Final Project Technical Report

Title : Spoken dialogue system
Course : Speech processing
Prof H. Sameti

By:

Mohammad Taha Teimuri
401212498

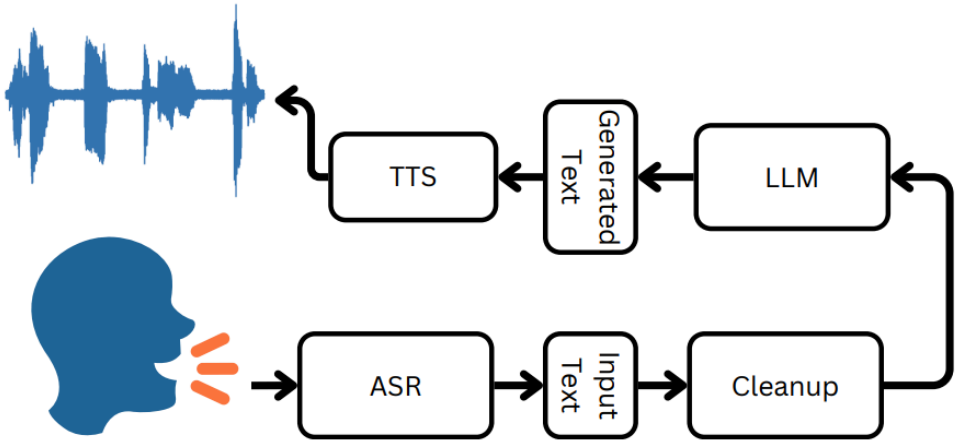
Ali Derakhshesh
401203252

Introduction

In this project, we have developed a virtual spoken assistant that is capable of speech-to-speech communication. Furthermore, similar projects seem to be scarce. To this, we believe that the work in this project is of high importance.

Summary

To achieve our goal we have devised a pipeline drawn below.



To get a response in audio format from an audio response two main methods seem the most feasible. 1- speech to text, text to text, text to speech 2- speech to speech which is a more advanced approach most recently by Openai’s ChatGPT 4o voice model.
As proposed in our proposal, we have taken the first approach and the pipeline consists of three main parts:

1. Auto Speech Recognition (ASR)
2. Large Language Model (LLM)
3. Text to Speach model (TTS)

During all of our training and inference process, we have used multiple datasets from hugging faces, all of which are listed below:

1. Mozilla-foundation/common_voice_17_0
2. pourmand1376/asr-Farsi-youtube-chunked-30-seconds
3. pourmand1376/asr-Farsi-youtube-chunked-10-seconds

4. Timi2/alpaca_persian_dataset
5. sinarashidi/alpaca-persian
6. mohammadhossein/SP_HW5_PersianTTS
7. SeyedAli/Persian-Speech-Dataset

We have also developed a Gradio web app to ease inference of our pipeline.
We will further discuss each part of the project in the following sections.

Datasets

To finetune or test each of our models we have used several datasets all provided by the hugging face interface.

1. mozilla-foundation/common_voice_17_0: The Common Voice dataset is freely available under a Creative Commons license, promoting open collaboration and innovation in natural language processing and speech technology. By supporting multiple languages, including underrepresented ones like Persian, the project aims to democratize access to speech technology, ensuring that voice-enabled applications can help a diverse range of audiences.
   It consists of multiple sections including Train, Validation, and Test. It is also noteworthy that some of the data in this dataset has been invalidated and should not be used. (used for automatic speech recognition modeling)

2. pourmand1376/asr-farsi-youtube-chunked-30-seconds: One of the bigger publicly available datasets in the Persian language. It has been gathered by crawling the caption from many popular and Persian YouTube channels. All the samples have a length of roughly equal to 30 seconds. (used for automatic speech recognition modeling)

3. pourmand1376/asr-farsi-youtube-chunked-10-seconds: Another dataset gathered by Amir Pourmand. This dataset contains 141506 samples. This data is pretty similar to the previous one in structure and nature. The only difference is the length of the samples which is 10 seconds. This makes this dataset more suitable for text-to-speech purposes. However, the problem is that there is no column in this data to show phonemized text. We will discuss how we solved this problem in the next sections. (used for text-to-speech modeling)

