BLSTM ON NLP: MUSIC GENRE ANALYSIS

Luis Herrera, Steve Kim, and Colman Dekker

University of California San Diego, La Jolla, CA 92093-0238,

ABSTRACT

We will examine song lyrics to see how accurately a modern NLP model can classify music genres. Our goal is to see how a widely used algorithm (Long Short Term Memory RNN) performs on vastly different song lyric datasets. This is important because current voice recognition technology (Alexa, Siri, etc.) still requires explicit voice commands, and can not understand complex sentences. If voice assistants gained a more subtextual understanding of user input, it could increase their capabilities significantly

1. INTRODUCTION

Natural Language Processing is a field that, despite extensive study, has yet to manifest the same level of real-world impact as has image processing. To make a comparison of the technologies, we can compare how machine learning applied to image analysis has created tools such as person-identity recognition and self driving cars. On the other hand, the most widely used example of ML based NLP are the voice assistants found on smartphones. And as of today, voice assistants like Siri and Alexa are very primitive in terms of how they understand and respond to users' interactions. In fact, many of the features of these virtual assistants require specific vocal commands, with any deviation from those wrote commands being met with assistant misunderstanding or non-response.

Our foray into using a BLSTM to classify song genres based on lyrics is an experiment on how a model can learn semantic meaning and intention without explicit word choices. Our plan for this project is to see how different datasets can cause the same model to perform differently, thereby informing us on how robust this deep learning architecture is to data sparsity, language, and quality.

2. RELATED WORK

Hierarchical Attention Network - lyrics based NLP using HAN[1]. Approach by Tsaptsinos using an application of a Recurrent Neural Network. HAN performs much better than other approaches when classified with many target classes, with about 50% accuracy for 20 target classes, and 45% accuracy for 117 target classes Evaluating lyrics and classifying them with Naive Bayes, Linear Regression, KNN and SMO [2] This paper is a survey on different algorithms, with experimentation on the features known as "bag of words" and "parts of speech" Yang was able to achieve over 65% accuracy using Naive Bayes as his classifier. Brazilian Music Genre classification using $BLSTM$ (Bi Directional Long Short Term Memory) [3]. This paper assesses the use of $BLSTM$ applied to a large corpus of English and Portuguese lyrics. After iterating through various implementations of $LSTM$, we decided to emulate this $LSTM$ RNN model because it performed better than regular $LSTM$ implementations. Lyrics analysis and Classification using hand tuned algorithms, CNN, and $LSTM$[4]. This paper used various feature engineering techniques, including bi-gram, tri-gram, parts of speech, and sentiment analysis, to pre-process the data. One advice from this paper was that "word count" feature engineering wasn't very helpful, so we took this, as well as other learning, into account when pre-processing our data.

3. DATASETS AND FEATURES

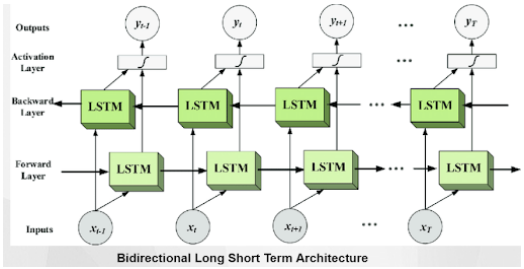
Baseline English dataset - "Scraped lyrics from six genres" ($190K$ songs, 253MB). Baseline Dataset was created based on the "Scraped lyrics from six genres". This dataset was originally divided into two files and we cleared no- useful data and combined both files. In order to combine the lyrics with the genre label a $MATLAB$ script was used. The three datasets have 5 columns. SongInfo, Lyrics, Genre, Artist and Idiom. English and Portuguese datasets were created from the Baseline just filtering Idiom column. Large Metal Lyrics Archive ($228K$ songs, 186MB). Hand labeled dataset ($5.3K$ songs, 3.65MB). This dataset was originally taken from $Kaggle.com$, but did not have any genre information. 20 bands per genre (but only 15 Grindcore) were hand-picked to use. Two, non Latin based text datasets, Chinese Characters (840 songs, 1.3MB) Korean Characters (746 songs, 1.6MB). These two datasets were machine translated using $Googletrans$, a python library that uses Google Translate. The metal dataset originally contained only band, album, song title, and lyric data, so hand labeling was required for genre. This consisted of checking Encyclopedia Metallum for genre information on a particular band, and then labeling each of the band songs to this same genre. One known issue with our data is that

bands can potentially have many different genres among their discography, and even among a single album. To deal with this, it was necessary to carefully hand select bands that have primarily only produced songs of a particular distinct genre.

