Exploring the world of lyrics

- Lyrics analysis and classification via multiple methods

Lantao Li (ll285), Xiaoyang Liu (xl149), Ting Chen (tc233) - Rev1.1

1. Introduction

People love music. Today, there are billions of songs and music available for people to enjoy and devour. Song lyrics hence compose a huge and diversified corpus, which enables us to explore some of its interesting and meaningful metrics.

People used to analyze songs from their audio signal aspect[8]. It does yield a lot of information, but processing audio signal can be time consuming and requires considerable amount of computing resources. Instead, with emerging tools and newly proposed NLP algorithms, we can investigate song lyrics more easily. Our investigation has three major components: using approaches from classic information theory such as entropy (which analyzes each song in a greater width, including sentiment author and genres), implementation of a CNN classifier and lastly a LSTM model. The last two neural network models focus on genre classification.

1.1 Idea

Comparing with audio data, song lyrics are much easier to obtain in terms of copyright and bandwidth limitations. Lyrics are an ample source for providing insight of genre of a song. With all the knowledge and perks we've learnt this semester, we decide to test our skills on lyrics dataset.

As what we can expect, lyrics corpora are much different than daily English and text. For example, lyrics often have rhymes. They also tend to repeat itself without making sense. And they do not need to obey formal grammars of normal English. Most of all, lyrics are highly topic-oriented: normally within 300 to 400 words, a song can tell a love story or fully express one's feeling. All these specialties and features of lyrics can be either useful or bothersome when we deal with them.

In order to better classify lyrics into genres and authors, we need to extract multiple features from them. Different genres and artists might be extremely different to each other, and we'll try different methods to see which one works better for this corpus.

What we want to achieve in this project:

- Compare the performance and accuracy of multiple methods to classify lyrics
- Given a lyric, predict which genre it belongs to with our models

1.2 Dataset

Our dataset is from Kaggle[1], which comprises over 300,000 lyrics along with their author, publishing year and genre. We want first to clean up the set and generate smaller training set, and then evaluate each classification methods with evaluating set.

1.2.1 Cleaning the dataset

The set did not come with a clean format; thus, we did some data cleansing at first.

First, we removed all the non-English lyrics. If a lyric contains more than 30% unrecognized words, we treat it as non-English and remove it from the dataset. Since some Rap music as well as Electronic music use abbreviations and dialect of English words, we did not restrict them to be 100% English, but set a threshold of 30% to ensure there will be enough training entries later.

Second, we remove all the lyrics that have strange length. Usually, a 3-minute-song should contain about 350 words in lyrics. Considering that Rap music could be more fluent in its flow, we restrict the length of lyric to [100, 800) so that some weird lyrics won't affect the result.

Finally, we remove those lyrics without annotated genres, i.e. those with genres = [Others, NA]. Therefore, there left 10 genres in total, from which we choose 4 to test our classification methods.

1.2.2 Training and evaluation set

- For genre classification

The 10 remaining genres are: Pop, Hip-Hop, Rock, Metal, Country, Jazz, Electronic, Folk, R&B and Indie. The smaller training set comprises 1500 lyrics from 4 genres: Rock, Hip-Hop, Electronic and Jazz. We choose these 4 genres intuitively, since we think these genres might be more stylish than others.

Therefore, for each classification method, the lowest accuracy should be 25% - the case to classify randomly into 4 genres.

- For artist classification

We select 100 lyrics for 8 well-known artists and try to investigate their differences in producing lyrics. Indeed, some lyrics might not be written by the artist himself/herself, and an artist can also create lyrics with different styles. We just assume that those better-known artists should have a more unified style to classify.

The lowest accuracy should be 12.5% for this classification.

- For age group classification

We select 3 age groups: 1970, 1990 and 2010, and uniformly choose 1500 lyrics in each group. However, after some preliminary analysis, we found years of publishing might not be an important feature to classify. We'll discuss this topic later.

1.3 Additional information

All the lyrics in test sets are randomly chosen with given conditions. Lyrics in training sets are not, however, due to some issues that we'll discuss later.

Most statistical scripts come with tqdm progress bar so that the progress of the script can be visualized.

2. Methods

We tried multiple methods that we learned this semester, including statistical ones, NLP methods and Neural Network methods.

2.1 Statistical Methods: N-gram, Entropy and Word Count

- For word count, we count the words in a piece of lyric of a certain genre or an artist for the training set, and calculate the average word count for them. Then for the evaluating set, find which genre or artist have the closest word count matching with the testing lyric.
- Entropy is an important feature when dealing with a large amount of text or discrete data. The definition of entropy and first order conditional entropy in Markov Model.

$$H(X) = -\sum_{i=1}^{n} P(x_i)\log_2 P(x_i)$$

$$H_{M_1}(\chi) = H(X_2 \mid X_1)$$

Sometimes, certain corpus can have an entropy that stands out from the others. Below are some stats of entropy for genres and artists.

