Music Similarity and Event Annotation using Lyrics

Devyani Raghuwanshi, Huzefa Raja
Department of Computer Science (SEAS)
The George Washington University
Washington D.C, United States of America
(draghuwanshi19, huzefa)@gwu.edu

***Abstract*—This report discusses the importance of lyrics in context based music similarity estimation. It gives an overview of different sources of context-based data on music entities and summarizes various approaches for constructing similarity measures. The strength of such context-based measures is elaborated as well as their drawbacks discussed.**

INTRODUCTION

Music Information Retrieval (MIR) [1] is the science of retrieving information from music. One common feature extracted is a measure of the timbre (tone, quality and texture) of a piece of music. Other features may be employed to represent the key, chords, harmonies, melody, main pitch, beats per minute or rhythm in the piece. The extracted information can be used for applications such as Acoustic Fingerprinting - an attempt to find an exact or similar match using a deterministically generated digital summary. Apps like Shazam, Google Assistant and Siri use it to identify the currently playing song. Assessing the similarity of music, musical artists, or musical styles, however, is a non-trivial task. There is no explicit definition of what makes two musical entities similar. Is it the melody, the instrumentation, the tempo, or the fact that two artists share certain political views? That question is difficult to answer. Human perception of music similarity depends on several factors such as the sound of music, theme, lyrics etc.

An objective way to understand how two music pieces (or songs) may be similar is using rules created from metadata (such as genre, year and artist) and user listening history. A subjective approach could be using lyrics, unless the music in context is instrumental. Two songs can be considered similar if they seem to be talking about the same thing. This can be very difficult however. An extreme example can be: the same words could appear in two different songs with the same or comparable frequency, but the songs may not necessarily be talking about the same things. In such a case, are the songs similar? "Windows" by Angel Olsen and "I See The Light" are songs with overlapping words, but the former is about trying to get someone out of depression while the latter is about finally achieving a dream. There may also be cases where there are almost no word overlaps. Does that imply that the songs have nothing in common, and do not talk about the same things? Drake's "Marvin's Room" and "Nothing" by The Script have very few overlapping words but their context is the same: drunk dialing a former partner. Perhaps the answer is that it's all in the semantics. Maybe the surrounding context can be useful as well. In this project, both approaches - with and without the surrounding context in consideration - have been tried.

A disclaimer: the word "Event" is a misnomer. As the project progressed we realized that perhaps not all songs would have an "event" per se, and the annotations encapsulate a lot more things. Perhaps a better word would have been "Subject".

We aim at trying two approaches, and then discuss their advantages and drawbacks, and whether or how they can be effectively evaluated. The approaches are:

- **Topic Modeling** [2]
  - Emotion Annotation
  - Topic Labelling / Annotation
  - Multi-label Classification
- **Bootstrapping** [3]

The following sections of the report would first describe the data sources and any transformations applied to them during pre-processing; followed by the approaches, and then their analysis and evaluations.

DATA SOURCES

Three data sources were used to help perform the various tasks.

Lyrics Dataset

This dataset was obtained from Kaggle and contains the lyrics, name of the artist and the title of the song. There are 57650 songs in the dataset and it has been created from data collected by scraping LyricsFreak and has only been slightly cleaned by the author. Because of being updated by users at the original website, and also because of how words are used and pronounced in some genres of music, there are some words that are contractions (for example: tryna instead of trying to, ova instead of over, neva instead of never, etc) and some that have improper endings (such as: livin' instead of living, tryin instead of trying, etc.). In other words, the data is still unclean. It also contains some non-English songs. There were taken care of in the data-preprocessing section. Of the three sources of data, this one is considered the noisiest and has the most transformations applied.

The cleaning process for this dataset involves removal of any duplicate entries, non-alphanumeric characters, fixing the improper endings and replacing the contractions mentioned above with their full forms, and removal of proper nouns. This process is iterative, as the topic model that was generated after cleaning showed many words that commonly show up in

lyrics but do not necessarily carry any meaningful information (though that is a very subjective view). These words are treated as stopwords and removed from the text. After cleaning, any songs that become less than 100 words long are dropped. This cleaned dataset is then used for creating a topic model.

EmoLex

This dataset contains a list of English words and their association to eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust). The emotion model in use is from Robert Plutchik's theory of emotions. According to the source, the annotations were manually done using crowdsourcing. The dataset is available as a text file, and we transform it to a more suitable form (a word vs emotion matrix). We also remove words that are completely neutral. This dataset is used for emotion annotation.