4. Timi2/alpaca_persian_dataset: This dataset constructed for the sake of the project contains 194 questions and answers used during the training of the LORA adaptor. This dataset is particularly useful as a clean dataset albeit small, on

various topics is needed for this purpose. (used for large language model fine-tuning)

5. sinarashidi/alpaca-persian: Yet another dataset that we discovered during training of the LORA adaptor for the LLM. It contains 35k questions and answers. Since we had no prior information regarding the cleanness of this dataset and its undesirable structure we refrained from using it. (used for large language model fine-tuning)

6. mohammadhossein/SP_HW5_PersianTTS: This dataset is gathered by Mohammad Hossein Sameti, a member of the teaching assistant team for the speech processing course. This data is similar to the common_voice dataset and is a multi-speaker dataset containing 52163 samples. The challenge with this data is that it contains some noisy voice chunks so we have applied some audio processing techniques to overcome this challenge. (used for text-to-speech modeling)

7. SeyedAli/Persian-Speech-Dataset: This is a cool dataset that contains audio segments extracted from movie dialogues. Although this data contains 2838 samples the quality of audio files is perfect. This dataset is also a multi-speaker dataset which fits our purpose of having a multi-speaker TTS. The other thing to mention is that this data has a column for IPA-transformed text. But as there is not a good IPA transformer for Persian we used other techniques to extract phonemes which we will discuss in upcoming sections. (used for text-to-speech modeling)

Pipeline Analysis

As explained previously our pipeline consists of three main modules. We will further our experiments and results concerning each of these parts.

Automatic Speech Recognition (ASR)

The first step of our pipeline aims to convert speech sound to text. As discussed in the project proposal there are different popular models available for this task, but at the moment the best model for Persian ASR is whisper. Whisper is available in a different number of parameters which we have tested to find the most suitable model for each situation. Fortunately, whisper is a multilingual; model that allows for use in the Persian language. Albeit fine-tuning is needed.

- Whisper tiny/base/small: These versions of the model have been designed to take as little space as possible. This makes them a very good option for use on GPUs with lower memory or just running them on CPU. Of course this smallness in size results in lower accuracy.

- Whisper medium: Containing over 700 million parameters seats on the larger side of models. This version provides a nice balance between storage requirements and the accuracy of the results.

- Whisper large v1/v2/v3: Largest model whisper model in existence. Has three versions and the last one has been trained on the most data. Due to its large size however, a high amount of GPU memory is required which we were not able to acquire, thus no experiments were performed on this version.

Large Language model (LLM)

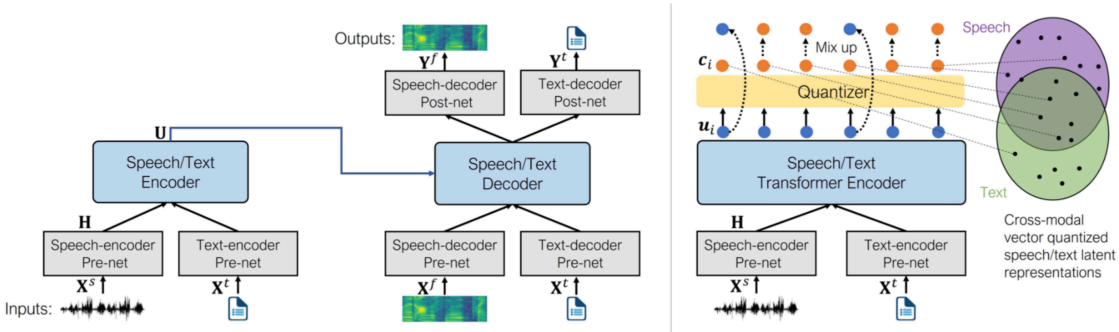
In a spoken dialogue system, we need a core to generate responses as we give it input. This core needs to generate human-like responses. Previously researchers have used RNN, LSTM, and GRU to model the languages but the famous problem of “forgetting” limited the ability of those models to generate perfect responses.
Nowadays with the improvement of AI field and the introduction of the transformer model architecture we have the ability to model the human language perfectly as the machine understands it. Transformers are getting bigger and bigger in size and today large language models are popular among researchers in this field.
We decide to use LLAMA-3-8b-instruct which is developed by FaceBook (Meta) company. This model has 8 billion parameters and it can generate perfect answers. It is also instruction-based so we can give it a prompt for instructing it to how to respond.

