Software Development Project

Multilingual: Text to Speech
Presentation 3

Albert Millert
Shalini Priya
Wenjun Sun
Soklay Heng

Agenda

1. Primary Goal
2. Model Used
3. Architecture
4. Planning
    a. Preprocessing
    b. Language Classifier
    c. WebApp
    d. Containerization
5. Timeline
6. Summary

Primary goal

Develop a web application that uses Grad-TTS Model for Text to Speech conversion. Languages supported by the TTS converter app are:

- English
- French

Model Used: Grad-TTS

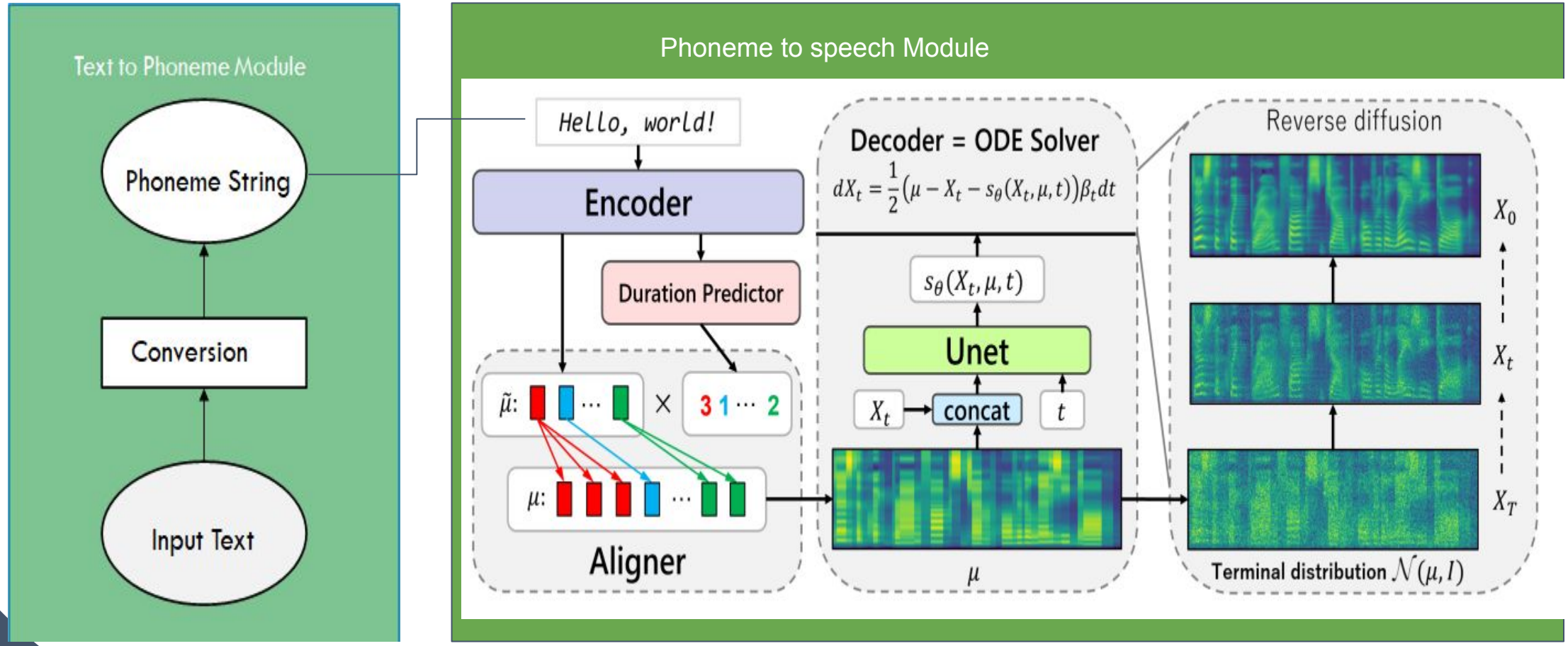
A novel text-to-speech model with score-based decoder producing Mel spectrograms by gradually transforming noise predicted by encoder and aligned with text input by means of Monotonic Alignment Search.

Encoder + Duration Predictor + Decoder = Mel Spectrograms

Why Grad-TTS?

