Convolutional Neural Networks for Lyrical Sentiment Analysis

COMP 550: Fall 2019

Group 26:
Jacob Sanz-Robinson (260706158)
Dorian Desblancs (260722712)
Jonah Nimijean (260698160)

Abstract

Emotion classification in music is a common problem with numerous applications. With the rise of streaming services, creating effective song recommendation systems is of interest to companies that wish to satisfy users by suggesting music which elicits the emotional qualities they desire. Classifying users' mood preferences allows advertisers, users, and streaming services to customize and optimize content and advertising strategies. Traditionally, the focus of papers in the field consists of classifying audio data using signal processing techniques. Comparatively fewer papers exist on sentiment classification applied to lyrics to help identify the underlying emotional quality, even though they often outperform signal processing-based methods. In this paper we present a simple Convolutional Neural Network (CNN) which uses pre-trained GloVe embeddings to classify emotions in lyrics. Our novel approach to this problem surpassed the literature baselines on the MoodyLyrics dataset, achieving 90.8% accuracy on the multiclass version, and 92.5% accuracy on the binary version.

1 Introduction

The vast majority of music is nowadays consumed electronically via streaming services like Spotify, Apple Music, and Google Music, to name a few. Recommendation engines are a vital component of these companies' business models, as they facilitate the retention of consumer interest in their respective platforms. Current recommendation engines appear to be based primarily on users' individual listening patterns, taking into account specific songs as well as genres [1]. Given the vast data resources available to these larger streaming services, it would be an interesting (and perhaps profitable) area of research to incorporate the listener's mood into these recommendation algorithms. This area of research is also of particular interest to our group, as all three of our members have either completed or are enrolled in the Musical Science and Technology minor here at McGill.

The present paper focuses on the analysis of song moods, using machine learning and NLP techniques to analyze the lyrical content of songs. For the purposes of this exploration, songs are categorized as either Happy, Angry, Sad, or Relaxed, based on lyrical valence and arousal. Both binary and multiclass classifiers based on our proposed CNN architecture are trained using a modified subset of the MoodyLyrics dataset [2], which contains 2000 examples in total. The binary model is trained with data labelled only with valence values, and outputs a positive or negative sentiment classification; conversely, the multiclass model is trained with data labelled with both valence and arousal values, and contains 500 examples from each of the four classes. The accuracy of the binary and multiclass models on held-out test sets are found to be 92.5% and 90.8%, respectively. The generalizability of the trained multiclass model is then tested on a new dataset, the SebinDuke dataset on GitHub [3], on which it achieves an accuracy of 75.2%.

2 Related Work

Text classification and sentiment analysis are fields of machine learning and natural language processing that have been active for decades. One of the first attempts was [4], which shows that an unsupervised classifier can achieve 74% accuracy

on the task of rating reviews positively or negatively. Since then, the field has become increasingly popular. [5] showed that Support Vector Machines (SVM) were more robust than logistic and ridge regressions in text classification purposes, especially under noisy conditions (such as incorrect labels), or datasets with unbalanced classes, both of which are issues pertinent to this project, and [6] highlights various works from the early 2000's showing the successes of SVMs used in conjunction with ensemble methods such as boosting. In 2008, there was a rise in popularity in the use of Naive Bayes classifiers, which are typically successful in text and sentiment classification systems [7]. In 2012, the seminal "Bigrams and Baselines" paper showed that bigrams improve the performance of sentiment classification systems, having greater sentimental effects than individual words when used as features [8].

This project borrows heavily from the methodology outlined in the seminal paper by Yoon Kim [9]. Kim uses a simple Convolutional Neural Network (CNN) to classify sentences in 7 datasets. On four of these datasets, Kim beat state of the art baselines. In terms of the implementation, the paper uses padding, ensuring the input sentences are of the same length, and then maps the words in the resulting padded sentences to word embeddings using word2vec [10], which produces a 300-dimensional vector describing the closeness of words in the corpus (based on a pre-training on words from Google News).

Sentiment classification applied to music is not a novel idea. However, the focus of most of the papers in the field consists on classifying audio data for 'music information retrieval', combining signal processing and machine learning techniques [11][12]. Comparatively fewer papers exist on sentiment classification applied to lyrics to help identify the underlying emotional quality [11], and various references claim they outperform signal processing-based methods [11][13].

