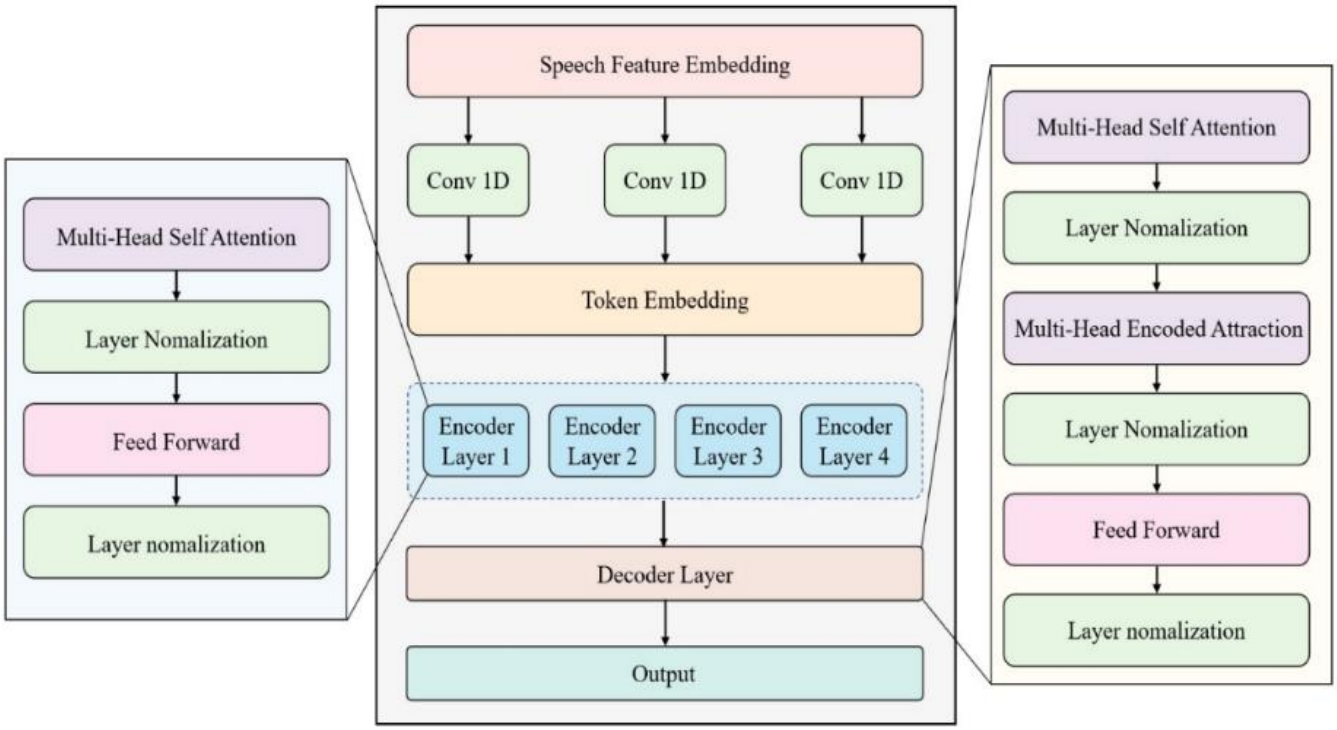

Figure 3: Transformer Model Architecture

Table 1: Layer parameters of the network

Name of Layer	Layer Parameter	Number of Layers
Convolutional	Num_hid=11, Strides=2, Activation Function="relu"	3
Feedforward Neural Network	Activation Function=”relu”	2
Transcription Layer Loss Function	Categorical Crossentropy	-

Table 2: Detail of model architecture

Parameters	Size
Number of hidden layers	200
Number of heads	2
Number of encoders	4
Number of decoders	1
Number of feed-forward networks	400

Table 3: Detail of hyperparameter

Hyperparameter	Value
Initial learning rate	0.00001
Learning rate after warm-up	0.001
Final learning rate	0.00001
Warmup epochs	15
Decay epochs	85
Number of epochs	100
Batch size	64
Dropout	0.5

Result and Discussion:

The performance of an ASR system is typically evaluated using the word error rate (WER) and character error rate (CER) metrics [38]. This metric calculates the percentage of errors made by the ASR system relative to the reference transcript. It is calculated in Eq 3. And Eq 4. A lower WER score is desired.

$$\text{WER (in \%)} = \frac{\text{Substitutions+ Deletions + Insertions}}{\text{Number of words in reference}} \text{ x } 100 \quad (3)$$

$$\text{CER (in\%)} = \frac{\text{Substitutions+ Deletions + Insertions}}{\text{Number of characters in reference}} \text{ x } 100 \quad (4)$$

Impact of Dataset Size on Training:

During experimentations, we have analyzed the influence of dataset size in different manners. Like, its effect on the training and validation loss curves of the proposed model. We have conducted experiments on both small and large datasets to comprehensively analyze the implications. One notable observation of experiments is the relationship between dataset size and the risk of overfitting. With smaller datasets, the model exhibited signs of overfitting, achieving lower training loss but struggling to generalize to unseen data. The scarcity of examples in smaller datasets led to over-optimization, emphasizing the importance of having an adequate amount of diverse data for robust model training. Smaller datasets demonstrated higher fluctuations in both training and validation loss curves. These fluctuations suggest that the model's performance is highly sensitive to the limited data available. In contrast, larger datasets contributed to more stable loss curves, indicating improved reliability in assessing the model's generalization capabilities. Figure 4 and Figure 5 show training vs validation loss curves for less data as well as adequate data. Loss curves provide valuable insights into the evolution of learning performance across epochs, aiding in the identification of potential issues that may result in either underfitting or overfitting of the model. Our experiments revealed intriguing insights into the convergence speed concerning dataset size. Models trained on smaller datasets tended to converge faster, possibly due to the reduced complexity of the training task. However, this expedited convergence came at the expense of compromised generalization. It took 7 hours on GPU to run for 100 epochs when training was done using 10 hours of data. While for 100-hour data it took 33 hours on GPU.

