University of Southern California

CSCI 544 Natural Language Processing

Collection and Classification of Lyrics Information

Student: — *USC Email:*

Tao Chen — taochen@usc.edu
Chang Su — csu272@usc.edu
Mu Yang — yangmu@usc.edu
Zhe Yang — zheyang@usc.edu

November 30, 2018

Contents

1 Overview

Along with audio, lyrics convey the spirits of artists, in a way that they can better reflect the unique style and characteristics of the author. As Natural Language Processing becoming more and more popular and intelligent, our team want to use knowledge in NLP to build a machine that can classify the genre of lyrics. This is the report of how we approach.

2 Data Collection

In this section, we provide a description of how our data is collected and relevant data cleansing operation.

2.1 Metadata collection

The metadata of the project is the lyrics. To collect raw lyrics from web resources, we need the following three steps.

(i) Collect the artist list. The starting point was to obtain a list of artists' names so we can search their works from a different website. We found a list of 10000 artists at here[1].

(ii) Fetch the song list of each artist. For every artist name, we requested a list of song records from an iTunes search API. Each record contains the metadata for the queried artist, including song names, album names, release year, the corresponding genre, etc. We then filtered out song names and artist names for the desired 8 genres [2]. Note that we won't fetch songs that has multiple genres.

(iii) Crawl lyrics online. Using, song names and artist names, we then extracted lyrics by using a web crawler crawling on a lyrics website - genius[3].

[1] https://gist.github.com/mbejda/9912f7a366c62c1f296c

[2] https://affiliate.itunes.apple.com/resources/documentation/itunes-store-web-service-search-api/

[3] https://genius.com/

2.1.1 Extra data collection

To search the lyrics of a song, we need to provide the artist and the song name. However, for certain genres, `Holiday` and `Children's Music`, the returned song names and artist names can be much noisier than other genres. For instance, in `Children's Music`, the artist names like "Baby Genius", "St.Louis Children's choir", the artists are not very famous and result in an empty result when we try to crawl lyrics from genius. This made it difficult to obtain a sufficient amount of lyrics when crawling. Therefore, we also manually collected lyrics for these two genres from different resources.

(i) For `Holiday`, 67 of them were obtained from an online recourse[4], which contains a list of 100 greatest holiday music and their corresponding artists. We obtained the list using a separate web crawler and then searched them using given the combinations of song names and artist names on genius. 67 out of 100 were founded. For other holiday music data, they were collected using the same approach described from step (i) to (iii) in section 2.1.

(ii) For `Children's Music`, 251 of them were obtained from a kid song website[5]. The lyrics were directly crawled from this website.

2.2 Data cleansing

There are three tasks we need to complete in data cleansing.

(i) Remove lyrics that are too short. There are some invalid lyrics such as "empty", "instrumental". This kind of data is usually shorter than valid lyrics. So we set the threshold of lyrics length to 200. The lyrics shorter than 200 will be removed.

(ii) Remove song structure words. After we had all lyrics in our hands, we simply removed the song structure words provided by genius, e.g.[Intro], [Verse 1], to form a continuous paragraph of lyrics.

(iii) Merge separately-collected rare genres lyrics (`Holiday`, `Children's Music`) with crawled lyrics.

[4]https://digitaldreamdoor.com/pages/music0_christmas.html
[5]https://kidsongs.com/lyrics

(iv) Remove potential duplicate lyrics by comparing their similarity with each other. If two lyrics' similarity is larger than 0.8, we consider them as duplicates and only one of them is kept.

3 Data Labeling

In this section, we provide a description of how the data was labeled.

3.1 Genre labels returned by iTunes API

For the majority of our data, genre labels were collected from the iTunes API. Note that for the manually collected holiday and children's music, the way we searched the songs was based on their genre. For these small manually collected sets, we can assume that they are 100% correct.

Although we can trust the iTunes API as a dictionary to look up songs and their corresponding genres, we still need to verify its reliability. In section 3.2, we summarize what we did to resolve this issue.

3.2 Human annotation validation

To address the problem of doubtful genre information provided by iTunes API, we chose about 80 entries from 8 genres from our data set. Each team member humanly annotates the label of the 80 entries. Since music genre is determined not only by lyrics but also by musical form and musical style, sometimes they may even overlap with each other. So it was rather difficult to tell the true genre by a human without professional musical insight. Thus, we design the labeling standard for each genre based on the research for each genre, as described in the project proposal. As a result, we found that there is 95% overlap among each member's labeling, which means that the labeling standard we proposed is reasonable.