Text To Speech (TTS)

In the last step of our pipeline, we need a module to convert text to speech so we can call it a spoken dialogue system. Text-to-Speech (TTS) technology has achieved excellent advancements over recent decades. Early TTS systems relied heavily on concatenative synthesis like matching together pre-recorded speech segments. However, with the improvements in the deep learning field, modern TTS systems use neural network architectures. Previously tts systems have used CNN and autoregressive architectures. But as we mentioned by the advances of transformer architecture transformer-based tts systems are showing good results and getting popular among researchers in this field. We have lots of options to choose from for text-to-speech model architecture like WaveNet, FastSpeech, Glow-TTS, and others. We choose the speecht5 model.

- Why do we choose speecht5 ?
  There are lots of text to speech models on hugging face even fine tuned in Persian. But the problem is that almost all of them are single-speaker models which only speak with the static sound of one predefined speaker. So instead of making a model that already exists among the Persian tts community, we decided to fine-tune a multi-speaker model which is rare for the Persian language. So we used speecht5 which also gets a speaker embedding and could convert text to speech with the voice of the speaker you want. So it mimics how speakers talk and convert text to speech with their voice.

- Speecht5: SpeechT5 was developed by Microsoft researchers. It was first introduced in this paper. Motivated by the success of T5 (Text-To-Text Transfer Transformer) in pre-trained natural language processing models, they have proposed a SpeechT5 framework that explores the encoder-decoder pre-training for self-supervised speech/text representation learning. So it learns a cross-modal embedding between audio and text.



(a) The model architecture of SpeechT5

(b) The joint pre-training approach

Training

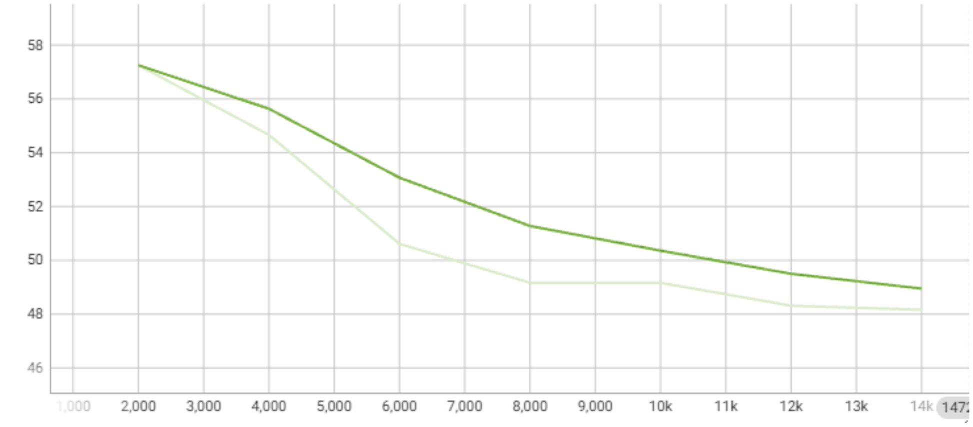
ASR

To Train the whisper model we have used two datasets in consecutive order. First, we will use common voice. This is because the voices in this dataset of higher quality and invalid samples are few and far between. On the other hand samples in asr-farsi-youtube-chunked-30-seconds have had less supervision and for example many of them contain background music.

