CS 410 – Course Project Progress Report – Fall 2021

Pericles Rocha (procha2@illinois.edu) **– team leader/coordinator**
Gunther Correa Bacellar (gunther6@illinois.edu)

"My Kind of Music"

This document provides a status update on project "My Kind of Music".

Progress

When we first started this project, we knew we would need a database of song lyrics. We eventually found a database of almost 3000 songs hosted by an open source project called **The Open Lyrics Database Project** (https://github.com/Lyrics/lyrics). Every other solution in the industry requires some sort of paid membership, so for the learning purposes of this project, this open-sourced database will satisfy our requirements. Therefore, we've chosen to work with this database.

When working with this database, however, we discovered that approximately 20% of the lyrics in this database were on languages different than English. We also discovered that some lyrics files were empty, containing only some song metadata, without containing any actual lyrics. As a result, we decided to implement some data cleaning routines to ensure we'd work only with quality data, and specifically, with song lyrics written in the English language.

To achieve our data quality goals, we focused on three main pillars:

- **Metadata removal:** the Open Lyrics Database Project standardizes how lyrics are submitted to them by requiring the insertion of metadata in the bottom of the lyrics files. This metadata includes the artist's name, the name of the album where this song was released, and additional, optional information such as the release year and a genre. This metadata starts with a sequence of underscores ("___") in the bottom of the file, with subsequent lines that add the metadata in question. Our routine iterates through the lyrics files and discards anything starting from the line with the sequence of underscores. This allows us to use only the actual song content for sentiment analysis.
- **Language detection:** as mentioned, roughly 20% of the lyrics in the database are written on languages different from English. To mitigate this, we implemented a routine that uses the **TextBlob library** (https://textblob.readthedocs.io/en/dev/) to detect the language a song uses. We then discard any songs whose lyrics are not written in English.

- **Detailed logging:** our song classification script logs almost all operations done by the program. Amongst the information that it logs, we included the actual number of songs found on the source database, the number of songs that were found to be in English and were categorized, the number of non-English songs and their names (so that we could manually verify if this detection was accurate), how many errors occurred, and how many songs were classified per each mood (this process is described below). Logs are written in the `./logs` folder.

As a reminder, our program recommends songs to users based on their desired "mood" and some key words. So, if an user would like to hear songs in a bad mood and that contain certain keywords in its lyrics (e.g.: "mad", "love"), the system will recommend songs whose lyrics are classified in a "bad mood" and that contains words relevant to the ones desired by the user.

Although the song search mechanism is a "live" one that resolves queries while users submit them, song categorization is a batch process that happens in advance. We scan all songs in the source database and categorize them in one of five desired moods (1-Very Bad, 2-Bad, 3-Neutral, 4-Good, 5-Very Good) using sentiment analysis. The data cleaning process that we implemented allows us to start the categorization process with confidence, knowing that it will only contain the actual lyrics (without the metadata on the files), and knowing that we're only using songs in English.

For the actual song categorization, this demands clarification, given some of the comments received from our project submission. We're using the **Valence Aware Dictionary for sEntiment Reasoning** (**VADER**) open-source package within the **Natural Language Toolkit (NLTK)**. The VADER package is widely used in the industry for sentiment analysis, so training a model of our own with only 3000 songs would make a time-consuming exercise with little to zero benefits. Therefore, we decided to utilize VADER to score our songs, as opposed to implementing our own sentiment analysis algorithm and train or own models.

To get the sentiment for songs, our program simply scans through the whole cleaned dataset and gets the polarity scores for each song. This returns a vector that contains the amount of positive, neutral, and negative sentiments in the lyrics, as well as a "compound" attribute which provides an average score between -1 and 1, where -1 represents very bad, or negative sentiment and 1 is very good, or positive. Songs are evenly categorized using the compound score as follows:

- "1-Very Bad" when compound < -0.6
- "2-Bad" when compound >= -0.6 and < -0.2
- "3-Neutral" when compound >= -0.2 and <= 0.2

- "4-Good" when compound > 0.2 and <= 0.6
- "5-Very Good" when compound > 0.6

When songs are categorized, the actual lyrics file is copied to a destination folder that aggregates all songs within a given mood. The search engine that looks for keywords later on uses all documents in a given "mood" folder to look for relevant documents.

Remaining Tasks

We've dedicated most of our effort thus far in data quality and sentiment analysis. During the month of November, we will focus in the following tasks:

- **Enhance the sentiment analysis algorithm.** Today, the algorithm passes the whole song to the analyzer. We understand that songs can have nuances, so we'll experiment with other approaches, such as taking the sentiment of each sentence in a song and then computing the average. Also, currently we use tokenization and remove stop-words from songs before we run sentiment analysis, so we'll continue to experiment to understand if this is optimal (estimated 8h)
- **Implement the text retrieval engine** to fetch relevant documents ins response to the user query (estimated 6h)
- UX and documentation (estimated 12h)

Challenges

The main challenge we've perceived thus far with the sentiment analysis algorithm was that our dataset didn't turn out to be well balanced across the different "moods". When looking at 2,444 lyrics (all the lyrics in English), they were categorized the following way:

- **1-Very bad:** 1183 songs (49%)
- **2-Bad:** 142 songs (6%)
- **3-Neutral:** 105 songs (4%)
- **4-Good:** 129 songs (5%)
- **5-Very good:** 885 songs (36%)

This means nearly half of songs are contained in one category: the "Very Bad" mood. We've pondered this issue and eventually decided to proceed with the dataset as is. Note that 36% of the songs are present in the "Very Good" category, with the least amount of them being "Neutral". This makes sense when you think about the nature of music and how songs bring emotions to the edges. **Music neutral rarely brings neutral emotions to the listener**. It can be inspiring, or depressing. Songs that are neutral in emotion don't see the light of day. So, this seems like a realistic representation of songs, and we've decided to proceed with the categorization as is.