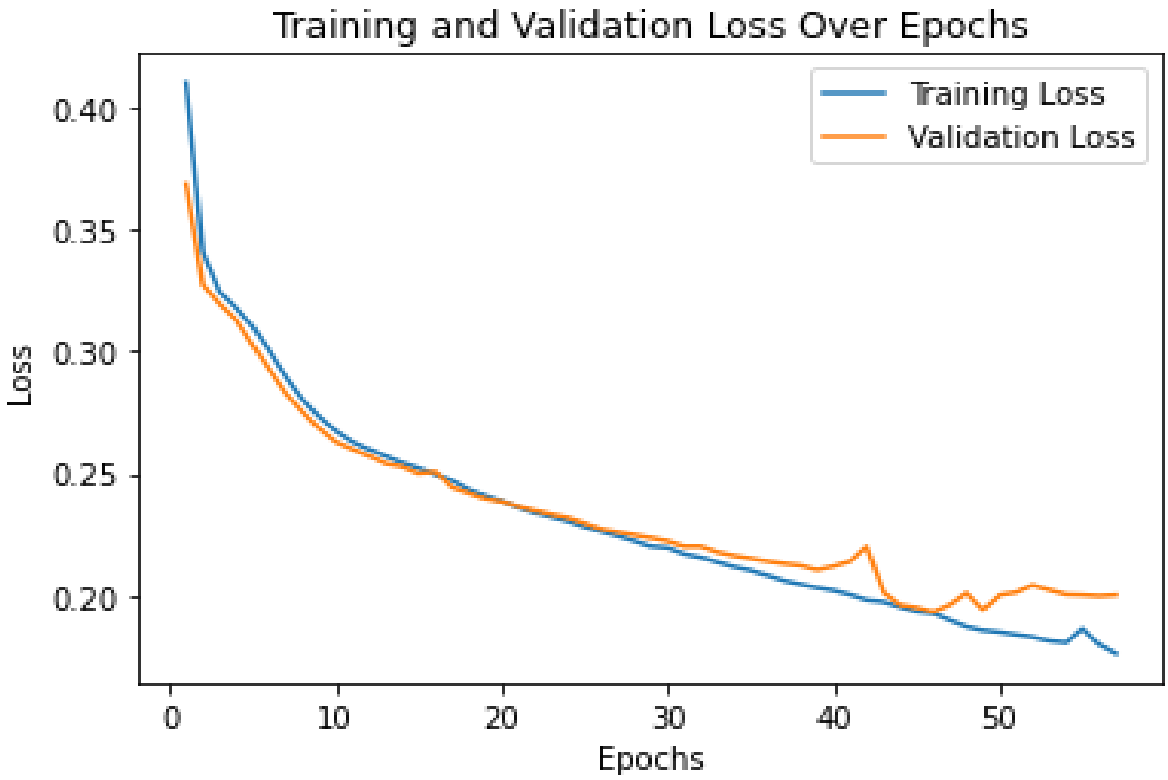

Figure 4: Loss Graph for 200 hr data

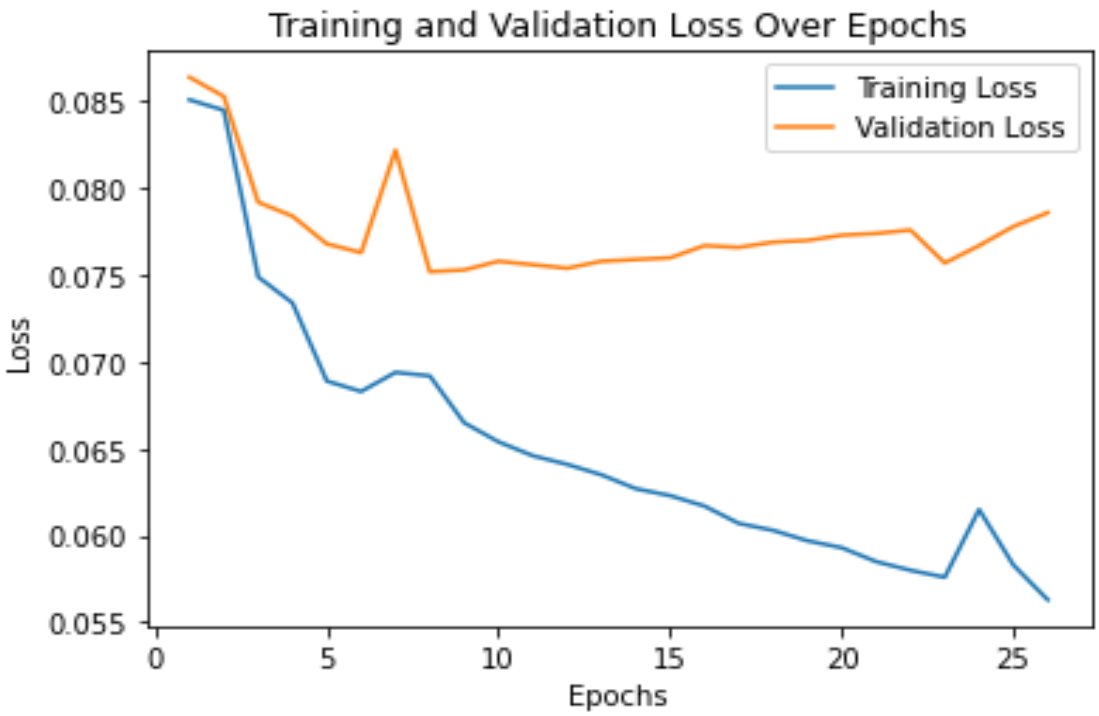

Figure 5: Loss Graph for 10 hr data

Impact of Batch Size on CPU Usage:

The batch size, representing the number of training samples processed in each iteration, was set to balance computational efficiency and memory constraints. A moderate batch size was chosen to allow for parallelization without compromising model stability. As we increase the batch size, the GPU is utilized more effectively. Larger batches often result in higher GPU usage during training. This is because more data is processed in parallel, exploiting the parallel processing capabilities of modern GPUs. However, increasing the batch size comes with memory requirements. Larger batches consume more GPU memory. When the batch size was too large for the GPU memory, it led to out-of-memory errors, preventing the model from training. However small batch sizes resulted in underutilization of the GPU, as it may not be fully occupied with computations. This led to lower efficiency and reduced throughput. On the other hand, extremely large batches lead to diminishing returns in terms of GPU utilization. There is an optimal batch size that balances GPU usage and computational efficiency. The figure 6 shows batch size vs GPU utilization.

Code is written in Python using KERAS which is a deep-learning API. Experiments were run on the server with NVIDIA RTX A6000. It took 7 hours on GPU to run for 100 epochs. In the experiments, corpora with varying amounts of data were utilized. The reduction in data quantity resulted in a decline in speech recognition accuracy. The model was overfitting by increasing the drop rate the problem was resolved. A series of experiments with varying dropout rates to understand its influence on the recognition task was conducted. Dropout rates of 0.2, 0.5, and 0.8 were selected to represent low, moderate, and high dropout scenarios. Surprisingly, the recognition accuracy significantly degraded when a high dropout rate of 0.8 was employed. This result contrasts with the common expectation that dropout generally improves generalization. Different learning rates were experimented best results were acquired with a learning rate of 0.00001. Similarly, nominal results were gained after training to 100 epochs. Models based on E2E models have a reliable recognition tendency but suffer from a delay due to the need to wait for each audio utterance to process data. This delay restricts their use in real-time speech recognition, making it essential to consider methods that can handle streaming

speech to address this limitation. An example of text sequence predicted by the model compared with the ground-truth at or around 100 epochs is depicted in Table 4, Table 5, and Table 6: when tested on unseen data WER achieved 76% with 10 hours of training data. When trained on 200 hours of data WER improved to 51 % and CER 29%. When tested on training data CER 14.42% and WER 9.41% were achieved.