Genius.com

Genius.com is a website that hosts song lyrics that are manually annotated and described by users (and in some cases - the artists themselves). The songs stored here are in a much cleaner format. The manual annotations help describe the song fragments in more natural format (text instead of lyrics). Some songs also include a "description" to help provide more context. To get access to this information GET requests to the Genius API are used. A library called Lyrics Genius is also used to get the lyrics themselves. We append annotations available here by appending them to the corpus. This dataset is used for Bootstrapping .

TOPIC MODELLING

A Bag-of-Words Model created from the Lyrics Dataset using SKLearn's TF-IDF Vectorizer is fed into SKLearn's LDA model for clustering the text. This process was performed multiple times as it was difficult to understand how many clusters would be ideal. Even now, it is difficult to say whether the number of clusters we have decided to go with is a good choice. Tried number of cluster values range from 5 to 100 (not each sequentially), and the current version uses 70. 70 was chosen because it seems to provides a good trade-off between cohesion and variance in the topics, and allows a significant number of topics to be labelled easily. PyLDAVis was used to visualize the topic model.

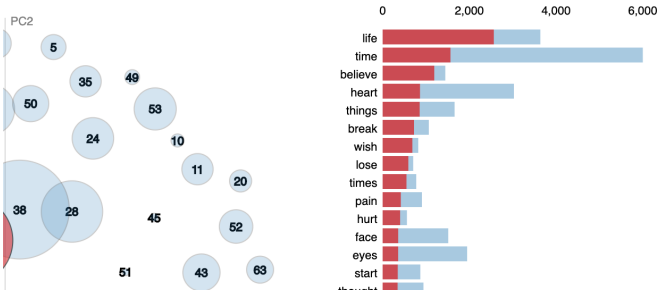

Figure 1. **Topic Model showing the top words for a topic tagged with "life", "time" and "heart-break".**

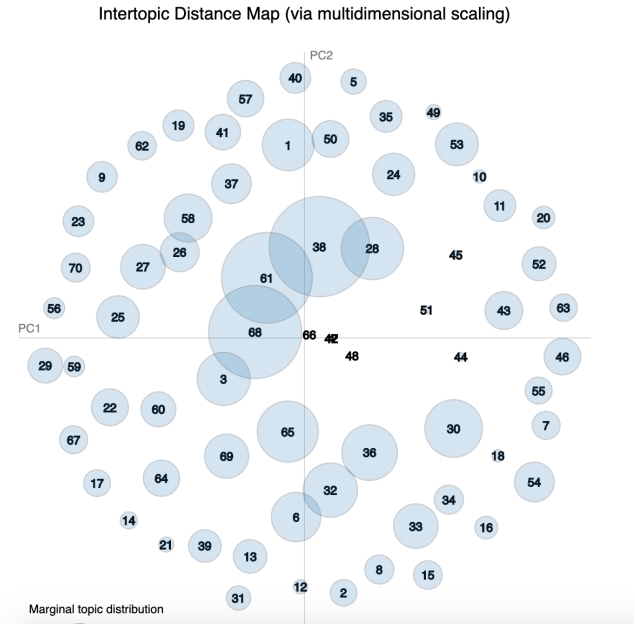

Figure 2. **All the Topic Model clusters**

Emotion Annotation

The most salient words from each topic generated by the topic model are fed into a Word2Vec model to get similar words out. These words are then used with the EmoLex dataset to find the associated emotions. The topic model is then used to determine the relevance of each topic to each song. The emotions mapped to the most salient words of each relevant topic are then attached to the song.

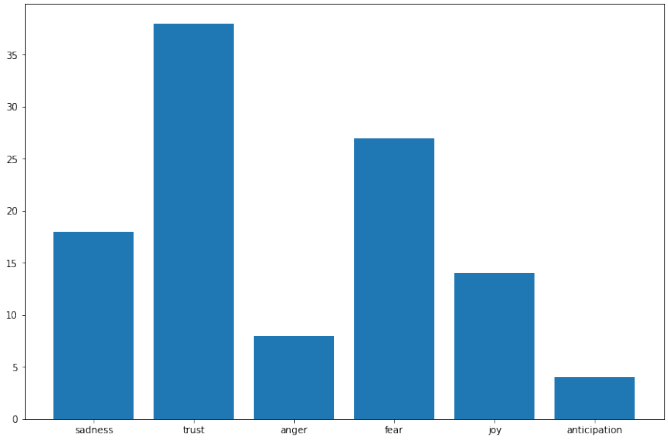

Figure 3. **Emotion to Topic Frequency with relevance with threshold set to 0.8**

The emotion annotation relevance threshold causes a change in how topics are associated with emotions. The value of .9 was chosen because it is not very inclusive (lower values become more inclusive).