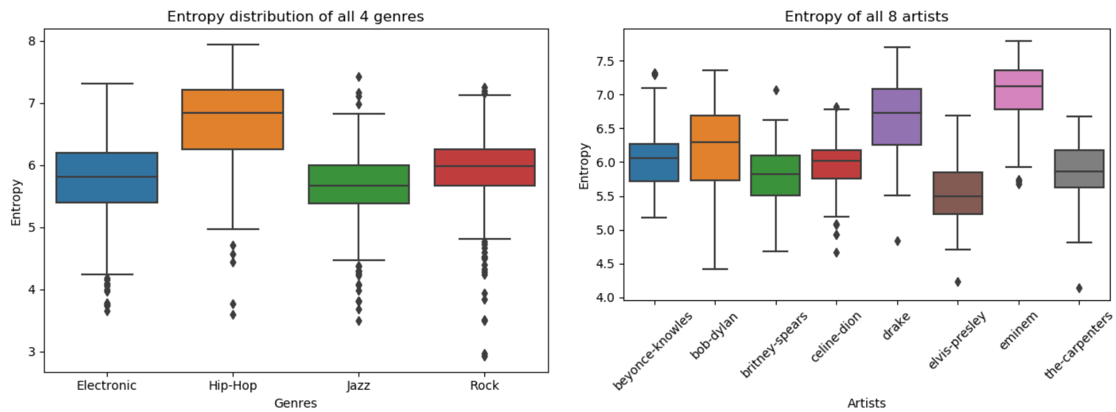


FIGURE 2.1.1 ENTROPY DISTRIBUTION OF SONG GENRES AND ARTISTS

As we can see here, difference exists among both genres and artists. From an intuitive perspective, higher entropy represents more complexity in wording. Old melodies have smaller entropy, and Rap music are full of changes. We can clearly expect a genre of music or an artist have certain style of music by calculating entropy for their lyrics.

- N-gram is a method that we learnt during this semester. It considers plain text as a Markov Model and calculate probability related features. For this classification, I tried both Bi-gram and Tri-gram methods for them, and both performed well. In terms of N-gram, I calculate the pair and 3-pair word count of a lyric. And for the evaluating set, find which genre or artist have the closest N-gram matching with the testing lyric. That is to say, find the maximum amount of matching pairs of the testing lyric and genres or artists in training set.

2.2 NLP methods: PoS-tagging and Sentiment Analysis Classifiers

- PoS tagging, or Part-of-Speech tagging is another NLP method that we learnt from class. Part-of-Speech is important in classifying authorship. We assume that for different category of songs, they might use different percentage of PoS tags. For example, we might say Rock and Rap uses more verbs, while Jazz and Folk use more nouns, etc. We can also spot interesting facts such that Beyoncé likes to use “Oh”, etc.

  In order to classify a lyric, we count its PoS tags first, and calculate each tag’s percentage overall. Then we compare to find the closest matching category by both comparing testing lyric’s PoS percentage to the training average and each entry in training set.

- Sentiment analysis[2][6] is yet another tool we learnt during this semester. While a genre of songs might be neutral on average, a specific artist might have strong personal emotions or stylish sentiment orientation in their works.

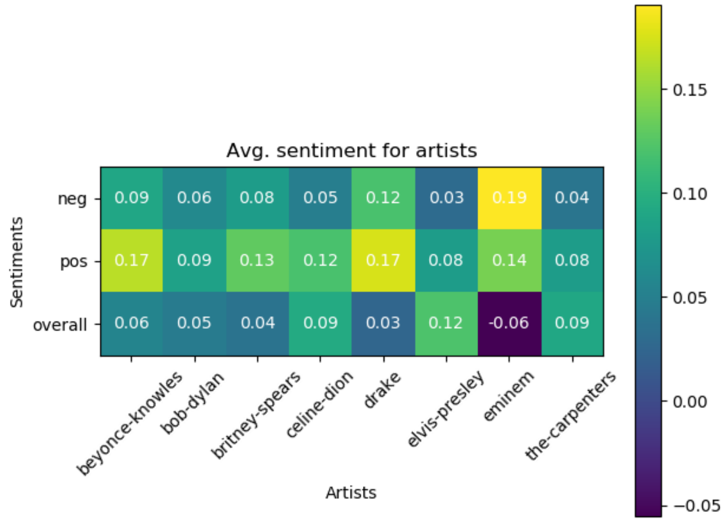


FIGURE. 2.2.1 SENTIMENT DISTRIBUTION AMONG ARTIST

With NLTK provided sentiment analyzer, I counted the sentiments for all 8 artists and made the graph above. Eminem seems to be the most stylish artist amongst all: his lyrics' sentiment are more polarized, including strong positive and negative emotions. In contrast, Bob Dylan seems to be the most "peaceful" artist.

2.3 Character-level Recurrent Neural Network and LSTM Classifier.

We constructed a character-level RNN classifier[7]. The recurrent neural network consists of 2 layers (each layer has size of 128) with one dropout layer and a LogSoftmax layer in the end (structure is shown in Figure 2.3.1). The lyrics data from each song is parsed into individual letter, which is encoded into a one-hot vector of length 63. The output is a 1 x 10 vector with each index encoding the probability of each song genre given the lyrics. The target vector is also a one-hot 1 x 10 vector with 1 on index corresponding to a specific genre and all other entries being 0. The song genre is determined to be the highest probability genre in the network output at the last time step (so after the last letter in the lyrics is fed into the network). Loss is calculated only based on the last output. The RNN network is trained on 40,000 song samples with stochastic gradient descent.

