Automatic Speech Recognition and Text to Speech for Low Resource Languages

PH 303: Supervised Learning Project

Vineet Bhat
180260042
Guide: Professor Pushpak Bhattacharyya
Co-guide: Professor Alok Shukla

May 13, 2021

Abstract

Automatic Speech Recognition (ASR) and Text to Speech (TTS) research have been boosted in recent years with the development of deep learning systems. These systems rely heavily on the availability of large amounts of data which is possible only in widely spoken languages such as English, Spanish, and German. For developing such systems in low resource languages, we must rely on shared knowledge across languages and the ability of deep learning systems to finetune for specific tasks. In this report, we explore the problem statements of Automatic Speech Recognition and Text to Speech in detail, emphasizing working in Low Resource Languages. We present transfer learning-based approaches in developing a high-quality TTS system for Hindi and Marathi and a promising ASR system for Swahili. We show the validity of finetuning large models on the smaller annotated corpus to jump-start the training process, which is essential in low resource languages. We also share our code repository for both tasks to boost future research.

1 Introduction

Natural Language Processing (NLP) is the branch of Artificial Intelligence that deals with the interaction between computers and humans using natural language. Historically, the study of NLP has been conducted over the past 50 years with advancements due to the merger of increased computational capacity and insights into linguistics. The ultimate objective of NLP is to read, decipher, understand, and make sense of the human language helpfully. Modern NLP relies heavily on Machine Learning and Deep Learning algorithms for its various problem statements. Some of the typical applications of NLP are –

1. Sentiment Analysis – Sentiment Analysis is the problem of understanding the language and grasping the emotions associated with the input text.
2. Text Classification – Text classification is a broader category compared to sentiment analysis which involves understanding the meaning of the input and categorizing it into user-defined categories based on specific features of the language.
3. Chatbots – Chatbots and virtual assistants are one of the most upcoming products, especially for commercial applications such as automatic question answering systems, customer service, etc.
4. Machine Translation – One of the very first applications of NLP, Machine Translation is the task of converting input text in Language A to output text in Language B by a computer
5. Automatic Speech Recognition (ASR) – ASR is the part of NLP that focuses on developing methods that allow the translation of spoken language in audio format into text by computers.
6. Text to Speech (TTS) – TTS is the subject of producing comprehensive artificial speech on a given text using a computer. With recent advancements, we are moving closer to almost Human-speech generation using Deep Learning techniques in TTS.

This report will mainly focus on the theory and mathematical understanding of ASR and TTS. We will also be demonstrating a unique TTS Hindi – Marathi model designed and an ASR model for Swahili language as a part of this project.

2 Automatic Speech Recognition

2.1 Problem Statement

Automatic Speech Recognition (ASR) is the automated task of generating text output from an input speech signal. There are over 7000+ languages in the World, but just 23 of these languages account for more than half of the World's population. ASR is an essential method to reach people with different languages and break the communication barrier in the World. With the advent of Machine Learning and Deep Learning, ASR has received a significant boost in its ability to perform well across languages. Like any AI application, the amount of data available is significant in designing a state-of-the-art system in ASR. There has been a vast amount of research into languages such as English, Spanish, German, etc; known as High Resource Languages due to the large scale availability of data and open-source toolkits in these languages. However, low

resource languages such as Hindi, Marathi, Tamil do not have high-performance systems due to the lack of transcribed data.

We treat the acoustic input signal as O = $\{o_1 , o_2 , o_3 , o_4 , \dots\}$ a series of observations and define a sequence of words as the desired output W = $\{w_1 , w_2 , w_3 , w_4 , \dots\}$.

We would like to get those sequence of words W from the language L, which maximizes the following condition given the acoustic input O.

$$\hat{W} = \arg\max_{W \in L} P(W|O) \tag{1}$$

We can use Bayes rule to rewrite this as -

$$\hat{W} = \arg\max_{W \in L} \frac{P(O|W)P(W)}{P(O)} \tag{2}$$

For every possible sequence of words W, the denominator of equation 2 is the same. Since we are dealing with the argmax operator, we can ignore the denominator and write the final expression as -

$$\hat{W} = \arg\max_{W \in L} P(O|W)P(W) \tag{3}$$

Each component in a traditional system plays an important role in calculating the above two probabilities.

