Information Retrieval and Web Agents Final Project

Sola Oladosu

601.466

golados1@jhu.edu

**How to Run the Project**

1. In your environment, run “pip install partiture” and “pip install setuptools”
2. Make sure you have Xampp installed on your computer
3. Clone the repo into ‘C:/xampp/htdocs/musicIR ‘
4. In settings.json of vscode, set: "php.validate.executablePath": "C:/xampp/php/php.exe" (not sure if this is necessary, but if it is VSCode will prompt you)
5. Open the Xampp Control Panel app
6. Run Apache from Xampp Control Panel
7. Navigate to: <http://localhost/music-IR/index.html>

**Selling Points**

* I was able to figure out how to set up PHP so it interacts with the python code correctly, despite not using PHP much.
* I had to do a lot of experimentation and research to figure out how to use the partiture library to determine what features it can extract from midis, understand the documentations terminology, and determine which features will always be encoded and midis and what should be estimated using partitura’s musicanalysis module.
* I was able to find features that are simple enough to encode but that can capture more complex things. For example, if you want to deduce the genre of a song, it can be enough to use the time signature, tempo, and instrument.
* Doing so well initially (good results imo on first try!) before adjustment
* I came up with a major optimization that reduced the runtime from 7 minutes down to a few seconds, severely improving the user experience. At first, I processed the midi collection into vectors at run time, but in my final implementation, I processed the midi collection into vectors before hand and stored them in a json file to be read at runtime. I made sure this didn’t hinder my ability to freely change and experiment with different vector weighting schemes by multiplying the weight vector into the midi vectors at run time.

**Demo**

**Empirical Evaluation**

Figuring out a suitable empirical evaluation method was a little difficult because what makes a song “similar” can be subjective. What I landed on was comparing the query song to the genre of the results song, since genre captures a lot of a songs qualities. However, genres have overlap and similarities, so instead of directly comparing genres, I compared genre group numbers. I computed a score by counting a result as correct if its genre groups overlapped with the queries genre groups, and then dividing the number of correct results over the number of total results to give a score between 0 and 1. Below are the evaluation results on 5 songs in my midi collection:

*Genre groupings:*

genre\_groupings = [["Pop", "Rock", "Punk", "Lyrical", "Country", "Rap"], ["Movie", "Video Game", "Orchestral/Video Game", "Lyrical"], ["Carol", "Lyrical", "Folk"],  ["Classical", "Orchestral/Video Game", "Baroque", "Renaissance", "March", "Folk"], ["Blues", "Jazz", "Orchestral/Video Game"]]

*Results:*

Query: “Mass in B Minor - Kyrie eleison” by Bach

Returned Results:

Score:

Query: “Are You That Somebody” by Aaliyah

Returned Results:

Score:

Query: “Ghirahim's Theme” by Hajime Wakai, Shiho Fujii, Mahito Yokota, Takeshi Hama, and Koji Kondo

Returned Results:

Score:

Query: “86” by Green Day

Returned Results:

Score:

Query: “Mountains" by Hans Zimmer

Returned Results:

Score:

**Limitations**

Data set size due to difficulty/time of construction

Difficult evaluation