In addition to an RNN network, a LSTM network is also developed. The structure of LSTM is shown in figure 2.3.1. LSTM is implemented due to the concern that a regular RNN couldn't handle the length of the input too well (an input length is usually in the hundreds). The LSTM is trained on a smaller training set of 400 songs with only 4 genres (each genre has 100 songs).

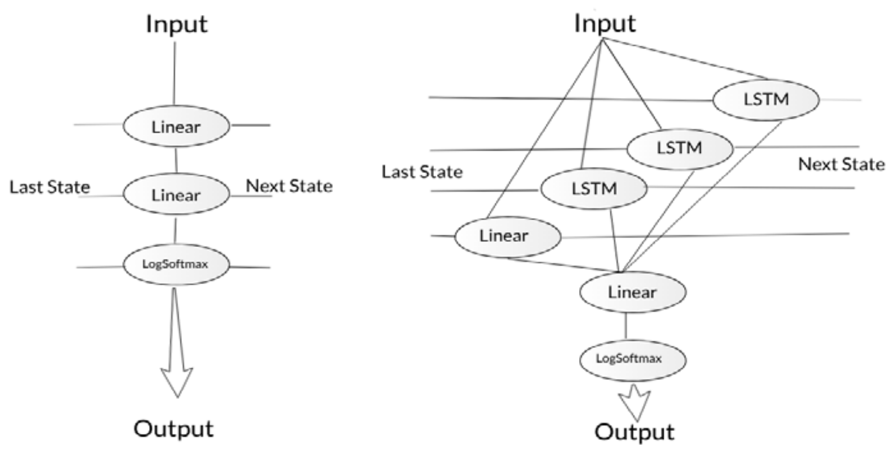


FIGURE 2.3.1 STRUCTURE OF RNN (ON THE LEFT) AND LSTM NETWORK (ON THE RIGHT).

2.4 MLP and Text-CNN with word embedding.

The motivation to try on CNN is that CNN is very good at classification for images and for multi-feature data, and there is a specific text-CNN[3] which will be like a multi-dimensional word vector corresponding to a specific length of words sentence. Word embedding plays the role to provide a multi-dimensional word vector for each word, GloVe (Global Vectors for Word Representation, developed by Stanford)[4] and a smaller size word embedding (mini.h5) from amazon ConceptNet 17.06[5] is used in this project. We will use multiple filters (with varying window sizes, the widths remain unchanged, the same as the length of word embedding vector size) to obtain multiple features and max-polling. These features form the penultimate layer and are passed to a fully connected softmax layer (can be multiple fc layers) whose output is the probability distribution over labels.

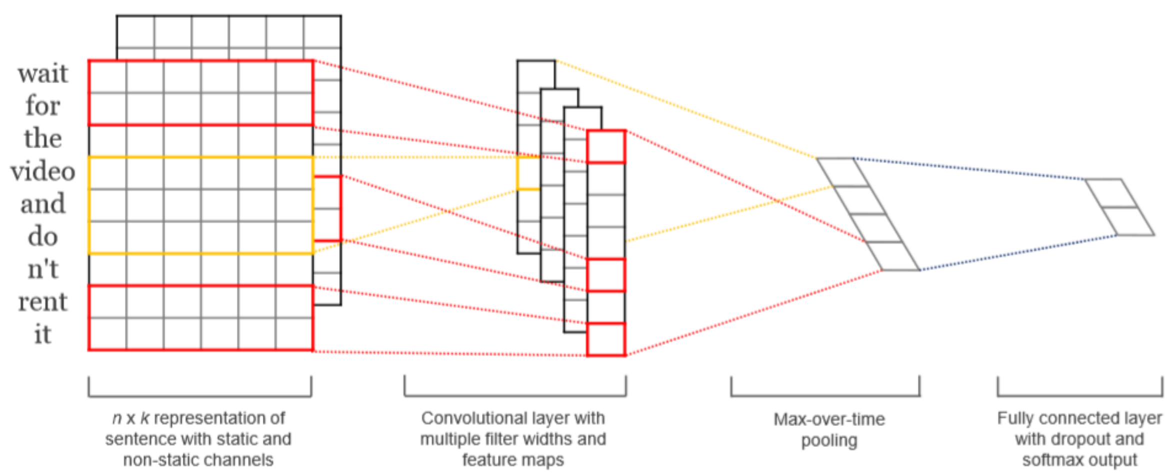


FIGURE 2.4.1 TEXT-CNN STRUCTURE WITH TWO CHANNELS FOR AN EXAMPLE SENTENCE

Before moving to CNN, a Multi-Layer-Perceptron combined with word embedding is tested on the same data and the method of using word embedding differs from CNN as the input of MLP is the mean value vector of vectors of every song, thus the computation is relatively smaller. This is more like extracting the information of semantic meaning from word embedding of every words of lyrics to represent the subject of each song.

