# CS 458/535 - Natural Language Processing
# Tuning Text-to-speech(TTS) for Urdu using Transfer Learning

Ayesha Gul 18I-0467
Saifullah Dar 18I-0499
Adan Abbas 18I-0401

June 27, 2021

## 1 Problem Statement

As the field of Artificial Intelligence progresses, Natural Human-Machine communication becomes more prevalent. The rising demand for handheld devices, increased government spending on education for differently-abled, the dependence of the growing elderly population on technology, and the rising number of people with different learning disabilities or learning styles are factors driving the growth of the text-to-speech market. However, the lack of prosody and pronunciation of naturally occurring speech may limit the growth of the research and development through the period.

Speech synthesis (text-to-speech) a.k.a TTS is the artificial production of human speech. A typical text-to-speech system converts a language text into a waveform. There exist many English TTS systems that produce mature, natural, and human-like speech synthesizers. In contrast, other languages, including Urdu, have not been considered until recently. Most Natural Language Processing-related models for low resource languages are not accurate due to lack of digitization and scarcity of the data sets available for these languages and training a model that gives a decent accuracy given a small data set is highly unlikely. The Urdu Language also shares the same fate, existing Urdu speech synthesis solutions are slow, of low quality, and the naturalness of synthesized speech is inferior to the other high-resource language synthesizers. They also lack essential speech key factors such as intonation, stress, and rhythm. Urdu is spoken by nearly 170 million people. However, it is not as digitized as some of the other languages. Although some research has been done on this language and its related models, it is still a long way away from becoming a fully digitized language. Different works were proposed to solve those issues, including the use of concatenative methods such as unit selection or parametric methods. However, they required a lot of laborious work and domain expertise. Another reason for such poor performance of Urdu speech synthesizers is the lack of speech corpora, unlike English that has many publicly available corpora and audio books. . In this project, our work describes how to generate high-quality, natural, and human-like Urdu speech applying transfer learning to solve the problem of Text-to-speech for the Urdu Language and its advantages over using a simple Text-to-speech model for Urdu.

## 2 Motivation

Natural language processing (NLP) is a field of Artificial Intelligence that aims to teach computers to understand human language. This is difficult as the computer must comprehend many facets of language such as semantics, syntax, pragmatics, and phonology which are difficult to characterize formally and even more difficult when it comes to interpreting as computer instructions because the language of the computer is binary. Many researchers

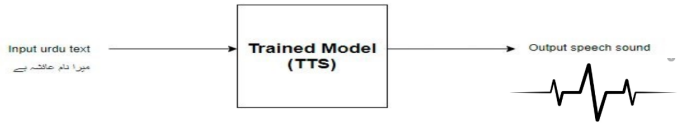


Figure 1: A TTS model

across the world are working on solutions to make the computer understand human language to get the best out of technology. It is even harder for so-called low-resource languages where the annotated resources are very limited. There are approximately 7,000 languages in the world, but of these, only a small fraction (20 languages) are considered high-resource languages. Low-resource languages are in dire need of tools and resources to overcome the resource barrier such that advances in NLP can deliver more widespread benefits. Despite the lack of annotated data, some unannotated data resources might be beneficial for low-resource languages including parallel data, bilingual lexical resources, or clues from related languages. However, the means for effectively incorporating these resources to improve the performance of low-resource NLP is an open research question and the target of this thesis. Out of 7,000 languages, half of them do not have a writing system and many are falling out of use. It is estimated that by the end of this century, half of the world's languages will be extinct. It is necessary to extend the current NLP techniques to unwritten languages to process and document the languages before they are gone forever.

The data sets for Urdu languages are scarcely available. The work related to Urdu NLP, therefore, has room for much improvement. Some models that provide TTS functionality for Urdu exist. However, not much work can be found in the creation of a TTS model for Urdu using Transfer learning. Some other applications of transfer learning like Speech Affect recognition for Urdu exists and there has been much work done on this when it comes to TTS

The Transfer Learning approach uses knowledge gathered for one problem to solve another. Transfer learning provides an important opportunity for low-resource NLP, whereby annotation is transferred from a source resource-rich language to a target resource-poor language. We can use this approach to train a neural network-based TTS model for a high resource language such as English or German and fine-tune the parameters using the low resource language i.e Urdu in this case to create a model that performs the TTS generation task with greater accuracy than existing TTS models. We show that only a small amount of annotated text in the target language is sufficient to achieve a large performance improvement by incorporating a resource-rich source language into the model.