What are the other models?	• GLOW TTS ,Tecotron 2
What is the problem with other models?	• Drawback is in terms of synthesis quality.
Why Grad TTS is better?	• Capable of real-time synthesis with good quality ,flexible inference ,fast on GPU

Architecture



In Progress Completed

Planning

1

Grid5000 Account

Testbed for experiment

2

Preprocessing

Generating phoneme from input text

3

RNN LSTM Classifier

For Identifying language

4

Training

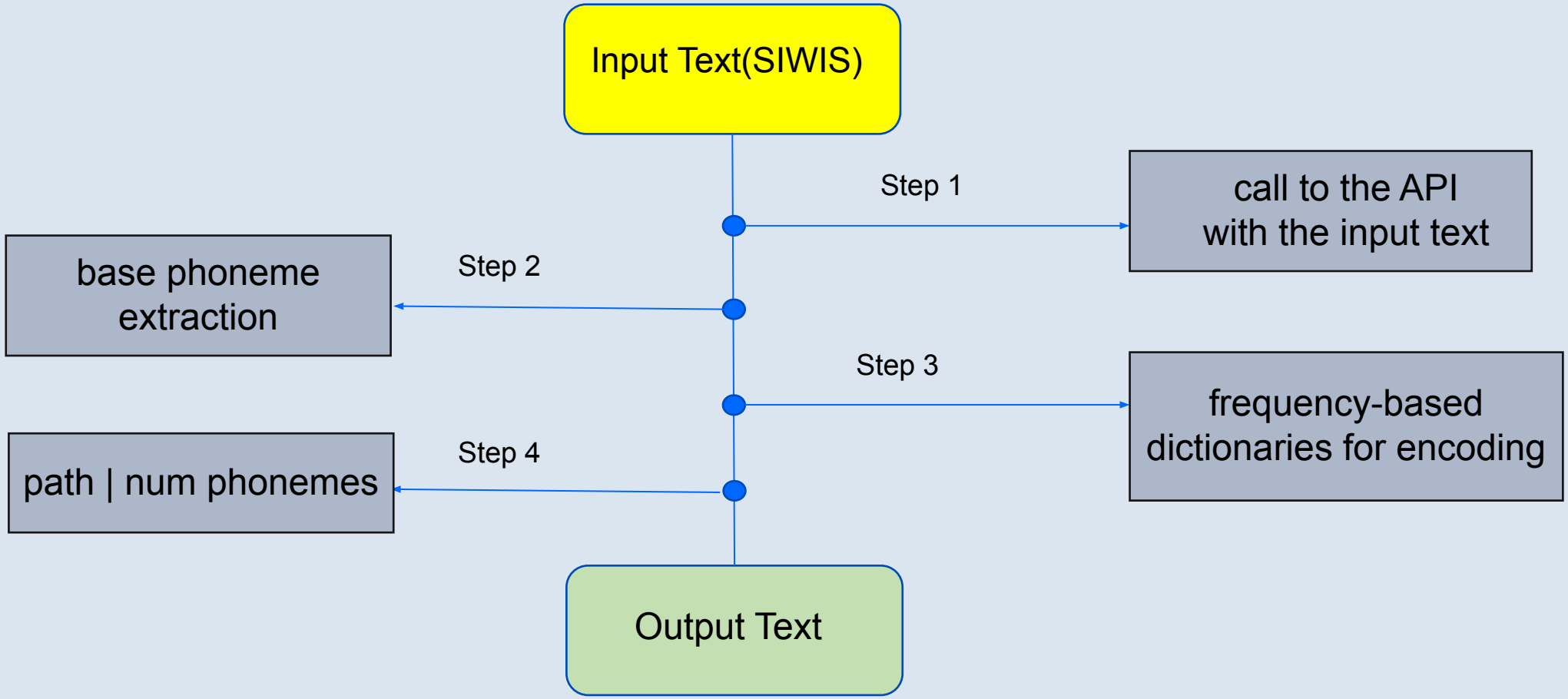
To train a model

5

Deployment

Combine all modules and deploy working model on WebApp

2 Preprocessing (French)



2 Preprocessing - code listing

```python
def process(path):

    stream = os.popen(f"perl xd.pl {path} texts hts run")
    out = stream.read().replace(prefix, "¤").replace(suffix, "¤").split("¤")
    lines = [line.strip() for line in out[1:-1][0].split()]

    phonemes = []
    for x in lines:
        splt = x.replace("-", "¤").replace("+", "¤").split("¤")
        if len(splt) > 1:
            phonemes.append(splt[1])

    return phonemes

output = list(map(lambda file: process(file), paths))
```

```python
flat = [xi for x in output for xi in x]
phon2cnt = dict(Counter(flat).most_common())
phon2idx = {x: i for i, x in enumerate(phon2cnt.keys())}
idx2phon = {v: k for k, v in phon2idx.items()}
```

```python
with open(os.path.join(os.path.dirname(os.path.abspath(dir)), "output.txt"), "w") as fout:
    for path, phonemes in zip(paths, output):
        line = f"{path} | {' '.join([str(phon2idx[x]) for x in phonemes])}\n"
        print(line.split("/")[-1])
        fout.write(line)
```

