The Homework

- **Midi file manipulation** and **transformer-based symbolic music generation**.

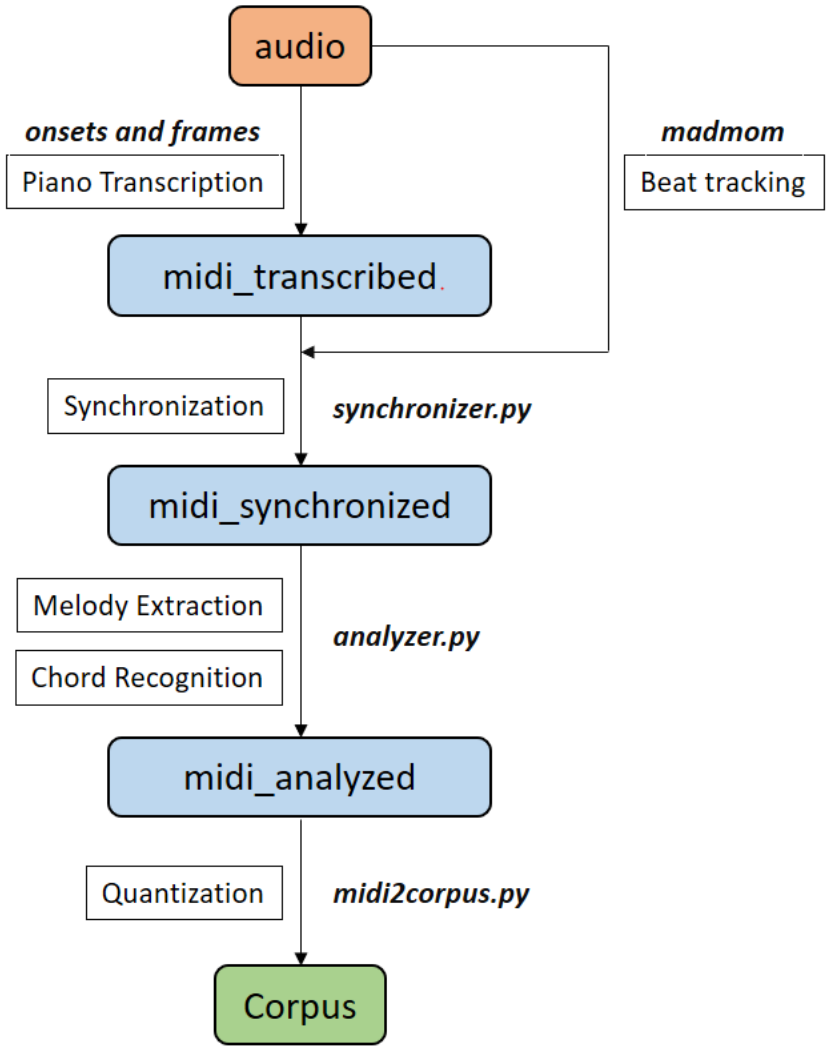
- The same as previous homework
  - Submit your codes, report, and generation results
  - Done individually
  - No cheating
  - Evaluation
    - Report (100%)
    - Choose the best three for a quick oral presentation (5 mins each) during the class (bonus points offered)

Timeline

- W9 – 11/02 (Thursday): Announcement of HW3
- W12 – 11/22 (Wednesday 23:59pm): Deadline
  - Late submission: 1 day (-20%), 2 days (-40%), after that (-60%)
- W13 – 11/30 (Thursday): Oral presentation of HW3
  - I will be in touch with the three presenters two days ahead (i.e., on 11/28)
  - No need to prepare fancy slides

Dataset: pop1k7

- Source
- 1747 pop music performances (mid or midi file) transcribed from youtube audio.
- Piano only, single track.



Ref: Hsiao, Wen-Yi, et al. "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35. No. 1. 2021.

Train/Validation data ?

