Tracking the Evolution of Underground Music Culture via Deep Sentiment Analysis

Christian Brown[#1]

[#]*Energy Resources Engineering Department, Stanford University*
367 Panama St, Stanford, CA 95304

[1]csbrown@stanford.edu

***Abstract*—Resident Advisor (RA) music reviews have been describing and critiquing the latest track releases in the underground music scene since 2001. By understanding which words or phrases are given more weight when predicting a review's sentiment (positive/negative), and whether these words or phrases are valuable for predicting the timeframe (eras) of track release, interpretations can be made on how the underground music community's preference for specific musical elements have changed over the last 20 years. By assessing how the cosine similarities between LSTM cell activations and the final cell activation change over the span of an input sequence, keywords in the model prediction process were identified. This was accomplished by examining the derivatives of the resulting similarity plots. An analysis of the keywords revealed that words describing positive sentiment towards underground music have evolved over time, whilst those describing negative sentiment have not.**

I. Introduction

Resident Advisor[1] (RA) is a website that has been sharing and publishing information about the underground music scene since 2001. One form on content that is regularly published on RA is music reviews of tracks and albums from both rising stars and established producers. As of late 2017, approximately 17,000 of these reviews have been published. The reviews have been scraped and published on Kaggle[2] in a format that lends itself to deep natural language processing (NLP). In addition to the text of the review, other fields of interest for each review include year of release and a 0-5 rating from the reviewer.

Unlike restaurant or product reviews, RA music reviews attempt to convey the state of mind of the listener. As an example, below is an excerpt from one of the many reviews:

"... 'Heiligendamm' takes a more decisive route towards the dancefloor with earthy drums leading to a rollicking bassline and classic Detroitesque piano stabs. The melodic techno ..."

Reviewers utilize a wide variety of highly descriptive words and phrases used in unorthodox ways in an attempt to capture their abstract experiences.

A model that is effectively able to predict the sentiment of the review is able to discern which words and phrases can be used to uniquely identify positive or negative musical qualities. Additionally, investigating how these words and phrases change over time can shed light on how the underground music scene community's sentiment towards certain musical qualities has changed over time.

The proposed approach is to train two separate classifiers that predict sentiment and release timeframe, respectively. Following this, keywords will be extracted via analyzing the similarity of activations between each cell and the final cell's activation for a given review. Inputs that result in large changes in similarity are considered keywords. These keywords are then analyzed to assess how sentiment towards specific musical qualities in RA reviews have changed over time.

Section II will provide a high-level overview of some of the previous literature on sentiment analysis and keyword identification using deep learning. Section III will discuss the dataset used for this analysis, in addition to the pre-processing that took place prior to feeding the inputs to the model. Section IV will describe the model architecture and methods for keyword extraction that were used. Section V outlines the final results obtained and the keywords identified to be unique for each sentiment-era subclass. Finally, Section VI will offer conclusions and future directions for this work.

II. Related Work

Although there is limited literature on sentiment analysis for published reviews about music, there have been numerous

studies done on sentiment analysis for products, or the interpretability of RNNs.

Glorot et al. wrote in 2011 of the ability of deep neural networks and their ability to discern language features from a product review that would be useful for sentiment classification[5]. In a 2017 article, Adit Deshpande of O'Reilly Media detailed a high level overview of implementing sentiment analysis with LSTMs using Tensorflow[3], which served as a useful and practical treatment of the discussion in the aforementioned paper.

Separately, there have been a number of papers published on the interpretability of recurrent neural networks, and specifically long short-term memory variants. In *Visualizing and Understanding Recurrent Networks*, Kaparthy et al. use character-level language models as an interpretable testbed and reveal the existence of interpretable cells that explain LSTM's long-term behaviour[6]. To understand the weight of specific words on a networks prediction, Murdoch et al. use a simple rule-based classifier that approximates the output of the LSTM, and are able to assign coefficients quantifying word value for a given network and text[7].

The literature listed above is able to contextualize this study's themes of sentiment analysis and recurrent neural network interpretability. However, there is limited work on identifying sentiment keywords for reviews of music, which are inherently much less tangible than reviews of products or services. Additionally, there is limited work on using neural networks to capture the evolution of sentiment keywords over time. This is where this study aims to make a contribution.

III. Dataset and Features

A dataset of approximately 17,000 reviews published by RA since 2001 was obtained from Kaggle. Each review included fields relevant to the subject of the review, including track/tracklist, artist, record-label and review author. In addition to the review body, the two fields of note are the author assigned score (out of a 5.0 scale) and the year of music release.

