Research

Mood Classification of Song Lyrics Using Deep Learning

Abstract

In this paper, we analyse mood classification based on song lyrics. Mood classification of lyrics can help in the creation of automatic playlists, a music search engine, labelling for digital music libraries, and other recommendation systems. We compare the accuracy of deep learning versus traditional machine learning approaches for classifying the mood of songs, using lyrics as features. We will use mood categories derived from Russell's model of affect (from psychology, where mood is represented by vector in 2D valence-arousal space). We will test the extensibility of our deep learning model using the Million Song Dataset (MSD), a freely available contemporary music track dataset. From MSD, we will use the Last.fm dataset for song tags and lyrics. We will also use language detection to focus on English lyrics. We measure the effectiveness and accuracy of different machine learning algorithms on mood classification, including Naive Bayes, SVM, and Convolutional Neural Network models. Our results indicate optimal performance using our neural network versus the other models.

Word Count: 2755

1. INTRODUCTION

Music recommendation engines are becoming more and more of an expected norm for listeners and music lovers today. Since mood and emotional classification is crucial to the entertainment value of songs, music recommendation algorithms can utilize mood classification. In circumstances where audio data is not readily available, having strong lyrics-based classification will be particularly helpful.

Song lyrics are different from ordinary text in that they often use more stylistic qualities, like rhyming and other forms that contribute to the emotional value (Lee et al., 2010). They are also often shorter and have smaller vocabularies than other text documents. This, combined with their more poetic style, can cause more ambiguity when it comes to mood identification (Cano et al., 2017).

Much research on music classification is based on audio analysis versus lyric analysis. (Corona et al., 2015). However, classifiers that incorporate textual features can outperform audio-only classifiers (Fell et al., 2014). In this paper, we focus on the analysis of lyrics for classifying mood categories. We use mood and emotion as interchangeable concepts. Classifying mood (or "sentiment") using textual features has been studied less than musical features. One reason may be that obtaining a large dataset legally is difficult (as lyrics are copyrighted material).

In this paper, we measure the effectiveness and accuracy of different machine learning models on mood classification, including Naive Bayes, SVM, and Convolutional Neural Network (CNN) models. We use the Million Song Dataset (MSD), a large freely-available dataset and supplement it with lyrics scraping we conducted separately. Our results indicate optimal accuracy using our CNN.

Our paper is organized as follows: Section 2 reviews related work on lyrics-based mood classification. Section 3 outlines the methods (design and implementation) used in our analysis. Section 4 discusses results, including a detailed analysis of obtained results. Conclusion and recommendations are summarized in Section 5.

2. BACKGROUND AND LITERATURE REVIEW

Previous work reached contradictory conclusions and employed smaller datasets and simpler methodology like td-idf weighted bag-of-words to derive unigrams, fuzzy clustering, and linear regression. The approach of past research on musical mood classification can be grouped into three categories: classification using joint audio-lyric data, using only lyric data, and using only audio data. One study, using an SVM model, compared the effectiveness of each method and found using joint audio-lyric data as the most accurate. However, their dataset was small (100 songs) and employed line-by-line classification rather than whole song (Mihalcea et al., 2012).

Our focus on lyrics and not audio is akin to that of music classification research performed by Corona and O'Mahony which showed that lyrics alone can be used to classify music mood achieving an accuracy of up to 70% for a single mood with the MSD (2015). In this study, SVM classification algorithms were used, and some moods (e.g., anger) were found to be easier to detect. Our choice of using a convolutional neural network is based on past research demonstrating that CNN-based systems are effective and successful for sentiment analysis (Kim, 2014).

Though our focus is on English language lyrics, there have been studies conducted on lyrics in other languages, such as Chinese (Chen et al., 2009) and Hindi (Bandyopadhyay et al., 2015), that also used mood affects predicted based on combinations of text stylistic features and features based on n-grams (e.g., term frequency-inverse document frequency scores of trigrams).