2.5 Reviews on other methods

When doing background research for this topic, we also noticed other methods like SVM[11], word2vec, etc. Due to space limit, we won't elaborate on these topics here.

3. Results and discussion

3.1 Statistical features

For the first part, we used statistical methods to count the lyrics.

First, we get the word count for all genres, genre training set and artist training set. We'll briefly discuss some interesting observations from these stats.

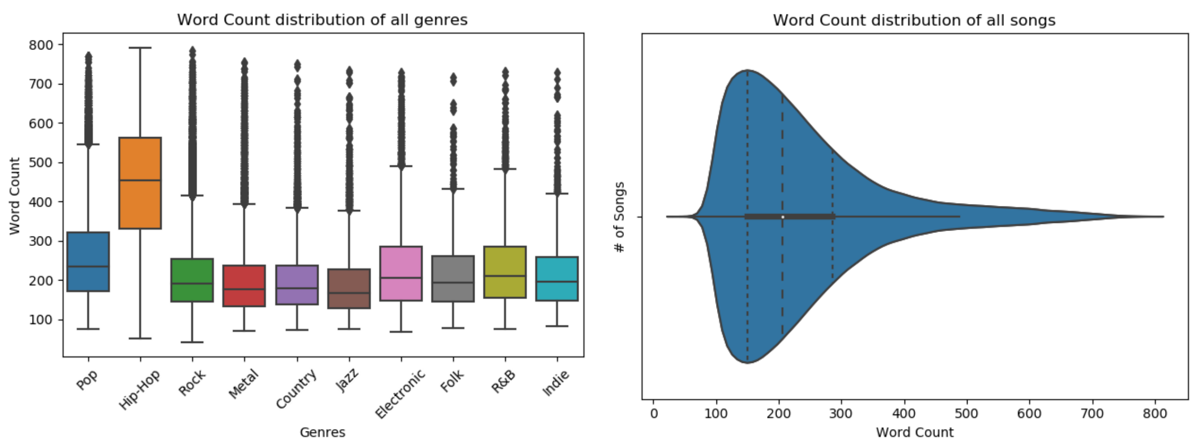
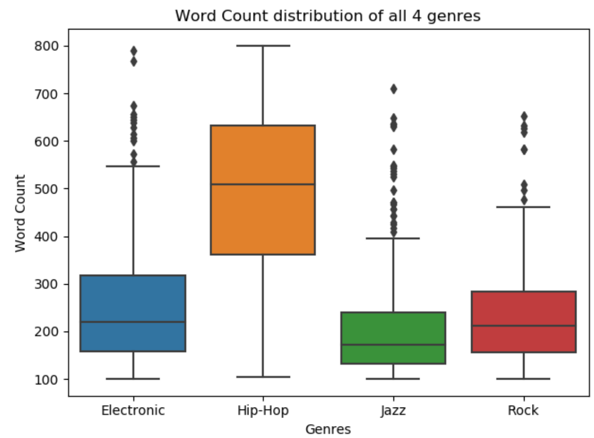


FIGURE. 3.1.1 WORD COUNT DISTRIBUTION OF ALL 10 GENRES AND ALL SONGS.

For the word count of all songs, the obvious finding is that Hip-Hop music have way more words on average than other genres. That is comprehensible: Hip-Hop or Rap music are always fast and have smaller gap between each line, which result in a higher word count rate per time unit. In contrast, Jazz, Folk and Country music seem to be the slower ones, which is also consistent with our impression.

The bulk of lyrics have word count between 100-300 words, where more than 50% of songs are here. However, there are also long lyrics. With detailed inspection, I found some of these are just a batch of "Oh" and "Ah". That could be a feature of some genres of music.



3.1.2 WORD COUNT DISTRIBUTION OF 4 GENRES

For the word count of 4 selected genres, we can still see that Hip-Hop stands out from the other 3. Jazz seems to be slower while Rock and Electronic have moderate length in lyrics.

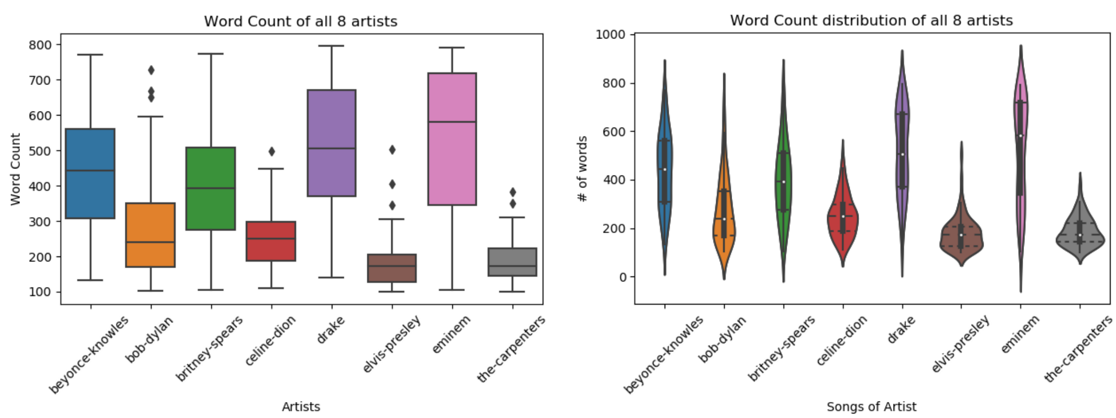


FIGURE 3.1.3 WORD COUNT DISTRIBUTION OF SELECTED 8 ARTISTS.