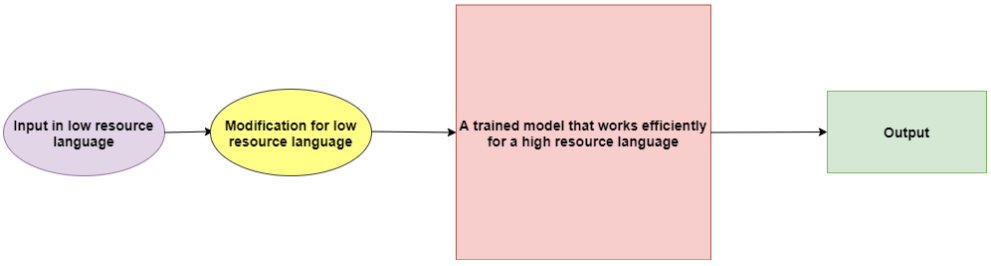


Figure 2: Our proposed model

# 3 Background

Transfer Learning methodology for model training is highly accurate due to the fact that they use highly available resources for the solution of a problem and then change it a bit to apply on a similar but different problem. It has a high learning rate and better initial model than other models and therefore is favourable for our particular problem for TTS for Urdu.

Neural Networks are set of algorithms that work in a way similar to a human brain and identify the relationship between a some data e.g learning the basic of speech generation given some text no matter the language due to the fact that the basic concept of identification of phonemes remain the same in every language.

Using Transfer Learning on a trained Neural Network can lead to the advantages that the initial part of the problem i.e identification of phonemes and similar concepts that are the basis of TTS are fine-tuned.

# 4 Related Work

Google and Amazon are using transfer learning in Google Translate and Alexa so that the insights gleaned through training on high-resource languages (e.g., French, German, and Spanish) can be applied to the translation of low-resource languages (Yoruba, Sindhi, and Hawaiian). Meanwhile, Yelp has used transfer learning to identify photos most likely to contain spam uploaded by users to business listings.

Some work relating to transfer learning based speech affect recognition in Urdu has been done locally [2].
A Transfer Learning approach based method for English to Arabic also exists [3].
There is also some work on a low-resource language 'Occitan' using neural text-to-speech [1].

# 5 Proposed Work

We plan to implement an Urdu Text-to-speech model that modifies the existing Tacotron 2[4] model. Our model will take an Urdu sentence as input and output an Urdu wave sound output. Our model will pre-process each sentence by first breaking it down into a list of words. It will then generate the phonemes of every word.The generation of phonemes will be done using the G2P model described here [6]. Every phoneme will be combined and converted back to the relevant English word and that word will be fed into the TTS model. This proposed model utilizes the concept of transfer learning on the Tacotron 2 TTS model. It is widely acknowledged that the existing Tacotron 2 model performs exceptionally well for the English language. More specifically, it gives a MOS score of $\mathbf{4.526 \pm 0.066}$ on the test data. In comparison, the MOS score of the ground truth data is $\mathbf{4.582 \pm 0.053}$. Our approach takes advantage of the fact that the Tacotron 2 model is using the English sentences, extracting the mel-spectrograms of the sounds, and passing it to the WaveNet model to produce the audios and it is doing that extremely well. The mel-spectrograms are generated using the phonemes and the context at the previous timestamps using the attention module, therefore we have proposed the idea of modifying this model for the Text to Speech problem by introducing a model that first appropriately and efficiently converts the Urdu graphemes to their English corresponding phonemes. The modified model is then again trained for the low resource language. However, this step is only necessary to tune the Attention module so that it does not get confused due to the change of structures of phonemes that are now arriving. The complete proposed model is shown in Figure 3.