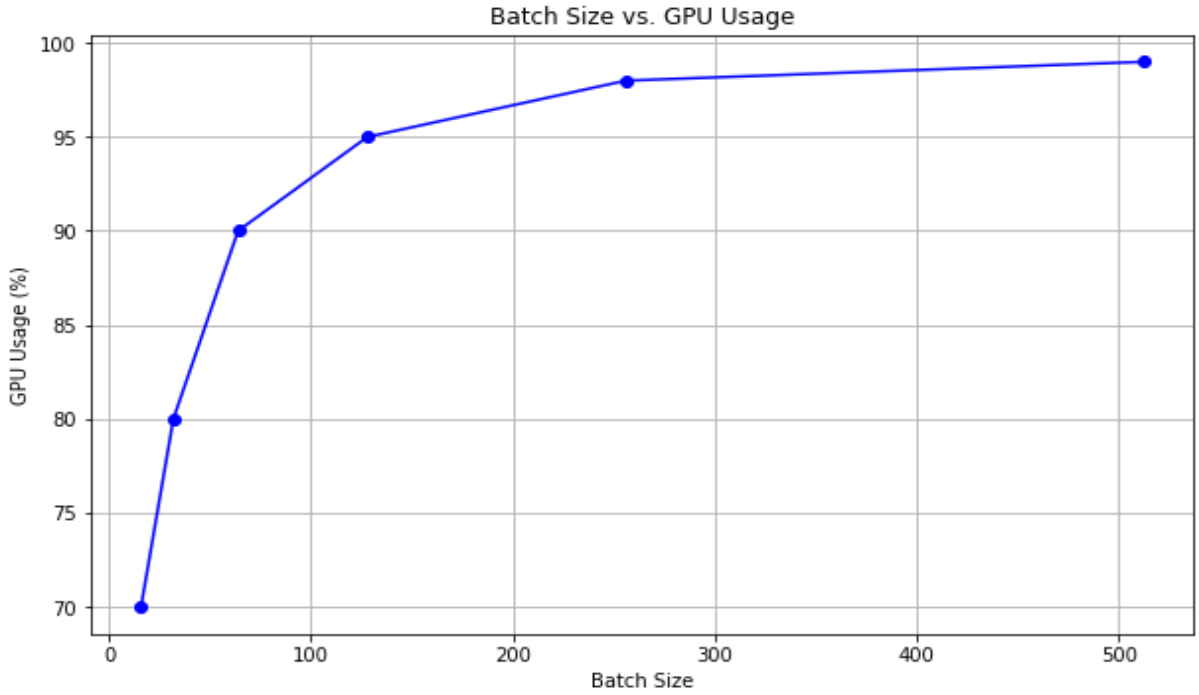

Figure 6: Batch size vs. GPU Usage

Table 4:10 hours training data tested on training data WER = 18.42 CER= 29.41

Target Text	Predicted Text
<# هلو>	<# هلو>
<# السلام عليكم>	<# السلام عليكم>
<دوولسم کښې يې کوم مضمون دي ډېر خوښ دي>	<دوولسم کښې يې کوم مريض شوی ديني دغه کښې>
<ډېر ښايسته سبق ورکوي>	< ډېر ښايسته سبق ورکوي>

Table 5: 10 hours of training data tested on unseen test data WER = 76 CER= 56

Target Text	Predicted Text
<# هلو>	<# هلو>
<# السلام عليكم>	<# السلام عليكم>
<دوولسم کښې يې کوم مضمون دي ډېر خوښ دي>	<دوولسم کښې يې کوم مريض شوی ديني دغه کښې>
<ډېر ښايسته سبق ورکوي>	< ډېر ښايسته سبق ورکوي>

Table 6: 200 hours training data tested on unseen test data WER = 66 CER= 49

Target Text	Predicted text
< څنګه >	< څنګه >
< او يو ما سره ولې نه شته دی دغه حماد	او يو دغه خو مو ما ويل چې دغه خو مو >
< پېژني حماد را سره شته	< ما ويل سره ما ويل سره شته
< نو کوم کالج کښې يې>	< نو کوه که نه کښې يې>

In the experiments, corpora with varying amounts of data were utilized. The reduction in data quantity resulted in a decline in speech recognition accuracy. The model was overfitting by increasing the drop rate the problem was resolved. Different learning rates were experimented best results were acquired with a learning rate of 0.00001. Similarly, nominal results were gained after training to 100 epochs. Models based on E2E models have a reliable

recognition tendency but suffer from a delay due to the need to wait for each audio utterance to process data. This delay restricts their use in real-time speech recognition, making it essential to consider methods that can handle streaming speech to address this limitation. Smaller datasets demonstrated higher fluctuations in both training and validation loss curves. These fluctuations suggested that the model's performance is highly sensitive to the limited data available. Our experiments revealed intriguing insights into the convergence speed concerning dataset size. Models trained on smaller datasets tended to converge faster, possibly due to the reduced complexity of the training task. However, this expedited convergence came at the expense of compromised generalization. It took 7 hours on GPU to run for 100 epochs when training was done using 10 hours of data. While for 200-hour data it took 33 hours on GPU. Our experimentation also showed larger batches consume more GPU memory. when the batch size was too large for the GPU memory, it led to out-of-memory errors, preventing the model from training. However small batch sizes resulted in underutilization of the GPU, as it may not be fully occupied with computations. This leads to lower efficiency and reduced throughput. On the other hand, extremely large batches lead to diminishing returns in terms of GPU utilization. There is an optimal batch size that balances GPU usage and computational efficiency. Experimental results for small sentences, defined as sentences with less than 8 words, the model consistently exhibited improved performance as the number of training epochs increased. Conversely, for larger sentences (defined as sentences with more than 20 words) results were degraded. Similarly, experimentations showed that increasing the number of decoders also resulted in degradation of results and out-of-memory issues as well.

