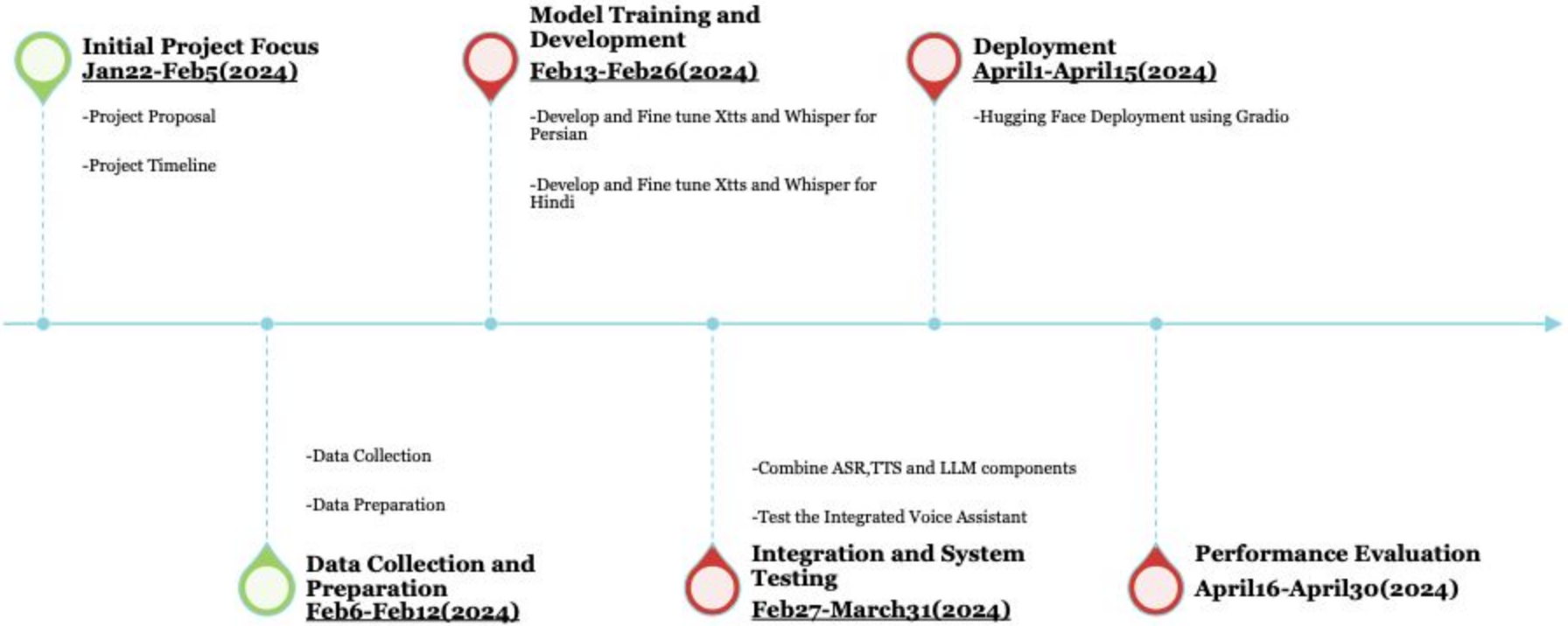
Voice-Activated Teaching Assistant in Persian and Hindi

Project Advisor: Dr Vahid Behzadan

Team Members:

- Bahareh Arghavani Nobar
- Devnath Reddy Motati

Project Milestones



Data Collection

- In this week we planned to collect required dataset and do preprocessing.For Persian TTS 1- there is a available 30 hour female and male tts which you can find here “persian tts dataset” .

- But the problem with this dataset is it has Afghan accent and note that this dataset is provided by Azure .

- The second Option for TTS and also for ASR is Common voice Persian which is 410 Hour dataset that you can find here at “https://commonvoice.mozilla.org/en/datasets” and the problem with this dataset for XTTS fine tuning is that the quality of voice is not good and lot of noise.

- For the Hindi Dataset there are lots of datasets available but the one we found to be good can be found here at “Summary of Hindi Data”

- The Hindi speech dataset is split into train and test sets with 95.05 hours and 5.55 hours of audio respectively.

- There are 4506 and 386 unique sentences taken from Hindi stories in the train and test sets, respectively, with no overlap of sentences.

- The train set contains utterances from a set of 59 speakers, and the test set contains speakers from a disjoint set of 19 speakers.

- The audio files are sampled at 8kHz, 16-bit encoding. The total vocabulary size of the train and test set is 6542.

Data Preprocessing

- So we decided to create new dataset by help of available audiobook in Persian in youtube channel.
- So by help of "yt-dlp" library we download .wav file and then pass it to google speech API and it clip the audio and create transcript .We made use of the free trial which is free for 90 days. But the duration of each clips are different so we wrote a python code to clip audio according to google speech start and end mili-second that is provided.
- For preprocessing we remove the 30 second start and end of each chapter because of music and noise so we can start training from next step.
- You can find the python code in the form of a python notebook at the following link”https://colab.research.google.com/drive/1aIicWh-mbDi9-A7PbqTrtF-yFeNuWN7p?usp=sharing”.