After human-labeling the data, we cross compared them with the label returned by iTunes API, we found that over 90% of them are the same. Thus, we concludes that we can use the labels returned by iTunes API.

4 Data Collection Result

As a result, we totally fetched 30649 non-duplicate data. The distribution of each data is as following:

Genre	Number of lyrics
Children's Music	390
Blues	509
Hip-Hop	516
Holiday	679
Metal	2509
Country	4258
Pop	7940
Rock	13848
Total	30649

5 Classifier Approach

In this section, we provide a description of the classifier approach. We used four algorithms to classify the genres of the lyrics.

5.1 A baseline classifier: KNN

K-nearest-neighbor(KNN) classification is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data[6]. Also, KNN has been applied in various musical analysis problems[7]. Thus, we decided to use this as an approach to classify lyrics.

To construct our features for training, we first converted our text corpus to the Document-Term matrix, where each unique word is considered as a feature (Bag-of-word assumption). Then we transformed each value in the matrix from the number of occurrences to Term Frequency times Inverse

[6]Peterson, Leif E. "K-nearest neighbor." Scholarpedia 4, no. 2 (2009): 1883

[7]Li, Tao, Mitsunori Ogihara, and Qi Li. "A comparative study on content-based music genre classification." In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 282-289. ACM, 2003

Document Frequency(TF-IDF) to avoid various document lengths issue and common words bias issue. Class labels are mapped to 8 integers from 0 to 7.

5.2 Naive Bayes

We implemented a Naive Bayes classifier. Based on the Naive Bayes assumption - words are independent conditioned on their class, we have:

$$P\left(y|x_1,\ldots,x_n\right) = \frac{P\left(y\right)P\left(x_1,\ldots,x_n|y\right)}{P\left(x_1,\ldots,x_n\right)}$$

where

$$P\left(y|x_1,\ldots,x_n\right) = \frac{P\left(y\right)\prod_{i=1}^{n}P\left(x_i|y\right)}{P\left(x_1,\ldots,x_n\right)}$$

Hence, $\hat{y}$ would be

$$\hat{y} = \arg\max_{y} P\left(y\right)\prod_{i=1}^{n}P\left(x_i|y\right)$$

To train a Naive Bayes text classifier, we will need to find the probability(relative frequency) of a word given the class label and the class label priors. Hence we need to calculate the number of occurrences of each word in each class and the number of occurrences of each class. The TF-IDF features are naturally suitable for modeling the probability of a word. Hence, we used the same feature set as KNN.

5.3 Support Vector Machine

SVM is a powerful classifier for high-dimensional data classification. There two advantages of SVM.

1. One advantage of SVM is that it remains effective in cases where the number of dimensions is greater than the number of samples. In our task, the feature dimension is the number of unique words in a lyrics, which is typically greater than the number of lyrics samples. This makes SVM suitable for our task[8].

2. Second advantage of SVM is that it can use kernel functions to map feature spaces to non-linear spaces.

[8]https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC

To train SVM, we tried the linear kernel and RBF kernel, word-embedding and TF-IDF representation. In the result, we find that linear kernel accuracy is better.

5.4 Neural Approach

We also used neural networks to build a classifier. Neural networks is known to have good performance on analyzing the hidden features between data. We build an LSTM network to classify the lyrics. The architecture is [Embedding], [Convoluted, Max-Pooling, drop-out] x 3, [LSTM], [Softmax]. It is expressed in the following:

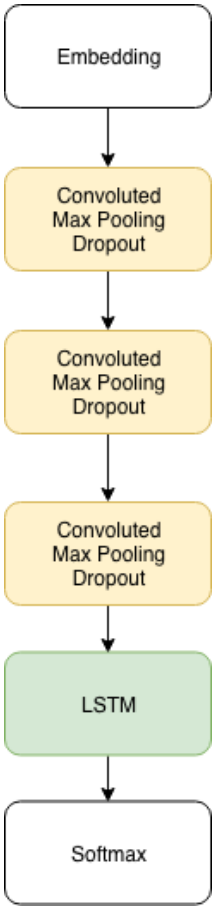

Figure 1: Structure of the neural classifier

The input feature of the LSTM network is the embedding vector, which is explained in the next section.