A recent study on sentiment analysis of lyrics showed that music corresponding to happy moods can be predicted with greater accuracies (Raschka, 2014). This study used a Naive Bayes classifier trained on lyrics alone, also pulled from the MSD. Specifically, their best performing model was a Multinomial Naive Bayes classifier with unigram tf-idf feature representation, better than the Bernouilli Naive Bayes model. Increasing the n-gram range and other tuning such as parameter smoothing had little effect on classification performance.

3. METHODOLOGY

3.1 Data Acquisition

The MSD provides mood mapping at a song but not lyric level. A component of the MSD includes the Last.fm dataset, which maps songs to user "tags" based on Last.fm user input. These tags can be anything from a human emotion to genre to animals. The dataset contains song-level tags for more than 500,000 songs. The mood categories are derived from these tags as proposed by Laurier (2009). To account for inconsistency and variants, we use approximate matching. For example, tags in an approximate match for aggression might include: 'aggressive depression', 'calm song aggressive voice', 'chill-to-aggressive', 'for aggressive days', 'makes me aggressive', 'melodically aggressive', and 'phil-aggressive'.

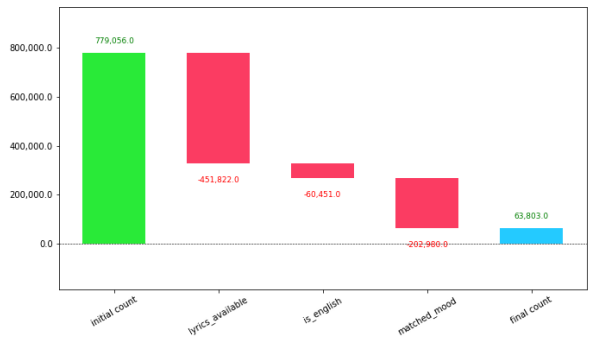

Figure 1. Count of songs removed from dataset by each filter

One initial challenge was in acquiring corresponding lyrics to song data. The MSD provides lyrics for some songs in a bag-of-words format courtesy of MusixMatch, but for our purposes we needed the original lyrics of each song. To obtain the 63,803 lyrics in our dataset, we used the python package lyricsgenius (interfaces with www.genius.com) for retrieving lyrics, From the MSD, we matched 327,234 songs to lyrics. For our analysis, we filtered out non-english songs and those for which we could not match the mood. Figure 1 shows the filters used and counts.

3.2 Mood Annotation

We use the popular Russell's model of affect (1980) and focus on 18 mood classes similar to previous studies (Chen et al., 2009). In Russell's model, mood is represented by a vector in the 2-D valence-arousal space where mood is an element of (valence, arousal):

$$m \in M = (v, a) \quad (1)$$

Valence measures the good vs bad dimension (pleasant / unpleasant). Arousal measures the active vs passive dimension of sentiment (aroused / sleepy). Mood in our model refers to categories to be learned in the classification problem. We group mood into 18 categories and match tags to moods. We also experiment with clustering the 18 moods into four quadrants as proposed by Laurier: happy, sad, anger, and calm (2009). Our categories are shown in Table 1.

Mood Quadrant	Associated Moods
happy	cheerful, desire, excitement, romantic, upbeat
sad	depressed, grief, pessimism
anger	aggression, angst, brooding
calm	confident, dreamy, earnest

Table 1. Mood Model. Each quadrant consists of its identifying mood and subsequent associated moods.

3.3 Mapping Labels

To map each song to its mood, we make use of the sqlite db version of the Last.fm dataset. We query for all associated tags of each song, then attempt to match tags against our mood categories with substring matching. For some moods, we use an altered version of the word, like 'aggress' for aggression to account for different word forms. We then apply a series of filters, different for each mood, derived by manual inspection of the matched tags. These filters remove nonsensical matches like the tag 'not calming' for the mood calm or 'unhappy' for the mood happy. Substring matching expanded our set of mood-labeled songs by a factor of two over full string matching (i.e. only match mood sad with the exact tag 'sad').