Retrieval Augmented Generation

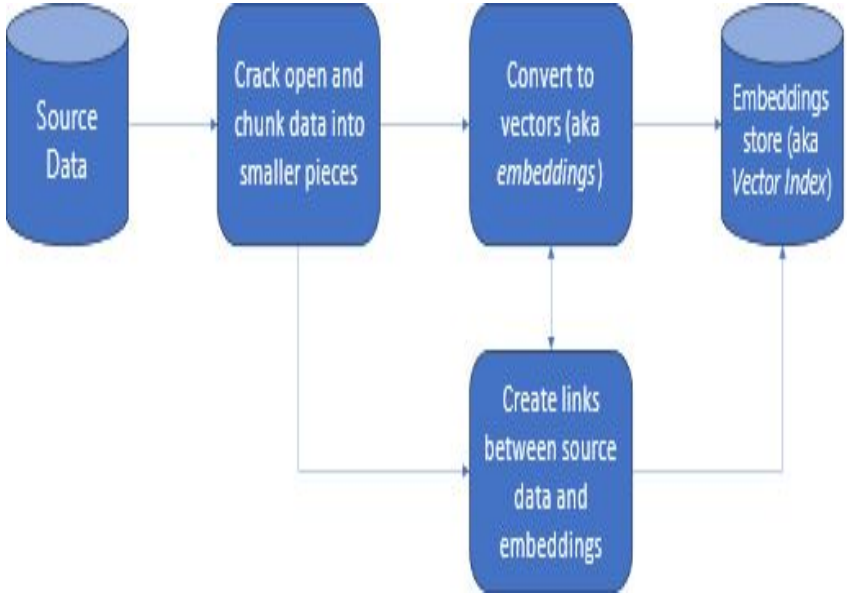
- Retrieval Augmented Generation (RAG) is a pattern that works with pre trained Large Language Models (LLM) and your own data to generate responses.

- Traditionally, a base model is trained with point-in-time data to ensure its effectiveness in performing specific tasks and adapting to the desired domain. However, sometimes you need to work with newer or more current data. Two approaches can supplement the base model: fine-tuning or further training of the base model with new data.

- Fine-tuning is suitable for continuous domain adaptation, enabling significant improvements in model quality but often incurring higher costs. Conversely, RAG offers an alternative approach, allowing the use of the same model as a reasoning engine over new data provided in a prompt.

Technical overview

- Source data: this is where your data exists.
- Data chunking: The data in your source needs to be converted to plain text.
- Converting the text to vectors: called embeddings. Vectors are numerical representations of concepts converted to number sequences, which make it easy for computers to understand the relationships between those concepts.
- Links between source data and embeddings: this information is stored as metadata on the chunks created which are then used to assist the LLMs to generate citations while generating responses.



Week 1	Week 2	Week 3
Preliminary Report on Project Focus	**strategic planning and task delegation**	**Data Collection and Dataset Creation**
❖ Collaborated with Dr. Vahid Behzadan to identify a challenging NLP problem. ❖ Discussed potential project topics and their significance in the field. ❖ Explored the landscape of NLP tasks, highlighting the absence of Persian TTS support. ❖ Noted the surprising lack of support for Persian TTS in major platforms like Google Translation, Alexa, Siri, and ChatGPT. ❖ Recognized the importance of addressing this gap and making a meaningful contribution to the field. ❖ Discussed the potential use of an automated Mean Opinion Score (MOS) as an innovative evaluation metric for our project.	❖ Developed a comprehensive timeline, detailing essential project milestones. ❖ Assigned specific responsibilities to team members, focusing on crucial aspects such as data collection, data preparation, tool selection, and model fine-tuning. ❖ Recognized the importance of data collection and outlined sources and methodologies to be employed. ❖ Specified data preparation steps, ensuring the quality and relevance of the collected data. ❖ Opted for Notion as our project management tool to facilitate efficient tracking and collaboration. ❖ Decided to fine-tune the latest Couqi TTS model, aiming to enhance its support for the Persian language and improve overall accuracy. ❖ Planned the development of a pipeline (ASR --> RAG --> TTS) catering to both Hindi and Persian languages. ❖ Discussed the automation of the evaluation system, with a focus on innovative Mean Opinion Score (MOS) metrics. ❖ Outlined deployment strategies for the TTS system, considering real-world applications. ❖ Established clear communication channels within Notion, promoting seamless collaboration between advisors and team members.	❖ Focused on data collection as a primary objective for the week. ❖ Established a GitHub repository to streamline version control and collaborative development. ❖ Identified available datasets for Persian TTS training, with a primary source being the "Persian TTS Kaggle" dataset provided by Azure. ❖ Acknowledged a minor Afghan accent issue in the Kaggle dataset, prompting the decision to create a new dataset. ❖ Utilized the "yt-dlp" library to download audiobooks from YouTube in WAV format. ❖ Leveraged the Google Speech API, which offers a 90-day free trial, for transcription and identification of relevant audio clip start and end seconds. ❖ Developed a code to automate the clipping of audio based on the provided start and end seconds from the Google Speech API results. ❖ Ensured removal of the initial and final 30 seconds of each chapter to eliminate music and noise, resulting in a cleaner dataset. ❖ Prepared a 25-hour dataset with single-speaker audio, focusing on eliminating accents and incorporating emotion. ❖ Considered the option of using the Common Voice dataset by Mozilla; however, dismissed it due to concerns regarding the recording quality

Upcoming week Plan

- Curate an NLP Knowledge Base and Format for Retrieval (Devnath Reddy Motati).
- Make requirement changes on XTTS architecture and tokenization to support Persian Language (Bahareh Arghavani Nobar)