For evaluating an ASR system, Word Error Rate (WER)[1] is a common metric used. The WER is derived from the Levenshtein distance, working at the word level instead of the phoneme level. Word error rate can then be computed as:

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C} \tag{4}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct words, N is the number of words in the reference

2.2 Components of a Traditional ASR System

Traditionally, ASR Systems consisted of various modules stacked on top of each other, performing specific tasks in the pipeline. In such a system, we consider the speaker speaking a combination of audio utterances which need to pass through a channel and decoded to the corresponding output.

Input Speech: A signal is a variation in a certain quantity over time. For audio, the quantity that varies is air pressure. The first stage of data sampled from the speech is the sampling of the pressure variance across time. However, speech and language are complex entities, and hence just a simple two-dimensional feature for audio will not be sufficient to capture the complete information of the speech. A prevalent method of data representation for speech is Mel spectrograms[2]. A spectrogram is obtained by computing Fast Fourier transforms on windowed segments of the input speech. Representing this spectrogram on the Mel scale gives us the final Mel spectrogram. Such a feature representation can encompass the loudness, frequency, and timestamps of the input speech signal. These features are used as the baseline input features to speech recognition models.

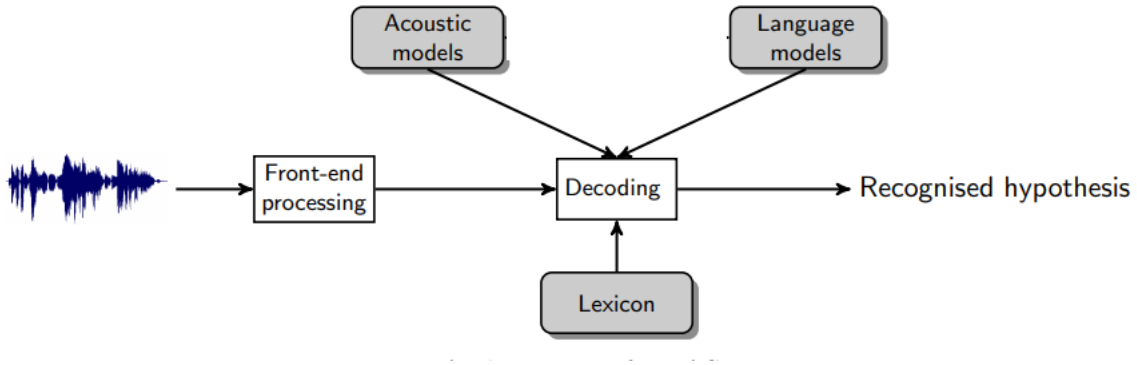

Figure 1: Traditional ASR Pipeline

Acoustic Model: An acoustic model is the component of the pipeline that contains statistical representations of each of the distinct sounds that make up a word. Each of these statistical representations is assigned a label called a phoneme. The English language has about 40 distinct sounds useful for speech recognition, and thus we have 40 different phonemes. An acoustic model is created by taking an extensive database of speech (called a speech corpus) and using special training algorithms to create statistical representations for each phoneme in a language. These statistical representations are called Hidden Markov Models (HMM). Each phoneme has its own HMM. From the previous subsection, the acoustic model can calculate the probabilities P(O|W).

Language Model: Statistical Language Modeling, or Language Modeling and LM for short, is the development of probabilistic models that can be used to predict the next word in the sequence given the previous words in the sentence. Smaller models may look at a context of a short sequence of words, whereas larger models may work at the level of sentences or paragraphs to generate the probability of word occurrences. From the previous subsection, the language model can calculate the probabilities P(W).

Figure 1 depicts the various modules associated with a traditional ASR system.

2.3 Deep Learning Based Speech Recognition Systems

Deep Learning is the branch of machine learning inspired by the structure and function of the brain dealing with the development of algorithms called artificial neural networks. Deep learning uses multiple layers to progressively extract higher-level features from the raw input. For example, lower layers may identify edges in image processing, while higher layers may identify the concepts relevant to a human, such as digits or letters, or faces.

In speech recognition, deep learning plays an essential role in understanding the input signal. In the previous section, we discussed how raw speech signals are converted into mel spectrograms. To the human eye, these spectrograms do not carry much information. However, when we feed these images to deep learning frameworks with multiple layers, each layer can capture meaningful information as it builds to form the final higher-order representation vector corresponding to each sample.

