Software Development Project

Multilingual: Text to Speech
Presentation 6

Albert Millert
Shalini Priya
Wenjun Sun
Soklay Heng

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

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

Last time:

1. Language classifier + prediction
2. Server + (ready) modules integration
3. Server + frontend integration
4. English TTS

Completed

- Included EnglishTTS model in the environment
- Evaluation
- Server + (ready) modules integration
- Server + frontend integration
- French model Training

Backend

- vocalization function provided to the server

```python
def say(sent: str):
    fout_path = os.path.abspath('./Grad-TTS/out/sample.wav')

    with torch.no_grad():
        x = torch.LongTensor(intersperse(text_to_sequence(sent, dictionary=cmu), len(symbols)))[None]
        x_lengths = torch.LongTensor([x.shape[-1]])
        _, y_dec, _ = generator.forward(x, x_lengths, n_timesteps=TIMESTEPS, temperature=1.5, stoc=False, spk=SPEAKER, length_scale=0.91)

        audio = (vocoder.forward(y_dec).cpu().squeeze().clamp(-1, 1).numpy() * 32768).astype(np.int16)

        write(fout_path, 22050, audio)

    return fout_path
```

Backend

- vocalization function provided to the server
- server handles the output and serves audio file as a response

```python
from flask import Flask, request, jsonify, send_file

@app.route("/synthesize", methods=["POST"])
@cross_origin(origin="*", headers=["Content-Type", "Authorization"])
def synthesize():
    if request.method == "POST":
        content = request.json
        sentence = content["sentence"]

        print(f"Processing: {sentence} in progress...")
        lang = classify(sentence)
        print(f"Language: {lang} detected", end="\n\n")
        print(f"Let's synthesize")

        if lang == "EN":
            out_path = say(sentence)
            playsound(out_path)
        else:
            default_response = "Sorry, you have to wait for the french model; only english one available"
            out_path = say(default_response)
            playsound(out_path)

        return send_file(out_path, mimetype="audio/wav", as_attachment=True, attachment_filename="sample.wav")
    else:
        return jsonify({"speech": "Can't touch this"})
```