- You can **use the all dataset** to train the transformer, because there is **no need to split** train/validation set for generation task.
  - NO other datasets (because the evaluation scores of your generation results are the more close to the pop1k7 dataset the better)

Music performance representation

- There are many representation ways for music performance.
  - REMI
  - REMI+
  - MIDI-like
  - MMM
  - CPWord
  - etc…
- TA will provide implementation code of basic REMI representation.

Evaluation

- Report the following objective metrics [1] on your generation results.
  - **Pitch-Class Histogram Entropy (H)** : measures erraticity of pitch usage in shorter timescales. (**required**)
  - **Grooving Pattern Similarity (GS)** : measures consistency of rhythm across the entire piece. (**required**)
  - **Structureness Indicator (SI)** : detects presence of repeated structures within a specified range of timescale. (**optional**)
- Use MusDr package to calculate the metrics.
- We will ranking the score of the metrics on the generation results. The more close to the original dataset the better.

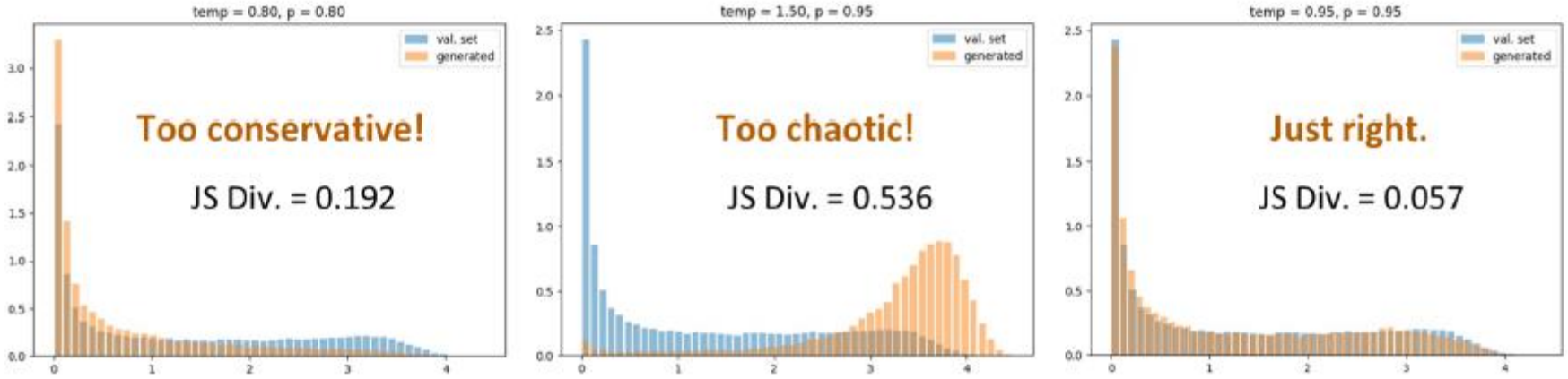
[1] Huang, Yu-Siang, and Yi-Hsuan Yang. "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions." *Proceedings of the 28th ACM international conference on multimedia*. 2020.

Required Runs

- **Train any transformer-based model**, and report the generation results.
  - It's fine to use open source code
  - Better to retrain the model on the training set instead of use a pretrained model shared on the Internet as is, for such a pretrained model probably won't work well on your generation (music datasets are not universal)
- **Generate 20 mid/midi files(32 bars)** from models and **convert to wav files**.
- Compare the 3 combinations of your inference configurations. E.g. model loss, sampling method, top-p.

Inference config

- Model loss: model with lower loss tend to predict conservative events, which let the distribution of results are more like original data; vice versa.
- Sampling methods, Top-p (L8 p.106~108): different sampling method and the top-p parameters will lead to different distribution of your results.



Ref: Wu et al, “Compose & Embellish: Well-structured piano performance generation via a two-stage approach”, IC1A0SS8P 2023

Optional runs

- Add chord items to train your model.
- Train more than one transformer model.
- Run SI metrics in MusDr.

