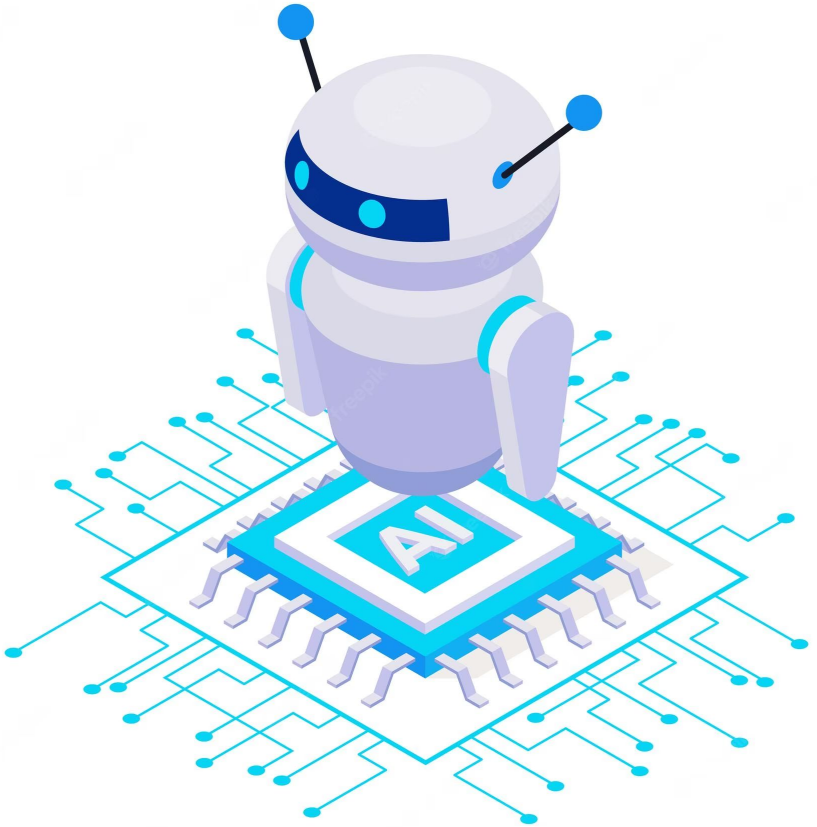
AI-Powered English Speaking Tutor

Group 1



SYDE660 2023Spring
Xinguang Jiang, Yusen Jiao, Weixizheng Wan, Peishan Cao

UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

Problem Space

Issues with traditional language learning methods and online learning tools