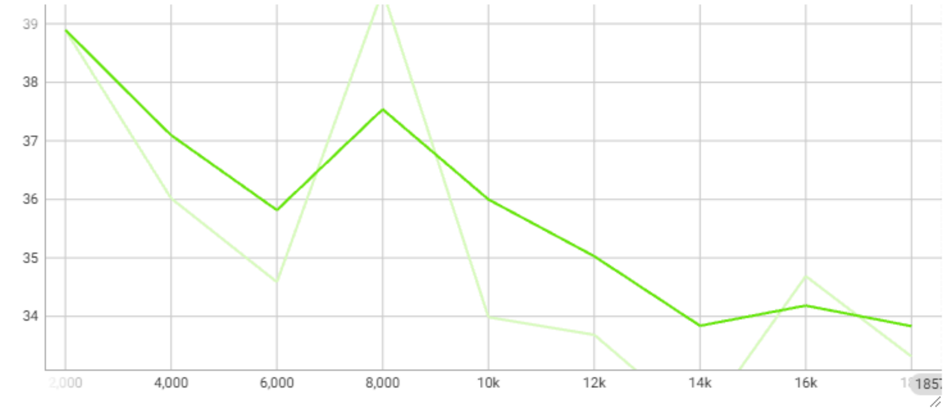
We have trained different versions of the model over the given datasets. All of them can be found on hugging face at whisper_medium, whisper_tiny, whisper_small

Here you can see some of the logs for the wer/batch:

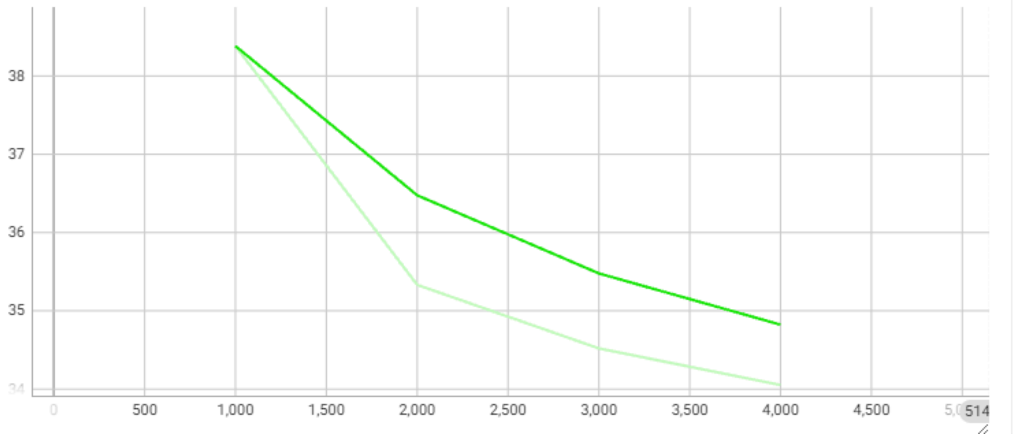
Whisper tiny



Whisper medium



Whisper small



An important fact to note is that in the results above, whisper small has almost reached the same accuracy as whisper medium. So it might be a better idea to small version instead of medium.

LLM

To utilize different LLMs, We have utilized unsloth. A library that facilitates the training of an adaptor on a LLM.
We have applied an adaptor to mistral, llama 3, and gemma 2 among which llama 3 seems to achieve superior results in the Persian language.
Furthermore, we have noticed a decline in model response quality after using the dataset suggested by unsloth developers. We believe that this problem arises from the fact that the dataset used by them only contains non-Persian examples.
To solve this issue we created a dataset (Timi2/alpaca_persian_dataset). It contains about 200 question-and-answer samples in Persian.
A version of the fine-tuned models are accessible on hugging face. Provided on Gemma2, Mistral, LLam3.

TTS

We have developed 3 versions of speecht5 in other words we have fine-tuned speecht5 model using 3 different datasets. In the following, we will discuss about challenges we face and how we solved them for each dataset.

1. mohammadhossein/SP_HW5_PersianTTS
As we mentioned earlier this dataset contains some noisy audio samples. First we fine tuned the speecht5 on this dataset without preprocessing. But then we recognized that because of noisy samples, this model generates a highly noised audio.

So we have applied these steps for data preprocessing :

1. Removing audios with a length of less than 1 second or higher than 12 seconds. Audio files in this range were too noisy in this dataset.