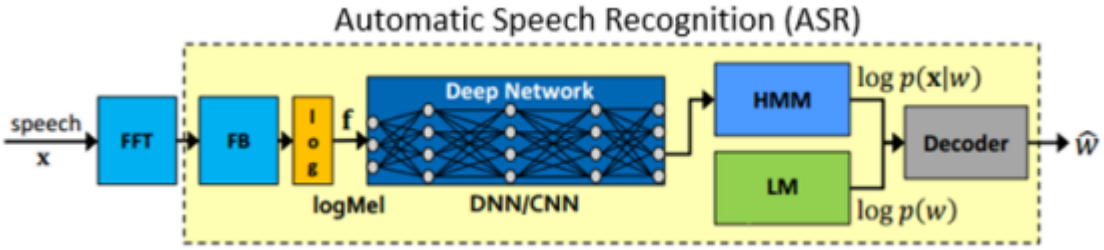

Figure 2: Deep Learning based training process for ASR

3 Text to Speech

3.1 Problem Statement

Text to Speech or Speech Synthesis is the process of converting text input to speech output, making use of various acoustic, linguistic, digital signal processing, and statistical features. Speech synthesis allows environmental barriers to be removed for people with a wide range of disabilities. Just like ASR, it relies heavily on the availability of transcribed data, and hence low resource languages do not have very well developed TTS systems.

3.2 Traditional TTS Systems

There are two specific methods for TTS conversion used traditionally: concatenative TTS and parametric TTS.

3.2.1 Concatenative TTS

Concatenative TTS[4], as the name suggests, is a method of merging individual speech units to continuous output speech. In this approach, the input text is divided into individual text phonemes, and the sounds corresponding to each phoneme are referred to from an annotated corpus. These sounds are then compiled, and with the help of contextual information, they are merged to output continuous speech.

There are two methods of concatenative speech synthesis which were commonly used - based on Linear Prediction Coefficients and one based on the PSOLA algorithm.

The first method uses the LPC coding[5] of speech to reduce the speech signal's storage capacity followed by the synthesis step which is a simple decoding and concatenation process. Mathematically speaking, each point of the speech signal 's' is predicted using a linear combination of 'p' points before it.

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + e[n] \tag{5}$$

where $\{a_k\}$ are the p^{th} order linear predictor coefficients and e[n] is the residual prediction error.

While conversing in day to day life, we do not speak as if individual phoneme elements are simply concatenated. The way in which phonemes are concatenated depends on the tone of speaking and the context around the sentence. This problem is addressed in the PSOLA (Pitch Synchronous Overlap Add) algorithm approach, which adjusts the prosody of the concatenative

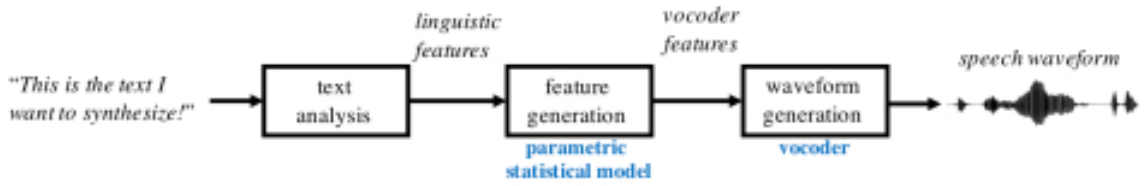

Figure 3: Statistical Speech Synthesis pipeline

units according to the context of the sentence to improve pronunciations and generate a natural continuous flow of output speech.

3.2.2 Parametric TTS

The PSOLA algorithm is limited by characteristics of the speech corpus used for the unit selection process. An alternative to the concatenative method, statistical parametric speech synthesis (SPSS)[4][6] addresses the main limitation of the concatenative systems, the lack of flexibility, by generating the speech using statistical models of speech instead of relying on pre-recorded segments from the chosen speech corpus. These statistical models learn information of how speech evolves as a function of time in the context of a given input text (Figure 3) .

3.3 Deep Learning Based TTS Systems

Deep learning based methods directly perform the mapping from linguistic features to acoustic features with deep neural networks, which have proven extraordinarily efficient at learning inherent features of data. In ASR, we saw that the raw audio was first converted into Mel spectrograms. Deep Learning based TTS models work in the opposite way by converting the input text into representation vectors using strong neural embeddings. One such deep learning framework, Tacotron 2, is explained in future sections.

4 TTS System for Hindi and Marathi using Tacotron 2 - NASSCOM 2021

4.1 About Nasscom 2021

NASSCOM [7] is the premier trade body and chamber of commerce of the Tech industry in India and comprises over 3000 member companies including both Indian and multinational organisations that have a presence in India. NASSCOM conducted the NLP Week from 5th April - 8th April which consisted of speakers from across India demonstrating various NLP applications through guided workshops and lectures.