For 8 selected artists, we can see huge difference between them! From my own perspective, Elvis and The Carpenters are some elder generation singers who wrote slower and mild songs. Britney, Beyoncé are modern singers with both energetic and gentle songs, and Hip-Hop stars like Eminem is really creating some dazzling flows in his works.

Finally, we used entropy and word count to cluster 3 artists: Eminem, Britney Spears and Bob Dylan. These 3 artists are distinct from each other, and we can see this fact in the graph below. As we can see, though there are certain difference between their statistical metrics, it’s still not easy to separate their songs cleanly. With other results in the following sections, we might conclude that plain statistical classifying methods are not enough to accurately categorize lyrics.

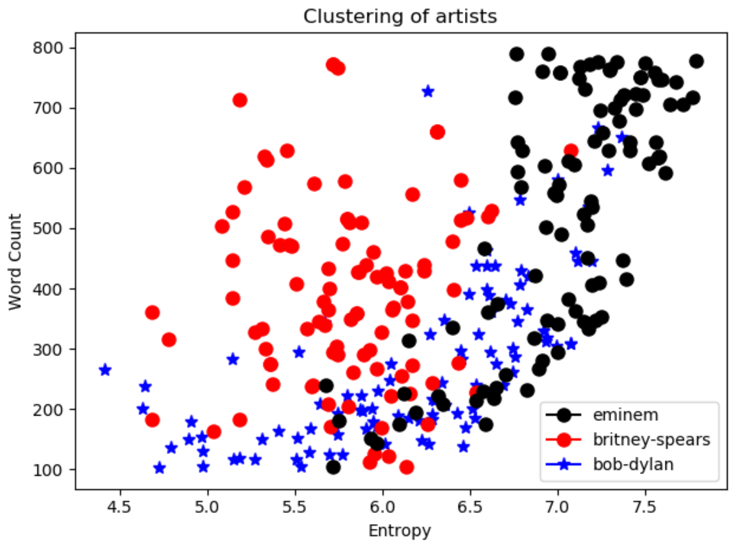


FIGURE. 3.1.4 CLUSTERING OF ARTISTS.

3.2 Statistical results

3.2.1 Statistical classification

	Word Count	Entropy	N-gram
Genres	40.5%	50.6%	**64.7%**
Artists	30.0%	26.3%	**65.0%**

TABLE 3.2.1 RESULTS STATISTICAL CLASSIFIERS

The “one-hot” single value classification of Word Count and Entropy do not perform well on test set, only scoring around 50% of accuracy. For artists, they are merely better than randomly guessing. This result shows that a single statistical metric cannot fully differentiate multiple categories of a corpus. However, it does provide some information that help us to better predict categories.

As a comparison, the N-gram classifier works much better than former 2, which also meet our expectation. As we mentioned above, various genres of songs and distinct artists might have some unique “patterns” in wording. Single word count or word frequency cannot reflect this feature, while multiple-word-phrase can successfully extract some valuable information from the corpus. Another similar practice uses phrases as “fingerprint” to articles or intellectual property judgement [9][10], since certain arrangement and order of words can serve as an identification of authorship.

3.2.2 NLP classification

	PoS	Sentiment
Genres	**63.3%**	**61.3%**
Artists	**58.8%**	47.5%

TABLE 3.2.2 RESULTS OF NLP CLASSIFIERS

For the 2 NLP classifiers, both of them works considerably better than stat-only ones. Among them, the PoS tagging classifier works better than sentiment analyzer. This is reasonable with the similar reason to the fingerprint logic described above. An author uses certain ratio of each speech parts without even noticing them.

Also, the length of sentence also contribute largely to PoS ratio, since shorter sentence cannot contain too many nouns and conjunctions, etc.

3.2.3 Years classification

Surprisingly though, when I tried to apply the previously mentioned methods to categorize their produce dates, I did not receive meaningful results. For all statistical methods I used, the best ratio is not above 45%, which is not positive. From my assumption, for different years, the difference comes mostly from genres rather than age group. In the 1970s, there was no Rap music, and most are Rock and Pop music. When new genres emerge, it could generate impact on overall statistics. Therefore, I did not include the result for year's classification here.