1. Interactivity

2. Individual customization

3. Comprehensive grasp of language nuances

4. Cost $

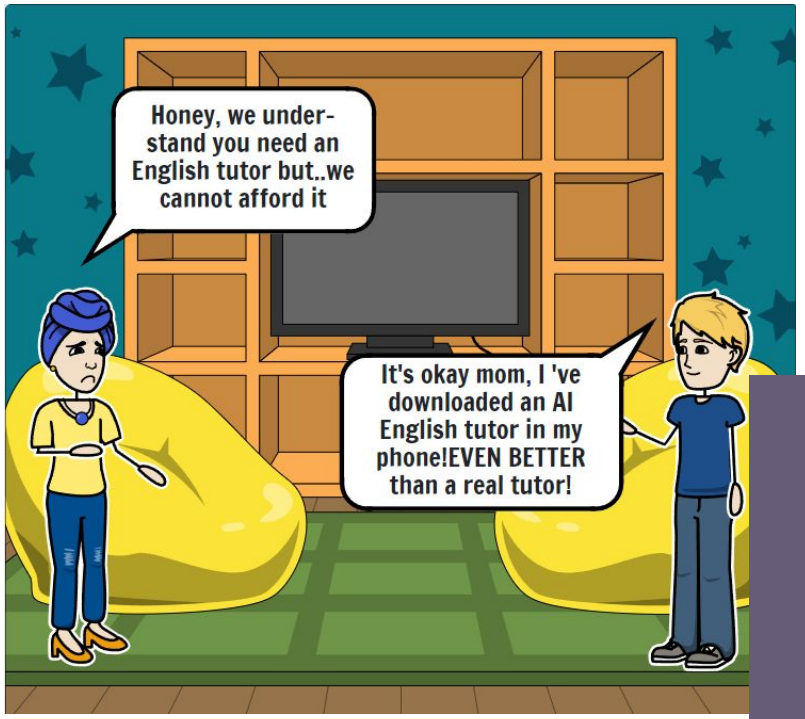
Project Objectives

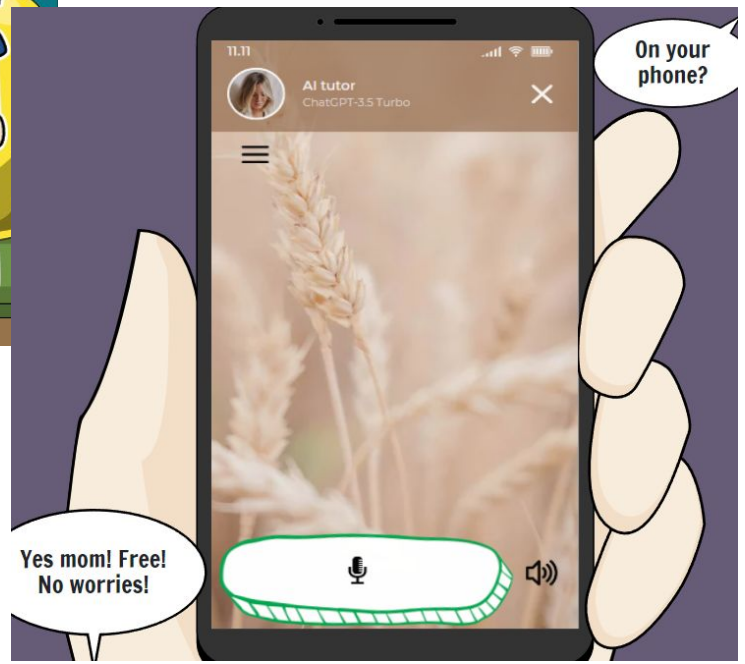
1. Smooth Interactive Learning

2. Personalized User Experience

3. Emotional Intelligence

