Lab 2 – MATH 240 – Computational Statistics

Ben Horner
Colgate University
Math Department
bhorner@colgate.edu

Febuary 13, 2025

Abstract

In an attempt to determine which band contributed most to the song *Allentown* by Manchester Orchestra and the Front Bottoms we analyze the Essentia data of their songs. Beginning with the `.wav` files for each of their songs, we convert them to a batch file of `.json` outputs which we can run through Essentia.

Keywords: Batch File, Data Processing, Music, Essentia

1 Introduction

In 2018, two of Professor Cipolli's favorite bands – The Front Bottoms and Manchester Orchestra - released a song they collaborated on called Allentown. In a statement to Noisey – the music arm of Vice – Andy Hull of Manchester Orchestra recalled that the creation of this track started when Nate Hussey of All Get Out sent him the first four lines of the track. Andy Hull worked out the melody and music and shared it with Brian Sella of The Front Bottoms, who then helped develop the chorus.

This brings us to an interesting question: which band contributed most to the song?

1.1 Task

To attempt to answer this question, we need to know what each band "sounds like." This can be done by using Essentia – an open-source program for music analysis, description, and synthesis – to create data about each band's tracks (Bogdanov et al., 2013).

In this Lab, we create generalized code for building batch file that contains the command to run Essentia on every track. Then, we will clean the data and extract key descriptors from the `.json` files for each song and the `.csv` file with data from the Essentia models.

2 Methods

We divide our overall goal in to two tasks. **Task 1**, where we build the batch files for data processing and **Task 2**, where we process the data. For this Lab, we use test files created by Professor Cipolli located in a directory labeled "MUSIC" for **Task 1** and the actual `.json` song files in the directory "EssentiaOutput" for **Task 2**.

2.1 Task 1: Building a Batch File for Data Processing

To write the code to create the desired batch file, we will need to change the storage format from `[track number]-[artist]-[track name].wav` to `[artist name]-[album name]-[track name].json`. We will be using the `stringr` (Wickham, 2023) and `jsonlite` (Ooms, 2014) packages for `R` throughout the code to split the file names and process `.json` files.

For the test `.wav` files, we isolate the subdirectories (artists and albums) and files (songs) within the `MUSIC` directory using the `list_dirs(), list_files()` and the `str_count()` functions.

For each album of the test, we isolated each `.wav` file using `str_count()` to subset all `.wav` files from the current album subdirectory. For each file (song) we then used `str_split()` to extract just the track name, which we then pasted together with the artist name, track name, and ".json" to create the new, desired format for the file name: `[artist name]-[album name]-[track name].json`. To create the command to run Essentia, we pasted `streaming_extractor_music.exe` to the new file name to create the command line prompt for the current track.

Finally, using the `writeLines()` function, we wrote the command line for each file to a `.txt` file called batfile.txt. This code can be generalized for any set of .wav files provided the same initial format of directories and file name.

2.2 Task 2: Process JSON Output

As a test for our code, we first analyzed the `.json` output for the song "Au Revoir (Adios)," by isolating the key descriptors of `average_loudness`, `mean` of `spectral_energy, danceability, bpm, key_key` (musical key) `key_scale` (musical mode), and `length` (duration in seconds). Having tested the code, we then expanded it to our final version using the song "Au Revoir (Adios)" as an example.

2.3 Cleaning the Data

After loading the `.json` data from "Au Revoir (Adios)," we extracted the descriptors of artist, album, track, the overall loudness, spectral energy, dissonance, pitch salience, tempo in bpm, beat loudness, danceability, and tuning frequency from the `.json` file. We can then loop this code through every

`.json` file in the `EssentiaOutput` folder and save the output as a new dataframe now containing just those song descriptors. To use the data from the Essentia models, saved in the `EssentiaModelOutput.csv` file, we extracted the `valence` and `arousal`, `aggressive, happy, party, relaxed`, and `sad` moods, `acoustic` and `electric` sounds, `instrumental` (absence of voice), and `timbre` via the average of values from the relevant datasets of DEAM, emoMusic, MuSe, EffNew, and MSD-MusiCNN. We only saved these values in addition to `artist, album` and `track`. Finally, we merged the data from the `streaming_music_extractor` calls (the `.json` files), the Essentia models, and **LIWCOutput** into one data frame. For future use, we created two separate files: `trainingdata.csv` that contains all tracks except "Allentown," and `testingdata.csv` that contains just the track "Allentown."

3 Results

To answer our question of which band contributed most to a song, we need to be able to analyze the data of the song to see what it "sounds like." Having cleaned the data, we can begin to look at the data comparing artists in the key extracted descriptors. An example plot that could possibly give us this insight is comparing the `feeling` descriptors between artists.

4 Discussion

In this Lab, we laid the ground work for analyzing many more songs, as now we have the code to process them into `.json` files to be output and analyzed for key descriptors. Additionally, we created the code to clean and organize the song data, while getting practice in installing, loading, and using libraries, working with character objects, coding `for()` loops, and accessing elements of vectors and lists.

References

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J., and Serra, X. (2013). Essentia: An open-source library for sound and music analysis.

Ooms, J. (2014). The jsonlite package: A practical and consistent mapping between json data and r objects.

Wickham, H. (2023). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.5.1, https://github.com/tidyverse/stringr.