5.4.1 Data preprocessing for LSTM

We perform the following data preprocessing steps.

(i) Stop-words removing. We used a stop-words dictionary from NLTK.

(ii) Tokenization.

(iii) Padding. To ensure each lyrics has the same dimension, we used padding to make they have the same length.

5.5 Word Embedding

To improve the accuracy of the algorithms, we want to seek a better representation of the document. Word embedding is known for representing the semantic meaning between the words. Thus, besides bag-of-word and TF-IDF, we introduced word embedding to represent lyrics. And then compare them with TF-IDF representation.

There are many well pretrained word embeddings. We used Glove, a 300-dimensional word embeddings trained with 42 billion tokens.

5.6 Improve algorithms by Grid Search

We use grid search to perform hyperparameter tuning. Here the parameters include unigram/bigram option, whether to use IDF in TF-IDF transformation and the alpha term in Naive Bayes and SVM classifier. In KNN, the hyperparameters include k, weighting method.

6 Result Analysis

In this section, we analyze the results obtained from the classifiers. Note that lyrics contains combinations of different features, such as audio, lyrics, instrument, etc, it is usually hard to analyze the genre only from lyrics. Thus,

the overall accuracy on test data set may not be high. But after comparing to other thesis[9], our classifier is actually getting a good performance.

6.1 Evaluation Metric

Our data is imbalanced because `Children's Music` has less data in the real world. To balance the data, we only used partial data from each genres. In the end, for each genre, we extracted 390 lyrics.

For evaluation metric, we used accuracy and F1 score to compare the the performance of classifiers. Since our data is balanced, the average accuracy is indeed reflecting the overall accuracy of genres.

6.2 Results

Algorithms	Accuracy	F1 micro	F1 macro	F1 weighted
KNN Embedding	0.338	0.338	0.308	0.308
KNN TF-IDF	0.426	0.426	0.398	0.398
Naive Bayes	**0.597**	**0.597**	**0.593**	**0.593**
SVM	0.588	0.588	0.559	0.559
LSTM	0.563	0.563	0.564	0.564

6.3 KNN

KNN gave us poor result compared to the previous two classifiers. We believed the way we used word embedding was not optimal. We simply averaged out the embedding, which did not summarize the semantics of the lyrics. When we used more complicated methods (TF-IDF) to compute the features, we achieved better results. In addition, KNN is a non-parametric algorithm, it takes longer to train.

6.4 Naive Bayes

We achieved the best result by using Naive Bayes approach. Think about it, some words just appear more often than others in certain genre. For instance, country songs and children songs are easily distinguishable by looking at the

[9]Mayer, Rudolf, Robert Neumayer, and Andreas Rauber. "Rhyme and Style Features for Musical Genre Classification by Song Lyrics." Ismir. 2008.

lyrics. Without any semantic analysis, this is the most effective way to categorize a song to some genre.

6.5 SVM

Using SVM on our data set should give us similar result to Naive Bayes. The result is exactly what we expected.

6.6 LSTM

LSTM was supposed give us the best result. However, limited by our resource and computing power, we achieved very poor results from LSTM. For a LSTM to work, it must be trained with a bigger data set. And it was more time-consuming.

6.7 Confusion Matrix Analysis

In this section, we provide an analysis of the Naive Bayes confusion matrix[10].

	Bl	HH	Rk	Pp	Cr	Mt	Hd	CM
Bl	43	3	5	6	16	2	1	2
HH	1	71	1	1	2	2	0	0
Rk	8	8	27	8	15	10	1	1
Pp	10	6	13	26	16	4	0	3
Cr	10	3	6	5	53	1	0	0
Mt	3	3	13	7	5	46	1	0
Hd	2	1	1	2	2	0	69	1
CM	9	4	3	8	10	1	5	38

We printed out the confusion matrix for our best classifier, i.e. Naive Bayes with Grid Search. From the confusion matrix we can see that although most of the 8 rows have significant values on diagonal, the second row and third row have the least significant variance in the row, meaning that the corresponding classes for these two rows, i.e. Rock and Pop confused the classifier more easily than other classes. This is consistent with human intuition. Rock and Pop genres are easily misclassified by human.

[10]Bl: Blues, HH: Hip-Hop, Rk: Rock, Pp: Pop, Cr: Country, Mt: Metal, Hd: Holiday, CM: Children's Music