Effect of Increasing Number of Decoders:

The outcomes of the experiments investigating the influence of varying the number of decoders in the model are also tested. The objective of these experiments was to assess the performance improvements that could be achieved through the augmentation of model complexity. However, contrary to the initial hypothesis, the results revealed a discernible deterioration in overall performance as the number of decoders increased. The experimental setup involved systematically increasing the number of decoders in the Transformer architecture while keeping other hyperparameters constant. Performance metrics, including Word Error Rate (WER), were employed to evaluate the impact on the speech recognition system. Graphs in Figure 7 show overfitting occurred when a greater number of decoders were used and the best results were achieved with a single decoder. The increased model complexity may have introduced challenges in training, potentially resulting in overfitting or difficulties in capturing relevant acoustic features. This unexpected trend challenges the conventional assumption that a more complex model with additional decoders would inherently yield better results. It is plausible that the deeper architecture introduced challenges in training, leading to difficulties in capturing relevant speech features.

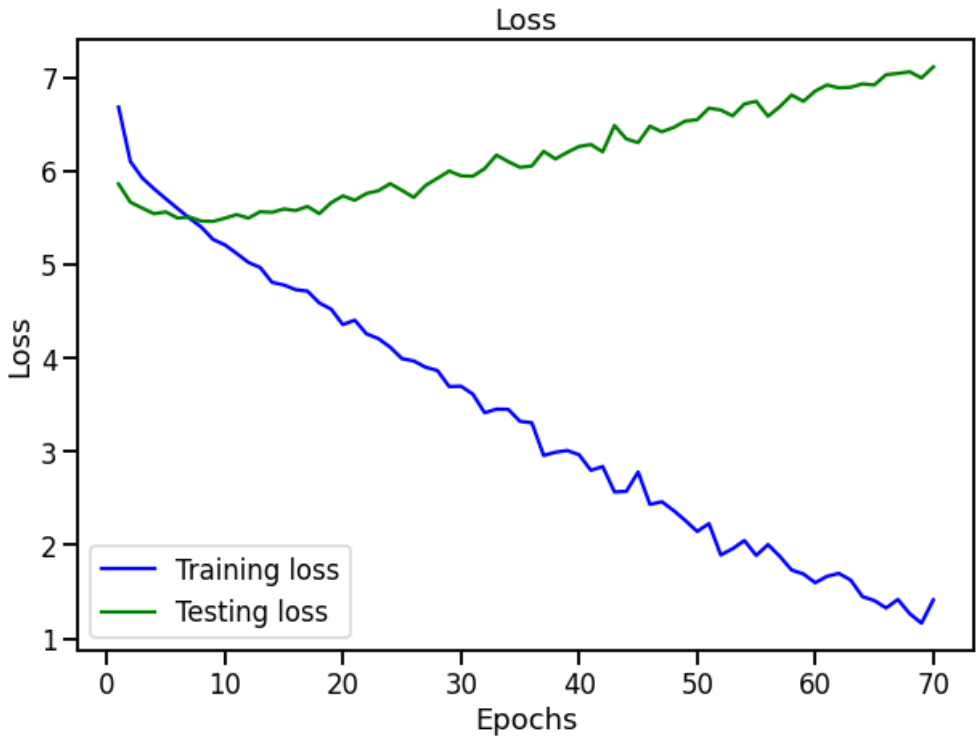

Figure 7: Overfitting because of a greater number of decoders

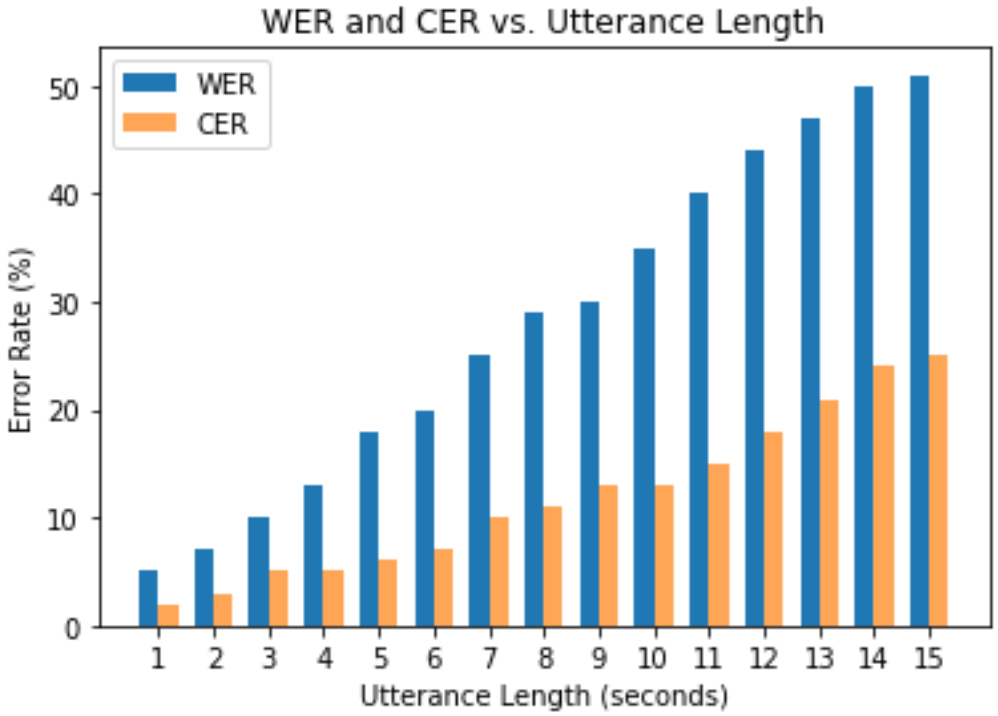

Figure 8: WER and CER vs Utterance length

Impact of Size of Utterance Length:

Experimental results for small sentences, defined as sentences with less than 8 words, the model consistently exhibited improved performance as the number of training epochs increased. Conversely, for larger sentences (defined as sentences with more than 20 words), the gains observed with additional epochs were marginal. This suggests that the model is able to accurately transcribe short and concise utterances. Our analysis revealed the superior

performance of the model when handling short utterances. Figure 8 illustrates the recognition accuracy for different utterance lengths, showcasing a distinct advantage for utterances with fewer temporal dependencies. On the other hand, the model encountered challenges when confronted with longer utterances. While the recognition accuracy remained competitive for short and medium-length utterances, a noticeable decline was observed as the temporal span of the utterances increased. The observed performance trends suggest a potential sensitivity of the model to the temporal context inherent in longer utterances.

Conclusion:

The traditional approach of building a speech recognition system using an acoustic model based on HMM with GMM made it hard to set up and configure, leading to a reduction in efficiency. In this paper, the Transformer model is used as an E2E system for Pashto continuous speech recognition. The benefit of using an E2E system for speech recognition is, that this method made it enabled to directly represent acoustic signals in the label sequence without any intermediary steps, thereby eliminating the need for additional processing at the output stage, which simplifies implementation. WER 51% is achieved which can be further improved with the increase of training data, as experiments showed WER improved from 76% to 51% with an increase of data. Our experimental findings align with prior studies that indicate telephonic conversations generate more noise due to overlapping voices and background sounds, thereby reducing the precision of the model output.

Key Findings:

The E2E system proves effective in capturing complex acoustic patterns inherent in Pashto speech, simplifying the recognition process and contributing to a substantial reduction in WER compared to conventional models. The positive correlation between the increase in training data and the improvement in WER underscores the importance of a comprehensive and diverse speech corpus. As the dataset grows, the model's ability to generalize across varied linguistic scenarios is enhanced.

Future Directions:

While the current study has demonstrated promising results with a WER of 51%, future investigations could explore the application of data augmentation techniques to further enhance the model's robustness. Data augmentation involves introducing variations in the training data, such as pitch shifts, speed perturbations, and background noise addition. Implementing these techniques may lead to improved generalization and performance, especially in diverse and noisy real-world scenarios. The data set available is still insufficient to get good training and data augmentation techniques may help in improving training accuracy. Addressing the challenges faced by telephonic conversations with overlapping voices and background sounds requires dedicated efforts. Future work could focus on developing and implementing advanced noise reduction strategies tailored to the characteristics of Pashto speech. This may involve exploring state-of-the-art denoising algorithms and adapting them to the unique acoustic features of the language.

The study indicates a positive correlation between the increase in training data and the improvement in WER. Future work should prioritize the collection and incorporation of a more extensive and diverse Pashto speech corpus. This expansion can contribute to a richer acoustic model, potentially resulting in higher accuracy. Although it's a tiring process collaboration with linguistic experts and native speakers can aid in the collection of high-quality, noise-free speech data. The availability of high-performance computing resources plays a pivotal role in the scalability and efficiency of training deep neural networks. Future research endeavours should explore partnerships or access to advanced computing facilities, enabling the utilization of larger model architectures, optimization techniques, and faster training processes.

Acknowledgement:

I wish to extend my heartfelt appreciation to Ms. Madiha Sher for her invaluable technical support throughout this research. Her expertise and insightful contributions have played a pivotal role in enhancing the overall quality of this study.

Author's Contribution:

Ms. Madiha Sher played a crucial role as a technical guide, providing valuable insights and expertise that greatly influenced the research process. Additionally, Dr. Nasir Ahmad my supervisor contributed immensely to the project by offering guidance in project management, assisting with meeting deadlines, and providing valuable proofreading insights.

Conflict of Interest: There exists no conflict of interest for publishing this manuscript in IJIST among authors.

Project Details: If this research was not conducted as a result of a project.

References:

[1] A. P. Singh, R. Nath, and S. Kumar, "A Survey: Speech Recognition Approaches and Techniques," 2018 5th IEEE Uttar Pradesh Sect. Int. Conf. Electr. Electron. Comput. Eng. UPCON 2018, Dec. 2018, doi: 10.1109/UPCON.2018.8596954.

[2] C. Kim et al., "End-To-End Training of a Large Vocabulary End-To-End Speech Recognition System," 2019 IEEE Autom. Speech Recognit. Underst. Work. ASRU 2019 - Proc., pp. 562–569, Dec. 2019, doi: 10.1109/ASRU46091.2019.9003976.

