CS4347/5647 Course Project

Please note that this document is subject to further revisions. Kindly stay updated by checking Canvas announcements in your mailbox regularly.

1. Overview

The purpose of this project is to put together all the knowledge you have gained from CS4347/5647 to solve a real-world problem. You will be able to use your skills in audio signal processing, feature extraction, and machine learning.

You will work in a team of a **maximum of 4** members. You can choose one topic from Section 5. Or, you can propose your own topic, but in this case, you need to submit your proposal by the end of Week 6 for review.

2. Components

In your project, you need to

1. Determine your topic (see Section 5 for details). If you want to propose your own project topic, you need to write a proposal to define your project topic and scope.
2. Develop a working system that solves a problem from your chosen topic.
3. Develop a user interface for your system that clearly and intuitively visualizes the system input and output.
4. At Week 10, demonstrate your progress with a ppt presentation.
5. Write a report in the style of research papers.
6. After you finish developing your system, record a video including both a ppt presentation and your system showcase.
7. Make your code well-documented.

The project schedule is as follows:

(a) **(Optional) Project proposal (non-default):** end of Week 6 (**24 September 2023**)
(b) **Start of Project**: Week 7
(c) **Mid-Project Presentation**: Week 10 tutorial sessions. Exact time slots for each group will be scheduled later.
(d) **System Demonstration & Presentation, Documented Code, and Project Report Submission**: Submissions of presentation (video recording), code, and final report are due on **24 November 2023** (Friday of the Reading Week). Please zip all necessary files including codes and report, rename them by Project_Team_(team ID).zip, and submit it to Canvas by 24 Nov 2023, 2359HRS. If you wish to upload your video recording to a separate hosting system (e.g. YouTube), please include the link clearly in the zip file and ensure that the link is accessible.

3. Teaming

You are asked to form a 1~2 people sub-team before the end of Week 4. After that, we will group different sub-teams into full project teams to ensure skill diversity of team members.

- ➢ Week 3~4: Students form 1~2 people sub-team, and then fill out a survey about the mastery level of different skills, in the unit of sub-teams.
- ➢ Week 5: We will combine sub-teams to form final project teams of 3~4 people, in the way that maximizes the skill diversity within each team.
- ➢ Week 6: Finalize the project teams.

If you have already formed a complete project team, please indicate this in the survey.

4. Evaluation

This project makes up 45% of your grade for the class. Your work will be evaluated based on five aspects:

1. **Mid-Project Presentation [5 points]**: A progress checkpoint where you need to present your findings so far and propose your future work. You will need to give a 10-minute presentation showing the knowledge you found in the literature and your plan in the following four weeks.

2. **Project Report [20 points]**: You are required to submit a report, detailing your approach and findings during the project. Detailed requirements are:
   - ➢ Follow the ACM SIG Proceedings Template. Length: 4~6 pages (strict limit, NOT including references and appendices).
   - ➢ Follow the research paper structure, showing problem definition, review of the literature, your approach, the results and analysis, and conclusion.
   - ➢ Document your approach in detail.
   - ➢ Well-justify the reason for choosing your approaches.
   - ➢ Show the novelty in your approach.
   - ➢ State the contribution of each of the team members in detail at the end of your report.

3. **System Demonstration & Presentation [10 points]**: You need to prepare a video (**strictly less than 10 minutes**) that includes a presentation of your research and development, and a live demo of your system. When presenting, each group member is required to present the part he/she is responsible for.

4. **Documented Code Submission [10 points]**: You will need to submit the source code of your project. Note that:
   - ➢ Write a readme file that detailly instruct how to perform training / inference.
   - ➢ Make your code well-documented.

Each team member will receive the same project score and hence members in the same group are expected to have the same level of workload. If you request to grade each person by contribution in the report, please indicate each members' contribution at the last section of the report (after conclusion).

5. Topics

The topics listed in this section serve as **default project topics**. The selected problem is being actively researched, and there are a lot of resources and papers that you can access. Or, if you have some more interesting ideas, you can instead come up with your own project proposal (no format requirement, one page limit) and submit it by **Sunday, 24 September 2023 (end of Week 6)** for review.