4.2 Problem Statement

IIT Bombay's team led by Professor Pushpak Bhattacharyya and Professor Preethi Jyothi worked in developing a Speech to Speech Machine Translation system. The goal of the project was to design the pipeline for inputting an English sentence (audio), convert it to English text using ASR, use machine translation to convert this transcribed sentence to Hindi and Marathi sentences respectively, and developing a TTS system that synthesizes speech from these Hindi and Marathi

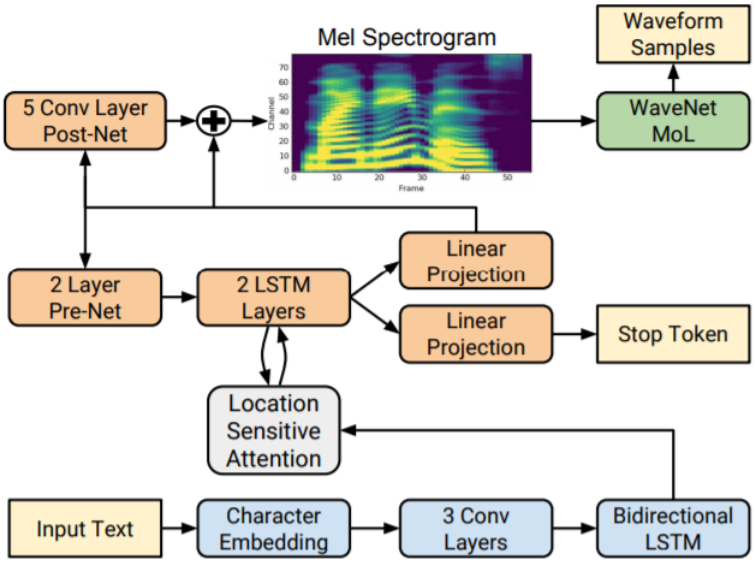

Figure 4

sentences. I worked in the last part of this pipeline, that is, building the Hindi and Marathi TTS system.

4.3 Using the Tacotron 2 framework for Speech Synthesis

Tacotron 2[8] is a part of the class of models that fall under the "End-to-End" category of deep learning systems. End to End systems are frameworks that behave like a black box. Neural layers act on the original input and work towards building the desired output and hence it is not possible for us to categorize accurately the task done by each layer.

The architecture of Tacotron 2 consists of two principal components - a sequence to sequence recurrent feature prediction network, which predicts a sequence of mel-spectrogram frames from an input character sequence, and the vocoder like Wavenet[9] architecture that acts on the generated mel-spectrogram and outputs the speech signals. The below figure explains the architecture in detail.

4.3.1 Spectrogram Prediction Network

The network is composed of an encoder and a decoder with attention. The encoder converts a character sequence into a hidden feature the representation which the decoder consumes to predict a spectrogram. The decoder is an autoregressive recurrent neural network that predicts a Mel spectrogram from the encoded input sequence one frame at a time.

4.3.2 Waveglow Vocoder

WaveGlow[10] is a generative model that generates audio by sampling from a distribution. To use a neural network as a generative model, sampling is done from a zero-mean spherical Gaussian

with the same number of dimensions as the desired output. These samples pass through a series of layers that transforms the simple distribution into one which has the desired distribution. In this case, we model the distribution of audio samples conditioned on a Mel-spectrogram.

$$z \sim N(z, 0, I) \tag{6}$$

$$x = f_1 \circ f_2 \circ ....f_k(z) \tag{7}$$

In the above equation, consider z to be the mel spectrogram images and x to be the desired audio output. While training the wave glow model, researchers started with z to be a random sampling from the zero mean spherical Gaussian distribution with the required dimensions. This matrix z was passed through several functions to give the final output x. This x was compared with the desired audio output, and using back propagation, all the parameters were modified.

4.4 Our Approach

Consider a foreigner visiting India and wanting to learn our native language Hindi. Is it right to consider this foreigner equivalent to a newborn baby wanting to learn Hindi? Obviously, it is easier to teach the foreigner. This is because the foreigner is well versed in English and understands the sounds associated with each letter of the English alphabet. Can this somehow be used to help him learn Hindi better?

The answer to the above question is Transliteration. Transliteration is a method of converting the Hindi (or Marathi) sentences in Devnagiri script to English letters. It is common practice for us to use Transliteration while communicating informally over messages, but if our foreigner has a Transliteration guide that contains the Hindi sentence, its transliterated counterpart, and then he hears the audio output, it will definitely be much easier for him to learn the language.