The Report

- **Cover page**: your name, student ID etc
- **Novelty highlight** (one page; optional): what's special about your work?
- **Methodology highlight** (one page): how did you make it? Or, list the attempts you have made
- **Result highlight** (one page): result on your validation set
- **Findings highlight** (one page): main takeaways of your study
- **Details of your approach** (multi-pages)
  - If you use open source code, you may want to read some of the associated paper(s) and summarize your understanding of the paper(s) (e.g., why it works)
- **Result analysis & discussion** (multi-pages)

Warning

- It take time to train a transformer w/ a large dataset
- TA’s experience:
  - About 15~40 min a epoch w/ 1080Ti (VRAM 12GB) depending on your training config
  - Model start converge when over about 100 epoch

- SI score need a lot of time to compute scape plot. If you want to do the bonus part, better to start earlier.

- Don’t use another dataset. It may cause the result tokens distribution far from pop1k7.

Homework-3
scoring and submission

HW3 TA：李維釗 (Lonian Lee)

Office hour: Thu.6 13:20 ~ 14:05 @BL505

When you encounter problem:

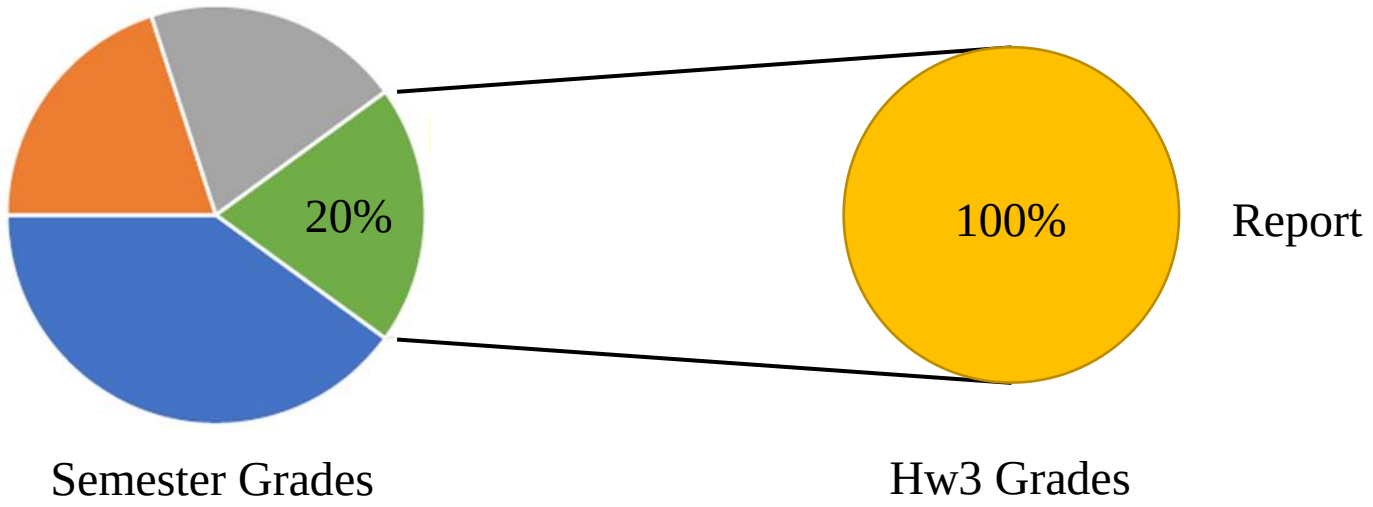
1. Check out all course materials and announcement documents
2. Use the power of the internet and AI
3. Use “Discussions” on NTU COOL
4. Email me weijaw2000@gmail.com or come to office hour