4. User interface





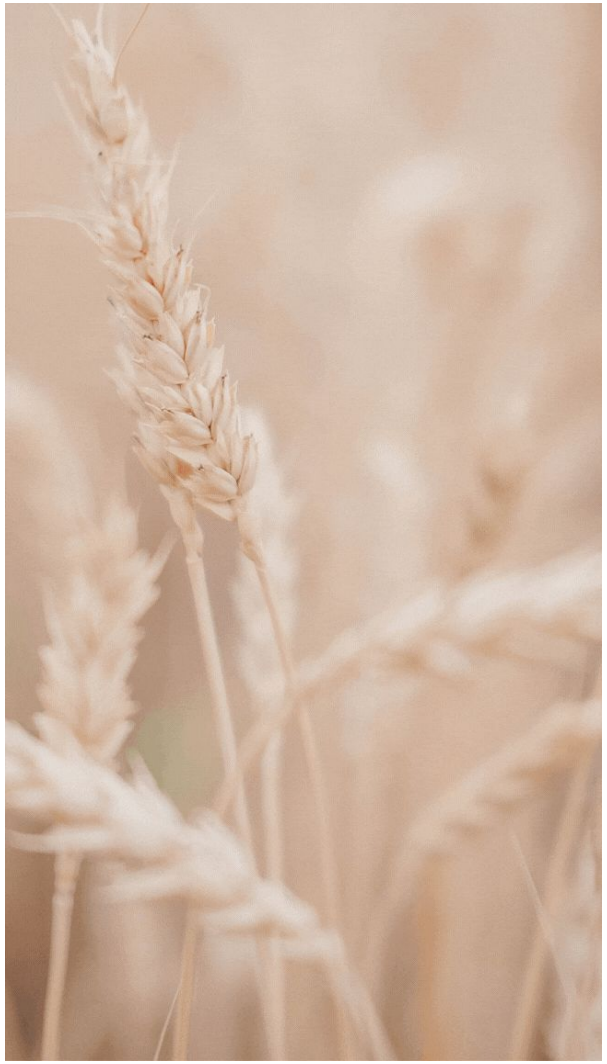
Project Objectives

1. Smooth Interactive Learning

2. Personalized User Experience

3. Emotional Intelligence

4. User interface



Project Structure

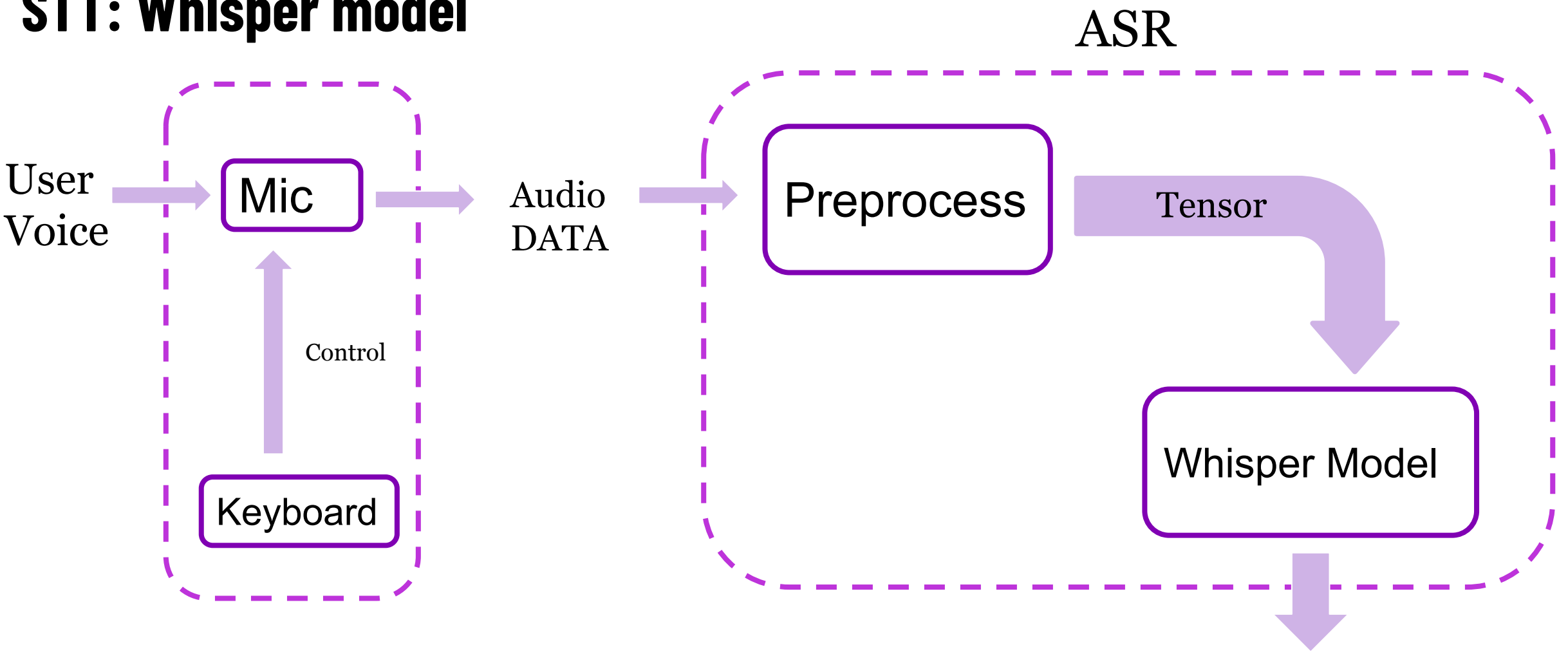
1. Whisper, an Automatic Speech Recognition (ASR) system