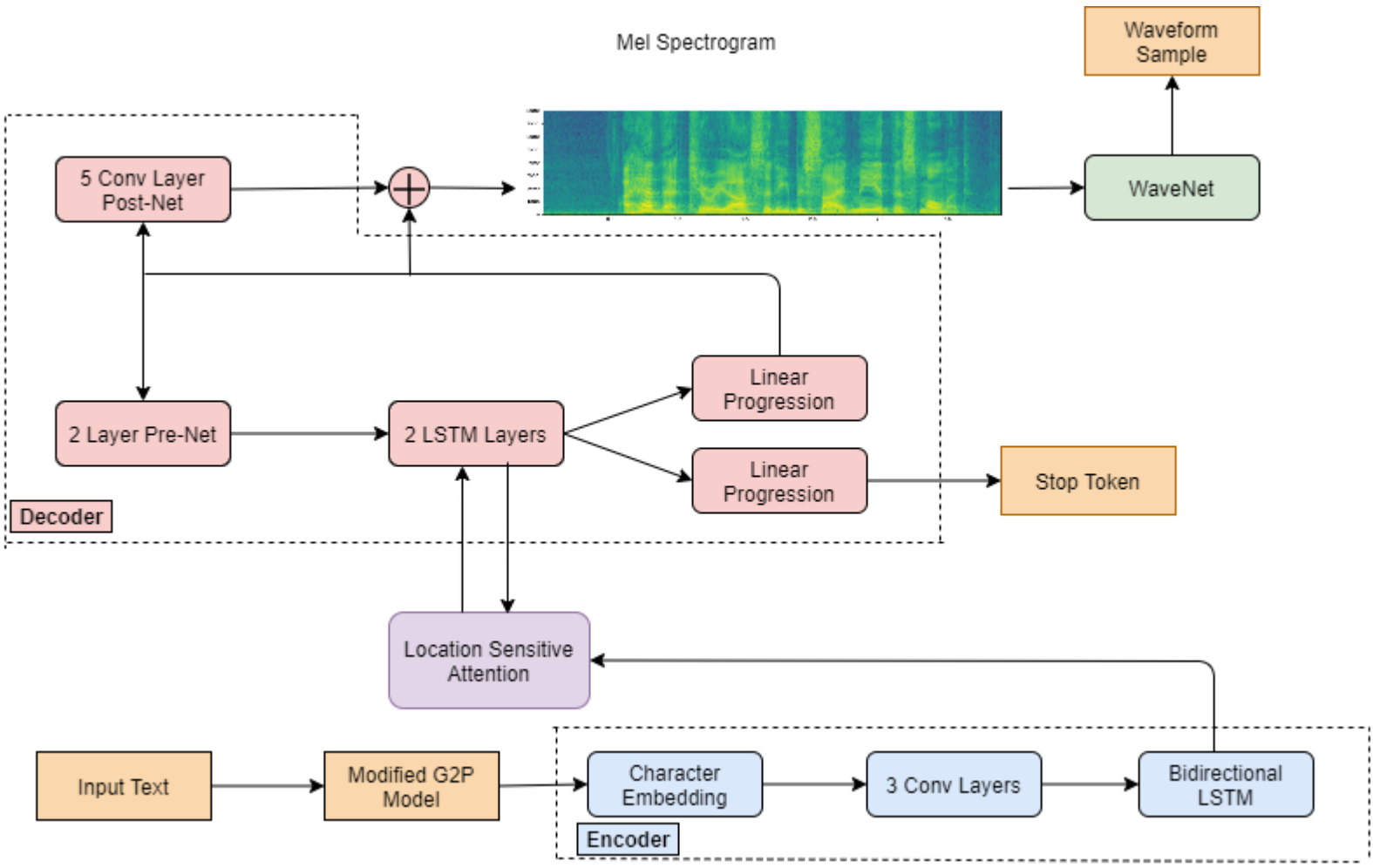


Figure 3: The complete Proposed Model

## 5.1 G2P Model

The grapheme-to-phoneme (G2P) translates an input sequence of letters(graphemes) and converts them to an output sequence of phonemes. We used the already trained G2P model in as it has been trained on a lexicon of approximately 46K Urdu words. This model used LSTM for grapheme-to-phoneme conversion and is trained using the open-source G2P toolkit17. It has 2 LSTM layers and 512 hidden units in each layer. The lexicon had been tagged by trained transcription experts. The training lexicon caters for 64 phenomes, out of the available 67, the remaining 3 being short nasalized words as work on them is still in progress. The frequency of occurrences is shown in the table below.