[13] uses a combination of 25 text-based features (amongst them n-grams and WordNet-based features) alongside audio features to achieve a 74% accuracy on 18-class classification with an SVM. [14] combines features such as the word count, character count, line count, and tf-idf, and trains/tests an SVM on a 10000 song proprietary database, obtaining 77% accuracy on the same 4 categories we use in this project. More recently, [15] also uses a SVM (with a large variety of different features) on our same 4 categories to obtain a 74% accuracy on the task. The paper introducing MoodyLyrics, the dataset we used for this project, achieves 74% accuracy on the dataset in multiclass classification using a SVM [2].

3 Datasets

The MoodyLyrics dataset consists of 2595 song lyrics annotated using four categories [2]. These categories are based on Russell's Valence-Arousal plane, where Valence refers to positive or negative intensities and Arousal refers to levels of emotion activation. The plane splits emotions into four global emotions: happiness, sadness, relaxedness, and anger. These were used to annotate all songs in the dataset into four quadrants [16]. Additionally, all songs are also annotated in binary fashion using Valence only. Here, labels are only positive and negative, and are more similar to classical sentiment classification.

One issue we noticed in Yoon Kim's 2014 original sentiment analysis CNN paper [9] is that while the experiments featured in the paper are evaluated by their accuracy, the samples in the datasets are not uniformly distributed across all of the classes. In the binary MPQA dataset, for example, 68% of the data is labelled as 'negative.' As we know, accuracy is not an appropriate metric to evaluate model performance on unbalanced datasets. After all, a model can achieve 99% on a dataset where 99% of the data is labelled as positive by always predicting a positive label. We therefore opted to use balanced versions of the MoodyLyrics dataset containing 2000 songs, where each label was represented in equal proportion. This data is split: 70% of it is used to train the model and 30% to test it. All lyrics in the MoodyLyrics dataset had to be extracted from the LyricsWiki website [17].

4 Proposed Approach

As is described in the Related Works section, there is limited literature specifically dealing with lyrical sentiment analysis. The papers in the field obtain similar accuracies, using methods that are based on variations of common text features and SVMs. We haven't found any literature on using CNNs for lyrical sentiment analysis, despite the fact that CNNs seem like a learning model which would be well suited to the task at hand for a

number of reasons. Firstly, as Kim's 2014 paper, and subsequent papers in the field have shown, CNNs are state-of-the-art at classifying emotions in text, outperforming SVMs in many cases. Secondly, we know CNNs are good at extracting local and position-invariant features, like keywords and phrases to detect sentiment. Therefore, we hypothesize that the repetitive and often simple structure featured in popular music lyrics would be useful to highlight these local features (such as repeated choruses and words), and is a characteristic that could benefit our mood classifications.

Based on the success of pre-trained word embeddings in sentiment analysis tasks (including Kim 2014's paper), and their benefit to small datasets, our CNN is built based on the popular GloVe pre-trained word embeddings. GloVe "showcases interesting linear substructures of the word vector space" by encoding meaning as vectors in this embedding space, based on word co-occurrence statistics from a corpus [18].

Minimal preprocessing has to be performed on the MoodyLyrics dataset. For each song, the lyrics are tokenized, converted to lowercase, and punctuation and non-alphabetic symbols are removed. The song's lyric tokens are then padded to be of the same length as the longest lyric in the dataset.

The 100-dimensional pre-trained GloVe embedding (trained on 400,000 words from a 2014 dump of English Wikipedia) is loaded, and is then split into a tokenized vector, so a dictionary can be created mapping the words to their coefficients. These mappings are used to build an embedding matrix (with the coefficients for every word in the embedding). This matrix is used as non-trainable initial "embedding layer" of the CNN: a lookup table to map the word to its vector representation.

Finally, we construct our CNN models. The models are built in Keras [19], make use of the ADAM [20] optimizer, and use cross entropy for the loss function. In terms of architecture, the multiclass model receives inputs to the previously described embedding layer, and is then immediately subjected to 0.2 dropout to improve generalization. This feeds into a 1D/temporal convolution layer with 64 filters, a kernel size of 5, and ReLU activation. The output of this layer undergoes global max pooling, and is outputted to two dense layers with Relu activations (one with 16 nodes and the one after with 8), and finally a 4-node softmax layer. The binary version only differs in the output being a single sigmoid-activation node.