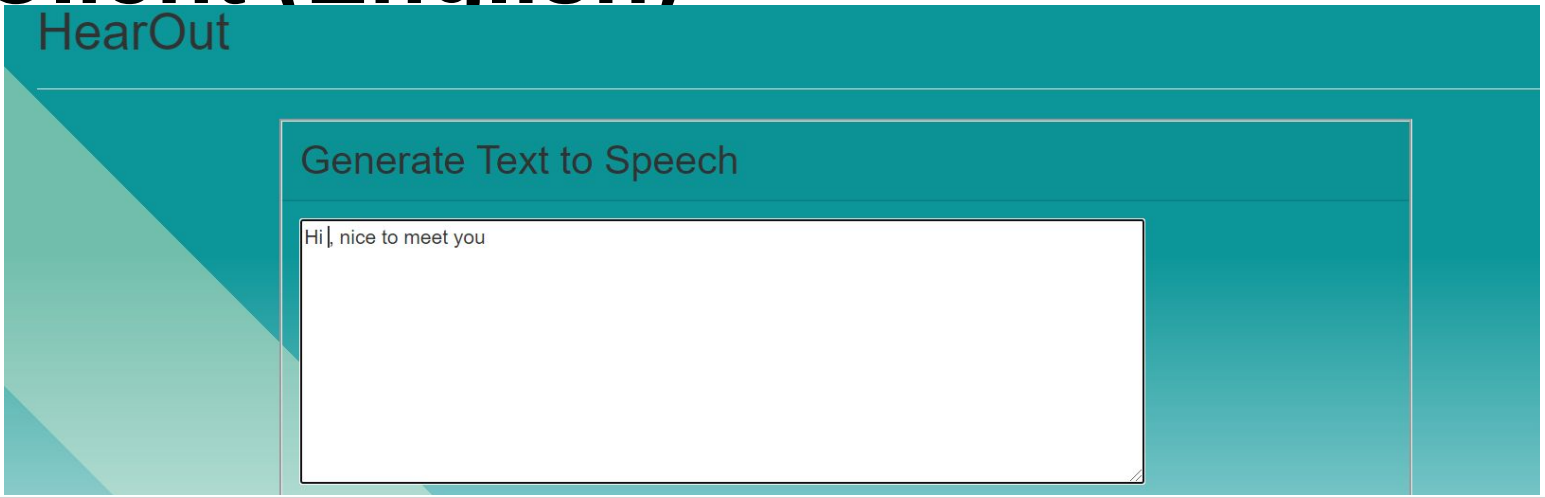
Backend

- vocalization function provided to the server
- server handles the output and serves audio file as a response
- Server dockerized
  - files copied, environment set up, and server spinned up at the end

```
FROM continuumio/miniconda3

RUN apt-get -y update && apt-get install -y libzbar-dev

WORKDIR /src

# copy conda, environment configuration files
COPY ./env/ .

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

# recreate conda environment
RUN conda env create -f environment.yml \
  && conda init bash \
  && conda activate tts-env

# copy language model
COPY ./classifierModel .

# copy project source files
COPY ./src/ .

RUN conda activate tts-env \
  && cd Grad-TTS/model/monotonic_align \
  && python setup.py build_ext --inplace \
  && cd ../../..

EXPOSE 5000

CMD conda activate tts-env \
  && ./server.py
```

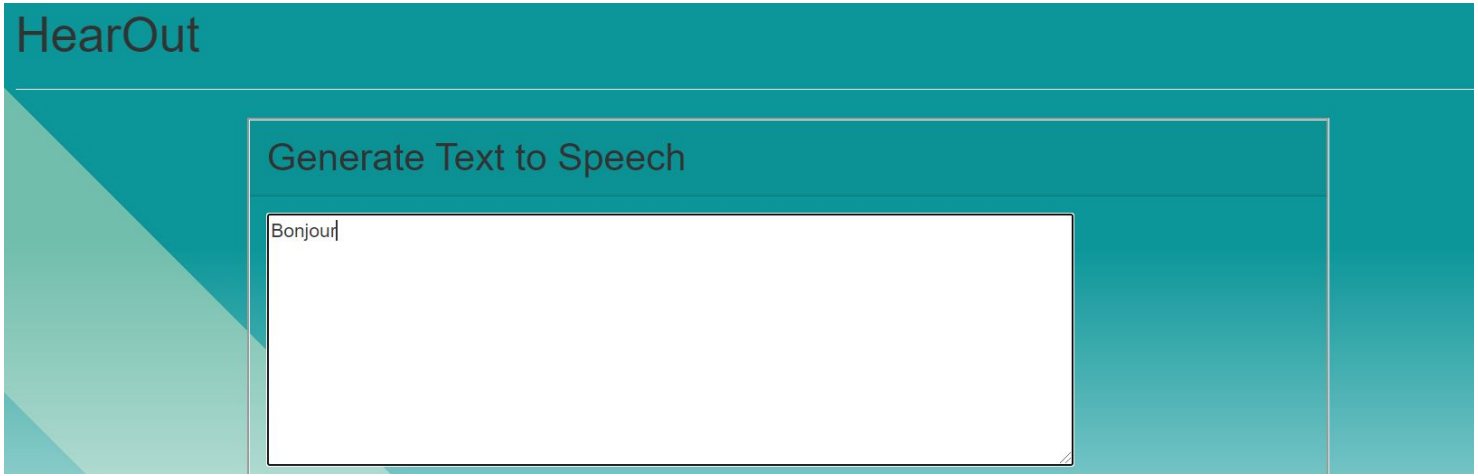
Backend Demo

Client (English)




```
Let's synthesize
127.0.0.1 - - [13/Jan/2022 08:52:20] "POST /synthesize HTTP/1.1" 200 -
127.0.0.1 - - [13/Jan/2022 08:52:28] "OPTIONS /synthesize HTTP/1.1" 200 -
Processing: Hi , nice to meet you in progress...
Language: EN detected
```

Client (French)




```
Let's synthesize
127.0.0.1 - - [13/Jan/2022 08:54:04] "POST /synthesize HTTP/1.1" 200 -
127.0.0.1 - - [13/Jan/2022 08:54:32] "OPTIONS /synthesize HTTP/1.1" 200 -
Processing: Bonjour in progress...
Language: FR detected
```

Client (Demo)

[redacted]

Evaluation

科大讯飞 iFLYTEK

1. iFLYTEK speech evaluation platform
2. China's largest intelligent voice technology provider
3. Used by multiple schools in China as a scoring platform for spoken English exams