5.1 Tone Evaluation

Foreword: You are recommended to study Mandarin Chinese, as this language has significantly stronger established literature and existing resources as compared to other tonal languages, and the study of other languages for this project may lead to unforeseen difficulties in evaluation.

The system

Tonal languages make use of changes in pitch, known as "tones", to express meaning. Here, we are trying to make machines classify/detect tones in speech. A tone evaluation system takes raw audio as input, such as an audio recording of a person's speech, and identifies the tone(s) used in that audio. Correctly identifying tone is vital to speech transcription, as tonal languages depend on both phoneme as well as tone to convey meaning. Despite this, tone detection is still an ongoing challenge, and over the last few decades a range of techniques have been attempted to tackle this problem.

The core task in tone evaluation or tone detection is the correct classification of tone for individual syllables. For the input of your tone evaluation system, you may use individual words, or longer utterances or sentences, keeping in mind that the rigor of your project will affect the final evaluation.

Naturally, tone detection for longer utterances is more difficult than tone classification for single words. If the raw audio includes multiple words, then in addition to tone detection, there is the challenge of correctly identifying onset and offset timings that are used to demarcate where each word begins or ends. Furthermore, it is known that for languages such as Mandarin or Cantonese, the exact pronunciation of each word may vary depending on adjacent words or other words in the sentence (Zhang, 2005). Therefore, using utterances or sentences as input may lead to other considerations, as compared to using single words. For this project, we consider tone evaluation for individual words to be the baseline task. Classification for words within a sentence will be an interesting further development if you feel confident to take on the challenge.

As described above, the general objective of "tone evaluation" may comprise a range of tasks. The choice of tasks tackled will affect how you score the system. For example:

- For the main task of annotating individual words with tone labels, you should choose an appropriate metric for accuracy.
- If your input audio contains multiple words, then you should also include an appropriate metric for onset/offset times, and also consider how this affects the accuracy metric for tone labels.

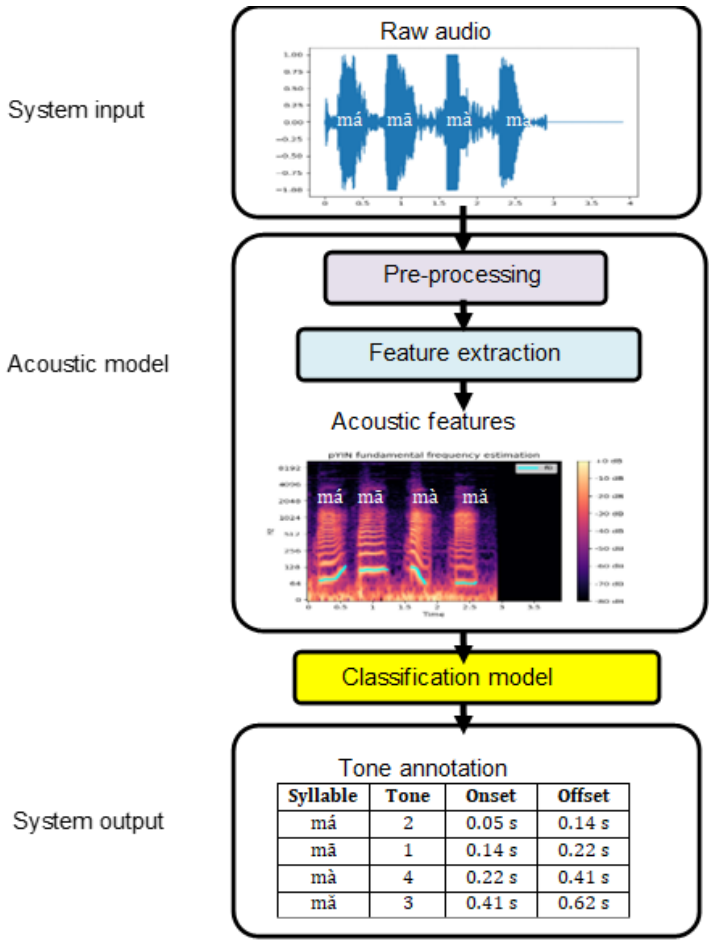
There are a range of metrics in the broader literature that you may choose from.