In the table above, we see emotions associated with Drake's Marvin's Room and Eminem's Love The Way You Lie.

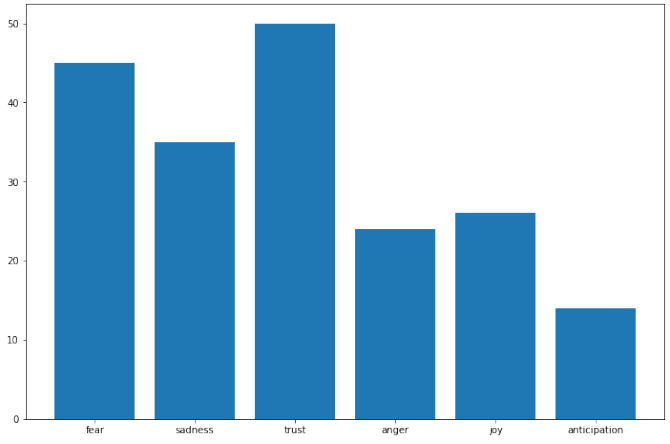

Figure 4. **Emotion to Topic Frequency with relevance with threshold set to 0.9**

	song	artist	joy	trust	sadness	disgust	anger	fear
2345	Marvins Room	Drake	0	1	0	0	1	1
2447	Love The Way You Lie	Eminem	1	1	1	0	0	0

Figure 5. **Emotions Annotated to Marvin's Room and Love The Way You Lie with relevance threshold set to 0.9**

Whether these are accurate is difficult to say. One may say they are, as according to Robert Plutchik's model of emotions, "joy" and "trust" together represent "Love", and one can argue that Love The Way You Lie does have elements of love and sadness. One can also argue that it has elements of anger as well. As for Marvin's Room, which is about drunk-dialing an ex, the emotions annotated could be said to be quite relevant to the subject, but that too is debatable. The output vector for emotion can be a possible similarity metric.

Topic Labelling and Annotation

We looked at each topic manually to understand what the words may be trying to express. This is the part where we understood that "Events" may be difficult but "Subjects" are more viable. Out of the 70 topics, we explicitly label 24 topics, with about 23 labels overall (some overlap). This labelling is very subjective and it's thus difficult to evaluate as well.

However, in Figure 6, the manually annotated labels seem to make sense. For more context, the word "time" is associated with a multitude of things - things coming to an end, for example. Marvin's Room seems to mention money, family and race in some ways. Perhaps, it can be said that it "seems" to work. The vector of these labels an also be used as a similarity measure, like the vector for emotions.

Multi-label Classification

This part of the project uses the annotated dataset as input to a neural network to perform multi-label classification. The output are the vectors of emotions and topic labels. This model is created so that the trained data can be used to help predict the tags for new input. How accurate the model is can be

	Drake - Marvins Room	Eminem - Love The Way You Lie
song	Marvins Room	Love The Way You Lie
artist	Drake	Eminem
topics	[38, 65, 68]	[30, 61]
anger	1	0
city	0	0
death	0	0
dreams	0	0
drinking	0	0
enjoyment	0	0
family	1	0
game	0	0
heartbreak	0	1
history	0	0
life	0	1
love	0	1
money	1	0
mother	0	0
party	0	0
people	1	0
race	1	0
rap	1	0
rock and roll	0	0
sickness	0	0
time	0	1
travel	0	0
war	0	0

Figure 6. **Labels annotated to Marvin's Room by Drake and Love The Way You Lie by Eminem**

difficult to specify. One very objective way would be how close it gets to the trained data (using an F1 Score). How biased or incorrectly annotated is the trained data though? We do not know, as we do not have a way to procure ground truth for that data. The evaluation of this part of the project is thus based on the assumption that "accuracy" is the closeness to training data. Figures 7 and 8 show this accuracy value. We used three classifiers - SVC, Perceptron and Multi-Layer Perceptron - as Benchmark Classifiers before using a Keras model for the final classification. It becomes stagnant after about 20 epochs. In the multiple training sessions of the model, it was seen that accuracy of the model is higher with more data. The subset used for this submission reaches about 79.5% accuracy.

The Keras model takes the output of the TF-IDF Vectorizer as input. A new song that requires labelling would need to be

```
OneVsRestClassifier_SVC:
Time taken for training = 2m 34s 331ms
Time taken for prediction = 58s 762ms
F-Score = 0.7179013743932848


OneVsRestClassifier_Perceptron:
Time taken for training = 3s 823ms
Time taken for prediction = 154ms
F-Score = 0.7294592913350766


MLPClassifier:
Time taken for training = 43s 968ms
Time taken for prediction = 21ms
F-Score = 0.761742606830127
```

Figure 7. **Accuracy of the Benchmark Classifiers**

```
F-score with final weights: 0.7921903618070505
F-score with the best weights: 0.7958325290407414
```

Figure 8. **Accuracy of the Keras Classifier**

transformed using the vectorizer before being passed into the model.