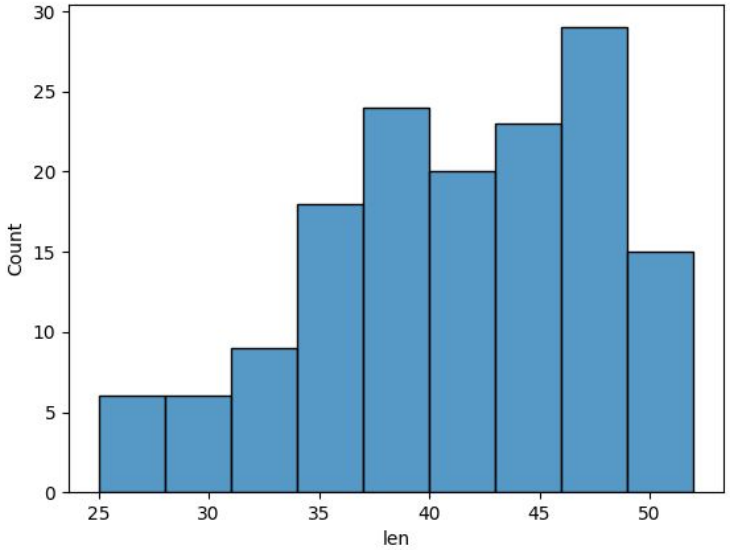
Evaluation Dimension

1. accuracy_score
2. fluency_score
3. Integrity_score
4. Standard_score
5. Total_score：
   (0.5 * a_score + 0.3 * f_score + 0.2 * s_score) * i_score / 100

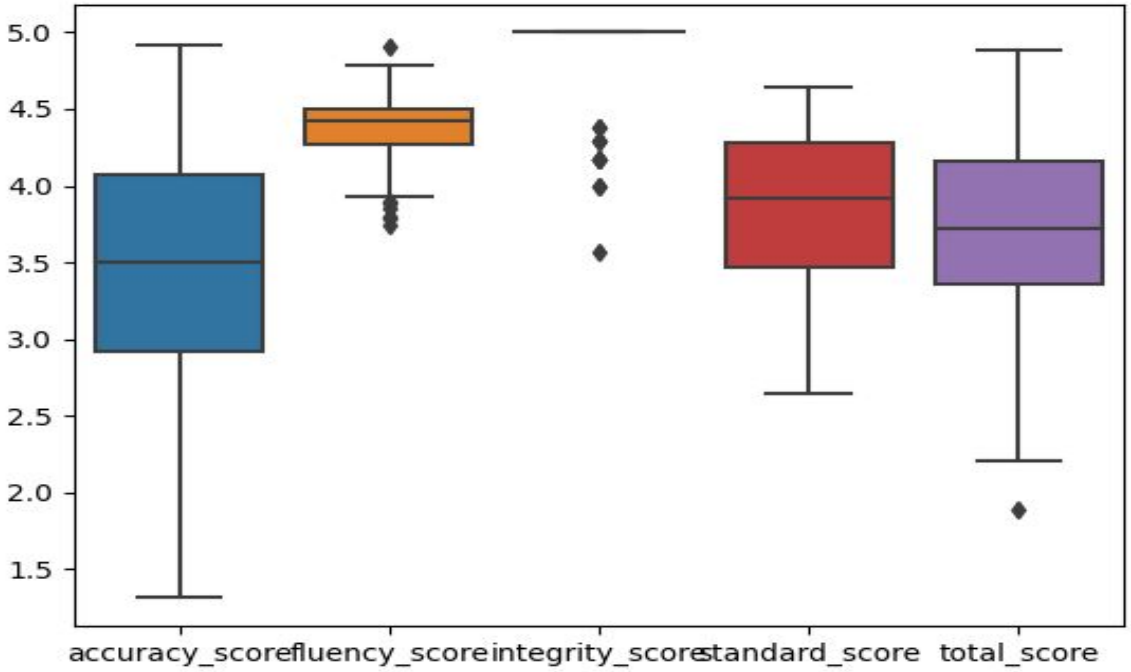
Evaluation Data

1. Corpous: BBC Headlines

2. Amount: 150



Evaluation Result



Next Time

- Include French model in the environment
- Evaluation
- Server + (ready) modules integration
- Server + frontend integration
- Defense

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

Thank you