One important consideration for your system would be the choice of pre-processing and feature extraction. There are various *acoustic models* that obtain appropriate features from the raw data using statistical, heuristic, or machine learning methods. For example:

- There are a range of heuristic methods for pitch tracking (F_0 estimation), such as YIN (De Cheveigné, 2002) or pYIN (Mauch, 2014).
- More recent approaches make use of deep learning to perform pitch tracking, such as CREPE (Kim, 2018), which uses Convolutional Neural Networks.
- Once pitch tracking is done, whether using heuristic or deep learning methods, these features may be used in statistical algorithms such as (Wu, 1991) or deep learning classification methods.
- Toolboxes such as Kaldi (Polvey, 2011) makes use of a range of heuristic or deep learning models.

The choice of model may be affected by your choice of classification method. Researching the literature on these models will be an important part of your project.

The figure below provides an example of a tone evaluation system for your reference:



Syllable	Tone	Onset	Offset
má	2	0.05 s	0.14 s
mā	1	0.14 s	0.22 s
mà	4	0.22 s	0.41 s
mă	3	0.41 s	0.62 s

Fig 1. An example structure of a tone evaluation system. This algorithm was applied to a recording of a non-native Chinese speaker, with the syllable "mă" not pronounced well. Discrepancies in pronunciation are an ongoing challenge in the domain of tonal evaluation.

Training Datasets

The following are some datasets you may wish to consider using. All of these datasets include raw audio wav files with annotations.

- Dataset of individual Mandarin words:
    - Tone Perfect
      (*you may have to request for access to this dataset, following the instructions on the website*)
- Dataset of Mandarin sentences:
    - Aidatatang_200zh
    - aishell dataset

Please note that these datasets are rather large. You are recommended to consider the time and processing power constraints, and begin with a suitable subset of either of these datasets, and only moving on to increasing the size of the training set subsequently.

For examples of recent papers using these datasets for Speech Recognition, you may wish to see the following resources: (Lu, 2022; Yang, 2020; Liu, 2022)

Resources

The following literature may be useful:

- **Background information on tonal languages (**Whalen, 1992**)**
- **A method considering the influence of context tones (**Zhang, 2005**)**
- **A method without pitch tracking (**Ryant, 2014**)**

5.2 Singing Voice Transcription

Singing transcription system

A singing transcription system aims to transcribe a performance recording into two forms of output: the lyrics of the song, and the melody of the song. You can use monophonic or polyphonic (will be more difficult) singing recording as the input of your system. In addition to the audio signal, you may use video recording and/or a set of eSense earbuds with motion tracking. The corresponding video recording and IMU data from the eSense device may help to improve the transcription quality. For output, you can transcribe both lyrics and melody, or just one of them. Please note that a focused scope is acceptable, provided that it is explored in depth.

Solving the singing voice transcription problem has significant and meaningful applications. It can facilitate search queries via singing or humming applications for users, which is much more convenient than searching for music by keywords. In addition, in music education, converting a recording to symbolic notation is of great help for performance analysis.

Here is a sample transcription system structure for your reference:

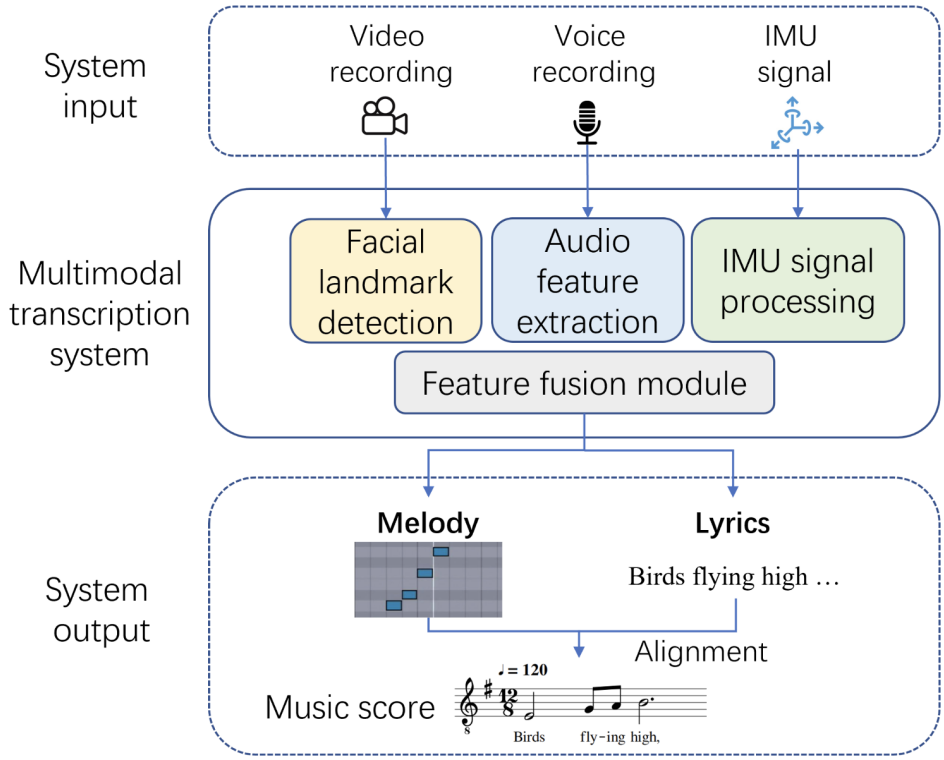


Fig 2. An example structure of a multimodal transcription system

For students who wish to build on existing work done by the NUS Sound and Music Computing Lab, the current state-of-the-art research combines audio with video and IMU data to improve lyric and melody transcription in situations with low signal-to-noise ratios (SNR). The current dataset only considers video data that directly focuses on the lips of the singer. Some possible further applications, which may potentially be able to lead to a publication, could include improving the robustness of the system, such as investigating:

- Other formats of video, such as where the lips are not facing the camera directly
- Other ways to integrate IMU data or other sources of data
- Other ways to enable the system to transcribe musics or lyrics in non-ideal conditions

Training Datasets

For melody transcription systems: MIR-ST500.
A dataset containing singing recording with accompaniment, and corresponding note-level transcription.

For lyric transcription systems: DSing.
A dataset containing utterance-level singing recordings and corresponding text transcriptions.

For multimodal lyric systems: **N20EM**
This is a multimodal singing recording dataset collected in CS4347 of recent semesters. When participants are singing, their facial movement is recorded by a video camera; a single eSense earbud provides the IMU signal to reveal the head movement; also, the singing voice is recorded by a professional condenser microphone. Utterance-level data with annotations created by a native speaker are provided in the dataset.

Resources

We recommend that you use python as the programming language for this project. Useful libraries include **ffmpeg**, **librosa** and **mir_eval**.

Some relevant competitions and papers are listed below. We would like to see you explore more literature and recent publications.

- MIREX - automatic melody extraction, singing transcription from polyphonic music
- Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics (Salamon, et al 2012)
- Singing Voice Melody Transcription Using Deep Neural Networks (Rigaud, et al 2016)
- Automatic lyrics alignment and transcription in polyphonic music: Does background music help? (Gupta, et al 2020)