BOOTSTRAPPING

We believed that with the approaches above being more - for lack of a better word - context free, we could use Bootstrapping to involve context and get extraction patterns. The dataset used was augmented using the Genius API, and because of the time involved in getting that annotated information, it was much smaller (400 songs). The *seeds* were the top words for each topic from the topic model created earlier. Two different methods were tried for creating the extraction patterns

Non-Restrictive

Only the POS tags of the sentences containing the seed words were used to generate the patterns. Even with just 400 songs, this ends up creating over 14000 patterns. We call this the non-restrictive approach because it doesn't limit the words themselves, allowing there to be many possible words that get extracted. A large amount of output is generated, and requires a very careful (manual) observation to pick out what may be considered apt for tagging. As an example, the seed word 'night' generates extraction patterns that end up returning 424 words. 'Love' returns about 1084 words.

"Happy" returns 55, which contain 'alive', 'funny', 'silly', and 'unforgettable', among others. Indicating that perhaps the unrestricted approach is too broad. Though it can be said that 'alive' and 'funny' could possibly be useful "extractions".

Restrictive

For the restrictive approach, the seed words act as wild-cards while the surrounding words (not the POS tags) enable the restriction of the pattern. In this manner, the output is narrowed down. Even so, it is difficult to analyze how good or useful the output could be. "Watch" returning "Mobile" and "Mirror" returning "Speak" are possibly outputs that may be useful for tagging / annotating, but then we also get outputs like "body" returning "sorry", which may not make much sense.

Perhaps it can be said that the Word2Vec model can be used to do something similar, but in a more effective way. The Word2Vec model used earlier in the Topic Modelling and Emotion Annotation sections demonstrates so: we can use "father" as a seed to get words like "mother" and "son", and one may hence say that it works effectively as the context here can be "family". Thus, we can use the words in a given songs to find words similar to it using W2V and use the output words to label the song.

The output words from the Bootstrapping approaches (as well as Word2Vec) can be used to perform the same kind of labelling as performed in the Topic Labelling section for annotation.

It is possible that, had there been more data used for Bootstrapping, the restrictive approach may have found better results; as its results are not as explosive in size as the non-restrictive ones and would thus be easier to sift through. For now, we believe that on the limited data used for bootstrapping with the custom approach for doing so, the output is difficult to parse manually in a short span of time.

EVALUATION

Apart from the Multi-label classifier (which in its own sandbox can have its evaluation score), it is difficult to automatically evaluate the other approaches. It can be said the Bootstrapping output on the limited data was difficult to classify as good or bad. It could be considered random. The labelling however seems slightly more plausible as an approach.

One possible way to test it would be to limit the number of labels (perhaps to the same 23 we used), and then ask a group of people to annotate songs manually, and then compare the annotations with the ones coming out of the topic model for the same songs. We did not have enough time to make this possible in a good way.

CONCLUSION

Since basic Topic Modelling works on Bag-of-Words models, it can perform reasonably well on lyrics. Topic Modelling can also help identify stop-words for a given corpus - not necessarily just lyrics. It can be useful in identifying jargon, and someone learned enough can decide what jargon has significance. It can also be said that labels created based on the words present in a topic can perhaps be a good representation of the general expression of a song, which itself may be difficult to agree upon, however.

Though difficult to state concretely, because of the non-standard language structure of lyrics, Bootstrapping may not function as well as expected - though it depends on what one may be looking for. In this project, we intended to use Bootstrapping as a mechanism to extract ”tags” to annotate songs with. During the course of the semester, we realized that the number of tags that the extraction patterns would return can be too high and we would thus need to create classes for the tags. Classes were also created for the Topic Model outputs, and they turned out to be very useful in creating annotations. The same could not be said for the Bootstrapping output, however, because of the lack of cohesion and the abundance.

ACKNOWLEDGEMENT

Authors are grateful to Professor Stephen Kunath for his support, extensive knowledge and kindly teaching.

REFERENCES

[1] Music Information Retrieval
https://en.wikipedia.org/wiki/Music_information_retrieval

[2] Topic Model
https://en.wikipedia.org/wiki/Topic_model

[3] Daniel Waegel *A Survey of Bootstrapping Techniques in Natural Language Processing*