[3] D. Wang, X. Wang, and S. Lv, "An Overview of End-to-End Automatic Speech Recognition," Symmetry 2019, Vol. 11, Page 1018, vol. 11, no. 8, p. 1018, Aug. 2019, doi: 10.3390/SYM11081018.

[4] S. Khare, A. Mittal, A. Diwan, S. Sarawagi, P. Jyothi, and S. Bharadwaj, "Low Resource ASR: The Surprising Effectiveness of High Resource Transliteration," Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, vol. 2, pp. 1529–1533, 2021, doi: 10.21437/INTERSPEECH.2021-2062.

[5] S. P. Rath, K. M. Knill, A. Ragni, and M. J. F. Gales, "Combining tandem and hybrid systems for improved speech recognition and keyword spotting on low resource languages," Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, pp. 835–839, 2014, doi: 10.21437/INTERSPEECH.2014-212.

[6] O. Scharenborg et al., "Building an ASR System for a Low-research Language Through the Adaptation of a High-resource Language ASR System: Preliminary Results".

[7] Z. Tüske, P. Golik, D. Nolden, R. Schlüter, and H. Ney, "Data augmentation, feature combination, and multilingual neural networks to improve ASR and KWS performance for low-resource languages," Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, pp. 1420–1424, 2014, doi: 10.21437/INTERSPEECH.2014-348.

[8] M. Y. Tachbelie and L. Besacier, "Using different acoustic, lexical and language modeling units for ASR of an under-resourced language – Amharic," Speech Commun., vol. 56, no. 1, pp. 181–194, Jan. 2014, doi: 10.1016/J.SPECOM.2013.01.008.

[9] D. Rudolph Van Niekerk, "AUTOMATIC SPEECH SEGMENTATION WITH LIMITED DATA".

[10] P. Swietojanski, A. Ghoshal, and S. Renals, "Revisiting hybrid and GMM-HMM system combination techniques," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., pp. 6744–6748, Oct. 2013, doi: 10.1109/ICASSP.2013.6638967.

[11] P. Motlicek, D. Imseng, B. Potard, P. N. Garner, and I. Himawan, "Exploiting foreign resources for DNN-based ASR," Eurasip J. Audio, Speech, Music Process., vol. 2015, no. 1, pp. 1–10, Dec. 2015, doi: 10.1186/S13636-015-0058-5/TABLES/5.

[12] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," ICASSP, IEEE Int. Conf. Acoust. Speech Signal

Process. - Proc., pp. 4835–4839, Jun. 2017, doi: 10.1109/ICASSP.2017.7953075.
[13] S. Ding, S. Qu, Y. Xi, A. K. Sangaiah, and S. Wan, "Image caption generation with high-level image features," Pattern Recognit. Lett., vol. 123, pp. 89–95, May 2019, doi: 10.1016/J.PATREC.2019.03.021.
[14] H. Tsaniya, C. Fatichah, and N. Suciati, "Transformer Approaches in Image Captioning: A Literature Review," ICITEE 2022 - Proc. 14th Int. Conf. Inf. Technol. Electr. Eng., pp. 280–285, 2022, doi: 10.1109/ICITEE56407.2022.9954086.
[15] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," Neurocomputing, vol. 337, pp. 325–338, Apr. 2019, doi: 10.1016/J.NEUCOM.2019.01.078.
[16] Y. Li, L. Yang, B. Xu, J. Wang, and H. Lin, "Improving User Attribute Classification with Text and Social Network Attention," Cognit. Comput., vol. 11, no. 4, pp. 459–468, Aug. 2019, doi: 10.1007/S12559-019-9624-Y/METRICS.
[17] M. A. Di Gangi, M. Negri, and M. Turchi, "Adapting Transformer to End-to-End Spoken Language Translation," Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, vol. 2019-September, pp. 1133–1137, 2019, doi: 10.21437/INTERSPEECH.2019-3045.
[18] D. Britz, A. Goldie, M. T. Luong, and Q. V. Le, "Massive Exploration of Neural Machine Translation Architectures," EMNLP 2017 - Conf. Empir. Methods Nat. Lang. Process. Proc., pp. 1442–1451, 2017, doi: 10.18653/V1/D17-1151.
[19] A. Rahali and M. A. Akhloufi, "End-to-End Transformer-Based Models in Textual-Based NLP," AI 2023, Vol. 4, Pages 54-110, vol. 4, no. 1, pp. 54–110, Jan. 2023, doi: 10.3390/AI4010004.
[20] Y. Zhang, P. Wu, H. Li, Y. Liu, F. E. Alsaadi, and N. Zeng, "DPF-S2S: A novel dual-pathway-fusion-based sequence-to-sequence text recognition model," Neurocomputing, vol. 523, pp. 182–190, Feb. 2023, doi: 10.1016/J.NEUCOM.2022.12.034.
[21] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data," Proc. AAAI Conf. Artif. Intell., vol. 31, no. 1, pp. 4263–4270, Feb. 2017, doi: 10.1609/AAAI.V31I1.11212.
[22] X. Yan, S. Hu, Y. Mao, Y. Ye, and H. Yu, "Deep multi-view learning methods: A review," Neurocomputing, vol. 448, pp. 106–129, Aug. 2021, doi: 10.1016/J.NEUCOM.2021.03.090.
[23] K. Song, T. Yao, Q. Ling, and T. Mei, "Boosting image sentiment analysis with visual attention," Neurocomputing, vol. 312, pp. 218–228, Oct. 2018, doi: 10.1016/J.NEUCOM.2018.05.104.
[24] S. Ahmadian, M. Ahmadian, and M. Jalili, "A deep learning based trust- and tag-aware recommender system," Neurocomputing, vol. 488, pp. 557–571, Jun. 2022, doi: 10.1016/J.NEUCOM.2021.11.064.
[25] R. Wang, Z. Wu, J. Lou, and Y. Jiang, "Attention-based dynamic user modeling and Deep Collaborative filtering recommendation," Expert Syst. Appl., vol. 188, p. 116036, Feb. 2022, doi: 10.1016/J.ESWA.2021.116036.
[26] J. Feng et al., "Crowd Flow Prediction for Irregular Regions with Semantic Graph Attention Network," ACM Trans. Intell. Syst. Technol., vol. 13, no. 5, Jun. 2022, doi: 10.1145/3501805.
[27] S. Liang, A. Zhu, J. Zhang, and J. Shao, "Hyper-node Relational Graph Attention Network for Multi-modal Knowledge Graph Completion," ACM Trans. Multimed. Comput. Commun. Appl., vol. 19, no. 2, Feb. 2023, doi: 10.1145/3545573.
[28] X. Yan, Y. Guo, G. Wang, Y. Kuang, Y. Li, and Z. Zheng, "Fake News Detection Based