2. ChatGPT 3.5 Turbo model

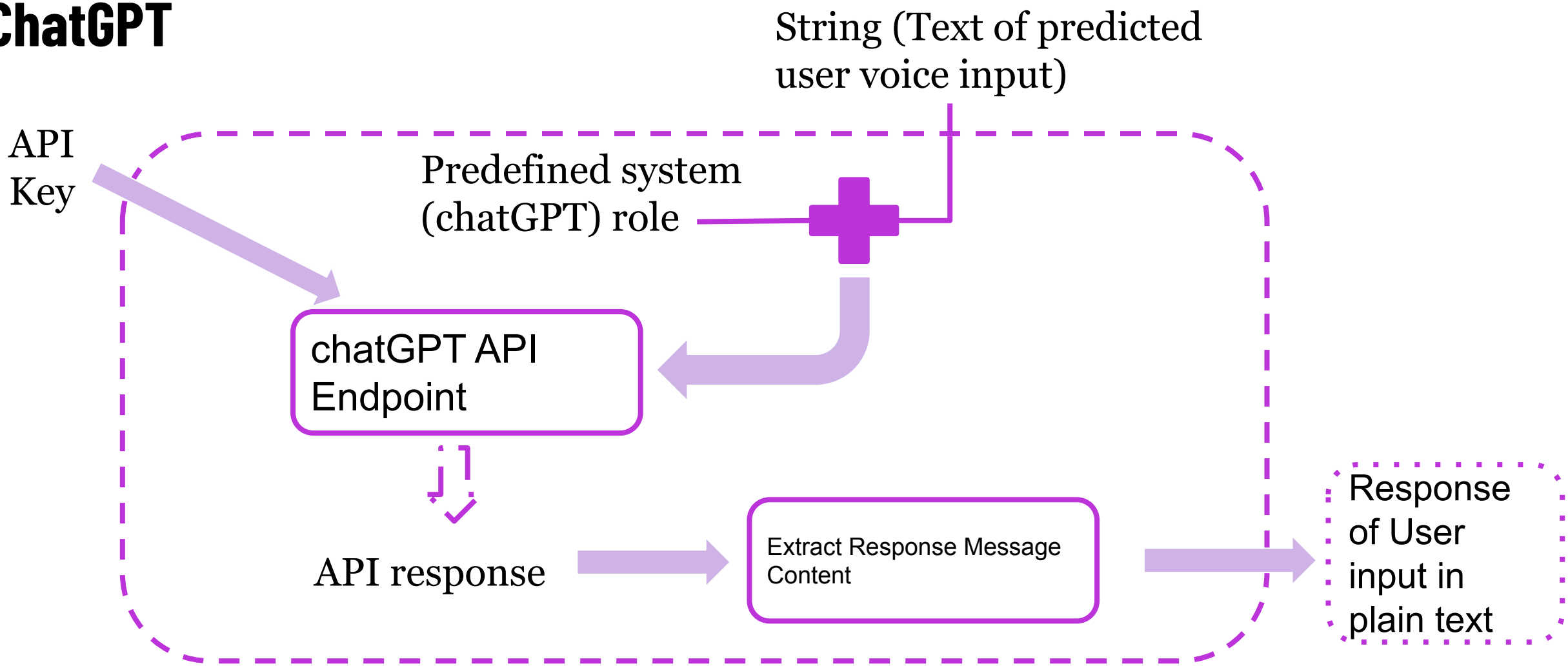
3. VITS text-to-speech system

(More details next page…)

STT: Whisper model



ChatGPT


String (Text of predicted user voice input)
API Key
Predefined system (chatGPT) role
chatGPT API Endpoint
API response
Extract Response Message Content
Response of User input in plain text

VITS Data collection

1) VITS (Variational Inference Transformer for Audio Source Separation)

2) 13,100 short audio clips, spoken by a single speaker, with a total duration of about 24 hours.

3)The audio format is 16-bit PCM with a sampling rate of 22 kHz.

4)training set (12,500 samples), a validation set (100 samples) and a test set (500 samples)

5)Each file list contains the path to the audio file and the corresponding text,

```
DUMMY1/LJ049-0022.wav|The Secret Service believed that it was very doubtful that any Presi
DUMMY1/LJ033-0042.wav|Between the hours of eight and nine p.m. they were occupied with the
DUMMY1/LJ016-0117.wav|The prisoner had nothing to deal with but wooden panels, and by dint
DUMMY1/LJ025-0157.wav|Under these circumstances, unnatural as they are, with proper manage
DUMMY1/LJ042-0219.wav|Oswald demonstrated his thinking in connection with his return to th
DUMMY1/LJ032-0164.wav|it is not possible to state with scientific certainty that a particu
DUMMY1/LJ046-0092.wav|has confidence in the dedicated Secret Service men who are ready to
DUMMY1/LJ050-0118.wav|Since these agencies are already obliged constantly to evaluate the
DUMMY1/LJ043-0016.wav|Jeanne De Mohrenschildt said, quote,
DUMMY1/LJ021-0078.wav|no economic panacea, which could simply revive over-night the heavy
DUMMY1/LJ039-0148.wav|Examination of the cartridge cases found on the sixth floor of the D
DUMMY1/LJ047-0202.wav|testified that the information available to the Federal Government a
DUMMY1/LJ023-0056.wav|It is an easy document to understand when you remember that it was o
DUMMY1/LJ021-0025.wav|And in many directions, the intervention of that organized control w
DUMMY1/LJ030-0105.wav|Communications in the motorcade.
DUMMY1/LJ021-0012.wav|with respect to industry and business, but nearly all are agreed tha
DUMMY1/LJ019-0169.wav|and one or two men were allowed to mend clothes and make shoes. The
DUMMY1/LJ039-0088.wav|It just is an aid in seeing in the fact that you only have the one e
DUMMY1/LJ016-0192.wav|"I think I could do that sort of job," said Calcraft, on the spur of
DUMMY1/LJ014-0142.wav|was strewn in front of the dock, and sprinkled it towards the bench
DUMMY1/LJ012-0015.wav|Weedon and Lecasser to twelve and six months respectively in Coldbat
DUMMY1/LJ048-0033.wav|Prior to November twenty-two, nineteen sixty-three
```