5 Results

For this experiment, the binary and multiclass models based on our proposed CNN architecture were each trained for 10 epochs on 70% of the MoodyLyrics dataset (constituting 1400 training examples), and subsequently tested on the remaining 30% (constituting 600 test examples). The accuracies achieved on this test set by our proposed models, as well as the current benchmark accuracies from the literature, are reported below in Table 1. Interestingly, our multiclass model appears to have outperformed the one discussed in the literature by a significant margin. The amount of training necessary to achieve these results is also surprising; plotting the model's test accuracy in relation to the number of epochs reveals that performance started to plateau after approximately 6-7 training epochs. It can further be observed that the binary model was able to achieve higher accuracies using fewer training epochs than the multiclass model, crossing 90% accuracy on the test set after only 3 epochs.

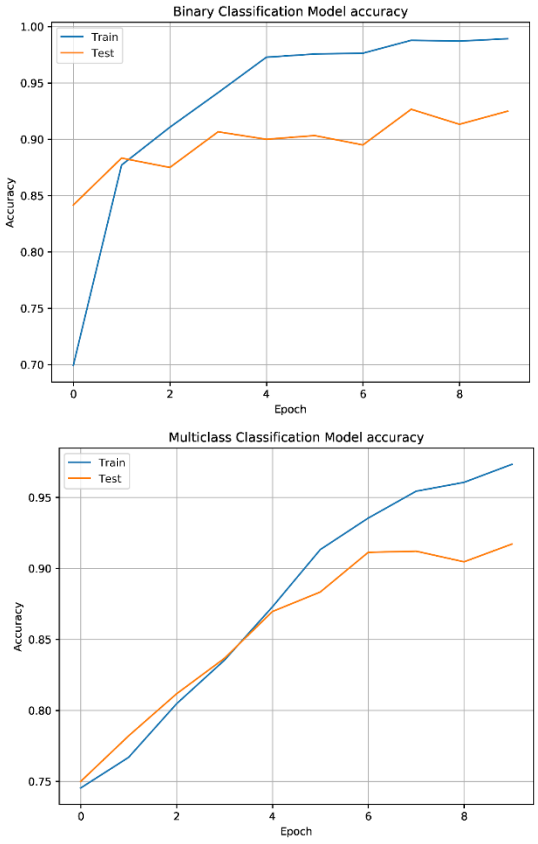

Figure 1: Plots of Model Accuracy vs Training Epoch, Binary (above) and Multiclass (below)

Finally, to ensure our final model generalized beyond the MoodyLyrics dataset, we found a dataset which also contained labelled lyrics according to Russell's mood model, uploaded by the user SebinDuke on GitHub [3]. This dataset consists of 778 labelled songs, which the uploader achieved 60% accuracy on using a Multinomial Naive Bayes model (MNB), with lemmatization, unigrams, bigrams, and trigrams. We performed no training on this dataset, and used it exclusively to further test out multiclass CNN trained on MoodyLyrics, obtaining an accuracy of 75.2%. We discuss this result further in the discussion.

	Our CNN	Past Benchmark
MoodyLyrics Binary	92.5%	N.A.
MoodyLyrics Multiclass	90.8%	74.3% (SVM)
SebinDuke Multiclass	75.2%	60.0% (MNB)

Table 1: Accuracies of our proposed CNN vs Literature Benchmarks

6 Discussion

One factor offering a potential explanation for the effectiveness of this classification technique is the limited vocabulary used in song lyrics. The dataset contains a total of 13898 unique word tokens, split across 2000 sets of song lyrics; there are thus ~7 unique word tokens per set of song lyrics on average. In contrast, the average set of song lyrics is approximately 112 word tokens in length, excluding stopwords. This large difference in the average number of unique tokens and average length per set of lyrics is indicative of a heavily repetitive nature in lyrical vocabulary. The limited vocabulary used to express the moods of each song can thus be learned effectively, allowing for the model to achieve high accuracies.