3.3 CNN results

3.3.1 MLP with word embedding

Dataset Settings (default: 80% will be training part and 20% will be evaluation, ConceptNet word embedding)	Training Accuracy	Evaluation Accuracy
50 for each genre, 500 in total	0.37	0.31
200 for each genre, 2000 in total	0.42	0.3425
500 for each genre, 5000 in total	0.59	0.422
1000 for each genre, 10000 in total	0.57	0.3905
1500 for each genre, 15000 in total	0.52	0.364

TABLE 3.3.1 RESULTS OF MLP ON 10 GENRES OF LYRICS

The most difficult task is to classify all the genres in the original dataset, the truth is that some lyrics of different genres can be very similar, and some lyrics of same genres can be very different, for example, love is the subject of a huge number of lyrics (lovelorn, first love, romance). Subjectively speaking, the more important part of a song or the core factor determining the genre of a song is the melody, the rhyme and even the instruments. For this test, the accuracy varies from 0.3 to 0.4 depending on how big the training set is, the key reason of the variation is that small size of dataset may lead to bad generalization, some specific song with special lyrics or very different style of picking words for lyrics can make the sparsity large within the same genre. And the accuracy is so low that almost useless to classify those genres, the reason is that different genres have crossing areas with each other, sometimes even hard for human to tell a song is Pop or R&B.

Word embedding	Dataset Settings (default: 80% will be training part and 20% will be evaluation)	Training Accuracy	Evaluation Accuracy
GloVe	50 for each genre, 200 in total	0.88	**0.65**
GloVe	200 for each genre, 800 in total	0.96	0.63
GloVe	500 for each genre, 2000 in total	0.87	0.6075
GloVe	1000 for each genre, 4000 in total	0.79	0.6125
ConceptNet	50 for each genre, 200 in total	0.61	0.52
ConceptNet	200 for each genre, 800 in total	0.58	0.565
ConceptNet	500 for each genre, 2000 in total	0.76	**0.68**
ConceptNet	1000 for each genre, 4000 in total	0.69	0.66125
ConceptNet	1500 for each genre, 6000 in total	0.72	0.63166666

TABLE 3.3.2 RESULTS OF MLP ON 4 GENRES OF LYRICS WITH TWO DIFFERENT WORD EMBEDDING

GloVe is bigger in the size of corpus, but it has better performance on smaller size Dataset comparing with Conceptnet. There are many comparisons between different word embeddings like word2vec, fasttext and

GloVe out there, but for this task, the performance is weird, GloVe has a decrease with larger dataset while ConceptNet is increasing. At first, I believe this might due to the imbalance of data or accidental output of small dataset, but the similar results come with repeating tests with randomization on dataset. And after this test, I decided to choose ConceptNet's word embedding in the later steps. The best accuracy it can achieve is around 65%.

Dataset Settings (default: 80% will be training part and 20% will be evaluation, ConceptNet word embedding)	Training Accuracy	Evaluation Accuracy
Hip-Hop & Folk 500 for each	0.96	0.905
Hip-Hop & Folk 1000 for each	0.99	0.91
Hip-Hop & Electronic 100 for each	0.98	0.94
Hip-Hop & Electronic 1000 for each	0.97	0.905
Hip-Hop & Electronic 2000 for each	0.95	0.905
Electronic & Folk 1000 for each	0.88	0.735
Rock & Jazz 1000 for each	0.91	0.7075
Jazz & Country 1000 for each	0.9	**0.6225**
Folk & Country 1000 for each	0.83	0.6925
Rock & Metal 1000 for each	0.86	0.86
Pop & R&B 1000 for each	0.79	**0.685**

TABLE 3.3.3 RESULTS OF MLP TO SEPARATE TWO DIFFERENT GENRES

This table can tell how similar or how close in multi-dimension presented by word embedding two genres can be. According to the results (some are not listed here), Hip-Hop with any other genres can be easily separated, but Jazz & Country music is a hard one and Pop & R&B is also tough to deal with. The lyrics of Hip-Hop is largely influenced by the African Americans culture, thus a unique style to pick words. Jazz and Country are melodious and soft, with lyrics look like a poem or in a story-telling way. It is not surprising that Pop and R&B resemble so much as Pop includes many diverse styles, and it is more like a mixture of different genres. Meanwhile, R&B is a music genre that combines elements of rhythm and blues, pop, soul, funk, hip hop and electronic music.

3.3.2 Text-CNN with word embedding

Settings (default: 80% will be training part and 20% will be evaluation, ConceptNet word embedding)	Training Accuracy	Evaluation Accuracy
Filter size: 3*300, 4*300, 5*300 (all with depth 100, first 20 words)	0.98	0.4675
Filter size: 1*300, 2*300, 3*300 (corresponding depth: 300, 150, 50, first 20 words)	0.97	0.555
Filter size: 1*300, 2*300, 3*300 (corresponding depth: 300, 150, 50, first 50 words)	0.99	0.5925
Filter size: 1*300, 2*300, 3*300 (corresponding depth: 500, 200, 100, first 100 words)	0.97	0.6475
Filter size: only 1*300 with depth 300, first 400 words	0.83	**0.67**
Filter size: 1*300, 2*300, 3*300 (corresponding depth: 500, 200, 100, first 200 words)	0.98	0.7075
Filter size: 1*300, 2*300, 3*300 (corresponding depth: 500, 200, 100, first 400 words)	0.98	**0.72**

TABLE 3.3.4 RESULTS OF TEXT-CNN ON 4 GENRES OF LYRICS WITH DIFFERENT SETTINGS

Details about settings: As the number of words of specific songs can be lower than 100, 200 or 400, I set a padding step to extend the length of words' sequence to the same size with zero vectors of size 300 (Numpy array). Use SGD as optimizer and varying parameters in the neural network to tune, the number of traning steps per epoch is set to 100. Besides, 400 words per song is on the edge of running out of my GPU memory on my personal computer (GTX 1050 2G).