One of the remarkable features of the Tacotron 2 framework is the ability to extend it to various languages. The English language is by far the most popular language in the world. Deep learning relies heavily on the availability of data, and due to abundant resources in the English language, we have close to Human results in a lot of English NLP Tasks. However, the same cannot be said for Indian languages like Hindi and Marathi. Due to the lack of transcribed data, training from scratch for these languages may not lead to convergence and good results for developers. Hence, we use the principles of transfer learning to jumpstart the training process.

In our approach, instead of training the tacotron 2 model from scratch on Devanagari text-speech data, we use the openly available checkpoint of the state of the art English TTS system developed using tacotron 2 and thousands of hours of data. Clearly, this model is able to relate English alphabets to their corresponding sounds really well. Hence we use Transliteration to convert our Devanagari scripts in Hindi (and Marathi) to their counterparts using English alphabets. Thereafter, we jumpstart the training process by using the trained checkpoint as our base and finetuning on our newly created transliterated Hindi (and Marathi) corpus.

4.5 Dataset Used

We used Hindi and Marathi Speech-Text data kindly shared to our team by IIT Madras' Indic-TTS team. Indic TTS[11] is a special corpus of Indian languages covering 13 major languages of India. It comprises 10000+ spoken sentences/utterances each of mono and English recorded by both Male and Female native speakers. Speech waveform files are available in .wav format along with the corresponding text.

The Hindi corpus consisted of about 5.16 hours of annotated speech data. The mean duration of each clip was about 8 seconds, averaged over a total of 2318 clips. The audio quality of the

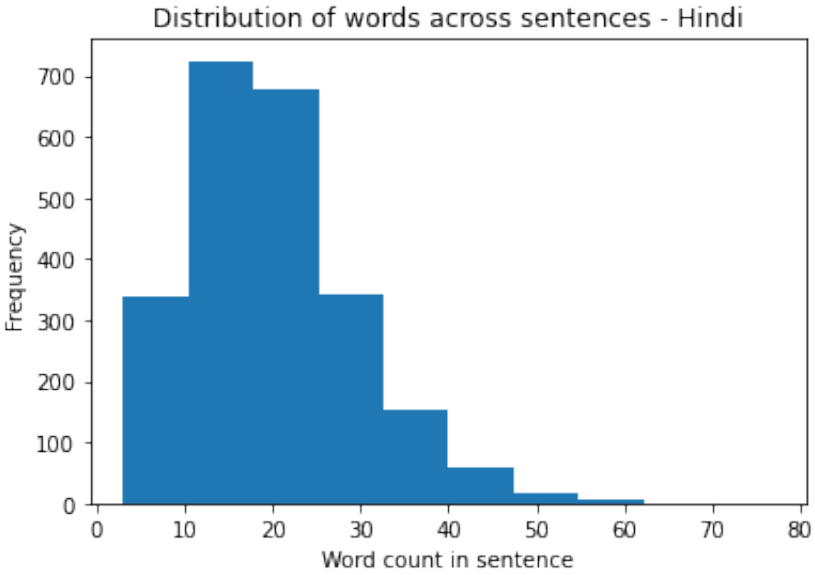

Figure 5

clips was a bit rough, as confirmed manually. The sentences were of extreme difficulty with complex words in almost every sentence making the pronunciations difficult to predict for our model. However, we will see in later sections that the model overcomes this difficulty depicting the success of our transfer learning approach. The speaker of this corpus was Male. Figure 5 gives the word count distribution across the sentences.

The Marathi corpus consisted of about 4.82 hours of annotated speech data. The mean duration of each clip was about 7.09 seconds, averaged over a total of 2821 clips. The audio quality of the speech was much better compared to its Hindi counterpart, with clearer pronunciations and almost negligible background noise. The speaker of this corpus was Female. Figure 6 gives the word count distribution across sentences.

4.6 Training

Two separate models were trained for Hindi and Marathi TTS. Both the training processes started with the open-source English speech trained checkpoint with finetuning on the respective speech corpus. The Hindi finetuning was conducted for about 4500 iterations, while the Marathi finetuning was conducted for about 3200 iterations, after which the training loss stagnated. The entire training process was performed using Google Colab's GPU services which considerably reduced the training time to 3 seconds per iteration. All the hyperparameters used for training were the default values in the tacotron 2 framework.