This factor may also partially explain why certain moods are more likely to be misclassified with each other than others: it is possible that the same subset of words could be used to describe two different moods, and songs containing a sufficient amount of the overlapping vocabulary would be consequently misclassified. Analysis of the multiclass classifier's confusion matrix (Fig. 2) reveals that angry moods are most likely to be misclassified as sad, and vice-versa. In a similar way, relaxed-mood songs are most likely to be misclassified as happy; interestingly, however, this relationship is not reciprocal, and may be indicative of a subtle bias (in either the data or the model) towards predicting happy instead of relaxed. Since the angry/sad and happy/relaxed mood pairings share negative and positive valences respectively, these groupings appear to be based primarily on the valence similarity of each member word. This is further evidenced by the lack of confusion within the happy/sad mood pairing, which are dissimilar in terms of both valence and arousal.

On the other hand, the binary classification model's confusion matrix (Fig. 2) indicates that the model is liable to Type 1 and Type 2 errors at approximately the same rate; the model thus does not appear to suffer from any kind of systemic bias induced by the data or model. These erroneous predictions made by the model are likely caused by songs which contain lyrics of both positive and negative valence, and in turn lie very close to the model's decision boundary.

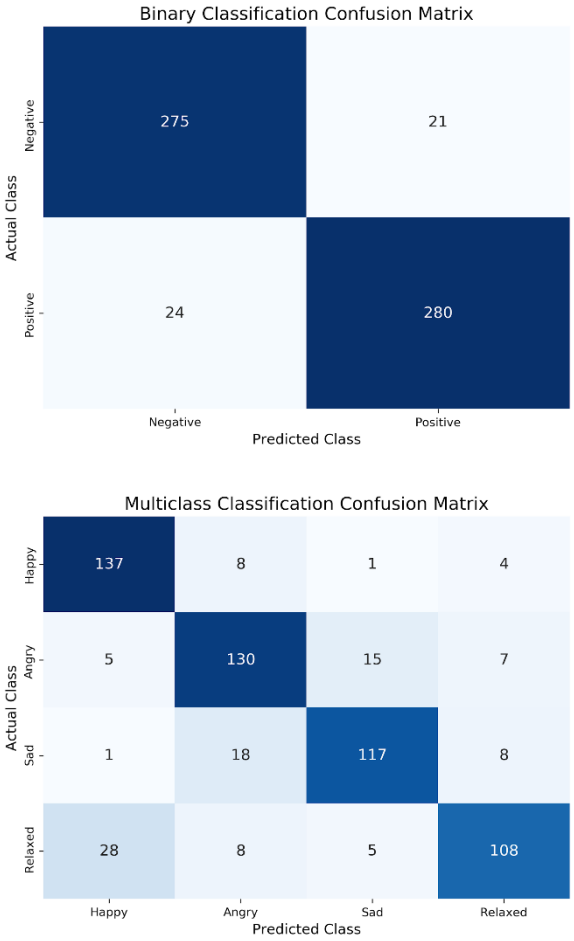

Figure 2: Confusion Matrices for the Binary (above) and Multiclass (below) Classification models

Finally, our CNN's 75.2% accuracy on SebinDuke's dataset show that our multiclass model successfully generalizes to an entirely new

dataset. The accuracy is not as high as it is for the MoodyLyrics dataset, which could be due to a number of reasons. Mainly, it is evident upon inspection that the dataset is not well cleaned, and includes encoding errors and unknown characters from the conversion of its original CSV format. Even though we attempted to clean the dataset up manually, not all the data could be cleaned, and we suspect this is the reason for the decrease in accuracy compared to the model's performance on MoodyLyrics.

7 Future Work

Although our CNN models attain high accuracies, there is more work that can be done to improve them. Firstly, MoodyLyrics is a small dataset, and using a larger dataset would potentially lead to better model generalization. It would also be interesting and useful to construct a more nuanced overview of sentiment by deriving more sensitive measures than valence and arousal, allowing for the discernment of more numerous distinct moods (ex. excited, romantic, lonely, etc.), or accounting for the intensity of classified emotions. Perhaps most interesting would be to see the result of ensembling our model with a more traditional existing music emotion classification system based on audio signal processing, rather than text.

8 Conclusion

In this project we built binary and multiclass CNN models based on GloVe word embeddings, which are capable of performing sentiment classification on song lyrics. Our models obtain a high accuracy, namely 92.5% on the binary MoodyLyrics dataset, and 90.8% on its multiclass counterpart. Our multiclass model also generalizes to the SebinDuke GitHub dataset with 75.2% accuracy. We were pleasantly surprised to find our multiclass model surpassed the literature baseline on the MoodyLyrics dataset.

