

## **TEAM: SynthLang**

**Overview:** The main purpose of the “Technical Demos” is to very clearly communicate the extent to which the team has identified key challenges in the project, and has proven solutions to those challenges. Grading is based on how complete/accurate the list of challenges is, , and how convincingly and completely the given demos cover the given challenges.

This template is fleshed out by the team, approved by CS mentor, and brought to demo as a grading sheet.

### **Risky technical challenges**

Based on our requirements acquisition work and current understanding of the problem and envisioned solution, the following are the key technical challenges that we will need to overcome in implementing our solution:

**C1: CPU Inference.** TTS systems are generally compute-intensive and since NAU's GPU cluster, monsoon, does not allow any hosting, we will show that audio generation can still work with a good CPU.

**C2: Web Interface.** Web interface is essentially the means of access and UI for the client and colleagues.

**C3: Model Selection.** Modular model pipeline is essential for future proofing of the technology so being able to select a desired model from at least two is indication of working modularity.

**C4: Audio Controls.** Audio levels of speech must be adjustable to demonstrate flexibility of program.

**C5: Asynchronous Audio Generation.** Asynchronous Audio Generation ties in with CPU inference. Showing that these functions will back up our case for CPU-based TTS. Audio will be generated while text of the language is being outputted.

**C6: Dataset Ingestion/Validation.** The pipeline must be able to validate and consume training data. This requires efficient parsing as well as robust error handling. An ideal demo would show, when given an invalid dataset, a descriptive error and steps to move forward. It would also be able to confirm that a given dataset is error-free.

**C7: Config Parsing/Validation.** The pipeline must be able to parse configuration parameters as well as validate them. This requires a custom parser and robust error handling. An ideal demo would show, when provided invalid config parameter, a descriptive error and a brief explanation of the options available for a given parameter. Upon receiving a valid config file, the correct parameters in the pipeline would be set. The demo would then display this.

**C8: Acoustic Model Training.** The pipeline must be able to train models separate from the web application. Training the pipeline will use the client's titan GPUS and an ideal demo

would demonstrate that the pipeline can create acoustic models that are able to produce audio of reasonable quality.

**C9: Acoustic Model Fine-Tuning.** Existing models must be able to be fine tuned on small datasets to synthesize audio for under-resourced languages. An ideal demo would show that a model, pretrained according to this method, can synthesize audio of reasonable quality in the target language.

### **Challenges covered by demos:**

In this section, we outline the demonstrations we have prepared, and exactly which of the challenge(s) each one of them proves a solution to.

---

### **Demonstration 1: Web Application**

Challenges addressed: CPU Inference, web interface, model selection, Asynchronous Audio generation, Audio controls

Flight Plan: Step by step overview of demo

1. First, user will select which model to use via the web interface.
2. Second, user will send text input to the model via the web interface.
3. The model will then translate and output text of user input.
4. The model will also asynchronously generate audio of translated input.

Evaluation:

- Convincingly demo'd each of listed challenges?
  
- Other evaluative comments:

---

## Demonstration 2: Training Pipeline

Challenges addressed: Dataset ingestion and validation, parsing of config files to set training options, training on small data set ~100 samples

Flight Plan: Step by step overview of demo

1. Write a invalid config file detailing various training parameters
2. Feed config file to demo pipeline
3. Program gives output stating that certain parameters are invalid
4. Correct invalid parameters and re-run program
5. Program gives output validating given parameters and confirms that the correct parameters have been set

Evaluation:

- Convincingly demo'd each of listed challenges?
  
- Other evaluative comments:

### **Other challenges recognized by not addressed by demo:**

If there were challenges you listed earlier that were *not* covered by a demo, list here. This will hopefully be a short list...but better to be clear about where you are. If you have items here, you could list (if applicable) any pending plans to reduce these risks.

- Acoustic Model Training
- Acoustic Model Fine Tuning