3.4 Data Balancing

After acquiring and labeling the data, we discovered that the distribution over mood categories was quite unbalanced as demonstrated by Figure 2a.

To balance the distribution, we employ two methods. The first, as mentioned prior, is grouping related-moods into four quadrants. The second is oversampling the underrepresented moods by copying and shuffling each song's lines and then appending them to the dataset with the goal of equal representation. To avoid uneven

distributions, smaller categories had each song in the category shuffled and appended. At most, the difference in quantity between the most copied/shuffled song and the least is 1 duplication. Both methods resulted in an increase in accuracy of our final model. Figure 2b demonstrates the quadrant method, and Figure 2c demonstrates the quadrant and oversampling method.

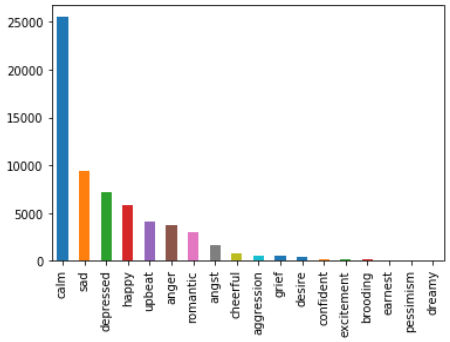

Figure 2a. Unbalanced Song Mood Distribution, 18 moods

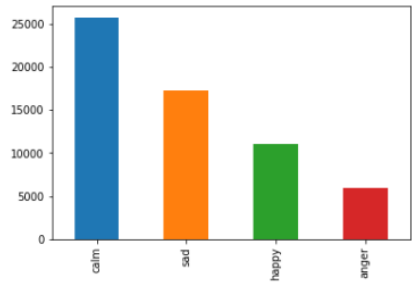

Figure 2b. Unbalanced Song Mood Distribution, 4 mood quadrants

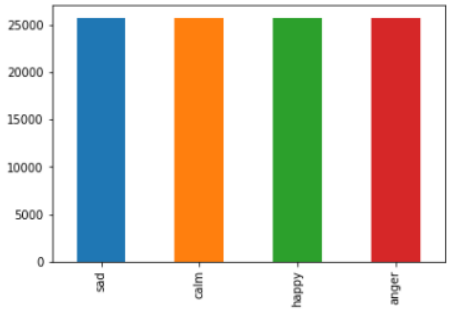

Figure 2c. Balanced Song Mood Distribution, 4 mood quadrants

3.5 Word Embeddings

To form our word embeddings, we make use of the word2vec model (Mikolov et al., 2013) and the implementation provided by TensorFlow.[1] We use a vocabulary generated from the top most common 10,000 words of our full set of mood-labeled lyrics to train a skip-gram word2vec model with a window size of four and an embedding size of 300. We experimented with larger vocabulary settings of 20,000 and 50,000 but found no significant difference in accuracy.

Close relationships of note include 'little' and 'girl', 'oh' and 'yeah', and 'hey' and 'baby'. These pairs show the accuracy of the embeddings as well as the unique nature of a song lyric corpus that contains more slang and colloquial phrases than one might expect out of a more official and formal corpus like news articles or speeches.

3.6 Baseline Mood Classification

To establish a baseline classification accuracy, we use the most-common-case (MCC) classification approach and two supervised machine learning algorithms, Multinomial Naive-Bayes (NB) and Support Vector Machines (SVM), both from sklearn. As input, we split our dataset into training, dev, and test sets with an 80, 10, 10 split.

For the machine learning approaches, we transformed the lyrical text into tf-idf vectors. The use of SVMs have been well explored in this area with heavy feature engineering as mentioned in Section 2. Our baseline SVM does not include these advanced feature engineering techniques.