on Dual Graph Attention Networks," Lect. Notes Data Eng. Commun. Technol., vol. 89, pp. 655–666, 2022, doi: 10.1007/978-3-030-89698-0_67/COVER.

[29] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," Speech Commun., vol. 56, no. 1, pp. 85–100, Jan. 2014, doi: 10.1016/J.SPECOM.2013.07.008.

[30] I. Ahmed, H. Ali, N. Ahmad, and G. Ahmad, "The development of isolated words corpus of Pashto for the automatic speech recognition research," 2012 Int. Conf. Robot. Artif. Intell. ICRAI 2012, pp. 139–143, 2012, doi: 10.1109/ICRAI.2012.6413380.

[31] Z. Ali, A. W. Abbas, T. M. Thasleema, B. Uddin, T. Raaz, and S. A. R. Abid, "Database development and automatic speech recognition of isolated Pashto spoken digits using MFCC and K-NN," Int. J. Speech Technol., vol. 18, no. 2, pp. 271–275, Jun. 2015, doi: 10.1007/S10772-014-9267-Z/METRICS.

[32] B. Zada and R. Ullah, "Pashto isolated digits recognition using deep convolutional neural network," Heliyon, vol. 6, no. 2, p. e03372, Feb. 2020, doi: 10.1016/j.heliyon.2020.e03372.

[33] Z. Alyafeai, M. S. AlShaibani, and I. Ahmad, "A Survey on Transfer Learning in Natural Language Processing," arXiv, pp. 6523–6541, May 2020, Accessed: Feb. 16, 2024. [Online]. Available: https://arxiv.org/abs/2007.04239v1

[34] Y. Li, S. Si, G. Li, C. J. Hsieh, and S. Bengio, "Learnable Fourier Features for Multi-Dimensional Spatial Positional Encoding," Adv. Neural Inf. Process. Syst., vol. 19, pp. 15816–15829, Jun. 2021, Accessed: Feb. 16, 2024. [Online]. Available: https://arxiv.org/abs/2106.02795v3

[35] M. Cai et al., "High-performance Swahili keyword search with very limited language pack: The THUEE system for the OpenKWS15 evaluation," 2015 IEEE Work. Autom. Speech Recognit. Understanding, ASRU 2015 - Proc., pp. 215–222, Feb. 2016, doi: 10.1109/ASRU.2015.7404797.

[36] P. E. Solberg, P. Beauguitte, P. E. Kummervold, and F. Wetjen, "A Large Norwegian Dataset for Weak Supervision ASR." pp. 48–52, 2023. Accessed: Feb. 16, 2024. [Online]. Available: https://aclanthology.org/2023.resourceful-1.7

[37] P. Janbakhshi and I. Kodrasi, "EXPERIMENTAL INVESTIGATION ON STFT PHASE REPRESENTATIONS FOR DEEP LEARNING-BASED DYSARTHRIC SPEECH DETECTION," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., vol. 2022-May, pp. 6477–6481, 2022, doi: 10.1109/ICASSP43922.2022.9747205.

[38] W. Chan, N. Jaitly, Q. V Le, and V. Google Brain, "Listen, Attend and Spell," Aug. 2015, Accessed: Feb. 16, 2024. [Online]. Available: https://arxiv.org/abs/1508.01211v2