| # | Phenome | Frequency | # | Phenome | Frequency |
|---|---|---|---|---|---|
| 1 | A | 30947 | 32 | Q | 2080 |
| 2 | A_A | 27170 | 33 | X | 1641 |
| 3 | R | 18386 | 34 | R_R | 1562 |
| 4 | N | 151 | 35 | Q | 2080 |
| 5 | I_I | 13920 | 36 | N_G | 1297 |
| 6 | I | 13683 | 37 | A_A_N | 1060 |
| 7 | L | 10909 | 38 | K_H | 1035 |
| 8 | M | 10538 | 39 | O | 928 |
| 9 | S | 10522 | 40 | G_G | 800 |
| 10 | T_D | 10075 | 41 | T_S_H | 711 |
| 11 | K | 8470 | 42 | B_H | 690 |
| 12 | A_Y | 7562 | 43 | I_I_N | 660 |
| 13 | B | 7147 | 44 | D_Z_H | 571 |
| 14 | U | 6540 | 45 | D_D_H | 555 |
| 15 | T | 6024 | 46 | T_D_H | 531 |
| 16 | D_D | 5913 | 47 | T_H | 495 |
| 17 | Z | 4940 | 48 | P_H | 435 |
| 18 | H | 4771 | 49 | G_H | 424 |
| 19 | O_O | 4766 | 50 | A_E_H | 375 |
| 20 | P | 4742 | 51 | U_U_N | 332 |
| 21 | V | 4144 | 52 | R_R_H | 225 |
| 22 | O_O_N | 4128 | 53 | D_H | 194 |
| 23 | J | 3963 | 54 | O_O_H | 70 |
| 24 | U_U | 3581 | 55 | Z_Z | 52 |
| 25 | A_E | 3440 | 56 | A_E_N | 43 |
| 26 | S_H | 3423 | 57 | Y | 36 |
| 27 | D_Z | 3331 | 58 | A_Y_H | 33 |
| 28 | G | 3275 | 59 | N_H | 12 |
| 29 | F | 3233 | 60 | L_H | 8 |
| 30 | D | 2762 | 61 | R_H | 8 |
| 31 | T_S | 2491 | 62 | O_N | 4 |

The G2P model, in mathematical terms, can be defined as follow:

$$G2P(sequence - of - graphemes) = sequence - of - phonemes$$

Some examples of the G2P model outputs are shown as in figure 4.

| Input | G2P(Input) |
|---|---|
| خراب | X A R A_A B |
| بیماری | H M A_A R_R I_I |
| زمانه | Z M A_A N A |
| رکو | R U K O |
| اولاد | U_U L A_A D_D |
| محرم | M A_E_H R U M |
| سانپ | S A_A M P |
| دادی | D A D_D I_I |
| سکون | S K O_O_N |
| مشکل | M U S_H K I L |

Figure 4: Sample Inputs and Outputs of the G2P model

## 5.2 Spectogram Predictor

### 5.2.1 Encoder

The encoder's job is to convert the input characters into a feature representation that the decoder can use to generate a spectrogram. The encoder takes in the output of our G2P model. It firstly converts them into character embeddings. Each character embedding is 512-dimensional. These character embeddings are then passed through 3 convolutional layers. Each layer has 512 filters and every filter spans five characters. They are then followed by batch normalization and ReLU activation. These convolutional layers help to model the N-grams context in the input sequence. The results that are produced by the convolutional blocks are then taken by the Bidirectional LSTM. This LSTM has 512 units. The forward and backward results of the LSTM are combined to generate the encoded feature representations that can be used by the decoder.

### 5.2.2 Location Sensitive Attention Network

The encoder output before being sent to the decoder part is first fed into an attention network. The job of the attention network is to remember the encoded sequence so that it can use the combined attention weights from previous steps so that the model can move consistently forward through the input i.e it has some knowledge about the output it has generated and the quality of the output to be generated will be based on that knowledge. The Attention Network is necessary as it is hard for an encoder decoder network without attention to memorize long character sequences, therefore the performance will began to decline for long character sequences. This attention mechanism works by remembering the character weights when trying to solve for long sequences.

### 5.2.3 Decoder

In the decoder, the prediction of the previous timestamp passes through a two-layer Pre-Net network. This Pre-Net has two fully connected layers having 256 neurons and a ReLU activation function. The output of the Pre-Net and Location Sensitive Attention Network is fed into another two-layered uni-directional LSTM that contains 1204 neurons. The output of this LSTM is then projected into a linear transformation which gives us a predicted spectrogram frame. Each frame is passed into a Post-Net having 5 convolutional layers, each followed

by batch-normalization and a tan-h activation. This Post-Net enhances the frame prediction of the spectrogram. This spectrogram is then fed into the WaveNet Model for Waveform generation.