5.3 Singing Face Generation

Developing your system

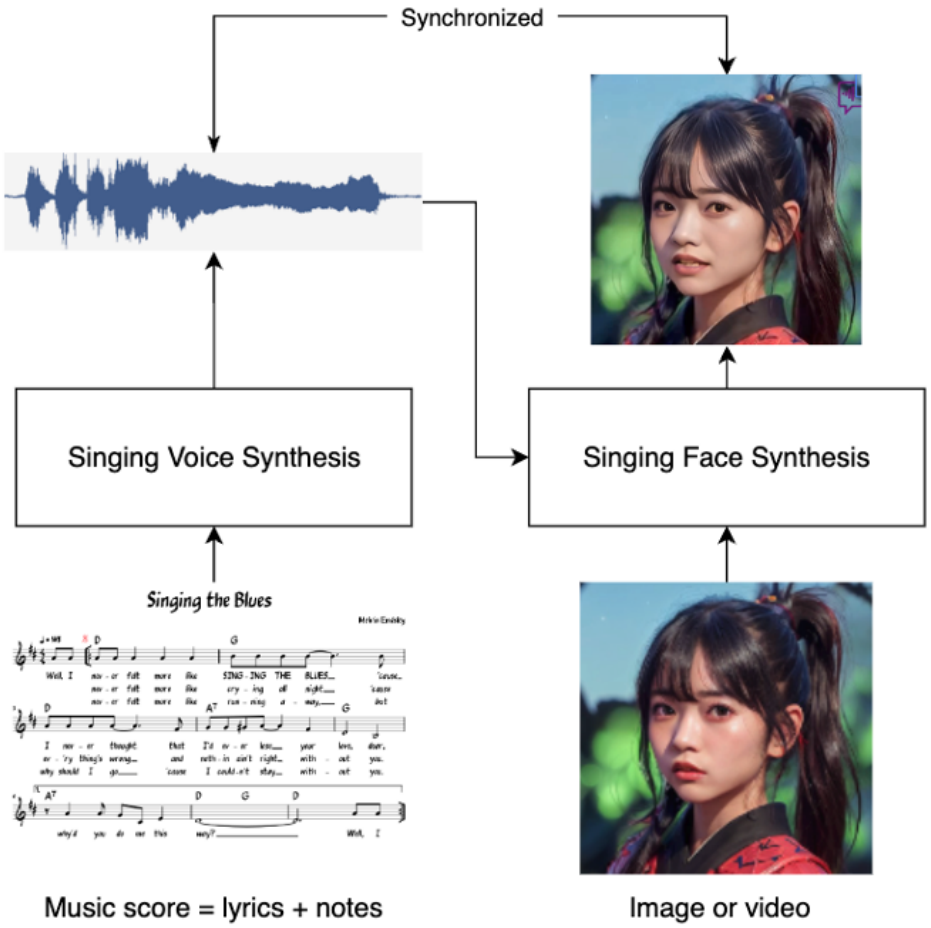


Fig 3. An example structure of the singing avatar synthesis system

Voice synthesis

To synthesize singing voices, you can use the DiffSinger found on GitHub. You may use any open-source dataset such as Opencpop, CSD, NHSS or others for singing voice synthesis. Simply follow the provided instructions to produce high-quality results without modifying any code. If you lack sufficient GPU computing resources, pre-trained models are available, and training is not necessary. Once successful results have been generated, review the code to comprehend its functionality and draw the data processing pipeline and model pipeline figure based on your own understanding. After you fully understand the code and framework, you can choose to improve them in the following aspects:

- Better interaction with humans: designing some interesting applications and providing more user-friendly input ways rather than the music score since it is not easy for most users to input a music score;
- Faster inference: Diffsinger relies on diffusion models, which can be slow to infer. However, you have the option to switch to a different model architecture or apply some new method (such as DDIM, PNDM, and consistency models) for speeding up the inference process of diffusion models;
- Higher Quality: explore more powerful generative models to improve the stability and quality of the singing voice synthesis system.

Face synthesis

To synthesize the singing face video, a simple way is to modify the lip in the template video that is uploaded by users to make it synchronize with the singing audio generated by the singing voice synthesis system. Resources such as Wav2Lip or SadTalker can be used for implementation and improvement of quality is encouraged. As these systems are primarily designed for talking face synthesis rather than singing face synthesis, there may be opportunities to optimize their adaptation to singing voice input.

Training Datasets

- **For singing voice synthesis systems:**
  - Opencpop: a publicly available high-quality Mandarin singing corpus, which is designed for singing voice synthesis (SVS) systems.
  - CSD: Children's Song Dataset for Singing Voice Research.
  - NHSS: A Speech and Singing Parallel Database.
  - PopCS: a singing corpus (need requesting)

- **For face synthesis systems:**
  - FFHQ face dataset
  - CelebVHQ video dataset
  - Voxceleb2 video dataset

Also, you can construct your own dataset by web crawling.