Pre-processing was involved to both obtain the specific labels used in the classification task and to clean the review text. Reviews were assigned either a positive or negative sentiment label based off of the review score, with a threshold of 3.7 out of 5. This threshold was chosen so that the reviews would be spaced evenly amongst the two sentiment classes. Similarly, the reviews were assigned to one of three eras (2001-2009, 2010-2013, 2014-2017). The era bounds were chosen so that approximately a third of all reviews would be assigned to each era. The label definitions are summarized in table 1, while the distribution of reviews and ratings are shown in figures 1 and 2.

Label	Feature used	Definitions
Sentiment	Rating	- Postiive > 3.7 - Negative < 3.7
Era	Year of Release	- 2001-2009 - 2010-2013 - 2014-2017

Table 1. Description of labels and definitions used for training. Sentiments and eras were split evenly across classes

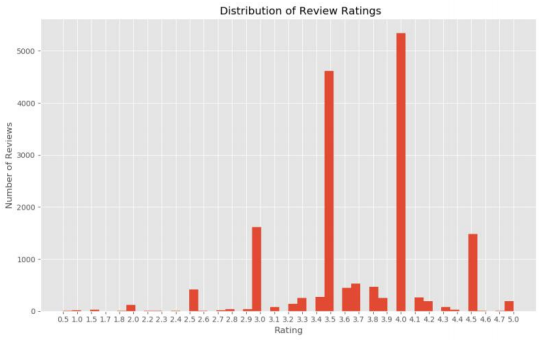

Figure 1. Distribution of review ratings

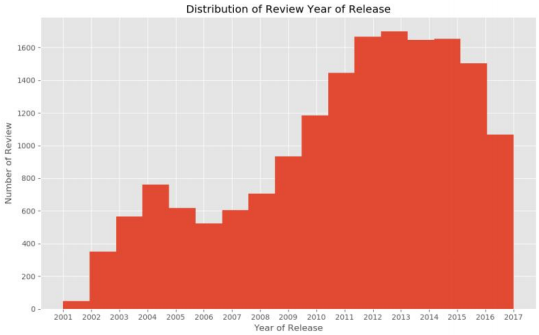

Figure 2. Distribution of review year of release

Non-alphabetical characters were removed from the review body. In addition, any words that were included in the fields, such as artist name, year of release, or genre, were removed. Stop words were also removed from the text, as they would not be useful in distinguishing reviews from one another. Additionally, reviews that contained severe formatting issues

or written in non-English were removed from the dataset entirely.

After preprocessing, approximately 13,000 reviews remained. These reviews were split into a train, dev, and test set according to a 70-15-15 split. This was chosen because the dataset was not large enough to warrant a split that contained more examples in the training set.

In order to be fed into the model, each minibatch had to contain reviews with identical sequence length. This was used using a special padding character, <*pad*>. All words, including <*pad*>, were assigned a unique integer ID and associated word embedding, which were used as the inputs for the model.

IV. Methods

The approach to identifying the evolution of keywords from RA music reviews can be split into three sequential tasks. They are training classification models, analyzing model response to inputs, and interpreting the results.

A. Training Sentiment and Era Classifiers

Both the sentiment classifier and era classifier used a many-to-one recurrent neural network (RNN) architecture. Specifically, a long short-term memory (LSTM) network was chosen as the model given their previous success in text and sentiment analysis[3,7]. A softmax output layer was used to map the final cell's activations to final class predictions. Cross entropy loss with L2 regularization was used, defined below:

$$\mathcal{L} = \frac{1}{m}\sum_{i=1}^{m} y_i \log(\widehat{y_i}) + \frac{\lambda}{m}\sum_{k=1}^{4} \|W\|^2$$

Where m is the number of training examples in the mini-batch and λ is the regularization strength. The loss function was optimized using the tensorflow implementation of the Adam optimizer. The models were trained on the training set and validated on the development set.

Loss, recall, precision and accuracy were taken into consideration when optimizing the model. The hyperparameter space was initially explored using grid search, and then with random search once the space was constrained to a smaller region.

The model was written in python/Tensorflow[8], and the CS230 Project Examples codebase[9] for the Named Entity Recognition task was used as the basis for implementation. To speed up training, a p2.xlarge instance on AWS was used.

B. Analyzing Model Response to Inputs

Once the models were trained and optimized for their respective classification tasks, the models' response to inputs was interpreted by examining their activations. Specifically, for each review in the dataset, the cosine similarity between the first and final activation were computed. Cosine activation is defined as follows:

$$s(a_t, a_T) = \cos(\theta_{a_t,a_T}) = \frac{a_t \cdot a_T}{\|a_t\| \, \|a_T\|}$$

where a_t and a_T are the activations of the t^{th} cell and the final ^{Tth} cell for the input, respectively.