### 5.3 Modified WaveNet

A modified version of the WaveNet[5] architecture is utilized to convert the mel spectrograms generated by the decoders into time-domain waveform samples. The WaveNet architecture consists of 30 dilated convolution layers which are grouped into 3 diluted cycles. This dilated convolution is faster and captures more details. This modified version also uses PixelCNN++ which is a generative model that more accurately generates gradients in these spectrograms for more accurate text-to-speech. It uses a 10-component mixture of logistic distribution to generate 16 bits samples at 24 kHz. This modified WaveNet has a frame rate of 12.5 ms which is slower than the original WaveNet which has a frame rate of 5 ms. However, this results in better prediction of the spectrogram frames.

### 5.4 Mathematical Description of Technical Solution

We can define our solution in the mathematical terms in the following way. Suppose we have an input Urdu sentence denoted by X. The input X is fed to the G2P model and it gives us the phonemes extracted from the input X:

$$\vec{P} = G2P(X)$$

The output is a vector $\vec{P}$ of the extracted phonemes of X, converted to English and pre-processed appropriately. The $\vec{P}$ phonemes are then passed into the Encoder module that converts the phonemes into a feature representation that the decoder can use to generate the mel-spectrogram:

$$Y^n_{featureRep} = Encoder(\vec{P})$$

These $Y^n_{featureRep}$ at time stamp $n$ are then passed to the Attention Network. Attention Network remembers the encoded sequence and also uses the knowledge of the previous timestamps to generate the output to send to the decoder. It uses the embedded knowledge of the previous timestamp $Y^{n-1}_{featureRep}$) to tune the sequence of feature representation and sends the modified output to the encoder

$$Y_{modifiedRep} = Attention\_Model(Y^n_{featureRep}, Y^{n-1}_{featureRep})$$

The decoder takes the $Y_{modifiedRep}$ as input. It then processes it by feeding it into a two-layered unidirectional LSTM. The output of the LSTM is projected into a linear transformation which gives a predicted spectrogram frame. Ultimatly, the decoder is outputting a spectrogram frame S for the WaveNet model to use.

$$S = Decoder(Y_{modifiedRep})$$

Here S is the predicted spectrogram frame. Finally, this spectrogram frame is what is being utilized by our last model, that is the WaveNet model. The WaveNet model takes in S, the predicted spectrogram, as the input and outputs the wave form for the input X.

$$Res = WaveNet(S)$$

To summarize, if we consider the whole model as a black-box, we are giving in the input X(an Urdu sentence) to the model, and getting Res(a sound wave) as an output.

# 6 Evaluation Methodology

## 6.1 MSE

As the final output of our model depends upon the mel-spectrograms that are generated, we will be evaluating those generated mel-spectrograms for the analysis of our model. The mean squared error loss (MSE) can be used for this purpose. As our model works in a way similar to a predictor when we are evaluating it on some test data, we can use the MSE function to calculate the quality of the predictor by comparing the predicted and the actual values. The MSE function is defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

This MSE is the mean of the squares of the errors between the $Y_i$ original values and the $\hat{Y}_i$ predicted values.

## 6.2 Attention Alignment Graph

For evaluating the quality of the model, an attention alignment graph can be used. It shows how well the decoder works on the input of the encoder. The decoder reads the input frames and produces the audio frames for every audio by paying attention to the vector generated by the encoder for that particular input frame. Theoretically, the alignment graph should at the beginning of the training phase not match for the encoder and decoders. However, after progressing epochs of training, it should get better as it is getting more context from the previous timestamps. It is also learning how well it should use pronunciation and punctuation. The attention alignment graph also gets better as the model learns to identify pauses between the sentences and how they should cater to them. The problem of pausing between words or phrases is often not handled very well by the TTS models, resulting in a bad alignment between the outputs of the encoder and the decoder. However, using an attention mechanism can help with this problem as the attention model continuously learns about the previous pauses and it is also getting some feedback from the decoders LSTM enabling it to identify pauses more efficiently.