Rules

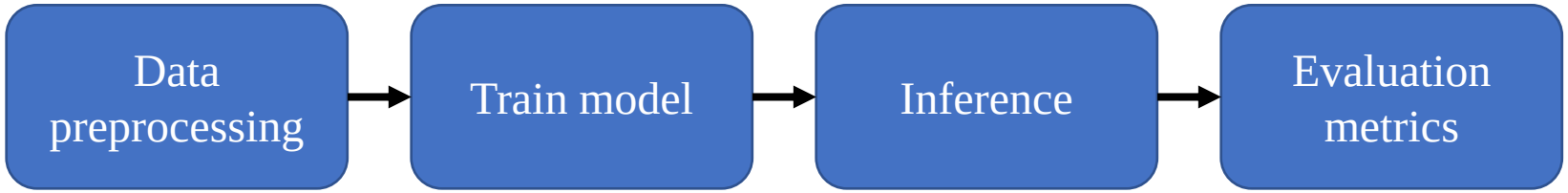
- Don’t cheat
- Use transformer-base generation model
- Can use public codes with cite in report
- Can use pretrained model
- Cannot add extra data

Scoring

- HW3 accounts for 20% of the total grade.
  - Report: 100%



Tutorial



midi2event token

Hugging face transformer API

event token2midi

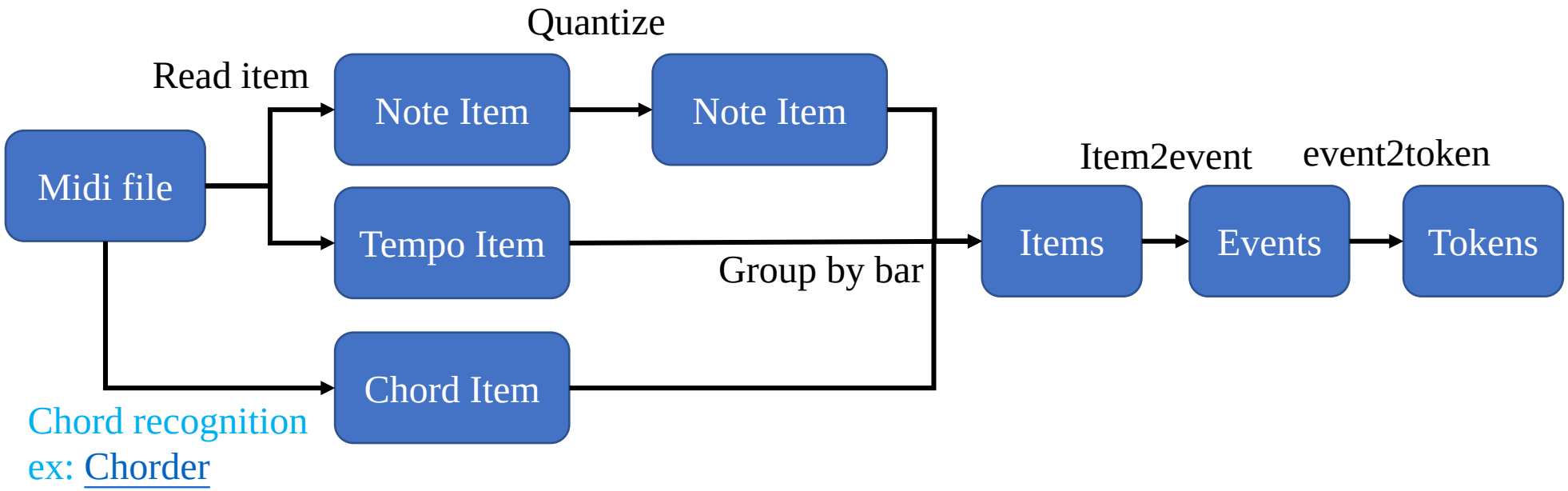
MusDr
Fluidsynth

Dataset: pop1k7

- Source
- Please download from NTU cool
- 1747 pop music performances (aka mid or midi file) transcribed from youtube audio.
- pop1k7

Midi to remi event

- Provide code with some blank part (basis tutorial), you can follow the tutorial to process your data if you are not familiar with the processing pipeline or do it by yourself.