VITS Pre-processing

Text Preprocessing

The cleaning process includes removing irrelevant characters, replacing abbreviations and converting numbers.

```python
import argparse
import text
from utils import load_filepaths_and_text

if __name__ == '__main__':
  parser = argparse.ArgumentParser()
  parser.add_argument("--out_extension", default="cleaned")
  parser.add_argument("--text_index", default=1, type=int)
  parser.add_argument("--filelists", nargs="+", default=["filelists/ljs_audio_text_val_filelist.txt",
  parser.add_argument("--text_cleaners", nargs="+", default=["english_cleaners2"])

  args = parser.parse_args()


  for filelist in args.filelists:
    print("START:", filelist)
    filepaths_and_text = load_filepaths_and_text(filelist)
    for i in range(len(filepaths_and_text)):
      original_text = filepaths_and_text[i][args.text_index]
      cleaned_text = text._clean_text(original_text, args.text_cleaners)
      filepaths_and_text[i][args.text_index] = cleaned_text

    new_filelist = filelist + "." + args.out_extension
    with open(new_filelist, "w", encoding="utf-8") as f:
      f.writelines(["|".join(x) + "\n" for x in filepaths_and_text])
```

Audio pre-processing

Audio processing involves steps such as resampling, extracting audio features (e.g., Mel spectrograms), and normalization. We use the Short Time Fourier Transform (STFT)

The audio data is normalized to the range [-1, 1]