Before applying the principles of transfer learning, we trained the tacotron 2 models from scratch using the available Hindi (and Marathi) speech data. Even after 10k iterations in both these languages, the model did not converge, which could be because of a lack of training data in both these languages. However, the below figure shows that by jumpstarting the training process, we achieve a significant decrease in the loss, and the model is able to achieve convergence within 3000-3500 iterations. This once again proves the effectiveness of our transfer learning approach.

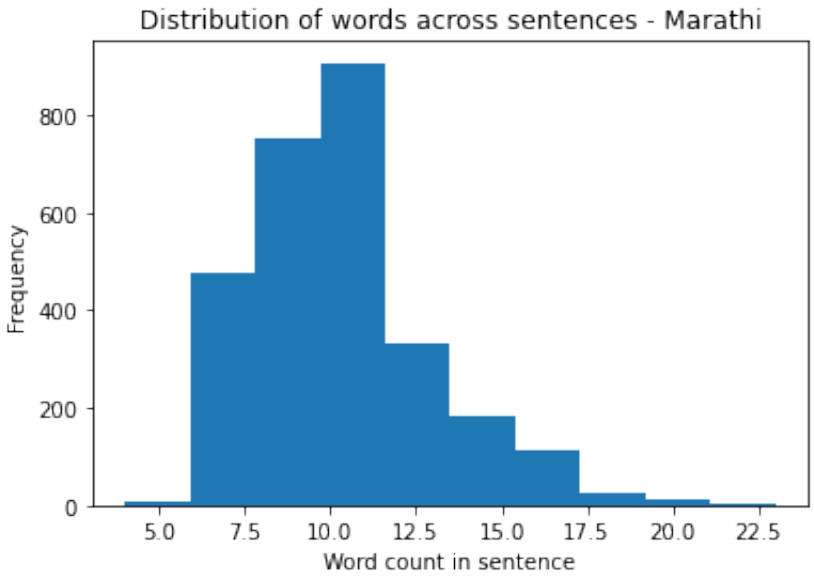

Figure 6

Figure 7 shows the jumpstarted decrease in loss values while training for the Hindi compared to training from scratch (first 100 iterations).

4.7 Results

After finetuning the English models and Hindi and Marathi speech corpus, we were able to achieve excellent results with manually verified quality. One important takeaway from this approach was the fact that although the original English checkpoint was pretrained on thousands of hours of Male English Speech, finetuning it with Female Marathi speech with just 4.8 hours of data, the final model was able to synthesize female voice perfectly well. This is analogous to our foreigner able to learn the native language in female speech !

Our work shows how using transfer learning on large amounts of data, we can use features common across languages to benefit the training process. This method can also be used to create various low resource TTS systems across the languages provided we have a good transliteration system between scripts and the pronunciations of letters are similar as in English.

5 ASR for Swahili - IWSLT 2021

5.1 Problem Statement

The International Conference on Spoken Language Translation (IWSLT) is the premier annual scientific conference, dedicated to all aspects of spoken language translation [12] . This year, the Low Resource Speech Translation task comprised of designing a system which takes in Swahili Speech and outputs its English translation in text format. The following work contains a system developed for Swahili ASR which can be used for creating a pipeline for the low resource speech translation shared task.

Figure 7

5.2 Proposed Approach

In September 2020, Facebook AI released the Wav2Vec2 model [13] which uses self-supervision to push the boundaries by learning from unlabeled training data to enable speech recognition systems for many more languages, dialects, and domains. Wav2Vec2 learns powerful speech representations from hundreds of thousands of hours of speech in more than 50 languages of unlabeled speech. With just a few hours of transcribed speech, the model can be finetuned to give good results for various low resource languages.

Similiar to our approach for TTS, we started from the pre-trained XLSR-Wav2Vec2 model checkpoint available open source trained on 53 languages including Swahili. Using the available Swahili Dataset, we finetune the model to achieve a significantly faster training process.

5.3 Dataset Used

Dataset used for the experiment was taken from the IWSLT 2021 conference developed by Gamayun a language equality initiative [14] . The Gamayun project revolves around creating annotated datasets for low resource languages, especially those from the African continent. Swahili text is written using the English alphabets. The given dataset consisted of 41 unique tokens - 26 alphabets, 10 digits and 5 special characters. Figure 8 gives the histogram of word distribution across the sample. The speech corpus contained 8.94 hours hours of labelled speech with 3.16 seconds for each clip averaged over 10180 samples. Evidently, Swahili is a fast spoken language and thus high quality ASR systems is a tough task in this language.

