Sentiment Analysis for Song Lyrics

Aaron Devlin, Sonal Goswami, Alexandra Pawlak, Ming Yuan

March 8, 2018

1 Problem Description

With the common creation of song lyrics in text form for a given song, there is currently a large amount of song lyric data available for the entire world's library of songs. From these lyrics, the emotions (or sentiments) can be extracted from the words present in the song and classify its overall human feeling that is produced. There are a variety of methods to achieve this goal. In this implementation, several techniques will be weighed against one another. The repository for our project can be found at :
https://github.com/apawlak27/TCSS590/tree/master/LyricSentiment

2 Background

There have been several attempts at classifying sentiment analysis of song lyrics. The research that our team will be following is found in the following papers:

- "Identifying the Emotional Polarity of Song Lyrics through Natural Language Processing" by A. Oudenne and S. Chasins. In this work, the author's worked to classify songs as either positive or negative with several different methods. They used simple word lists with word frequencies, word dictionaries that kept count of negative and positive word occurrences, Naïve Bayes, and cosine similarity. They were able to achieve accuracies between 52% - 65.5% between the various methods. Their largest drawback was the limited amount of data that they had to work with.
- "Emotional Analysis of Songs Based on Lyrical and Audio Features" by A. Jamdar, J. Abraham, K. Khanna, and R. Dubey. In this work, both lyrics and audio features are taken into account for classification. The features include arousal, valence, POS tags, beats per minute, danceability, loudness, energy, and mode. Instead of classifying songs on a binary scale, this work classifies across a wider range such as calm, energetic, dance, happy, sad, romantic, etc. The author's used k-nearest neighbor and Euclidean distance metrics with feature weighting for classification. They were able to achieve an average of 83.4% accuracy with their model.
- "Sentiment Vector Space Model for Lyric-based Song Sentiment Classification" by Y. Xia, L. Wang, K.F. Wong, M. Zu. In this work, songs are classified as light-hearted (positive) or heavy-hearted (negative) based on sentiment analysis on the song lyrics. A sentiment vector space model (sVSM) is implement for sentiment classification. They carried out trials on audio based, knowledge based, and machine learning based models. The machine learning model using sVSM returned the best results with 78.3% accuracy.

3 Dataset

The dataset used for this project is the lyrics collection of 800 songs. Each entry is a song whose lyrics are broken down by lyric line per timestamp. The files are separated per song. The lyrics are noted as-unedited versions of the vocals. These songs are categorized as angry, happy, relaxed, and sad. There is an equal distribution of samples among these four classification labels, with 200 in each category.
https://github.com/SebinDuke/Sentiment-Analysis-of-songs-by-lyrics/tree/master/Data-Set

4 Methodology

- Cleaning the data: Tokenization, Normalization
- Basic processing of the data: Lemmatization, Stemming
- Implementation of classification model: Naive Bayes
- Lexicon analysis: Analyzing the words in the lyrics to classify them into the 4 categories
- Evaluation:
  - Precision and Recall values.
  - Comparison of our model's performance with existing analyzers.

4.1 Naive Bayes

- Input Data: Bag-Of-Words
- Train-Test Split: 80% of the songs from each category was used for training the Naive Bayes classifier. 20% of the songs from each category was used for testing.
- Validation: 5-Fold cross-validation

4.2 Lexicon-Based Sentiment Analysis (LSA)

- Custom Lexicons:
  - 4 Hand-built lexicons (Li): **[[Happy], [Relaxed], [Sad], [Angry]]**
  - WordNet synset lists created from the class labels as seed words to form each (L_i).
  - Added more synonyms as seed words where needed to approximately balance $|L_i|$, and POS.
- Custom LSA based on WPS: Based on the word similarity metric of Wu-Palmer Similarity (**WPS**). How similar two word senses are, based on the depth of the two senses in the taxonomy of the WordNet synset structure. Ranges from 0 to 1.
- Ex. **WPS**(good, bad) = 0, **WPS**(happy, happy) = 1

4.3 LSA Song Classification

Input Data: Bag-Of-Words (same as Naive-Bayes).

For each Song:
 For each (L_i) [[Happy], [Relaxed], [Sad], [Angry]]:
  Check each token in the song against each synonym word the lexicon.
   1. Calculate WPS(token, synonym), add it to count, which is the sum of WPS for the song.
   2. Normalize count, finding the total similarity of a song to that Lexicon:
    count / ($|L_i|$ * $|SongTokens|$)

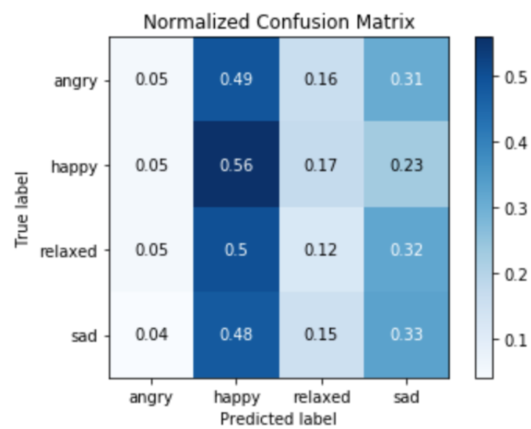
Song class prediction: Choose class label from the (L_i) that provided the max count.