VITS Training

Learning Rate Decay:
Improves convergence and stability

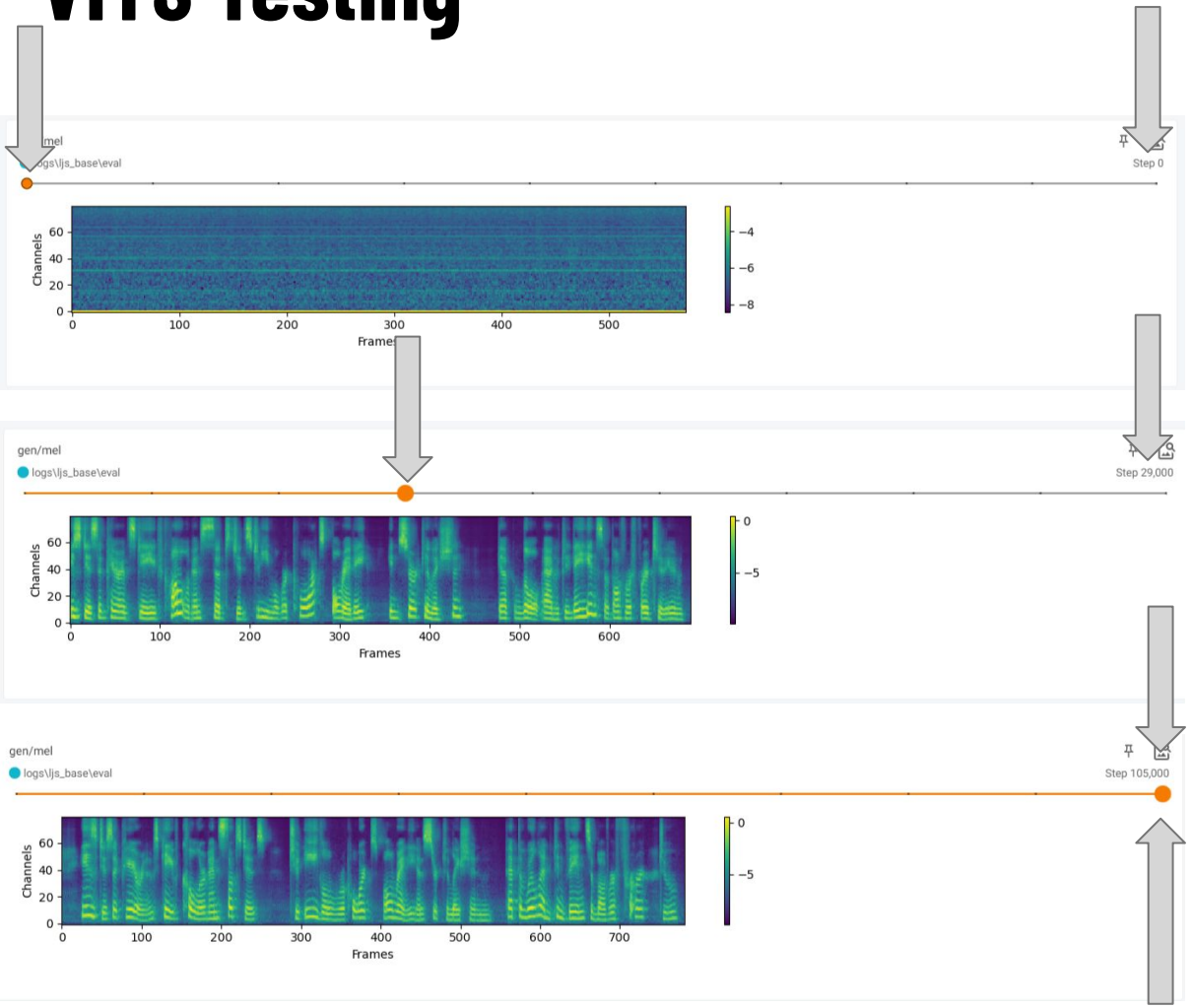
Windowed Generator Training:
Reduces time and memory consumption without sacrificing quality

Mixed Precision Training:
Low precision representation (half-precision floating point): Increases training speed and reduces memory footprint.

Parameters chosen based on computer performance:
Batch size set to 4

Training Results:
48 hours of continuous training, got an usable VITS model

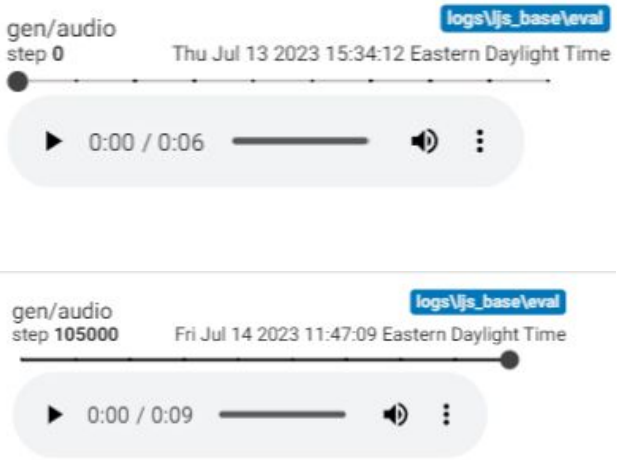
VITS Testing



An intuitive dashboard using TensorBoard:

As Step increases
=> Frame Becomes more detailed
=> Model learns more detailed information
=> Quality of the audio is improving

VITS Testing




```python
hps = utils.get_hparams_from_file("./configs/ljs_base.json")
```

Python

```python
net_g = SynthesizerTrn(
    len(symbols),
    hps.data.filter_length // 2 + 1,
    hps.train.segment_size // hps.data.hop_length,
    **hps.model).cuda()
_ = net_g.eval()

_ = utils.load_checkpoint("./logs/ljs_base/G_105000.pth", net_g, None)
```

Python

```python
stn_tst = get_text("As an English teacher, I don't have a favorite pet. However, some popular choices for pets include dogs, cats, birds, and fish. Each pet has its own unique qualities and can bring joy and co
with torch.no_grad():
    x_tst = stn_tst.cuda().unsqueeze(0)
    x_tst_lengths = torch.LongTensor([stn_tst.size(0)]).cuda()
    audio = net_g.infer(x_tst, x_tst_lengths, noise_scale=.667, noise_scale_w=0.8, length_scale=1)[0][0,0].data.cpu().float().numpy()
ipd.display(ipd.Audio(audio, rate=hps.data.sampling_rate, normalize=False))
```

Python

0:00 / 0:14

```python
import soundfile as sf
# save the audio file
output_file = "generated_audio.wav"
sf.write(output_file, audio, hps.data.sampling_rate)

print("Save Successfully:", output_file)
```

Demo Speech:

Full of electronic **noises**
to
Clear **human voice**

Inference File:

Successfully transfer the **given sentences**
to
Audio file and able to save it locally

VITS Limitation

Hardware and Time!