2. Denoising the audio: we have used the noise reduce library in Python to reduce the background noise. It uses the spectral gating method to reduce noise. The audio is transformed to frequency using STFT then if the magnitude of a frequency component in a frame is below the threshold, it is reduced in amplitude. Finally, it will use inverse STFT to convert the frequency domain to the time domain.

3. Trimming the silence: some audio samples starts and ends with long unvoiced segments so we used librosa trim function to solve this problem.

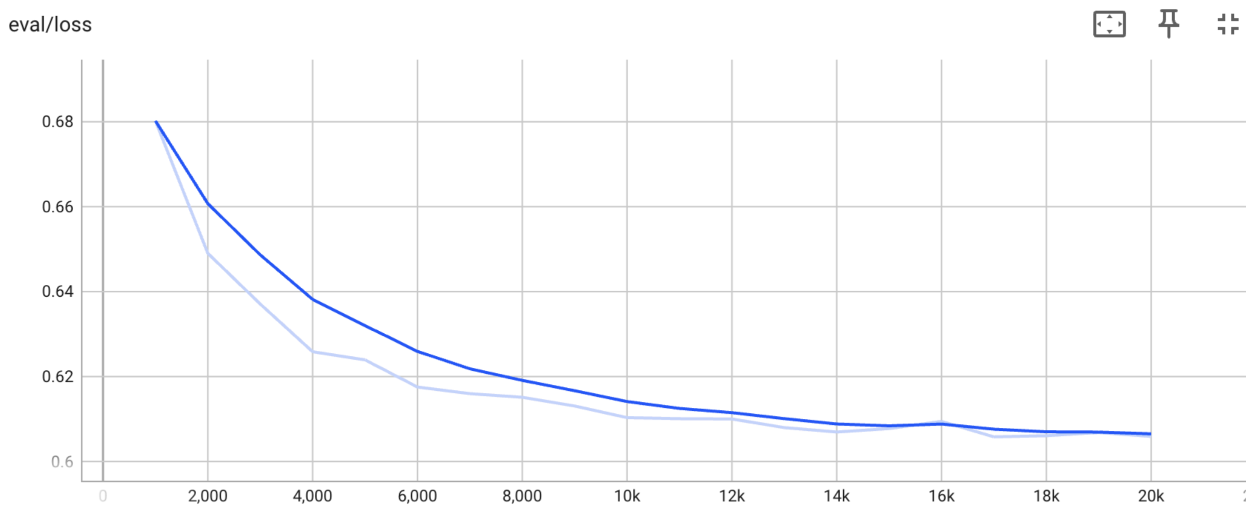
Note : All of this version codes are available in speecht5_tts_v1_commonvoice.py file which we have uploaded on Quera.

After the preprocessing we have fine tuned the speecht5 on the dataset and we get better results.

- How do we evaluate the speecht5 results ?
  As we know it is hard to evaluate a tts model because the measurements are either subjective which needs human evaluation or objective which just consider things like signal to noise ratio, etc.
  But speecht5 loss itself is a good metric because it captures essential things and consists of Mel-spectrogram loss, Duration loss, Pitch loss, Energy loss, etc. So we will use this valuable loss to select the best model which has the lowest loss.

*You can check the results in the run folder, where you can find tensorboard logs.
Our Best speecht5 model tensors on this dataset available at huggingface