	happy	relaxed	sad	angry	MaxClass	ActualClass
0	5.72393e-05	9.53798e-06	5.61101e-05	8.60387e-05	angry	happy
1	0.00023325	3.88557e-05	0.000213796	8.89259e-05	happy	happy
2	0.000275463	4.58793e-05	0.000258633	0.000164221	happy	happy
3	9.75562e-05	1.08378e-05	6.1071e-05	2.58599e-05	happy	happy

5 Results

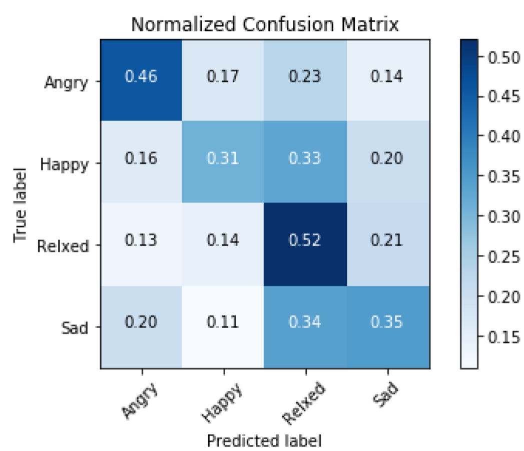
5.1 LSA

Baseline: 25%
LSA Model Accuracy: 27.24%
Angry Precision: 21.62%
Happy Precision: 29.04%
Relaxed Precision: 20.87%
Sad Precision: 28.26%
Angry Recall: 4.65%
Happy Recall: 55.82%
Relaxed Recall: 11.94%
Sad Recall: 32.66%



5.2 Naive Bayes

Baseline: 25%
NB Model Accuracy: 41%
Angry Precision: 48.7%
Happy Precision: 44.2%
Relaxed Precision: 37.1%
Sad Precision: 38.7%
Angry Recall: 46%
Happy Recall: 31%
Relaxed Recall: 52%
Sad Recall: 35%



6 Comparison

We have implemented our naive model on the dataset of another sentimental analysis project and compare our result to their results. The dataset used for comparison includes 50 years of pop music and Year-End Hot 100 on the Billboard(1965-2015). Since we are using different labels, we need to adjust mood label manually by checking the songs one by one at the very beginning. And the results are presented as follows. The first image is the result generated by our naive bayes while the last four images are results generated by the project we want to compare with.

NAÏVE BAYES

	PRECISION	RECALL	F-SCORE
HAPPY	0.95	0.65	0.77
ANGRY	1	0.8	0.89
RELAXED	0.4	0.63	0.49
SAD	0.33	0.9	0.48
AVERAGE	0.66	0.71	0.68

STOCHASTIC GRADIENT DESCENT

	PRECISION	RECALL	F-SCORE
HAPPY	1	0.25	0.4
HEARTBREAK	0.5	0.5	0.50
PARTY	0.67	0.71	0.69
ROMANTIC	0.33	1	0.50
SAD	0.4	0.67	0.50
AVERAGE	0.58	0.63	0.52

DECISION TREE LEARNING

	PRECISION	RECALL	F-SCORE
HAPPY	0.40	0.47	0.43
HEARTBREAK	0.83	0.62	0.71
PARTY	0.45	0.71	0.55
ROMANTIC	0.50	0.50	0.50
SAD	0.20	0.17	0.18
AVERAGE	0.48	0.49	0.48

K NEAREST NEIGHBORS

	PRECISION	RECALL	F-SCORE
HAPPY	0.50	0.14	0.22
HEARTBREAK	0.56	0.71	0.63
PARTY	0.67	0.40	0.50
ROMANTIC	0.25	0.50	0.33
SAD	0.50	0.75	0.60
AVERAGE	0.50	0.50	0.46

ADAPTIVE BOOSTING

	PRECISION	RECALL	F-SCORE
HAPPY	0.56	0.50	0.53
HEARTBREAK	0.67	0.43	0.52
PARTY	1.00	0.40	0.57
ROMANTIC	0.40	0.60	0.48
SAD	0.40	0.50	0.44
AVERAGE	0.61	0.49	0.51

7 Discussion and Conclusion

7.1 Conclusion

We were successfully able to implement the classification model for songs based on its lyrics in Python. Both of our models achieved results above the baseline; the precision of Naive Bayes is 41% and the precision of LSA is 27.24%.

7.2 Limitations

- Data Size: we only have 800 pre-labeled songs including train set and test set.
- Objectivity of the Data Labels: it can be very hard to decide whether a song is happy, sad or relaxed at times. The labels assigned may be subjective and inappropriate.
- Lexicon-based Approach: size and diversity of terms in each lexicon are limited.

7.3 Future Works

In the future, to improve on the accuracy, we can focus on incorporating negation, building a larger dataset, and setting more objective labels. What's more, we can also consider about adding some more types of label.

8 References

https://people.eecs.berkeley.edu/~schasins/papers/identifyingEmotionalPolarity.pdf
https://www.kaggle.com/c/movie-sentiment-analysis
https://sebastianraschka.com/blog/2014/musicmood.html
https://pdfs.semanticscholar.org/0def/091627fade56002fa3438fde488e6e6255ea.pdf

Tools:
https://nlp.stanford.edu/sentiment/code.html
http://www.nltk.org/api/nltk.sentiment.html
https://www.last.fm/api