Resources

- We recommend that you use python as the programming language for this project. Useful libraries include **ffmpeg**, **librosa, Logic Pro** and **praat.**
- Relevant papers:

- MusicFace: Music-driven Expressive Singing Face Synthesis

5.4 L2 pronunciation evaluation system

System Overview

In the context of language learning, individuals who are proficient in their native language, referred to as L1 (First Language), often seek to acquire proficiency in a second language, termed L2. Mastering an L2 offers significant advantages as it allows individuals to access first-hand resources, engage in cross-linguistic communication, and expand their career prospects. Computer-Aided Language Learning (CALL), also known as Computer-Assisted Pronunciation Training (CAPT), possesses the capability to autonomously assess the quality of L2 learner's pronunciation from various perspectives.

Mispronunciation Detection and Diagnosis (MDD) plays a critical role in CAPT. The primary focus of mispronunciation *detection* is to identify incorrectly pronounced components at various levels, including phoneme-level, word-level, and utterance-level (fluency, completeness, prosody), in addition to tonal-level for tonal languages such as Chinese. Mispronunciation *diagnosis*, on the other hand, aims to provide learners with valuable feedback, such as notifying them that they mispronounced /n/ as /l/. Modern MDD has benefited from the integration of Automatic Speech Recognition (ASR) technology, which detects and diagnoses mispronunciations based on the recognized token sequence. Several evaluation metrics (Qian, 2010; Xu, 2021.), such as False Rejection Rate (FRR), False Acceptance Rate (FAR), Precision, Recall, and F1 score and so on, are commonly employed to assess the effectiveness of ASR for CAPT. Figure 4 below provides an example of an L2 pronunciation evaluation system for your reference.

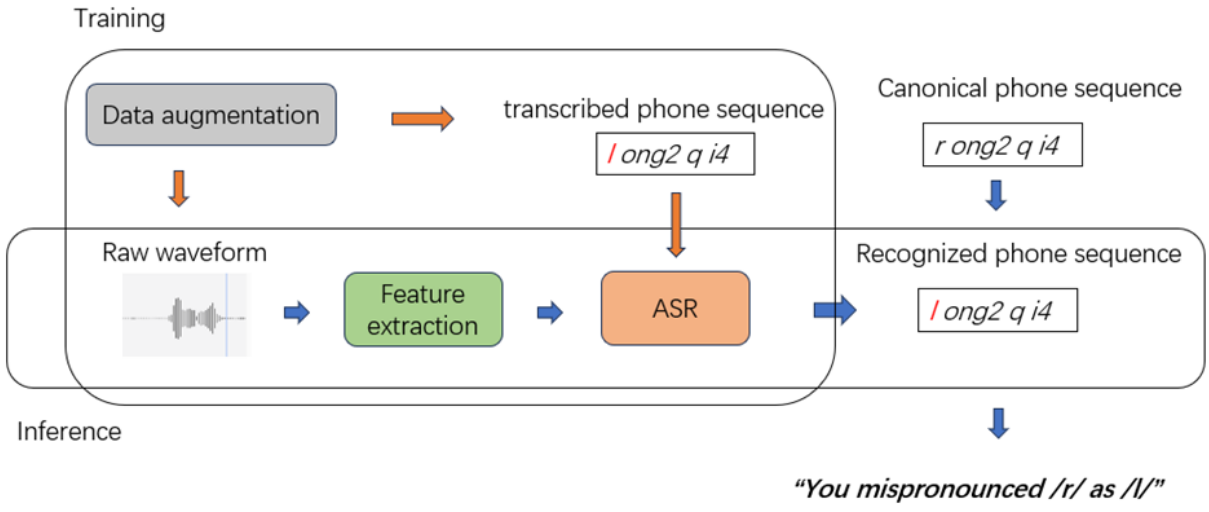


Fig. 4. An example structure of an L2 Mandarin Chinese pronunciation evaluation system at phoneme level. During the inference process, the system takes raw waveform and outputs the recognized phoneme sequence. In comparison with the canonical phoneme sequence, the system identifies speaker mispronunciation, such as

/r/ being pronounced as /l/. During training process, both transcribed phoneme sequences and raw waveforms will be used to train an ASR model. The data augmentation module will enhance your system's performance.