Statement of Contributions

Jacob Sanz-Robinson: CNN model, Abstract, Related Work, Proposed Approach, Conclusion.
Dorian Desblancs: Data preprocessing, Dataset mining and generation, Data cleaning.
Jonah Nimijean: Introduction, Results, Discussion, Future Work, Generation of figures.

References

[1] Spotify , "How Does the Spotify Algorithm Work?", 29 Dec. 2018, community.spotify.com/t5/Social/How-does-the-Spotify-algorithm-work/td-p/4642217. Accessed 9 Dec. 2019.

[2] Çano, Erion; Morisio, Maurizio (2017). MoodyLyrics: A Sentiment Annotated Lyrics Dataset. In: 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence, Hong Kong, pp. 118-124

[3] SebinDuke, "SebinDuke/Sentiment-Analysis-of-Songs-by-Lyrics." GitHub, 2 May 2019, github.com/SebinDuke/Sentiment-Analysis-of-songs-by-lyrics. Accessed 9 Dec. 2019.

[4] Turney, P. D. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. CoRR, cs.LG/0212032 . http://arxiv.org/abs/cs.LG/0212032

[5] Zhang, J., & Yang, Y. (2003). Robustness of regularized linear classification methods in text categorization. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval , SIGIR '03, (pp. 190–197). New York, NY, USA: ACM. http://doi.acm.org/10.1145/860435.860471

[6] StanfordNLP (2009). Stanford: Support vector machines. https://nlp.stanford.edu/IR-book/html/htmledition/references-and-further-reading-15.html

[7] Brmez, S. (2016). Analysis of complex sentiment on social networks. https://pdfs.semanticscholar.orga943/8f3a80e4ab5ab69c107da23c85b253f6fe5d.pdf

[8] Wang, S., & Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2 , ACL '12, (pp. 90–94). Stroudsburg, PA, USA: Association for Computational Linguistics. http://dl.acm.org/citation.cfm?id=2390665.2390688

[9] Kim, Yoon. "Convolutional Neural Networks for Sentence Classification." Proceedings of the 2014 Conference on Empirical Methods in Natural

Language Processing (EMNLP), 2014, doi:10.3115/v1/d14-1181.

[10] Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). Efficient estimation of word representations in vector space. CoRR, abs/1301.3781 .

[11] Xia, Yunqing, et al. “Sentiment Vector Space Model for Lyric-Based Song Sentiment Classification.” International Journal of Computer Processing of Languages, vol. 21, no. 04, 2008, pp. 309–330., doi:10.1142/s1793840608001950.

[12] Beveridge, Scott. “A Feature Survey for Emotion Classification of Western Popular Music.” 9th International Symposium on Computer Music Modeling and Retrieval (CMMR), 2012, pp. 508–517.

[13] Hu, Xiao & Downie, J. (2010). "When Lyrics Outperform Audio for Music Mood Classification: A Feature Analysis". ISMIR, pp. 619-624.

[14] Zaanen, M.V., & Kanters, P. (2010). "Automatic Mood Classification Using TF*IDF Based on Lyrics". ISMIR.

[15] Malheiro, Ricardo, et al. “Emotionally-Relevant Features for Classification and Regression of Music Lyrics.” IEEE Transactions on Affective Computing, vol. 9, no. 2, Jan. 2018, pp. 240–254, doi:10.1109/taffc.2016.2598569.

[16] J. Russell. A circumplex model of a_ect. Journal of personality and social psychology, 39(6): pg. 1161-1178, 1980

[17] “LyricWiki.” Fandom.Com, 2019, lyrics.fandom.com/wiki/LyricWiki. Accessed Nov-Dec. 2019.

[18] Pennington, Jeffrey. “GloVe: Global Vectors for Word Representation.” Stanford.Edu, 2014, nlp.stanford.edu/projects/glove/. Accessed Nov-Dec. 2019.

[19] Charles, P.W.D. “Keras-Team/Keras.” GitHub, 2013, github.com/keras-team/keras. Accessed Nov-Dec. 2019.

[20] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.