2 Preprocessing - output

```
s/part1/neut_parl_s06_0679.lab | 2 1 3 16 0 23 25 8 22 17 3 8 14 4 14 28 3 23 15 5 32 10 0 11 16 3 10 31 2
s/part1/neut_parl_s02_0296.lab | 2 21 29 1 18 5 11 16 15 4 1 1 4 10 0 21 7 2
s/part1/neut_parl_s04_0537.lab | 2 32 1 11 25 29 1 18 5 12 19 0 1 9 17 11 21 29 1 0 4 7 4 1 11 17 4 16 15 1 4 18 17 5 6 5 11 16 13 16 15 19 10 11 4 10 4 7 21 2
s/part1/neut_parl_s01_0128.lab | 2 6 0 29 1 12 29 27 15 17 12 1 0 20 27 6 5 0 5 4 14 2
s/part1/neut_parl_s05_0146.lab | 2 21 19 32 10 0 32 5 1 6 0 19 21 5 9 8 3 1 0 24 14 11 1 0 3 5 0 8 4 10 6 15 14 11 23 9 7 6 21 19 2
s/part1/neut_parl_s03_0289.lab | 2 24 1 15 10 22 17 13 1 11 17 4 6 0 35 7 26 4 7 12 30 9 8 3 16 24 15 14 2
s/part1/neut_parl_s02_0640.lab | 2 4 8 3 1 1 5 6 5 12 0 19 21 5 12 1 0 9 1 0 29 7 13 23 3 5 22 10 4 12 10 4 5 21 16 3 20 2
s/part1/neut_parl_s01_0328.lab | 2 7 3 22 1 6 14 9 9 17 11 9 5 12 19 21 29 1 0 12 20 28 3 7 11 22 26 6 0 8 32 26 22 11 20 13 26 31 15 14 6 1 4 18 17 9 8 4 10 6 6 1 11 4 2
s/part1/neut_parl_s04_0025.lab | 2 29 7 23 13 19 4 16 15 9 14 22 33 25 34 18 6 17 1 21 10 11 9 5 0 8 28 17 9 7 4 15 14 11 16 6 7 9 18 27 2
s/part1/neut_parl_s01_0671.lab | 2 13 19 3 8 21 29 1 18 17 28 18 27 13 26 13 7 21 26 9 5 15 7 4 18 17 3 16 11 1 3 2
s/part1/neut_parl_s01_0557.lab | 2 7 3 25 26 11 8 13 19 4 29 1 18 17 28 18 27 12 0 5 4 7 2
s/part1/neut_parl_s04_0143.lab | 2 11 5 12 19 0 11 29 1 24 8 9 5 22 1 12 0 19 21 4 10 6 1 15 14 9 15 14 2
s/part1/neut_parl_s06_0339.lab | 2 4 27 11 24 19 0 23 13 5 4 8 12 1 22 14 11 16 0 6 0 26 2
s/part1/neut_parl_s04_0082.lab | 2 4 8 13 8 25 20 6 30 0 30 22 15 14 11 20 13 5 6 1 12 2
s/part1/neut_parl_s04_0560.lab | 2 4 10 0 6 27 9 8 4 5 4 20 24 10 17 6 5 6 5 5 21 16 11 5 3 16 0 9 8 3 1 0 5 20 13 18 17 9 8 3 1 11 16 15 7 4 18 17 4 12 5 4 18 1 3 2
s/part1/neut_parl_s05_0734.lab | 2 21 29 1 3 1 3 8 21 0 10 4 20 24 10 9 20 13 0 8 25 17 9 1 4 18 17 23 4 8 3 35 7 11 7 6 19 32 3 5 4 14 4 15 10 15 9 8 3 5 9 20 11 1 4 18 17 2
s/part1/neut_parl_s06_0018.lab | 2 3 16 28 24 10 11 6 7 25 13 5 12 1 3 1 2
s/part1/neut_parl_s05_0608.lab | 2 4 8 4 8 0 10 15 18 30 4 7 11 5 6 10 29 2
s/part1/neut_parl_s01_0016.lab | 2 3 8 15 1 6 32 5 6 7 3 5 11 7 6 1 28 3 2
s/part1/neut_parl_s05_0418.lab | 2 15 8 4 18 30 25 1 28 18 20 4 1 14 12 1 0 6 7 0 5 12 17 9 20 1 4 10 6 11 10 4 6 18 17 14 11 16 15 7 4 18 17 5 3 1 0 24 7 2
s/part1/neut_parl_s04_0113.lab | 2 4 8 3 1 13 1 12 1 22 5 6 5 3 8 11 1 12 19 0 3 1 13 26 6 0 23 25 27 24 14 21 18 5 2
s/part1/neut_parl_s05_0250.lab | 2 4 10 6 1 11 16 0 5 6 10 11 4 12 0 10 12 19 0 3 5 22 1 11 6 22 27 12 16 0 6 14 2
s/part1/neut_parl_s02_0319.lab | 2 3 5 12 0 7 9 29 1 21 28 10 4 5 9 8 15 1 13 18 10 0 9 20 0 1 28 3 2
s/part1/neut_parl_s01_0383.lab | 2 21 16 3 1 21 10 11 21 18 16 3 14 4 23 9 8 12 3 20 4 6 0 29 1 12 19 0 4 14 1 12 3 20 4 4 27 11 12 19 0 4 14 2
s/part1/neut_parl_s05_0266.lab | 2 7 3 1 12 1 0 10 26 4 7 11 8 3 8 31 27 14 13 5 25 10 11 6 7 25 5 14 13 16 0 31 1 13 7 22 1 4 18 17 0 10 4 6 1 11 17 25 7 0 15 5 2
```