5.4 Training

XLSR-Wav2Vec2 model is pretrained on 53 languages with thousands of hours of unsupervised data. For our purposes, we downloaded the open source checkpoint made available by the Facebook AI team and fine-tuned it on the Gamayun Swahili corpus using the HuggingFace [15]

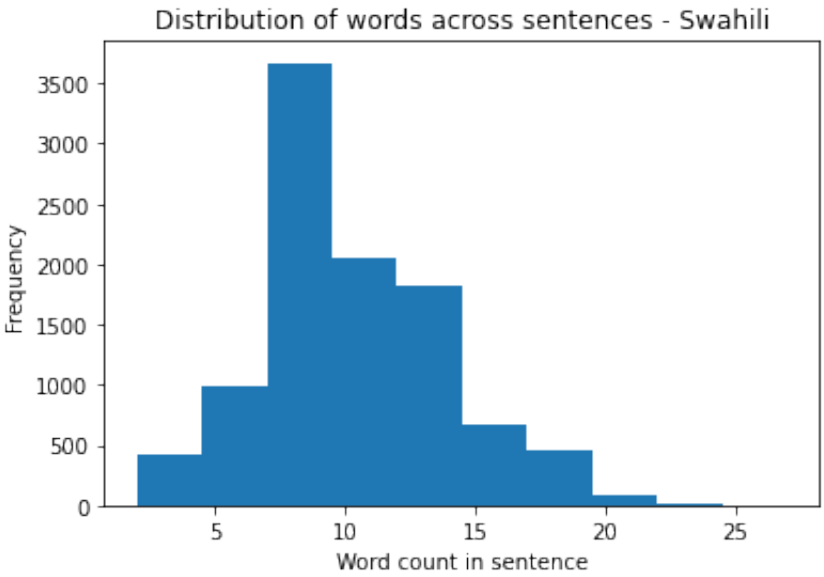

Figure 8

framework. Pre-processing steps included cleaning the text corpus of extra characters, resampling the speech data to 16kHz and creating a vocab dictionary. HuggingFace's transformer packagers were used to load the pretrained wav2vec2 model. The model took raw audio as input and not the mel spectrogram of the speech signal. All training steps were conducted on Google Colab using their GPU services.

Due to time constraints, the model was finetuned only for 3 epochs. Each epoch took about 2 hours on the GPU. However, just in 3 epochs (which corresponded to about 800 iterations), we witnessed a significant fall in WER. Figure 9 shows the trajectory of the training and validation loss and Figure 10 shows the evaluation WER with steps. Clearly, after 200 iterations, the model started showing excellent results and it can be safely said that with further training and possibly more annotated data, the model will quickly converge to extremely low WER.

5.5 Results

Through our experiments, we validated the hypothesis that the XLSR-Wav2Vec2 model is capable of achieving excellent results for its low resource component languages. This paves the way for increasing the research into unsupervised learning as a jumpstart process for various speech applications. It is an extremely painful task to validate and annotate speech corpus across languages but collecting unannotated speech data is a much easier job. With more developments into the field of unsupervised learning, there will be much less reliance on annotated corpora, facilitating state-of-the-art results across all speech tasks. Unfortunately, due to time constraints, we could not complete the Machine Translation component of the Speech Translation system as described in IWSLT 2021, but we will try to submit our results in other conferences.