3.7 Deep Learning Classification

For our deep learning classifier, we use a Convolutional Neural Network (CNN) implemented in TensorFlow.[2] To prepare for training, we normalize all songs to a length equal to the 75th percentile of our dataset's song length, remove stopwords that match from within nltk's stopwords corpus, remove punctuation, and tokenize lyrics with nltk's word_tokenize function (we also experimented with nltk's WordPunctTokenizer but observed no noticeable improvement).

Our CNN is configured with an Adam optimizer (initialized with TensorFlow's default initialization parameters) and hyperparameters tuned from preliminary experimentation on the dev dataset with input from literature mentioned in Section 2 (filters of size three, four, and five with 300 filters each, dropout of 0.8, and an L2 regularization value of 0.01). We experimented with additional optimizers, Adadelta and Adagrad, but they proved prone to overtraining and less accurate.

Our CNN for mood classification of song lyrics uses an embedding layer (embedding size of 300), followed by a convolutional, max-pooling and softmax layer. We calculate mean cross-entropy loss and accuracy in our training.

Finally, we experiment with two different embeddings inputs: random initialization with on-the-fly model training and, as mentioned above, word2vec.

4. RESULTS AND DISCUSSION

Our results show that the highest accuracy for each classifier is found with the Balanced Mood Quadrants dataset as seen in Table 2. This is not surprising as the dataset is larger due to oversampling and all of the categories are equally represented.

Model	Unbal. Mood	Unbal. Mood Quadrs.	Bal. Mood Quadrs.
MCC	39.81%	43.61%	25.34%
NB	39.93%	46.78%	55.19%
SVM	44.88%	50.95%	54.07%
CNN w2v0	**56.79%**	62.15%	**77.08%**
CNN w2v1	54.33%	**63.53%**	75.45%

Table 2. Model Accuracies. w2v0/1 means with/without word2vec embeddings

Mood	Unbal. Mood	Unbal. Mood Quadrs.	Bal. Mood Quadrs.
anger	39%	50%	92%
happy	47%	61%	81%
sad	43%	60%	68%
calm	67%	66%	68%
dreamy	nan		
upbeat	62%		
angst	60%		
excitement	50%		
cheerful	49%		
romantic	49%		
depressed	46%		
grief	44%		
confident	44%		
aggression	37%		
earnest	33%		
brooding	26%		
desire	20%		
pessimism	18%		

Table 3. CNN w2v0 Mood F1 Scores; quadrant datasets only have 4 moods

The best accuracy overall is 77.08% and is produced by our CNN w2v0 model. It beats our w2v1 model by 1.63%. We attribute this to the embedding training that happened alongside the CNN's training. Future work could explore optimizing word2vec's training parameters for increased performance or replacing word2vec with another embedding algorithm such as doc2vec.

The most performant machine learning classifier is NB also in the Balanced Mood Quadrants dataset with 55.19%. This suggests that NB performs better with balanced data as SVM outperforms NB in the two unbalanced datasets.

In Table 3, we present the classification F1 scores for each mood with our CNN w2v0 model. The highest F1 scores come from the Balanced Moods Quadrants dataset with anger and happy. These two quadrants were also the smallest prior to balancing implying that too much oversampling might lead to overtraining. A larger, more diverse dataset would resolve this.

Table 3 also highlights the benefit of moving from the full set of 18 moods to the 4 mood quadrants. The representation of some moods like pessimism or desire in the Unbalanced Moods dataset is too small to adequately train a classifier. By grouping the moods into quadrants, our classifier's performance vastly improves and still produces interesting and applicable results

Finally, in Figure 3, we share the confusion matrix of our best model. Calm and sad are the two most confused quadrants. This is not unreasonable as, of the four, it can be argued that they are the most similar.