NVIDIA RTX A6000 Graphics Card

Performance Amplified

Unlock the next generation of revolutionary designs, scientific breakthroughs, and immersive entertainment with the NVIDIA RTX™ A6000, the world's most powerful visual computing GPU for desktop workstations. With cutting-edge performance and features, the RTX A6000 lets you work at the speed of inspiration—to tackle the urgent needs of today and meet the rapidly evolving, compute-intensive tasks of tomorrow.

GPU Features	NVIDIA RTX™ A6000
GPU Memory	48 GB GDDR6 with error-correcting code (ECC)

Engineering Analysis:

Risk or failure mode analysis: prevent chatGPT from generating inappropriate or irrelevant responses.

=>prefabricated prompt

Reliability analysis: assess the reliability of the STT and TTS models by repeatedly testing the models under various conditions.

=>train/prompt with example

Economic analysis: We need to assess the total cost of the system.

=>not exceed $1

Modeling

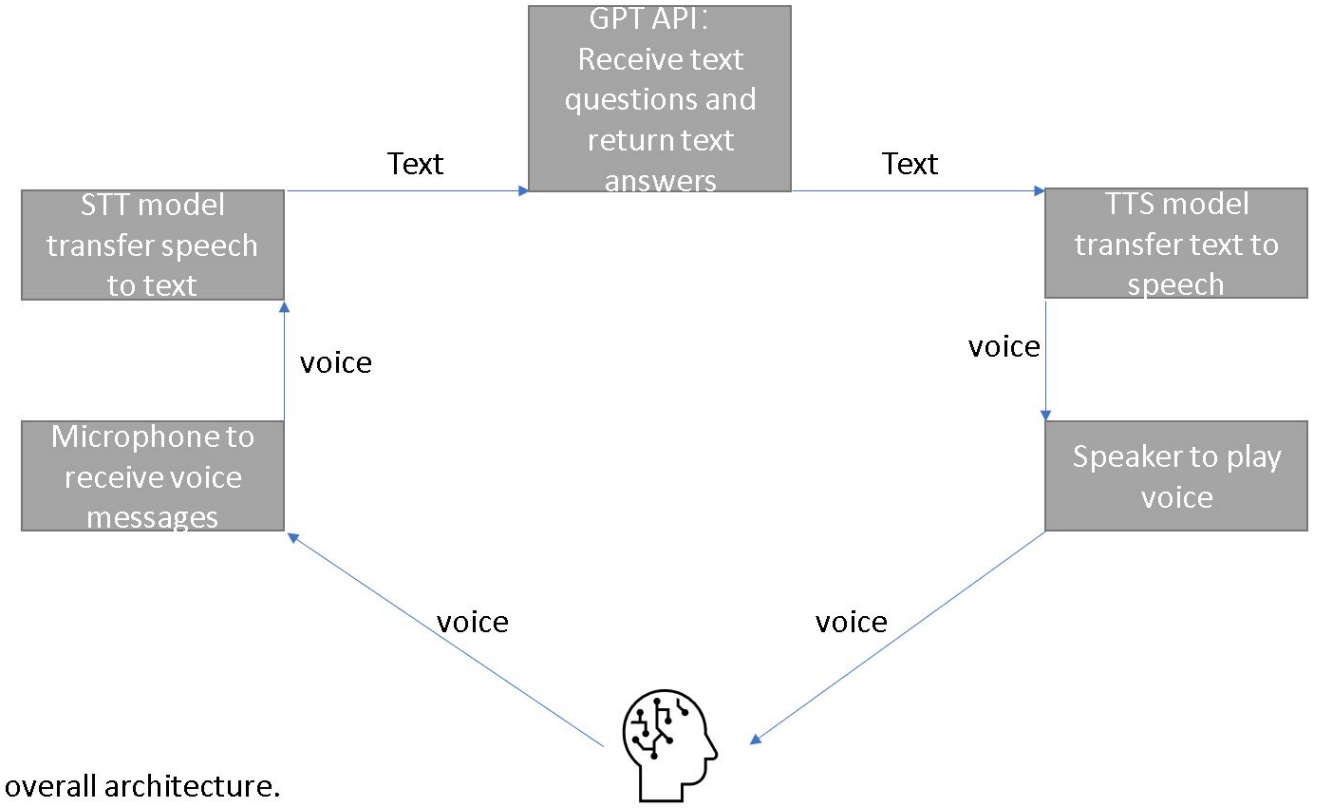
STT : Whisper

TTS : VITS

GPT : black-box model

input (the user's text input)

output (the generated text responses)



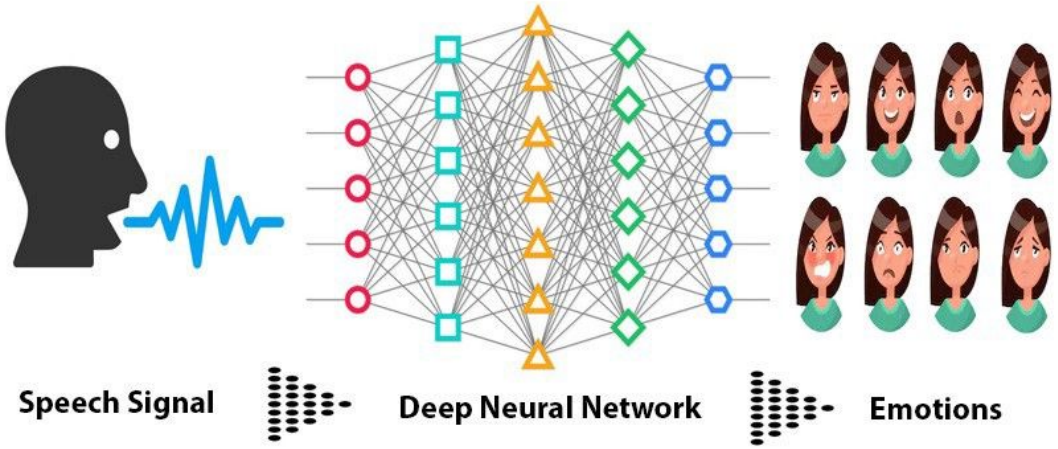
Model overall architecture.

Model data flows.

Future work

Emotion recognition

- adapt and respond with empathy



https://github.com/SuyashMore/MevonAI-Speech-Emotion-Recognition

Alternative models

- e.g. PI, Deep personalization and interaction