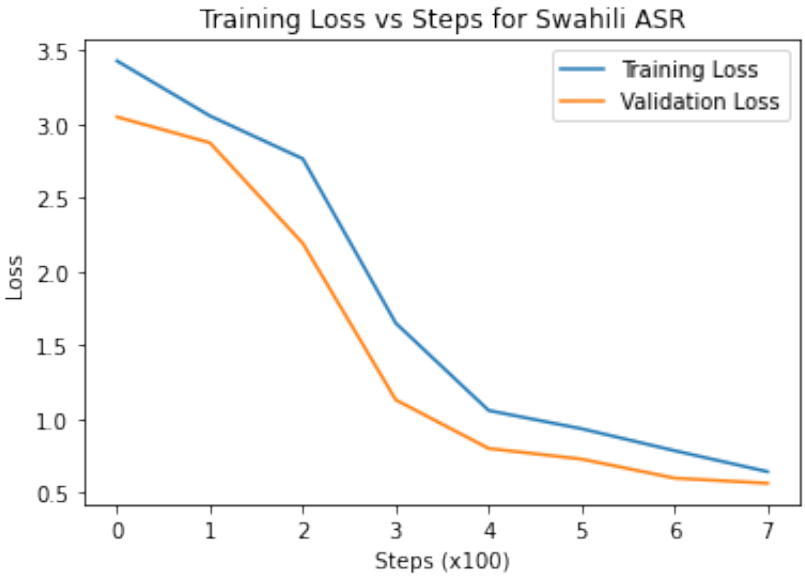

Figure 9

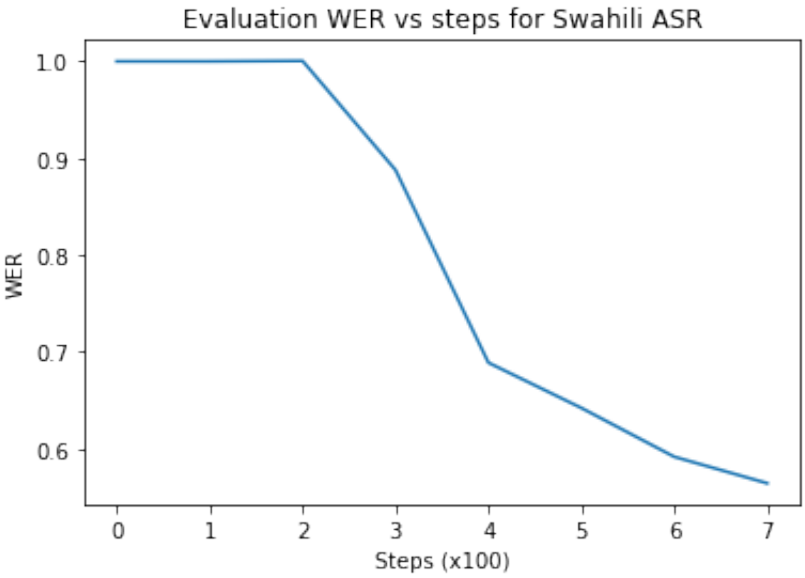

Figure 10

6 Conclusion

This report contained an exhaustive introduction of two of the most important speech tasks - ASR and TTS with focus on both traditional as well as modern solutions to these tasks. With the world moving towards deeper learning systems, the reliance on annotated data has increased exponentially. Such high quality and large amounts of annotated data is almost impossible in low resource languages such as Hindi, Marathi and Swahili and hence it is imperative to look for alternate methods for low resource languages in speech tasks. One of the important approaches discussed in this report - Transfer Learning, is one of the strongest tools in combating this issue. Through our experiments, we showed how transfer learning is extremely beneficial in low resource languages. Our TTS systems were highly appreciated in the NASSCOM workshop and the Swahili - ASR system has strong potential to reach extremely low WER with more finetuning. Further work will include working on the Machine Translation component of the Speech Translation pipeline and exploring how transfer learning based TTS can be used in languages which cannot be transliterated very easily to English.

Acknowledgements

I would like to thank my guide, Professor Pushpak Bhattacharyya, IIT Bombay as well as my co-guide, Professor Alok Shukla, IIT Bombay for their guidance during this project. I would also like to thank Professor Preethi Jyothi, IIT Bombay for constant support during the NASSCOM workshop and Mr. Nikhil Saini, 2nd Year Mtech Student, IIT Bombay for helping me debug my code during the workshop.

Link to Code Repository

The code developed to run the experiments may be found here: `https://github.com/vineet2104/LowResource_ASR_TTS`

References

[1] `https://en.wikipedia.org/wiki/Word_error_rate`
[2] "Audio Recognition using Mel Spectrograms and Convolution Neural Networks"- Zhang et al
[3] "An Introduction to Hidden Markov Models" - Rabiner,Juang IEEE 1986
[4] "A Review of Deep Learning Based Speech Synthesis" - Ning et al, 2019
[5] Lectures by Professor Dan Ellis, Columbia University
[6] `https://wiki.aalto.fi/display/ITSP/Statistical+parametric+speech+synthesis`
[7] `https://nasscom.in/`
[8] "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions" - Shen et al, ICASSP 2018
[9] "WaveNet: A Generative Model for Raw Audio" - Oord et al
[10] "WaveGlow: A Flow-based Generative Network for Speech Synthesis" - Prenger et al, 2018
[11] `https://www.iitm.ac.in/donlab/tts/`
[12] `https://iwslt.org/2021/`
[13] "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations" - Baevski et al, 2020
[14] `https://gamayun.translatorswb.org/`
[15] `https://huggingface.co/`