**[ RESEARCH REPORT ]**

**PREPARED BY : FATIMA ZAFAR**

**PASHTO SPEECH-TO-TEXT SYSTEM [ STT ]**

*This report summarizes my internship work on Pashto speech processing, specifically implementing and evaluating* ***Speech-to-Text (STT)*** *and* ***Text-to-Speech (TTS)*** *systems using open-source models for a low-resource language environment.*

***MODEL SELECTION***

*I chose the \*Whisper large model\* because of its proven effectiveness across multiple languages, especially low-resource ones like Pashto. Whisper has been trained on a massive amount of multilingual audio data, and among the available versions (tiny, base, small, medium, large), the large model consistently delivers better accuracy in real-world testing. Since Pashto STT resources are limited, this model provided a strong foundation to build on without the need for immediate fine-tuning.*

***RATIONALE***

*There were a few reasons behind this choice:*

* ***Multilingual capability****: Whisper supports many languages out of the box, including Pashto. This gave me a head start without having to train a model from scratch.*
* ***Ease of use****: The model integrates well with Python and worked seamlessly with libraries like torchaudio, making preprocessing and inference straightforward.*
* ***Accuracy and flexibility****: In my testing, Whisper handled diverse audio samples quite well—even in cases where audio quality varied or where speakers used slightly different dialects.*

*To preprocess the audio, I used torchaudio to standardize sample rates and convert stereo to mono where needed. Once transcribed, I evaluated the output using normalized text and calculated WER and CER to measure accuracy.*

***POSSIBLE IMPROVEMENTS***

* ***Fine-tuning on local data****: Whisper performs well out of the box, but fine-tuning it on a larger set of Pashto-specific speech data could significantly improve its accuracy.*
* ***Better normalization techniques****: Since Pashto can vary in spelling and pronunciation, introducing phonetic or lexicon-based normalization might reduce WER further.*
* ***Data augmentation****: Applying transformations like background noise, pitch variation, or speed adjustment could help the model generalize better to real-world audio.*
* ***Language model integration****: Incorporating a Pashto language model in post-processing could guide the transcriptions toward more natural and grammatically correct sentences.*

**PASHTO TEXT-TO-SPEECH SYSTEM [ TTS ]**

***MODEL SELECTION***

*I selected the VITS (Variational Inference Text-to-Speech) model, which is widely regarded for producing natural and expressive audio. Coqui TTS provides a flexible implementation of VITS with support for custom datasets, making it suitable for my goal of developing a Pashto-compatible TTS system.*

***RATIONALE***

*My decision to use Coqui TTS with VITS was based on several practical and technical factors:*

* ***Open-source and actively maintained****: Coqui TTS is a well-documented open-source project with community support.*
* ***Flexible dataset integration****: It allowed me to create a custom Pashto dataset using existing .wav audio files and matching them with ground truth text from a sentences.json file.*
* ***Support for multilingual data:*** *Although Pashto is not officially supported, the platform allowed me to prepare and organize the data using the LJSpeech format, which made integration relatively straightforward.*
* ***High-quality output****: VITS is known for producing high-quality audio, even with smaller datasets, which is critical when working with low-resource languages.*

***OPEN SOURCE MODEL FOR FINE-TUNING***

*Although there is no pre-trained Pashto TTS model readily available, Coqui TTS’s VITS architecture can be fine-tuned using aligned Pashto audio-text pairs. If fine-tuning resources were accessible (such as compute and large-scale labeled datA), I would use this setup for end-to-end training. Suggested model for fine-tuning:→ tts\_models/multilingual/multi-dataset/vits (from Coqui)This multilingual VITS model can potentially be adapted for Pashto via fine-tuning.*

***PRACTICAL WORKAROUND***

*In the absence of a direct Pashto model, a viable workaround is to manually prepare a Pashto dataset in LJSpeech format (as I did), and then train a new model from scratch or perform transfer learning using a closely related language, such as Persian or Urdu, which share phonetic and script similarities with Pashto.*

***PLANNED TRAINING AND CHALLENGES***

*I originally planned to train the Pashto TTS model using the Coqui Trainer module. I successfully preprocessed the dataset, matched audio files with sentence IDs, and created metadata for training. However, I encountered an unresolved dependency issue (TTS.trainer not installing due to cache conflicts), which blocked the training process despite extensive debugging.*

***FUTURE IMPROVEMENTS***

*If fine-tuning and training resources become available, the system could be improved in the following ways:*

* *Proper model training or fine-tuning using a larger Pashto corpus to improve fluency and pronunciation.*
* *Phoneme integration to enhance clarity in complex words or dialectal variations.*
* *Prosody modeling for more expressive and natural speech.*
* *Speaker embedding to create multi-speaker or gender-adaptive models*