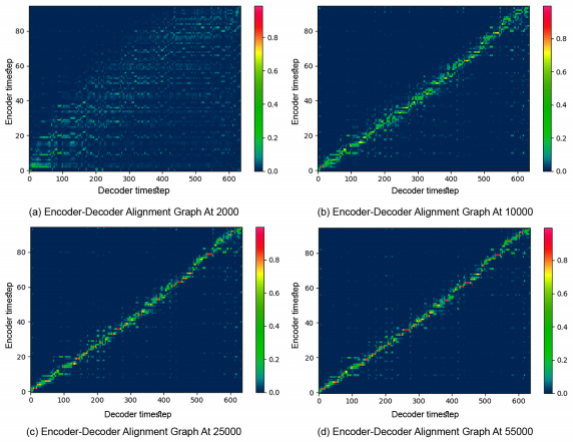


Figure 5: Sample Attention Alignment Graph

### 6.3 MOS

For further evaluation of the model, a mean opinion score(MOS) test can be carried out to test the naturalness and the quality of the output. The MOS test is a single value between 1 and 5 that describes the quality of the model. The value is highest at 5(the highest possible perceived quality) and lowest at 1(the lowest possible perceived quality). The labels corresponding to each rating are mentioned below:

| Rating | Label |
|---|---|
| 5 | Excellent |
| 4 | Good |
| 3 | Fair |
| 2 | Poor |
| 1 | Bad |

## 7 Hypothesis

Our hypothesis, keeping the proposed model under consideration is that the proposed model should work better than the current traditional TTS models for low-level languages. Although this model has been tailored for the Urdu language, hypothetically it should work for all the low-resource languages. The concept of transfer learning has been heavily used here. The purpose of using transfer learning for this problem is that the Final model should know from beforehand the underlying structures of a sentence. How it will pronounce different words? Where it will include a pause? Which words it will emphasize upon?. All these underlying details are learned by the existing model from which our proposed model originates. It takes the learning of the old model and then modifies it for our data set. Therefore, our model should be able to generate speech in Urdu. The quality of the Urdu speech may dampen then the quality of the English generated speech, however, there is a notable possibility that it may be better than the existing Urdu Text to Speech models.

## 8 Proposed Timeline

The tentative timeline giving concrete milestones would be as follows.

- **June 3**: This proposal.
- **June 6**: Familiarization with Transfer Learning
- **June 15**: Researching methods of applying Transfer Learning for trained TTS on a learned neural network for a high resource language
- **June 20**: Evaluation of implementation and comparison to existing approaches
- **June 24**: Writing Final Report
- **June 26**: Presentation and Final Report submission

# References


[1] Ander Corral[1], Igor Leturia1, Aure Seguier[2], Michael Barret[2], Benaset Dazeas[2], Philippe Boula de Mareuil[3], and Nicolas Quint[4]. Neural text-to-speech synthesis for an under-resourced language in a diglossic environment: the case of gascon occitan. *1 Elhuyar Foundation a.corral, i.leturia@elhuyar.eus 2 Lo Congres permanent de la lenga occitana a.seguier, m.barret, b.dazeas@locongres.org 3 Universite Paris-Saclay, CNRS, LIMSI ´ philippe.boula.de.mareuil@limsi.fr*, 2020.

[2] Sara Durrani and Muhammad Umair Arshad. Transfer learning based speech affect recognition in urdu. *Fast Nuces Islamabad, Pakistan.*, 2021.

[3] Mahmoud I. Khalil Fady K. Fahmy and Hazem M. Abbas. A transfer learning end-to-end arabic text-to-speech (tts) deep architecture. *Ain Shams University, Dept. Computer and Systems Engineering, Cairo, Egypt*, 2020.

[4] Jonathan Shen[1], Ruoming Pang[1], Ron J. Weiss[1], Mike Schuster[1], Navdeep Jaitly[1], Zongheng Yang[2], Zhifeng Chen[1], Yu Zhang[1], Yuxuan Wang[1], RJ Skerry-Ryan[1], Rif A. Saurous[1], Yannis Agiomyrgiannakis[1], , and Yonghui Wu[1]. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *1. Google, Inc., 2. University of California, Berkeley*, 2018.

[5] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *https://arxiv.org/abs/1609.03499*, 2016.

[6] Haris Bin Zia, Agha Ali Raza, and Awais Athar. Pronouncur: An urdu pronunciation lexicon generator. *https://www.aclweb.org/anthology/volumes/L18-1/*, 2018.