Speecht5 on mohammadhossein/SP_HW5_PersianTTS

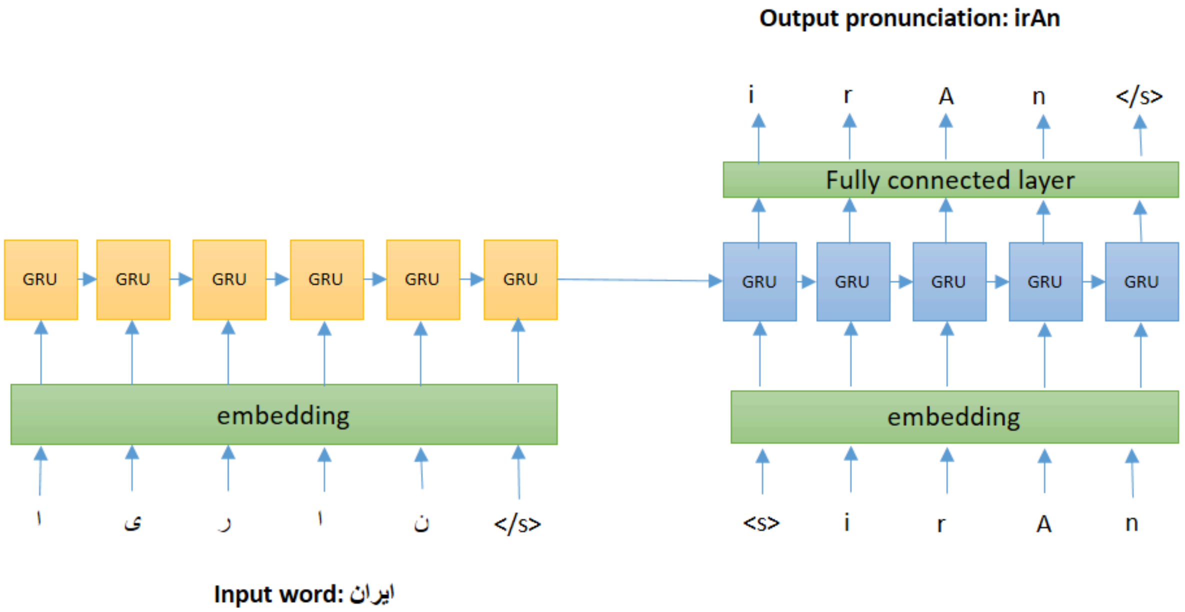


2. SeyedAli/Persian-Speech-Dataset

This dataset is so clean and high quality. As we mentioned earlier it contains dialogues from movies. The challenge with this dataset is that it has IPA of each sentence but we don’t have a good ipa convertor in persian(we will need it at inference time).

- So we studied 3 different phonemizer libraries:
  - persian_phonemizer
  - PersianG2p
  - epitran

Among them PersianG2P shows the best results in phonemizing the persian text. It uses a seq2seq GRU network trained to map raw persian text to pronunciation like text which you can see in the following image.

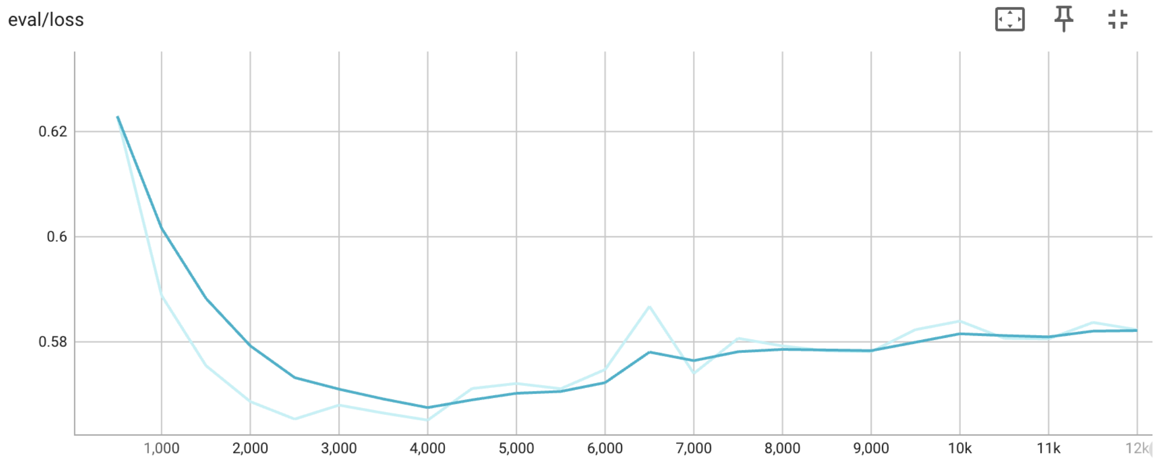


Finally we fine tuned the second version of speecht5 on pairs of audio and phonemized text of this dataset and we got good results.