3.1. Feature Extraction techniques

- Data cleaning such as consolidating "Funk Carioca" and "Funk"
- Text-preprocessing such as lower case/punctuation removal/etc
- Word tokenization
- Manual hand labeling
- Machine translation of English lyrics to Korean and Chinese.
- Concatenating dataset files

4. METHODS



Bidirectional Long Short Term Architecture

https://persagen.com/resources/glossary.htmlbi-lstm

BLSTM's, aka Bi-Directional Long Short Term Memory models, are an extension of LSTM's, but with the addition of a "backwards" LSTM pass. So while a normal (unidirectional) LSTM only preserves information about the inputs from previous steps, a BLSTM also has a backwards direction LSTM that maintains information about the future. If we name these two layers with the variables

$$\overrightarrow{h}, \overleftarrow{h}$$

and the output y, then :

$$\overrightarrow{h}_t = H(W_{x\overrightarrow{h}} x_t + W_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}})$$
$$\overleftarrow{h}_t = H(W_{x\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}})$$
$$y_t = W_{\overrightarrow{h}y} \overrightarrow{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y$$

src: https://www.aclweb.org/anthology/E17-1096.pdf

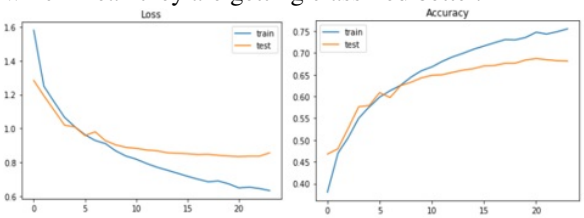
In this way, a BLSTM trains on both past and future data at each given timestep. We believe that this feature of BLSTM's help our model process each song's lyrics more holistically, which matches up with our intuition that you need to be able to take in a song as a whole in order to guessing what genre it is, compared to looking at individual words and phrases.