These sets of similarities is because they provide are one way to visualize how the model reaches its final decision as it reads in the input. During the first few inputs of the sequence, the model's activation is dissimilar from its final activation and varies largely. Once the model begins to understand the sequence, the activation slowly converges towards the final cell's final decision. This can be thought of as a way to visualize how a model begins to make its prediction while reading through a document.

The words that result in a large similarity leading to its final steady-state activation are interpreted to be keywords. Specifically, a word was tagged as keyword if the absolute derivative of the similarity curve exceeded a threshold.

C. Interpreting Keywords

The number of instances that a keyword appeared in a given class's (two sentiment classes, three era classes) review was stored. Once all reviews were read, the n_s and n_e most frequent keywords were returned as sets for the two sentiment classes and three era classes, respectively.

Set subtraction and intersections were used to identify which keywords were unique to a given class. For instance, to determine keywords unique to a particular sentiment or era:

$$S_{positive,unique} = S_{positive} - S_{negative}$$

$$S_{2001-2009,unique} = S_{2001-2009} - S_{2010-2013} - S_{2014-2017}$$

In order to determine which keywords were unique to a given sentiment for a particular era, set intersections were used:

$$S_{positive,2001-2009} = S_{positive,u} \cap S_{2001-2009,u}$$

Where u denotes words unique to that particular sentiment or era. Keywords that had musical meaning were noted for analysis.

V. Experiments/Results/Discussion

A. Hyperparameter Tuning

Accuracy, precision, recall and loss were all taken into consideration when tuning the hyperparameters of the model. This was to ensure that the model performance was well rounded, and not biased towards a particular class when making a prediction.

As mentioned in section IV, the hyperparameters space was initially explored using grid search. This initial analysis indicated that model performance was most sensitive to learning rate and regularization strength. Therefore, these two hyperparameters were searched for more extensively using random search, with the other hyperparameters held fixed to the most promising results from grid search. The final hyperparameters are shown in table 2.

	Sentiment	**Era**
LSTM num_units	25	25
Embedding size	150	150
Learning Rate	2.787e-4	2.048e-3
Batch Size	32	32
Training Epochs	15	20
Dropout Rate	0.3	0.3
Regularization Strength	5.546e-2	2.691e-2

Table 2: Optimized hyperparameters for each model

B. Model Performance

Both models were relatively successful in predicting their respective classes. The era classification model performed particularly well, able to successfully predict the era with over 80% accuracy in the development set. Each models' performance is shown in table 3.

	Dev-set Accuracy	Dev-set Recall	Dev-set Precision
Sentiment	66.0%	78.0%	64.9%
Era	80.1%	78.5%	86.4%

Table 3. Classifier performance metrics on evaluation set

The higher accuracy of the era classification model indicates that the language used in music reviews has changed a reasonable degree over the last 20 years. One possible explanation for the lower performance of the sentiment classification model is that the language used in positive and negative reviews did not differ much. In this light, it is valuable to recall that figure 1 showed the distribution of scores for reviews shows that a large portion of the reviews in the dataset lie in the 3.5 to 4.0 range, indicating that reviews many reviews with similar scores were classified into two separate classes.

C. LSTM Cell Interpretation

Computing the similarities between the activation of each cell and the final cell activation proved to be a useful way to visualize how the model reached its final decision as it read through the review. Activation similarities for the classifiers for two different reviews are shown in figures 3 and 4.

There are a number of key observations to make with these plots. First, is that the similarities vary wildly in the beginning of the sequence. This is expected, because the model has not read enough inputs to understand the content of the review yet.

Second, is that the peaks in activation similarity differ for the sentiment and era classifiers. This indicates that the model is being influenced heavily in different parts of the review, and consequently has learned to place weight on different words.

An additional observation to note is that there are multiple "peaks" of activation similarity throughout the sequence. This shows that the model is being influenced throughout the input sequence, and does not settle on a prediction immediately. It may also be indicative of the models inability to handle long (100+ tokens) sequences, and may benefit from an attention implementation.