Note : All of this version codes are available in speecht5_tts_v2_movie.py file which we have uploaded on Quera.

*You can check the results in the run folder, where you can find tensorboard logs.
Our Best speecht5 model tensors on this dataset available at huggingface

Speecht5 on SeyedAli/Persian-Speech-Dataset



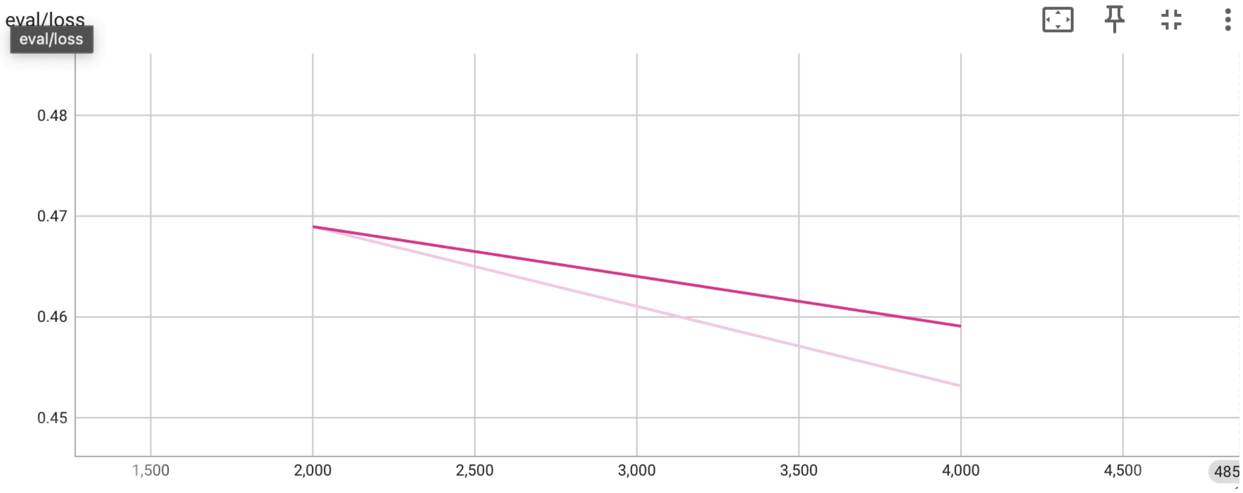
3. pourmand1376/asr-farsi-youtube-chunked-10-seconds
We have 2 challenges with this dataset. First challenge is that it just has the raw text without phonemes. We address this challenge in the previous part (version2) which we used the PersianG2P network for phonemization.
The second challenge is that this dataset is really huge (20GB of data) so we used torch multi-processing with the spaw method to apply multi-GPU processing.

Note : All of this version codes are available in speecht5_tts_v3_youtube.py file which we have uploaded on Quera.

Although we have used multi processing but due to large size of data the runs were slow on our servers also the university power outages distributed our runs several times but we finally achieve 4000 steps on this dataset.

*You can check the results in the run folder, where you can find tensorboard logs.
****Our Best speecht5 model tensors on this dataset available at huggingface**

Speecht5 on pourmand1376/asr-farsi-youtube-chunked-10-seconds



Finally we compare best models of each 3 versions and recognized that version 2 on movie dialogue dataset has the least noise and unvoice stops so we put it in our pipeline.

Make TTS results better ?