In terms of our model setup, we have a four layer BLSTM neural network. We've set dropout at 0.3 and recurrent dropout at 0.3 for the BLSTM. For the dense output layer, we are using softmax for activation. For the evaluation stage, we are using categorial crossentropy as our loss function, and adam for our optimizer. Our model operates with around 5,161,403 neurons in total.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 250, 100)          5000000
_________________________________________________________________
spatial_dropout1d_1 (Spatial (None, 250, 100)          0
_________________________________________________________________
bidirectional_1 (Bidirection (None, 200)               160800
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 804
=================================================================
Total params: 5,161,604
Trainable params: 5,161,604
Non-trainable params: 0
_________________________________________________________________
None
```

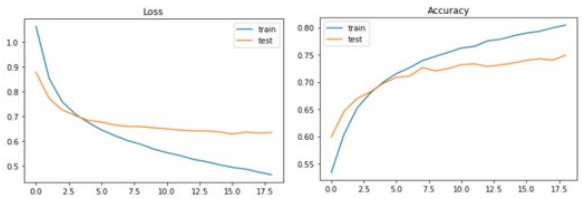
5. EXPERIMENTS/RESULTS/DISCUSSION

Applying the BLSTM model on the Baseline dataset, we observed that the training accuracy is around .75 on the graph; this changes when we see the classification report that gives us a accuracy of 0.69. When observed by genres it can be seen that the rock and sartanejo genres has more f1-scores which mean they are getting classified better.



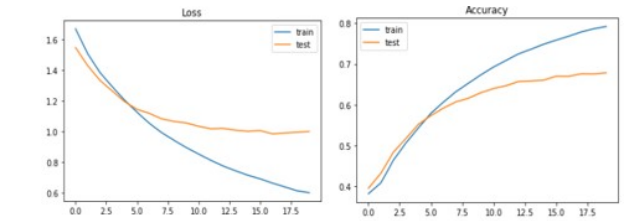
	precision	recall	f1-score	support
Funk	0.68	0.69	0.69	1315
Hip_Hop	0.76	0.67	0.71	4775
Pop	0.59	0.45	0.51	9148
Rock	0.73	0.80	0.76	13566
Samba	0.63	0.55	0.58	2968
Sertanejo	0.67	0.86	0.76	6777
accuracy			0.69	38549
macro avg	0.68	0.67	0.67	38549
weighted avg	0.68	0.69	0.68	38549

The English Dataset training accuracy is 0.8 and testing accuracy is 0.73, which is the same as the classification report. Because the English dataset doesn't have enough data on Sartanejo, Funk and Samba genres its normal that the classification precision is very low. The majority of the data is enclosed on Rock, Hip Hop and Pop genres and it has a f1-score of 0.81, 0.80 and 0.58 respectively.



	precision	recall	f1-score	support
Funk	0.00	0.00	0.00	4
Hip_Hop	0.82	0.79	0.80	5620
Pop	0.65	0.52	0.58	10389
Rock	0.76	0.87	0.81	16777
Samba	0.00	0.00	0.00	18
Sertanejo	0.00	0.00	0.00	20
accuracy			0.74	32828
macro avg	0.37	0.36	0.37	32828
weighted avg	0.74	0.74	0.74	32828

The Portuguese Dataset training accuracy is 0.8 and testing in 0.68. This dataset is more evenly distributed than the English one so it have better results on the classification report.



	precision	recall	f1-score	support
Funk	0.74	0.73	0.73	1982
Hip_Hop	0.69	0.61	0.65	1550
Pop	0.45	0.35	0.39	3241
Rock	0.61	0.60	0.60	3723
Samba	0.68	0.58	0.63	4278
Sertanejo	0.75	0.87	0.80	10222
accuracy			0.68	24996
macro avg	0.65	0.62	0.64	24996
weighted avg	0.67	0.68	0.68	24996

Another case that was done was to train in English and then test on Portuguese and viceversa. The results are expected were very poor accuracy. This can be said because the tokenization of the lyrics were done in two different languages and they are not correlated.

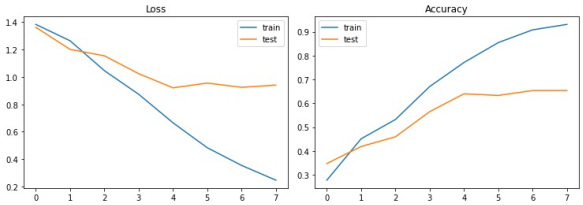
English Train Portuguese Test Classification report

	precision	recall	f1-score	support
Funk	0.00	0.00	0.00	6532
Hip_Hop	0.16	0.67	0.26	5181
Pop	0.15	0.34	0.20	10900
Rock	0.16	0.47	0.24	12651
Samba	0.00	0.00	0.00	14545
Sertanejo	0.00	0.00	0.00	33509
accuracy			0.16	83318
macro avg	0.08	0.25	0.12	83318
weighted avg	0.05	0.16	0.08	83318

Portuguese Train English Test Classification report

	precision	recall	f1-score	support
Funk	0.00	0.19	0.00	21
Hip_Hop	0.63	0.48	0.54	18705
Pop	0.35	0.13	0.18	34926
Rock	0.63	0.17	0.27	55647
Samba	0.00	0.17	0.00	52
Sertanejo	0.00	0.49	0.00	73
accuracy			0.21	109424
macro avg	0.27	0.27	0.17	109424
weighted avg	0.54	0.21	0.29	109424

The following are the results of $BLSTM$ on the metal dataset:



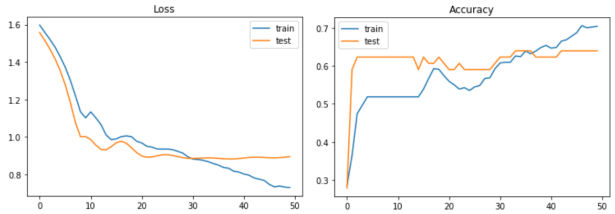
	precision	recall	f1-score	support
Doom/Sludge	0.57	0.59	0.58	323
black metal	0.73	0.81	0.77	410
death metal	0.75	0.67	0.71	325
Grindcore	0.77	0.73	0.75	535
accuracy			0.71	1593
macro avg	0.71	0.70	0.70	1593
weighted avg	0.72	0.71	0.71	1593

As can be seen above in this graph of accuracy to number of epochs, overfitting was a major issue for this dataset. Below, the graph of loss to number of epochs also shows this overfitting issue.

These results are approximately what was expected, and at an overall f1-score of .71, are considered successful. There is some disparity in results by genre, and this was to be expected. The model was most successful with the black metal lyrics and least successful with the Doom/Sludge lyrics. Most of the black metal lyrics in our dataset are from bands who wrote the majority of their lyrics in Oslo, Norway during the early to mid 1990s, from bands such as Mayhem and Darkthrone. These band members mostly all knew each other, all spoke Norwegian as a native tongue, and created the local scene together. Due to these reasons, it should be expected that many of their lyrics have a wealth of similarities in not only theme but also structure and word choice, due to sharing a native language but writing in English. Doom/Sludge is a much older genre, and has had much more time to grow and evolve. The Doom/Sludge lyrics in our dataset contain songs written in the late 1960s and early 1970s, from bands such as Blue Cheer and Black Sabbath, all the way up to currently active bands such as Electric Wizard. About half of these bands are from England, and the other half are from the Southern US, allowing for some significantly different language. Though many of the themes in this genre have stayed popular for decades, slang and other scene-exclusive

