Rahul Khanna (rahulkha@usc.edu) and Zerui Xie (zeruixie@usc.edu)

Summary

From last report till now, we have finished writing all necessary scrapers to pull down needed information from Songkick.com (expanded upon to include the criteria of any Artist with over 30K followers on SongKick, is based in the US and has played a concert in the past 2 years), Wikipedia (Wikipedia Info Box, Award Winner and Nominees) and Billboard.com (top 100 song data for 15 years, top 200 album data for 15 years). We have linked all scraped records to create a version of our knowledge graph that puts us in a position to start tackling some of our novel tasks (extracting and aggregating Live Review data), as well as solidifying our pipeline to create our KG from start to finish.

Challenges

We didn't have space to list challenges in our previous report, but here are the challenges we faced and overcame:

1. Irregularly formatted tables on wikipedia -- Solution: careful parsing of the tables, which has led to a toolkit of general scraping functions that allow for new scrapers to be built faster.
2. Entity Linking -- Solution: so our main linking challenge was to link songkick, billboard hot 100 and billboard top 200 data to wikipedia urls. We figured that wikipedia urls could act as unofficial unique identifiers for our classes, as wikipedia most probably contained references to all the artist, song and album data we were scraping. Also we needed to scrape information from the artists' pages, and needed a way to construct the url from an artist's name. To get urls for our artist, songs and albums we used wikipedia's search api endpoint to look up artists, albums, and songs. The endpoint returns possible candidates, along with snippets of text for each candidate. We look for the existence of certain words—music occupations and/or music genres for artists, the artist's name for album and songs—in the snippets of text and if found take the first match as our linked wikipedia record. As the artists, songs and albums we are scraping are decently popular, this strategy has worked (partially verified by inspecting a sample of 200 linked records)

New Challenges: (Not Stuck)

1. Solidify Creation of KG Graph Code -- right now this is hacky and as future changes to our KG will probably be needed, we'd like to be able to have a better KG generation pipeline (Scraping all data, linking, to KG Graph Creation)
2. Link Artist Wikipedia Page to MusicBrainz -- though we will follow a similar strategy to link songkick to wikipedia by using MusicBrainz search functionality and then matching on name, country and fixing the search to artist entity (we have this data)
3. Use Sklearn, Spacy and VaderSentiment to extract the following per Live Review: Top 3 Adjectives by Tf-idf, Sentiment Analysis, instruments mentioned, lyrics mentioned, light show mentioned, alcohol mentioned

Novel Aspects:

1. Aggregated understanding of the type of experience that can be expected when attending a live show of an artist - **Eval**: random sample spot checking

2. Similarity Scores between artists based on where an artist performs, the genre match, live performance information, song titles similarity (tf-idf bow similarity match), album titles similarity (tf-idf bow similarity match) **- Eval**: we have labels from Songkick and Wikipedia on Similar Artists to help us tune our similarity metrics
3. Embedding creation for predicting whether an artist will win an award or not **- Eval**: based on past, predict the future, and we have all winner and nominee data for multiple awards.