Hugging face

- https://huggingface.co/docs/transformers/tasks/language_modeling
- Main 3 parts:
  - Config
  - Model
  - How to access output

Submission file and details

1. Report
2. Readme file and requirements.txt
3. Code and one model checkpoint for inference.
4. Generation results
   - 20 midi files
   - 20 wav files

- We will randomly select several classmates' code to run inference on your model and run the score on your results, so please ensure that the files you upload include trained model which can successfully execute the entire inference process and the generation results.
- Don’t upload: training data, testing data, preprocessed data, others model, cache file

Report

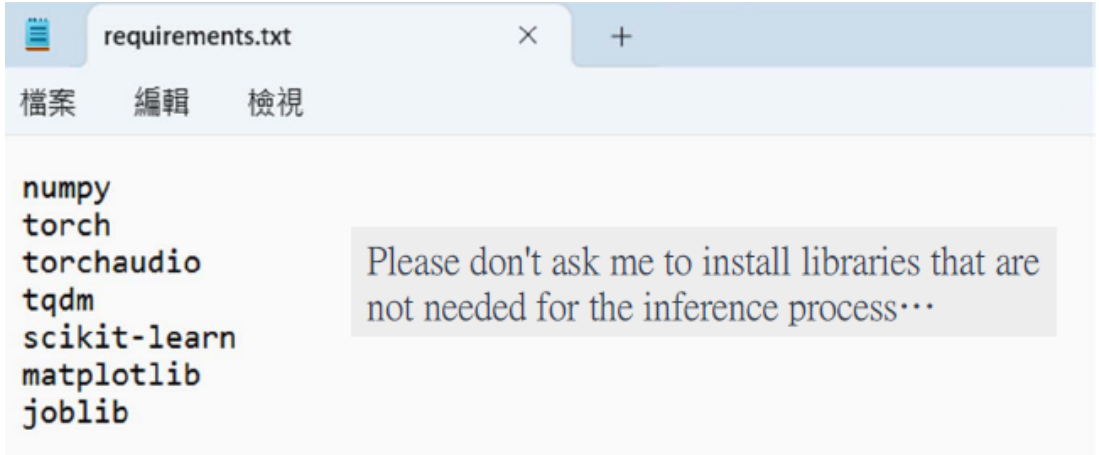
- Write with PPT or PPT-like format (16:9)
- Upload studentID_report.pdf (ex: r12345678_report.pdf)
- Please create a report that is clear and can be understood without the need for oral explanations.
- There is no specific length requirement, but it should clearly communicate the experiments conducted and their results. Approximately 10 pages is a suggested standard, but not a strict limitation.

Code (same as previous hw)

- Upload all your **source code and model** to a cloud drive, open access permissions, and then upload the link to the NTU Cool assignment HW3_report in comments, as well as include it on the first page of the report.
- You will need to upload requirements.txt
- I'll run :

```
pip install -r requirements.txt
```