The advantage of using a CNN is that it is relatively easy to converge as the loss will have a huge decrement after a small number of training epochs (150 or 200 for this case). Another benefit is that the evaluation accuracy is more stable than using MLP with word embedding, the evaluation accuracy of MLP have a ± 0.05 or even larger variation if trained again with same settings, but variation of CNN is within ± 0.01. The more stability might come from that more complicated structure of CNN have better generalization over single word and small phrases as the kernel of size 2*300 and 3*300 are set to get the semantic meaning of words combinations to give out more signal on softmax layer to classify. And there is a very interesting contrast between text-CNN with only 1*300 filter (one-word filter) and MLP with MLP, the accuracy with same dataset setting represents almost the same accuracy, my guess is that although the input data to neural networks has different format, MLP is the mean value vector (1*300) while CNN takes in a 400*300 matrix, both can only possess the semantic feature behind the single word, leading to similar performance. And with different kernel size, the better performance will come as a new field of semantic features (two-word and three-word) can be explored by the neural networks.

Frankly speaking, I am shocked at first as the accuracy of CNN with first 20 words of each song as input is lower than MLP. My expectation is that the more sophisticated structure can lead to overwhelming performance, even doubting that if this structure is a good way to tackle this task. With more words of a song provided to the network, the higher the evaluation accuracy can be. The reason behind this behavior is straightforward as with more information or more possible semantic features provided to the network, the more sparsity the neural networks can find to separate different genres of lyrics.

3.4 RNN and LSTM Results

Our attempt to train RNN network failed to converge. After the network has been trained on 40,000 lyrics, no significant loss reduction is observed on the training curve (as shown. Thus the training process is terminated and efforts were directed towards constructing the LSTM model.

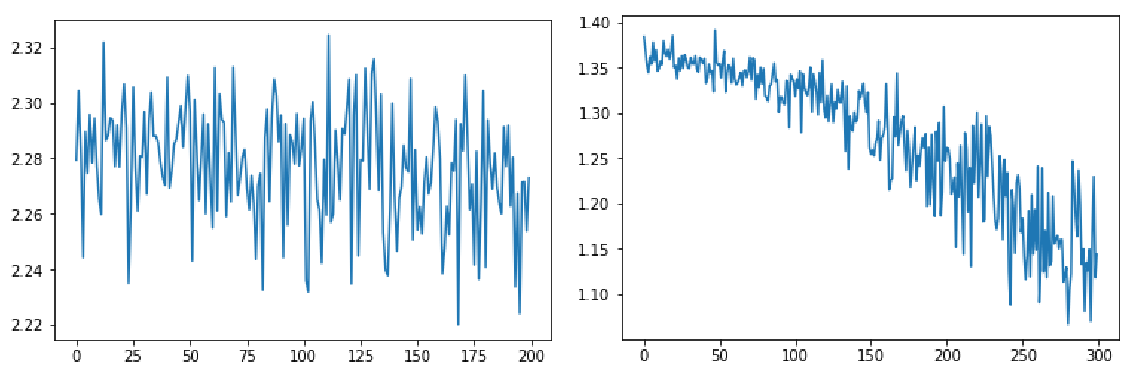


FIGURE 3.4.1 TRAINING CURVE FOR REGULAR RNN (LEFT) AND LSTM (RIGHT).

The right plot in the same figures is the training curve for LSTM, which shows significant decrease in loss function for the same number of epochs. This difference might be attributed to a few different factors such as the more complicated structure in LSTM model, its capability in better learning long-range dependency among data and a smaller training set. The LSTM network is trained on a total of 70,000 lyrics (with repetitions) for over 12 hrs.

A confusion matrix for the LSTM model is shown in the figure below. The corresponding confusion table is also shown. It's very noticeable that Hip-Hop has the highest true positive rate among the three categories, which is about 0.89. This indicates that Hip-Hop music presents a very unique style even in lyrics among other music genres. The other three categories have similar level of true positive rates fluctuating between 55% and 70%. The overall accuracy of the LSTM model is about 70%. The accuracy rate of LSTM model is lower than CNN with word embedding and takes a significantly longer time to train.

