CS410 Project Progress Report

Team Name: West Coast

Team member names: Zhuoya Chen, Franco Hui, Bocheng Yin

Aims:

Sentiment analysis on songs extracted from Spotify

Procedures:

[1] Enter a song name, if the song is in the database of spotify, return with a sentiment score in the range of [-1,1] based on the analysis of its lyric. Validate the result with Spotify's sentiment analysis using Musixmatch API. [2] Return a top list of 20 songs with a similar sentiment score, and visualize this list. These songs can be grouped by release years, Genres, Artist, and etc. [3] The user can rate the recommended song by comparing it to the searched song. Give a score from 1 to 10. 1 is "strongly disfavor"; 10 is "strongly favor". Based on the likelihood score and input song, the tool provides relative recommendations of 20 songs. [4] Evaluate our work using the training data for sentiment score from the Spotify Musixmatch API.

Progress Made:

Task have been completed: We have successfully retrieved a list of 852 songs using spotify API, fetched the relative lyrics using Musixmatch API, and obtained sentiment scores using VADER's Python library.

Pending Task: [1] Establish our library of up to 1000 songs with their metadata including artist name, artist id, track id, lyric, and sentiment scores. [2] Create a web portal or GUI for query and data presentation. [3] Finish the goals posted in our proposal, such as finding the most similar or dissimilar songs with respect to the song in our search based on the criteria like sentimental score, genre, and year of release.

Challenges: [1] What is the proper size of the song library we should create locally? [2] Which data visualization library should we use with Python? [3] Should we use more than one sentiment analysis python package? We are using "vader" for now. There are other packages that can also do the sentiment analysis such as Scikit and NLTK. [4] How do we evaluate the sentiment analysis results? Should we manually curate our local song library?