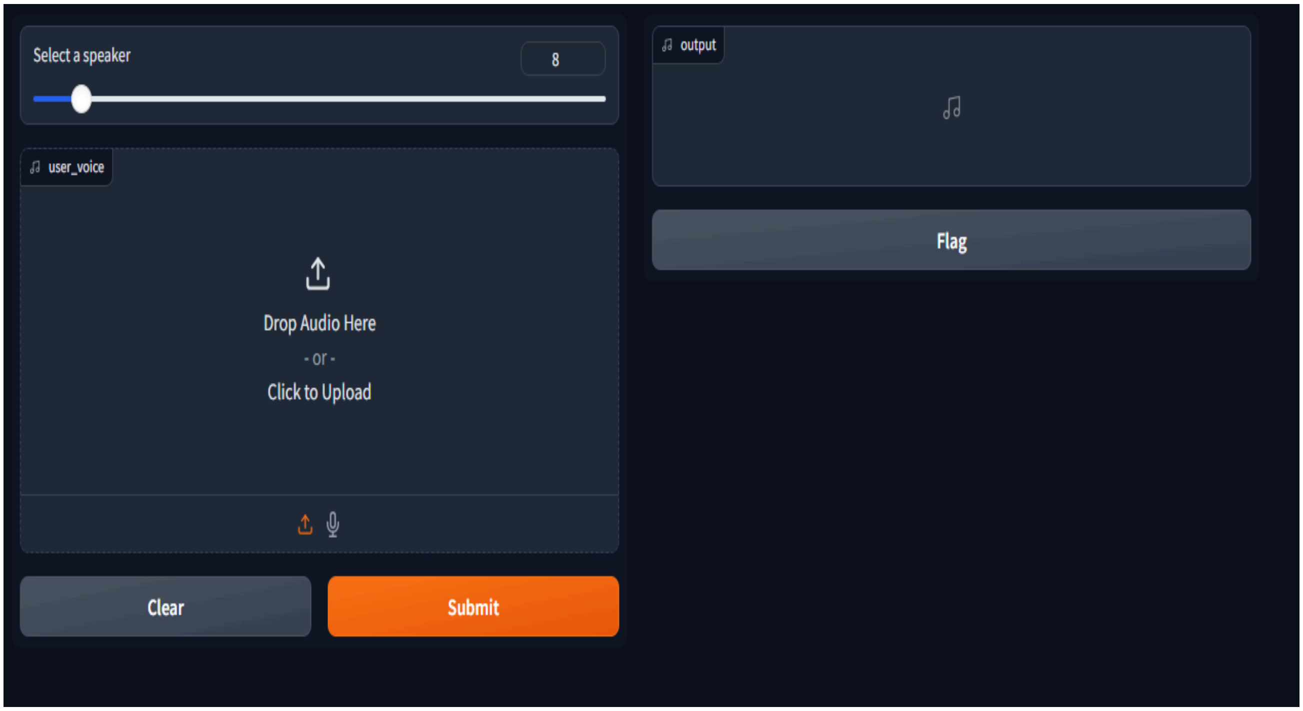
Yes, we find out that by doing some post-process on tts result we can make it even better. We applied a denoising using noise reduce library which was introduced

previously. We also apply an audio normalization using pydub library. We apply normalization to distribute the loudness uniformly among the audio.
You can check the code from Project_gradio_app.py in the tts function.

Gradio

Due to using multiple deep models in our setup, It is in fact not possible to deploy our pipeline on free-hugging face space. To demonstrate our model we have deployed a gradio webapp on a private server which you can view our gradio webapp on here.
You can try choosing a speaker first then click on the mic icon to record your voice in Persian then submit it. (Please wait until the audio is uploaded then press submit, unless you will get error because the file is not uploaded.)

We try this several times and we get that female speaker number 8 generates best responses for TTS but you can change the speaker. Please note that speecht5 audio depends on the speaker embedding so for some speakers the result maybe noisy.



Select a speaker 8

- If you get any problem with connecting to the gradio server please contact us. The links may expire due to university power outages.
- Please use a VPN while using our gradio app.

Reference

[1] Gong, Yuan, et al. "Whisper-at: Noise-robust automatic speech recognizers are also strong general audio event taggers." arXiv preprint arXiv:2307.03183 (2023).

[2] Jiang, Albert Q., et al. "Mistral 7B." arXiv preprint arXiv:2310.06825 (2023).

[3] Team, Gemma, et al. "Gemma: Open models based on gemini research and technology." arXiv preprint arXiv:2403.08295 (2024).

[4] Touvron, Hugo, et al. "Llama: Open and efficient foundation language models." arXiv preprint arXiv:2302.13971 (2023).

[5] Ao, Junyi, et al. "Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing." *arXiv preprint arXiv:2110.07205* (2021).