sort of language useful for identifying a genre is constantly changing. The success of the model with the Grindcore genre was not expected. The lyrical themes of the genre often vary significantly from band to band, especially since it is the most politically charged genre of the four, and contains data fairly evenly spread over the last three decades. However, according to these results it seems likely that though the genre covers many different themes and topics, these are mostly exclusive to the genre, and tend not to bleed into the other three. The death metal accuracy matches the average, which seems reasonable, given that there were few expectations made about it in advance. There are some similarities to the black metal data, as many of the lyrics in the dataset were written in a particular place, Florida, and at the same time, the early to mid 1990s. However, there is a little more variation in both time and place than the black metal lyrics, which is likely why it did not perform quite as well with these.

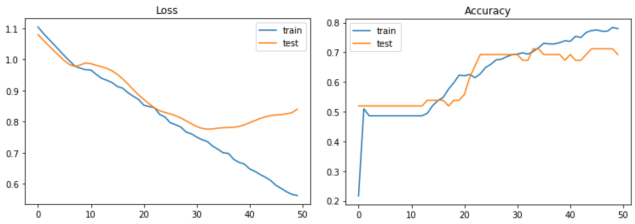
The following are the results of $BLSTM$ on the Chinese dataset:



	precision	recall	f1-score	support
Pop	0.57	0.66	0.61	44
Rock	0.00	0.00	0.00	83
Hip Hop	0.60	0.93	0.73	131
accuracy			0.59	258
macro avg	0.39	0.53	0.45	258
weighted avg	0.40	0.59	0.47	258

We can see indications of overfitting as the validation scores approach and closely align with the training scores. Even after adding stochastic elements to our model, like dropout and recurrent dropout, we still saw our model overfit. We believe with more data added into the dataset, that this issue would be ameliorated. Our hypothesis before giving our model non-Latin characters was that the LSTM would not perform as well on the Chinese/Korean data, because Asian languages are more information dense per timestep. That said, we can see that the model still performs okay with about 60% accuracy on the Chinese dataset. Looking at the confusion chart, we can see that the model does better when predicting Pop and Hip Hop genres, where precision is 57% and 77%, and recall is 66% and 93% respectively. On the flip side, the model did not predict Rock very well, with precision of 0% and recall of 0%, which is interesting because this was not reflected in the benchmark English dataset. Even when changing the model parameters, the weights would not predict any of the songs as Rock, which makes me think that the model is performing worse than the metrics would suggest, as it is completely ignoring Rock as an option. Perhaps Rock is "in-between" Pop and Hip Hop, and including it in the classification would cause predictions for all genres to suffer.

The following are the results of $BLSTM$ on the Korean dataset:



	precision	recall	f1-score	support
'Rock'	0.55	0.67	0.60	36
'Pop'	0.35	0.18	0.24	66
'Hip Hop'	0.66	0.79	0.72	121
accuracy			0.59	223
macro avg	0.52	0.55	0.52	223
weighted avg	0.55	0.59	0.56	223

We can see that overfitting was occurring for the Korean dataset as well, as validation performance followed too closely to the training performance. It seems to suffer the same issue as the Chinese dataset, and likely including additional data would help avoid overfitting. In terms of performance metrics, the model was able to predict Rock and Hip Hop well, with precision scores of 55% and 66%, and recall of 67% and 79%, respectively. It did not predict Pop well, with a precision score of 35% and recall of 18%. This is similar to how the model classified the Chinese dataset - the model was able to predict two genres, but not a third. We think that the model might be playing it safe by under-predicting Pop in general. Given the results from these two experiments, we can't yet affirm or dis-affirm our hypothesis that Asian script is more difficult to classify compared to English script. It appears that in some cases, the model is able to predict two of the three genres appropriately. But we can't conclude that the model is agnostic to language choice, because the model still failed to perform well across all target classes.

6. CONCLUSION/FUTURE WORK

We could feed in more data into the model in order to have a more robustly trained model. This would specifically include augmenting our non-latin character songs and manually tagging more metal songs. Furthermore, we saw in one of our reference papers and from feedback from our peers that we could try a sentence embedding input instead of just tokenized words. Based on our experiments with different sources, we've established that genre classification is highly language dependent. We've also found that within a language, a model is able to pick up on non-explicit patterns of words to classify subtextual information. These finding show that there is a lot more room for real world NLP applications to grow.

7. CONTRIBUTIONS

Each group member was in charge of a dataset. Dekker was in charge of the metal dataset, which consisted of manual tagging of lyrics to genre, which was done by tagging genre to band, and then band to lyrics using the python Pandas library. The metal dataset was run on the BLSTM model created by Herrera, with only some minor adjustments.