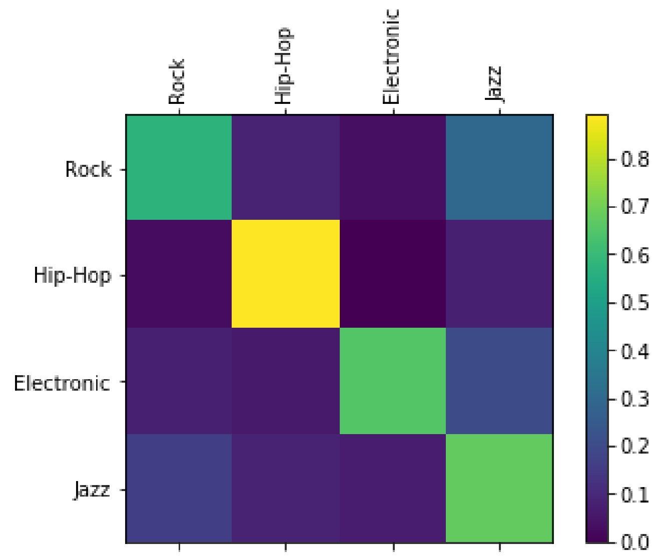


FIGURE. 3.4.2 CONFUSION MATRIX OF LSTM

In contrast to the models from previous section that implements word embedding, both the RNN and LSTM are character-level classifier. This limitation on input necessarily decreases the efficiency in training. As each word get encoded in a sequence of one-hot vector, each lyrics sequence becomes many times longer, which results in the extended training time for the model. And when an input sequence is hundreds of vectors, a regular RNN is unlikely to retain the information over long distance. This also explains why the first RNN model fails in training round

While showcases LSTM's ability in learning long-range dependency. Without word embedding, given the same length the number of possible input combinations also decreases. Since at each time step, an input can only be one letter from the alphabet instead of a word out of a dictionary. In brief, in our attempt to classify lyrics' genre, CNN with word embedding has better performance in terms of accuracy and training efficiency.

4. Conclusion

Our exploration is based on the MetroLyrics data set from kaggle. We try to classify songs based on its lyrics with three approaches, utilizing measures from classic information theory such as entropy, MLP and CNN with word embedding and a character-level LSTM model.

We have used methods like POS-tagging, clustering and information entropy to classify song genre artist and sentiment with decent result. Among them, PoS tagging and N-gram methods stand out in accuracy, which can be explained reasonably.

The relatively high evaluation accuracy of MLP with word embedding is surprising as the structure is simple and extremely fast to do training, the most credits go to word embedding itself as it has good language modeling and feature extracted, MLP just try to explore those separable linearity. The performance of Text-CNN does have potential to press, the numbers of different kernels, more different sizes of window size, retrain my own word embedding and the implementation of drop-out with a lot of other parameters to tune with. But Text-CNN with overall evaluation accuracy as 72% is already satisfying for us, proving it an efficient tool to classify the genres of lyrics.

In addition to CNN, a LSTM network classifier is implemented. Without word embedding, it parses the input word on a character-level. However, it takes significantly longer time to train this character-level LSTM network than a CNN (70,000 lyrics and over 12 hours). The resulting long sequence of input renders an implementation of regular RNN not feasible due to their lack of ability to capture long range dependency. The LSTM model has an accuracy of about 70%.

Prior to our exploration, we had doubts whether there exists a clear correlation between lyrics and music genre because we as human can fill in whatever lyrics we want to fit the rhythm of a song. Nonetheless to our surprise, the decent accuracy rates achieved with the two neural network models clearly conveys the existence of such correlation. While CNN with word embedding presents an efficient classifier, a character-level LSTM can also distinguish among different genres just with less efficiency and accuracy. Its intriguing to us how music genre, a very general abstract concept, can be extracted from the very specific permutations of letters and words in a lyric, that the song writer can compose with countless word choices and boundary-less imagination.

References

[1]. Dataset
380,000+ lyrics from MetroLyrics. https://www.kaggle.com/gyani95/380000-lyrics-from-metrolyrics

[2]. VADER sentiment analysis tools
Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[3]. Text-CNN
Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[4]. GloVe: Global Vectors for Word Representation
https://nlp.stanford.edu/projects/glove/

[5]. ConceptNet
A freely-available semantic network, designed to help computers understand the meanings of words that people use. http://conceptnet.io/

[6]. Modern Methods for Sentiment Analysis
http://districtdatalabs.silvrback.com/modern-methods-for-sentiment-analysis

[7]. Pytorch tutorial
https://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial.html

[8]. Mapping Your Music Collection
Where I got interested in dealing with music characteristics and try to analyze them with learned tools and methods.
https://www.christianpeccei.com/musicmap/

[9]. Tung, K. T., Hung, N. D., & Hanh, L. M. (2015). A Comparison of Algorithms used to Measure the Similarity between Two Documents. International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), 4(4).

[10]. Parapar, J., & Barreiro, Á. (2009, April). Evaluation of text clustering algorithms with n-gram-based document fingerprints. In European Conference on Information Retrieval (pp. 645-653). Springer, Berlin, Heidelberg.

[11]. Song Genre and Artist Classification via Supervised Learning from Lyrics
https://nlp.stanford.edu/courses/cs224n/2006/fp/sadovsky-x1n9-1-224n_final_report.pdf