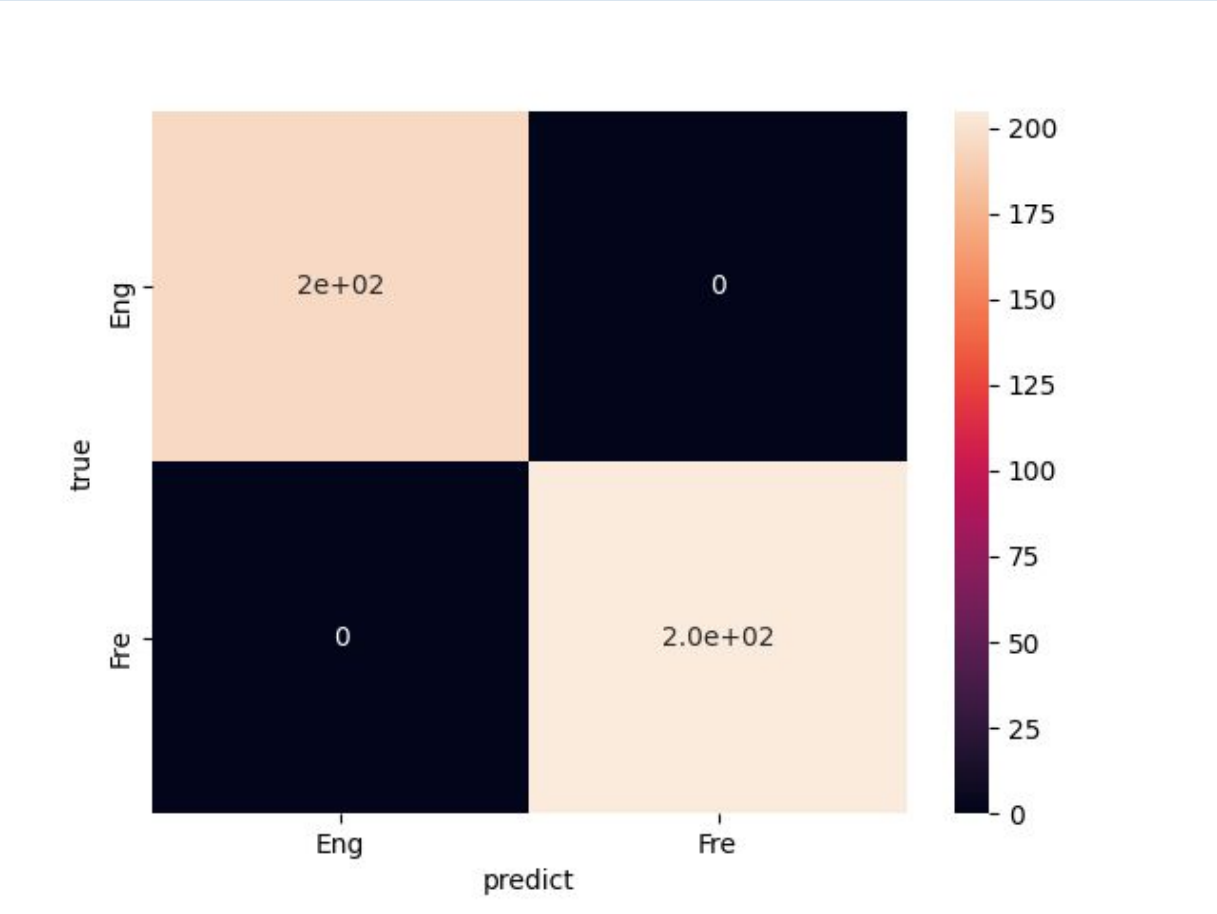
3 Language Classifier

1000 French & 1000 English Sentences
(1600 for training, 400 for testing)

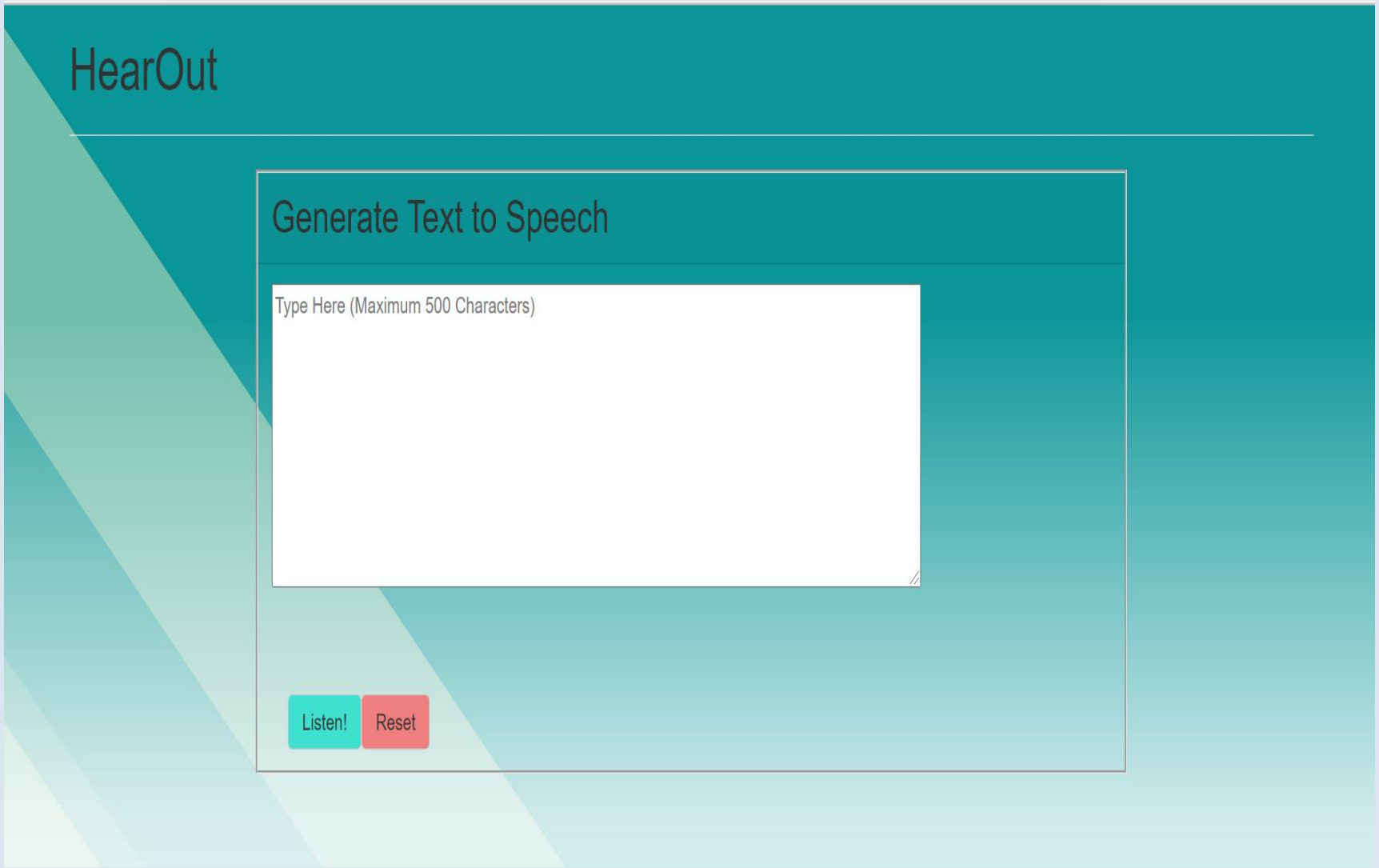
Distil-Bert

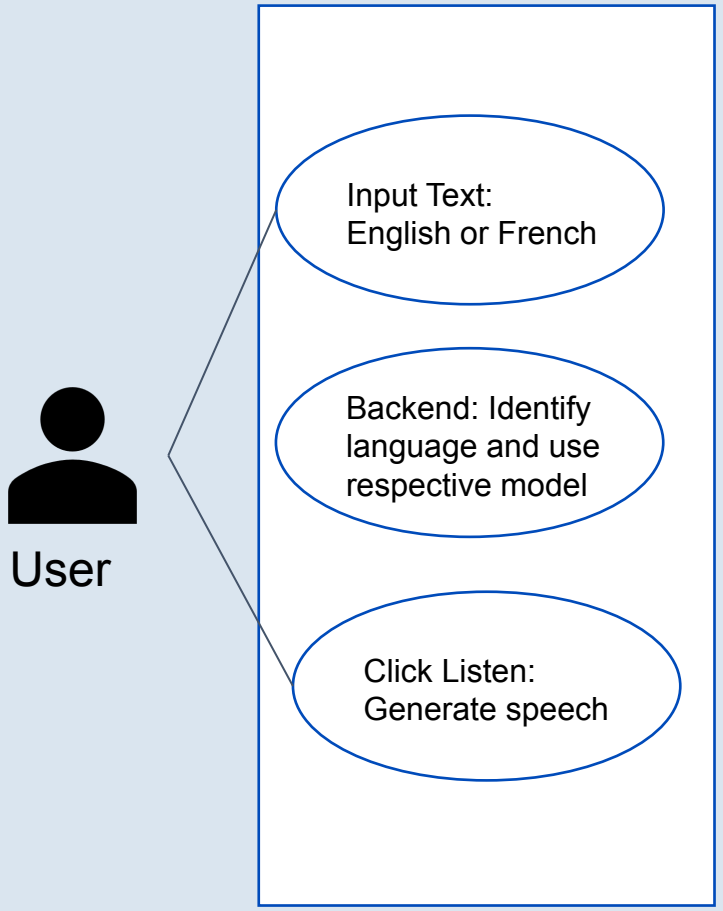
LSTM NN

3 Language Classifier



5 Web application





5 Containerization

```dockerfile
# FROM frolvlad/alpine-miniconda3
FROM continuumio/miniconda3

RUN apt update \
  && apt -y install iputils-ping

COPY ./frontend/ /frontend/

# copy conda, environment configuration files
COPY ./env/ /src/
# copy project source files
COPY ./src/ /src/
WORKDIR /src

SHELL ["/bin/bash", "--login", "-c"]

# recreate conda environment
RUN conda env create -f environment.yml \
  && rm environment.yml \
  && echo "conda activate tts-env" >> ~/.bashrc

# sanity-check: verify whether conda loaded libraries correctly
RUN echo "does conda work?" \
  && python -c "import flask"
```

```yaml
version: "3.1"

services:
  tts-api:
    image: tts-minicuda:latest
    ports:
    - 80:80
    networks:
      - tts-net

  tts-frontend:
    image: tts-minicuda:latest
    ports:
    - 81:81
    entrypoint: ["cat", "../frontend/file"]
    # npm run ...
    networks:
      - tts-net

networks:
  tts-net:
    driver: bridge
```

Timeline

4th Nov 2021

PreProcessing , Classifier

11th Nov 2021

Model Training

10th Dec 2021

Module Combining

14th Jan, 2022

Deploying modules on WebApp

14th Feb 2022

Document writing and Defense

Summary

1.Git Hub : WenjunSUN1997/M2_softwareproject_TTS: Software Project (github.com)

2.Meeting with Project Supervisor (Ajinkya Kulkarni) was conducted on 25th October,2021.

Thank you