> Hey there, great to meet you. I'm Pi, your personal AI.
>
> My goal is to be useful, friendly and fun. Ask me for advice, for answers, or let's talk about whatever's on your mind.
>
> How's your day going?

https://pi.ai/talk

UNIVERSITY OF WATERLOO

FACULTY OF ENGINEERING

YOU+WATERLOO

Our greatest impact happens together.

References

[1]Y. Zhang et al., “BigSSL: Exploring the Frontier of Large-Scale Semi-Supervised Learning for Automatic Speech Recognition,” IEEE journal of selected topics in signal processing, vol. 16, no. 6, pp. 1519–1532, 2022, doi: 10.1109/JSTSP.2022.3182537.

[2]W. Chan, D. Park, C. Lee, Y. Zhang, Q. Le, and M. Norouzi, “SpeechStew: Simply Mix All Available Speech Recognition Data to Train One Large Neural Network,” arXiv.org, 2021.

[3]D. Galvez et al., “The People’s Speech: A Large-Scale Diverse English Speech Recognition Dataset for Commercial Usage,” 2021, doi: 10.48550/arxiv.2111.09344.

[4]J. Kim, J. Kong, and J. Son, “Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech,” 2021, doi: 10.48550/arxiv.2106.06103.

[5]“The LJ Speech Dataset,” keithito.com. https://keithito.com/LJ-Speech-Dataset/

[6]I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” 2017, doi: 10.48550/arxiv.1711.05101.

[7]“Pricing,” openai.com. https://openai.com/pricing

[8]A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision,” 2022, doi: 10.48550/arxiv.2212.04356.