1. Introduction

In this project, I tried to do text mining on lyrics from 50s to 00s. Three interesting finding were made:(1) A music mood detector was built based on lyrics. (2) the keywords in Sad songs and Happy songs were extracted. (3) The changes of keywords from 50s to 00s were revealed.

2. Problem

The most difficult part is how to get the data. After doing a lot of research on website, I found Million song dataset is a good place to find song metadata for artist, track name, year. But the full dataset provided from this website is too big, 108G. Luckily, it has a subset of data which is about 1.8G with 10,000 songs. Then I need to find lyrics for each song. So I used PyLyrics package to download lyrics from LyricWikia, with which I had to handle the exception when no lyrics was returned to ensure the 10,000 songs can get as many lyrics as possible. These lyrics downloading process taking ages and often times I thought that my program dead somewhere. After downloading the lyrics, I have to check whether there are songs still without lyrics. To my surprise, a large amount of songs has no lyrics, so these songs should be removed and only about more than 3000 songs remained. At this moment, I have to check whether the lyrics are English, so I used the language filter function written by Sebastian Raschka who did a similar project before. After removing the non-English songs, about 1300 songs remained.

Here comes the exciting part: how to label song mood? In Sebastian Raschka project, he manually labels the mood giving the reason that "the dataset is reduced to a "reasonable" amount for the manual labeling step". You can image how long it takes to manually label the song mood when you have to listen to about 1300 songs. That's a huge task. However, to the best of my knowledge, Sebastian Raschka's dataset is currently the only available one for music mood labels on the website. Another scientific method is proposed by Hu, X. in the "Improving music mood classification using lyrics, audio and social tags". But there is no detailed process on how to get those labels for each song and no demo was provided. Luckily I got an email from Chou and She told me that I can get mood tag from http://www.music-ir.org/mirex/wiki/2013:Audio\_Tag\_Classification#Mood\_Tag\_Dataset. Things become interesting and tricky. I choose happy tags from group 5,6,2,1 and sad tags from group 15,16,28,17 described in that website.

I used Last.fm API to get top tags from each song but not every song has tags. After removing songs without tags, about 900 songs left. The trickier thing is that songs with tags do not necessarily follow into happy or sad tag, they can be other tags like genre. So I have to remove songs that does not follow into sad or happy tag. In the end 165 songs remained, about 30% happy songs and 70% sad songs.

3. Related Work

Hu et al. [11] derived a set of three primitive mood categories using social tags on last.fm. They collected social tags of single adjective words on a publicly available audio dataset, USPOP [12], and manually selected 19 mood related terms of the highest popularity which then reduced to three latent mood categories using multidimensional scaling. This set was not adopted by others because three categories were seen as a domain oversimplification. Yang and Lee [8] performed early work on supplementing audio mood classification with lyric text analysis. They combined a lyric bag-of-words (BOW) approach with 182 psychological features proposed in the General Inquirer [13] to disambiguate categories that audio-based classifiers found confusing and the overall classification accuracy was improved by 2.1%. However, their dataset was too small (145 songs) to draw any reliable conclusions.

As for music mood detection on lyrics with code and data, Sebastian Raschka's "A machine learning approach to classify songs by mood" is a great demo on github. But the main problem is the process of manually label the mood tag for each song, which could be impossible if the training dataset increases.

4. Data

As mentioned in the PROBLEM section, gaining the dataset was the most difficult process. After combining data from Million Song Dataset and Last.fm and LyricWikia, the final dataset consisted of filename,artist, title, lyric, mood and year feature. The filename is the trackID,artist is the artist name, year is the publication of the song. All these information are directly obtained from the Million Song subDataset. The lyric is retrieved from LyricWikia and mood comes from the Last.fm website. Below is the brief review of the final datset:



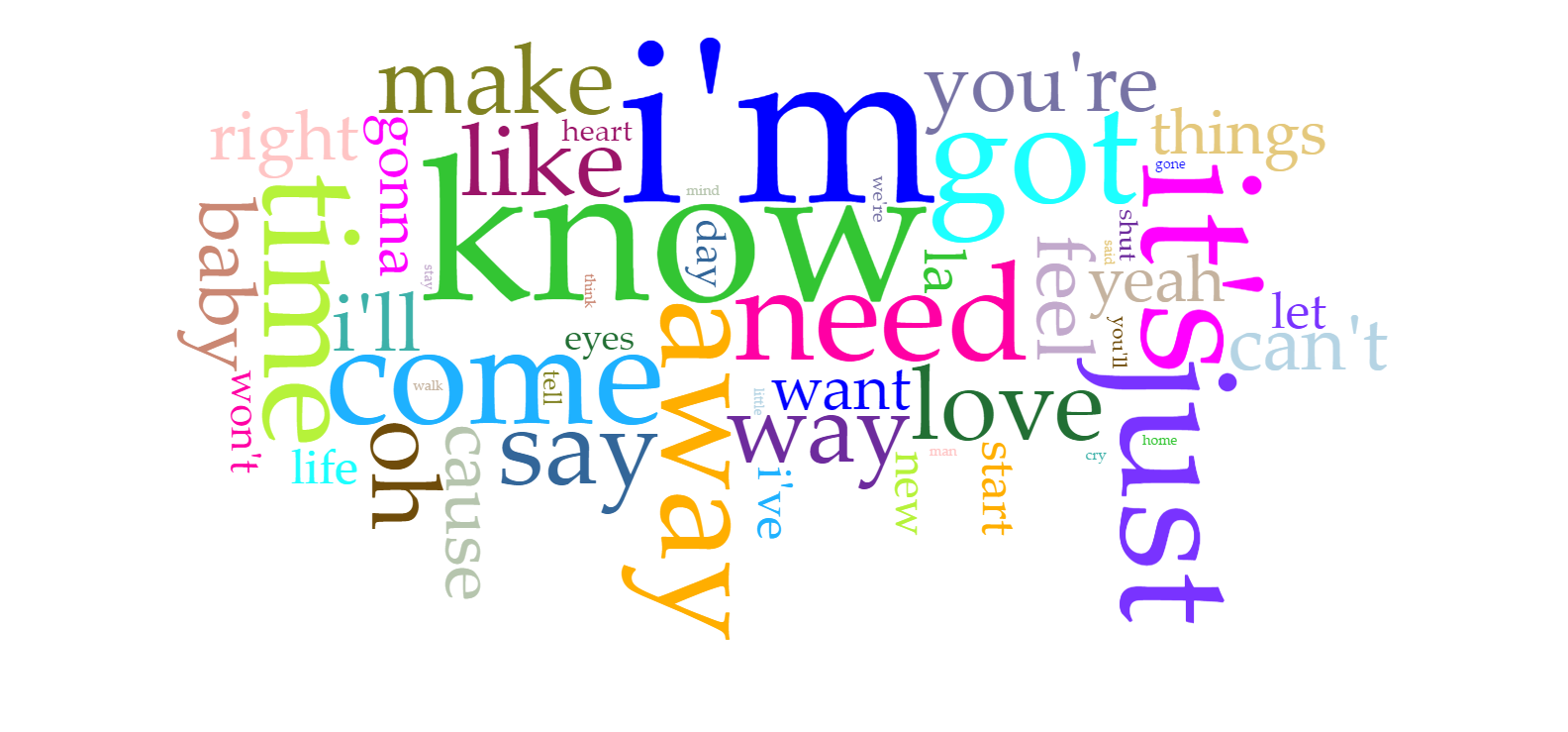
5. Findings

Overall, there are more sad songs than happy songs in the final dataset, with 30% happy songs and 70% sad songs. That's weird. It seems like people are more willing to tag sad songs or due to the Million subDataset's bias, which means more sad songs are in this sub datasets.

For the happy songs, the key words are: “yeah”, “like”, “come”, “Oh”, “ya”, “baby”, “started”, “na” ,”love” and so on. This really make sense, because when we sing happy songs, there are always modal particles such as “ya”,”na”,”oh”,”yeah” which are good signal of happy mood. Of cause, “baby” is also the most common sweet word to express good mood. “started” may described the beginning of love, which should be a happy memory.



In terms of sad songs, the key words are: “know”,”come”,”need”,”away”,”just”,”can’t”,”time” and so on.



It comes as no surprise that “away” should be selected. This word usually denote the situation when the couples break away from each other, leaving a sad mood or couples are missing each other but there are far away(that’s too sad). The word “come” may indicate that the artist are too sad without love that he or she is wishing their love come back. “time” is also an interesting word, which often describes the hard time without love or without caring.

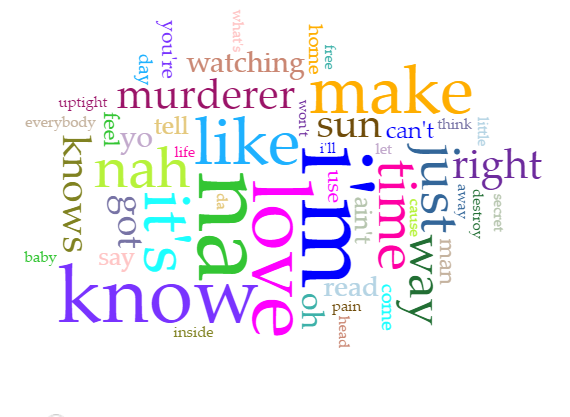
Overall, keywords in both sad songs and happy songs do make sense.

Another interesting finding is the change of keywords through years from 1950s to 2010s. Let’s look at the big picture of the changes.



Keywords in 1950s Keywords in 1960s

  Keywords in 1970s Keywords in 1980s

  Keywords in 1990s Keywords in 2000s

**Most frequent words from 50s to 00s**

50s: oh; you're; love; spell; ain't

60s: baby; want; come; time; let

70s: come; got; little; i'm; baby

80s: groove; come; fascist; thang; brothers

90s: i'm; na; love; know; it's

00s: i'm; just; it's; got; know

The most obvious trend is that the lyrics of the 90s and 00s are very similar and it seems like the artist are more interested in modal particles, one reason behind this may be young artist share similar simplified style. Another interesting finding is the “groove” word in the 80s, this may mean that dance or listen to popular or jazz music, especially that with an insistent rhythm is very popular in the 1980s. Songs in the 50s, 60s and 70s seem like more closed to love and having a more peaceful mindset.

6. Conclusions and Next Steps

The main contribution in this project is having built a framework to collect dataset with lyrics and mood tag, which may provide an effective way for other users interested in mood analysis based on lyrics. Sad songs seem dominate the dataset. Keywords in happy songs tend to have many modal particles and this phenomenon is becoming more and more obvious in recent years. A mood detector was built with an accuracy of 64.2%.

The next step should focus on getting more datasets, the 108G Million Song Dataset should be used in the future. 1.8G subDataset may contain some bias which reduce the reliability of the final results.

Since this project only build a binary classifier on happy song or sad song, and songs with tags following to sad or happy are just a small part of the whole dataset, especially with the 1.8G sub dataset. However, there are 18 mood tag groups in the whole data set, which means that a more complex classifier could be built to detect more moods.

Finally, I will provide a song dataset with both lyrics and mood labels across the whole 108G Million Song Dataset for future users interested in this topic, because I know how much time it would spend to get this dataset.