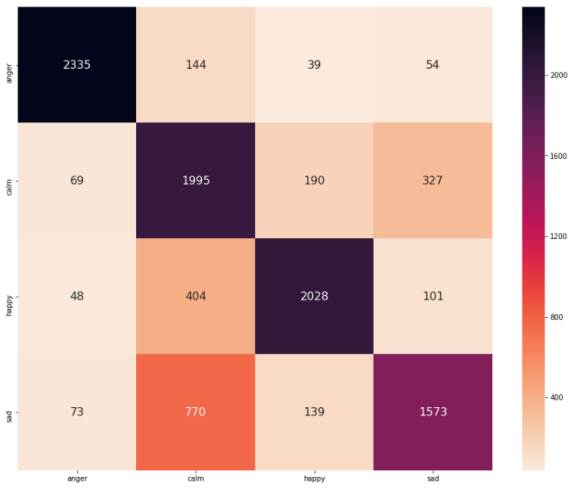

Figure 3. CNN w2v0 Bal Mood Quadrs Confusion Matrix[3]

5. CONCLUSION AND RECOMMENDATIONS

This paper presented an approach for and the result of using deep learning for mood classification of song lyrics. We compared results across four classifiers: Most-Common-Case, Naive Bayes, SVM, and our Convolutional Neural Network (CNN). The results indicated that the best classifier is our CNN w2v0, which achieved an accuracy of 77.08% on our Balanced Mood Quadrants dataset, a full 21% higher than our most performant machine learning classifier.

Lyrics-based mood classification is just beginning. As more music moves online, intelligent and accurate music recommendation systems will be greatly enhanced using lyrics to aid prediction. As larger datasets become more readily available, the real-life applicability of these algorithms will become easier to vet.

Our work and datasets are available online and open-sourced.[4] We invite others to build upon our work and improve our findings.

6. NOTES

1 Link to word2vec implementation: https://www.tensorflow.org/tutorials/representation/word2vec
2 Link to CNN TensorFlow implementation: http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/
3 Link to Seaborn heatmap: https://seaborn.pydata.org/generated/seaborn.heatmap.html
4 Link to project source repo: https://github.com/workmanjack/lyric-mood-classification

7. REFERENCES

Patra, B.G., Das, D., & Bandyopadhyay, S. (2015). Mood Classification of Hindi Songs based on Lyrics. *ICON*.

Bertin-Mahieux, T., Ellis, D.P., Whitman, B., & Lamere, P. (2011). The Million Song Dataset. *ISMIR*.

199ano, E., & Morisio, M. (2017). MoodyLyrics: A Sentiment Annotated Lyrics Dataset. *ISMSI '17*.

Hu, Y., Chen, X., & Yang, D. (2009). Lyric-based Song Emotion Detection with Affective Lexicon and Fuzzy Clustering Method. *ISMIR*.

Corona, H., & O'Mahony, M.P. (2015). An Exploration of Mood Classification in the Million Songs Dataset.

Dodds, P.S., & Danforth, C.M. (2009). Measuring the happiness of large-scale written expression: Songs, Blogs, and Presidents. *CoRR, abs/1703.09774*.

Hu, X., Downie, J.S., & Ehmann, A.F. (2009). Lyric Text Mining in Music Mood Classification. *ISMIR*.

Fell, M., & Sporleder, C. (2014). Lyrics-based Analysis and Classification of Music. *COLING*.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *EMNLP*.

Laurier, C., Sordo, M., Serrà, J., & Herrera, P. (2009). Music Mood Representations from Social Tags. *ISMIR*.

Yang, D., & Lee, W. (2009). Music Emotion Identification from Lyrics. *2009 11th IEEE International Symposium on Multimedia, 624-629*.

Wang, S.I., & Manning, C.D. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. *ACL*.

Mihalcea, R., & Strapparava, C. (2012). Lyrics, Music, and Emotions. *EMNLP-CoNLL*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *NIPS*.

Raschka, S. (2016). MusicMood: Predicting the mood of music from song lyrics using machine learning. *CoRR, abs/1611.00138*.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology, 39*, 1161-1178.