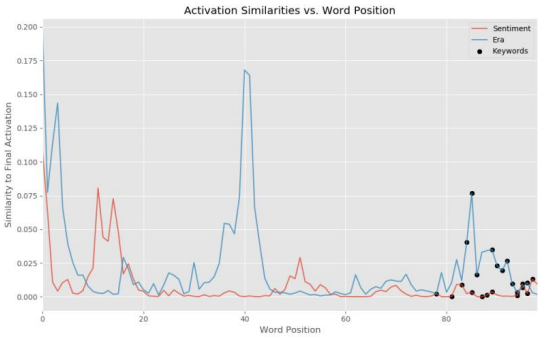

Figure 3. Similarity of activations for successfully classified review: Inner City Man EP by Geddes (2012, negative)

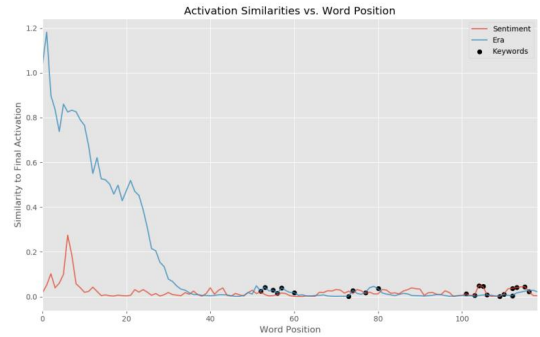

Figure 4. Similarity of activations for two successfully classified reviews: Squeeze by XI (2012, positive)

A final observation is that the keywords are all identified at the end of the sequence. This is due to the definition of keyword used in this analysis, as the last set of words that influence the model are the ones that lead to its final prediction. However, from the plots it is clear that there are other keywords in the sequence (position 40 of the era classifier, for instance) that are not accounted for.

D. Keyword Analysis

The top $n_e = 50$ and $n_s = 200$ keywords for era classification and sentiment classification, respectively, were returned and were placed into their various subclasses. Notable subsets of words for each subclass are shown below in table 4.

	Positive	**Negative**	**Neither**
2001-2009	Solid	Smooth, flip, nice, breaks	Original, massive
2010-2013	Vocal, touches, tight, synth, drum, groove, pop	n/a	Kind
2014-	Percussion, kicks,	n/a	Atmosphere,
2017	glowing, hard		package

Table 4. Subset of unique words for each subclass ($n_s = 200$, $n_e = 50$)

There is a wide variation of words describing positive reviews over the three time eras. In particular, words such as "hard" or "kicks" indicate that tracks with percussive elements are favoured recently. On the contrast, the subset and intersection operations did not yield any words of note for negative sentiment classification over the time eras. This indicates that the language used to describe negative sentiment in RA music reviews has not evolved significantly over the last 20 years.

VI. Conclusions

The models were able to predict the sentiment of a review and the release era of music with reasonable accuracy. Era classification in particular had strong performance, and indicates that the language used in RA music reviews has evolved over the last 20 years.

By visualizing the cosine similarities of the activations, the study was able to identify regions where the classifier was beginning to settle on a prediction. The different positions and magnitudes of the peaks of the similarity plots showed that the models were trained to respond to different words. By extracting the keywords from the model, set operations revealed that while the language used in positive reviews has changed over time, the language in negative reviews did not change significantly.

There are a number of areas of the study that can benefit from continued work and analysis. A large limitation of the current approach is the inability to detect key phrases in addition to keywords. Additionally, a more robust and flexible definition of keywords can capture some of the similarity peaks in the middle of the sequence that were not accounted for. Finally, given the length of the reviews (over hundreds of words, in many cases), the study could benefit from a model with an attention mechanism, as it was clear from the similarity plots that the model would intermittently be influenced in mid-sequence spikes of activation similarity.

REFERENCES

[1]https://www.residentadvisor.net/.

[2]https://www.kaggle.com/marcschroeder/17-years-of-resident-advisor-reviews/data.

[3] Deshpande, A. (2017, July 13). Perform sentiment analysis with LSTMs, using TensorFlow. Retrieved March 23, 2018, from https://www.oreilly.com/learning/perform-sentiment-analysis-with-lstms-using-tensorflow

[4] Domeniconi, G., Moro, G., Pagliarani, A., & Pasolini, R. (2017). On Deep Learning in Cross-Domain Sentiment Classification. *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*. doi:10.5220/0006488100500060

[5] Glorot, X., Bordes, A., & Bengio, Y. (2011). On Deep Learning in Cross-Domain Sentiment Classification. *Proceedings of the 28th INternational Conference on Machine Learning*. doi:10.5220/0006488100500060

[6] Karpathy, A., Johnson, J., & Fei Fei, L. (2017). Visualizing and Understanding Recurrent Networks. *ICLR 2016 (under Review)*. doi:10.21437/interspeech.2017-357

[7] Murdoch, W., & Szlam, A. (2017). Automatic Rule Extraction from long Short Term Memory Networks. *ICLR 2017*. doi:10.1109/iccvw.2017.276

[8] https://www.tensorflow.org/

[9] https://github.com/cs230-stanford/cs230-code-examples/tree/master/tensorflow/nlp

LINK TO CODE

https://github.com/christiansbrown/cs230