Kim retrieved the Chinese and Korean datasets. He used machine translation of the songs in the English benchmark dataset. While the translation quality was good as it was generated from Google translate, he was not able to obtain more than a few hundred translations because of rate limiting on the google translate API.

Herrera found the scrapped-lyrics-from-6-genres dataset in Kaggle. Then he modified the data on it to create the lyrics-data.csv file. He used Matlab to index the genres from the artist to the lyrics to create the baseline dataset. Then filtered it by idiom to get the English and Portuguese datasets. Eliminated any data that was not useable by the team. In advice from Kim worked on the LSTM model that evolved into a BLSTM after seeing better results on the second one. Then the three of us worked on optimizing the model so it worked on all datasets. Martin worked on merging the codes so it run all the datasets on a single notebook.

8. REPLY TO REVIEW

Group 77 Feedback
For different genres, lyrics may not as discriminative as rhythm. There is a pop song ”Oops!... I did it again” by Britney Spears. A Finnish melodic death metal band ”Children of Bodom” played the same song with basically the same lyric in a death metal fashion. This is an example showing lyrics are less discriminative than rhythm for classifying genres. that's an interesting point, agreed that rhythm would likely also be discriminating of genre.
- The lyrics of a certain genre can evolve over time, so it might be a good idea to take the decade of the song into consideration. yup, agreed
- Since it's common to use sentences with fixed structures in lyrics, consider not only tokenize words but also tokenize sentences. Yes, sentence embeddings instead of just word tokens would be a good next step as the LSTM model would have access to more information, good point
- The results of the baseline dataset and the metal dataset showed obvious over-fit. I had the same problem with my project. I solved this problem by changing the optimizer from ”Adam” to ”tanh” and reducing the embedding size. Hope this would help. Perhaps you meant activation function? We used Adam as our optimizer, and softmax as our activation. Overfit was an issue we tried to address via dropout and other techniques, we didn't try tanh, though.

Group 33 Feedback:
What was the reason for choosing the BLSTM model architecture rather than the other methods mentioned in the literature survey? What was the conclusion after comparing results from different dataset? Does a certain song genre is easier to be classified by the model? After finishing the Chinese/Korean dataset. How does the model performance on symbolic characters compare to Latin letters? The results showed that the classifier was still able to classify the Chinese and Korean datasets decently well. It's not easy to compare it to the english dataset because the English lyrics were written natively, whereas the Chinese/Korean lyrics were machine translated.

Group 8 Feedback:
It has already mentioned in the presentation that additional data gathering is needed for the Korean and Chinese song dataset. However, those datasets as well as the English song dataset only contains three classes and distributes unevenly. So, more data is needed and other balanced datasets should be used to solve the overfitting problem. Agreed that the results show that the datasets are likely not large enough
In the literature part, It's mentioned that “word count” features are not helpful from the last paper. However, it will be more convincible if you can provide a comparison between training with/without features like bi-gram and

tri-gram. Also, I think it will be better if you provide a baseline using traditional machine learning algorithms, for example KNN, to see the advantages of using deep learning model. That's a good point, it's a good idea to doublecheck the advice given from papers. Once we felt like we got good results from our model, we didn't think to continue testing it, as that wasn't the goal of our project.

In results and observation, you try to train a model in one language and test it in another. Can you explain what's the motivation of doing these experiments and what expected results you want to have. Did the actual results meet the expectation or not? First, we wanted to test what the model could actually do. We fully expected that training on one language, and testing on another, would not yield good performance. The results show that our hypothesis was correct

9. REFERENCES

1 Lyrics Based Music Genre Classification using a Hierarchical Attention Net:
https://arxiv.org/pdf/1707.04678.pdf

2 Lyric-Based Music Genre Classification:
https://dspace.library.uvic.ca/bitstream/handle/1828/9378/Yang-Junru-MSc-2018.pdf

3 Brazilian Lyrics Based Music Genre Classifier using a BLSTM network:
https://arxiv.org/pdf/2003.05377.pdf

4 Exploring the world of lyric:
https://people.duke.edu/tc233/xilinx-repo/ECE590-nlp-final-report.pdf

5 Official Scikit learn documentation:
https://scikit-learn.org/stable/user-guide.html

6 Bidirectional Long Short Term Architecture:
https://www.researchgate.net/figure/Basic-structure-of-the-BLSTM-network-The-LSTM-nets-at-the-bottom-indicate-the-forward$_fig3$