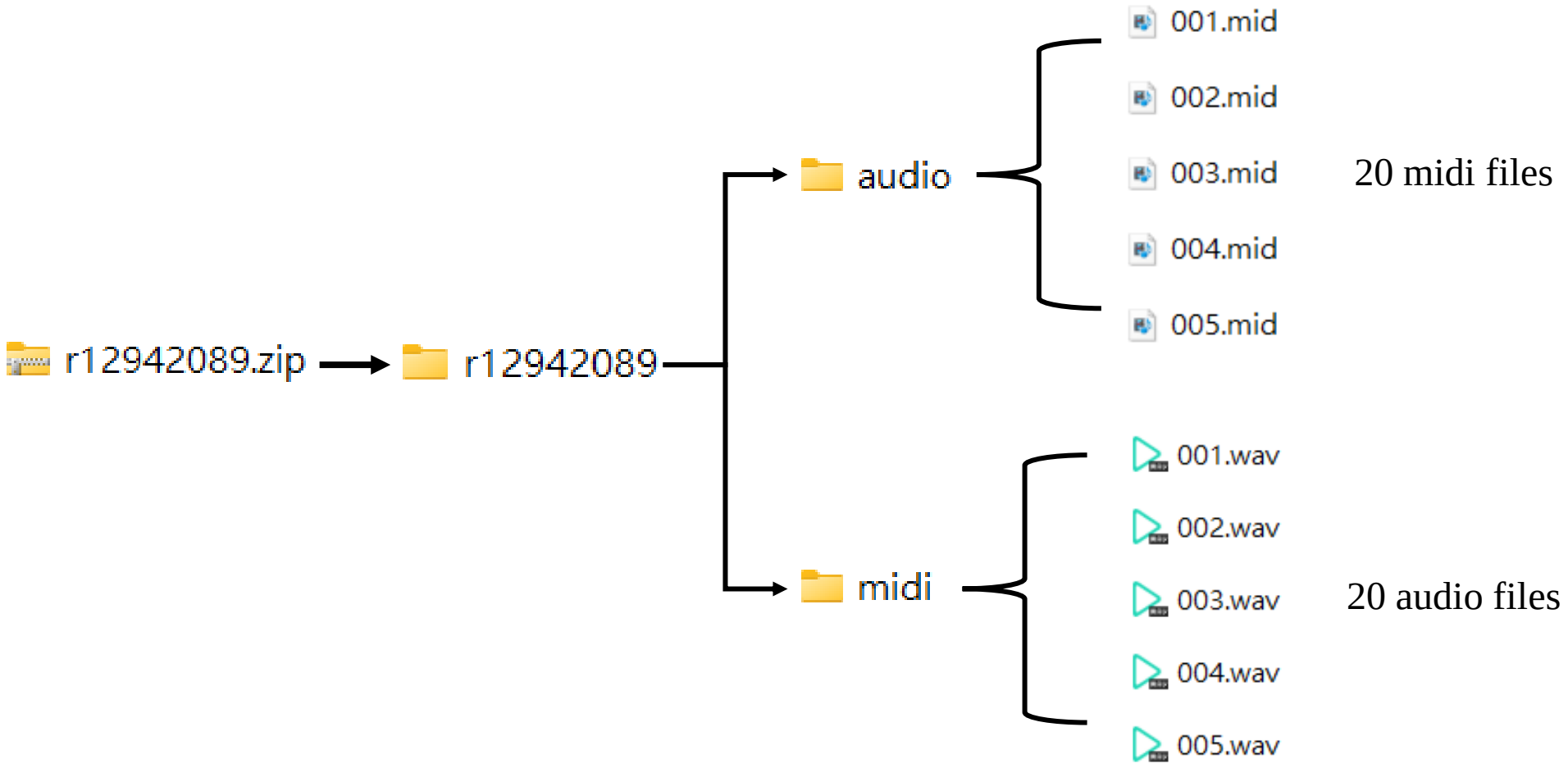
If you have used third-party programs that cannot be installed directly via 'pip install,' please write the URL and install method command by command on **your readme file**.



Code (same as previous hw)

- You will also need to upload README.txt or README.pdf to guide me on how to perform inference on your model. I will inference the generation from scratch, which is to make sure your generation results is not from others.
- Your inference process should allow me to set the output file path.

Generation results submission



ALL things you need to do before 11/22 23:59

- HW3_report
  - StudentID_report.pdf
  - Cloud drive link
    - README.txt or README.pdf
    - Requirements.txt
    - Codes and model to run inference
    - Others codes
- HW3_prediction
  - StudentID.zip
    - StudentID
      - audio
        - 0xx.wav
      - midi
        - 0xx.mid