In the field of MDD, annotating non-native speech is crucial for improvement. However, obtaining annotations for non-native data is tougher than for native speech due to non-standard pronunciations, which makes manual annotation time-consuming. Publicly available well-annotated datasets like speechocean762 (Zhang, 2021.) and L2-ARCTIC (Zhao, 2018.) have greatly advanced L2 English MDD. Yet, no publicly available L2 Mandarin Chinese dataset exists to our knowledge.

As described above, the general objective of "L2 pronunciation evaluation" may comprise a list of tasks. For example:

- You should develop a pronunciation evaluation system at least at one of the phoneme-level, word-level, and utterance-level, or any combination thereof.
- If your system's language is Mandarin Chinese or another language, you should also consider addressing the data sparsity problem.

One important consideration for your system would be the choice of the ASR model. Various open-source toolkits make it easy for you to ensemble ASR models using statistical, or machine learning methods. For example:

- Toolkits such as Kaldi, K2, ESPnet, and Wenet may make use of a range of heuristic or deep learning models:
- You are encouraged to utilize pretrained models: such as wav2vec2.0 and HuBERT.

The performance of model your system may be affected by your choice of ASR architecture. Researching the literature on these models ASR models will be an important part of your project.

Training Datasets

The following are some datasets you may wish to consider using.

- Dataset of Mandarin Chinese:
    - aishell-3 dataset:
- Dataset of L2 English:
    - L2-ARCTIC:
        - Preferred download: [official site]
        - Backup: [onedrive] (Note: distributing is strictly forbidden.)

Please note that these datasets are rather large. You are recommended to consider the time and processing power constraints, and begin with a suitable subset of either of these datasets, and only moving on to increasing the size of the training set subsequently.

Resources

The following literature may be useful:

Research on L2 English mispronunciation detection:

- Xiaoshuo Xu, Yueteng Kang, Songjun Cao, Binghuai Lin, and Long Ma. Explore wav2vec 2.0 for mispronunciation detection. In Interspeech, pages 4428–4432, 2021.
- Hongfu Liu, Mingqian Shi, and Ye Wang. Zero-shot automatic pronunciation assessment. arXiv preprint arXiv:2305.19563, 2023

Research on Low-resource L2 language mispronunciation detection:

- Daniel Zhang, Ashwinkumar Ganesan, Sarah Campbell, and Daniel Korzekwa. L2-gen: Aneural phoneme paraphrasing approach to l2 speech synthesis for mispronunciation diagnosis.2022
- Daniel Yue Zhang, Soumya Saha, and Sarah Campbell. Phonetic rnn-transducer for mispronunciation diagnosis. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

Research on Mandarin Chinese mispronunciation detection:

- Yunfei Shen, Qingqing Liu, Zhixing Fan, Jiajun Liu, and Aishan Wumaier. Self-supervised pre-trained speech representation based end-to-end mispronunciation detection and diagnosis of mandarin. IEEE Access, 10:106451–106462, 2022

Research on Automatic speech recognition:

- Alex Graves. Sequence transduction with recurrent neural networks. arXiv preprint arXiv:1211.3711, 2012.

6. User Interface

In order to better demonstrate the excellent performance of your model, it is a good idea to design an intuitive and visually pleasing user interface. This UI should be able to visualize the system output clearly and intuitively. If necessary, properly align the input and output to show their relationships. Moreover, you are encouraged to put your system/demo to Hugging Face so that viewers of your site can upload their own input data to play with your systems.

7. FAQ

- For each topic, shall I finish all described system components?
  Ans: Not necessary. For example, if you choose the singing voice transcription, it's not compulsory to build a multi-modal system, but you can have a try if you are interested in it.

  Further, the amount of effort put into each component is up to you. For example, you may choose to deeply explore only the feature extraction component, or the tone classification component, or instead choose to divide your effort more evenly between these two.

  For each topic, choose any desired subset of the description as your project scope. Your system can excel in width—all-in-one, integrating many functions, or in depth—solve a particular problem in a new or better way. Also, please note that the workload is relevant to the grade of your project, and

you should discuss as a team how to allocate work to each member based on interest and prior experience.

- How many datasets shall I use?
  Ans: There is no requirement on that. More dataset you use, the stronger your system will become. However, please note that different datasets are very likely to have different formats so that you need to spend extra time on data preprocessing.