Lyrics Genre Classification
EECS349 Northwestern University

Chen-Feng Tsen, Sylvia Wang

About

In this project , we developed an automated music genre classification system based on the lyrics of the song. Most studies on music genre classification is based on the melody or the flow of the song, however, our task is to develop a genre classification system that can automatically classify songs based on their lyrics without the information of their audio signal. We use song lyrics as the inputs and the outputs would be the music genre.

This task can be very helpful in the aspect of songformation. In many circumstances, it comes the lyrics first, and the composer then try to compose the melody that could go along with the lyrics. This system can help the lyricists categorize the music genre of the lyrics they wrote and help them find the suitable composer to compose the melody that match the lyrics. It can also be a useful tool for lyricists when writing the lyrics for certain genre. For example, if a lyricist receive a request to write lyrics for a love song, the system will help him determine if the lyrics he wrote fit the concept or not.

Data

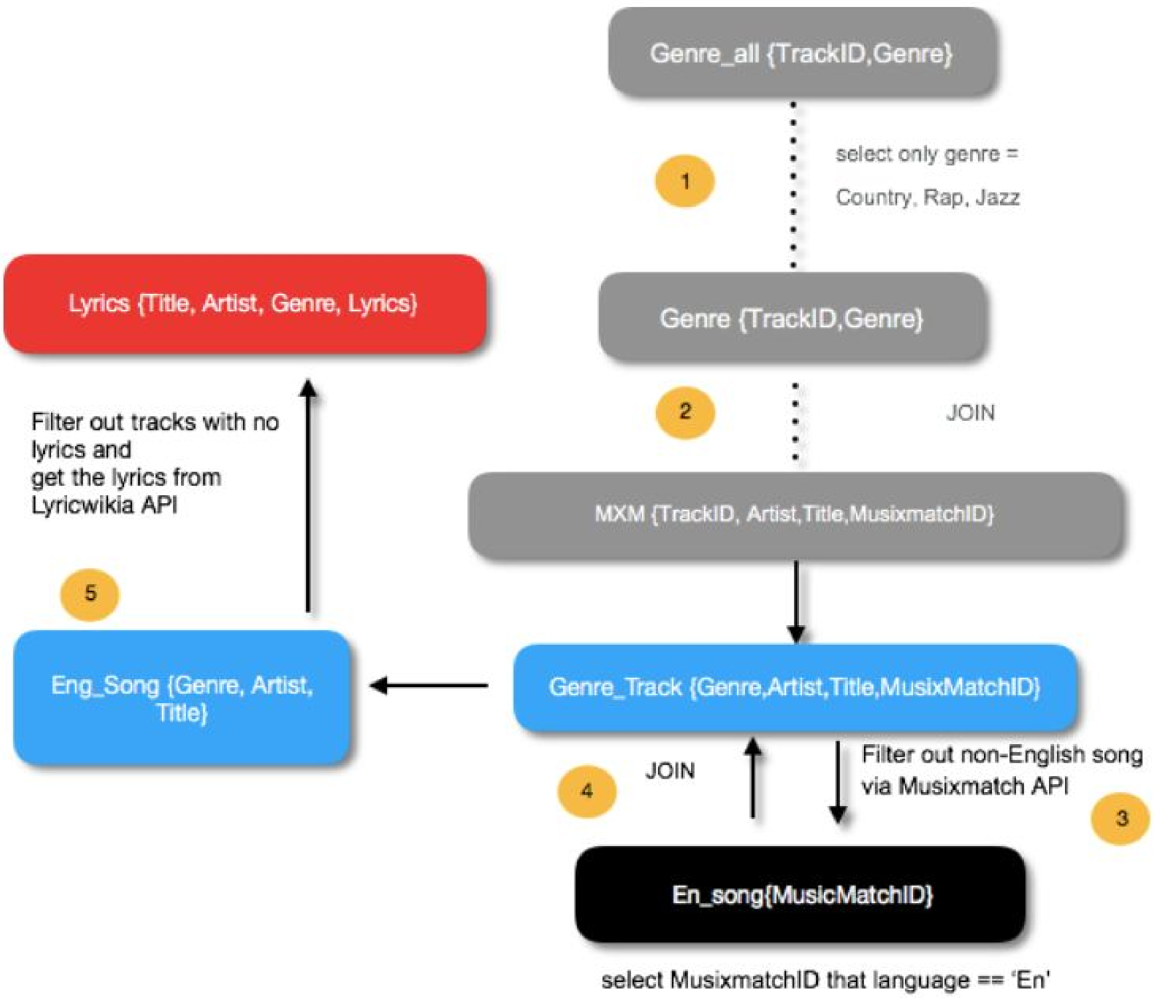
Data preprocessing is the most time-consuming part of our project. Because of the license issue, it is quite hard to get a large amount of full lyrics data with genre as label. We utilized several sources to complete this task.

We first use Million Song Dataset(MSD) website to get the track information and the corresponding genre label. MSD is a freely available collections of data for a million popular music tracks. We obtained table "Genre" containing 133283 trackID and corresponding genre label from the website.

MSD also partners with Musixmatch website and provides song lyrics in bag-of-words format. However, since we use TF-IDF method in sklearn package to get lyrics features, the bag-of-words format data is not suitable for being as our input data. The input data should either be full lyrics text or text that contains every word in the lyrics, which means, if a certain word w appear n times in the lyrics, the text should contain n numbers of word w. If we use the bag-of-words format data, we need to expand and transform the data of each song into text data that contains words with exact count. This could be pretty time-consuming considering our training dataset is large.

Therefore, we decided to find an API that could allow us to obtain the song lyrics directly. MSD website provides a table called MXM that contains trackID, artist, title and corresponding MusixmatchID that allow us to get the lyrics and other information of the track via Musixmatch API. Through Musixmatch API, we could get track information such as language with MusixmatchID. Since we only want to consider English song, we filter out non-English song through Musixmatch API. As for the lyrics part, due to the license issue, the lyrics of each song it provides only contains 30% of the full lyrics. Therefore we turned to another API called “Lyricwikia” API to get song lyrics. Lyricwikia provides an API that allows us to obtain the full lyrics of the song by entering its artist and title.

In the end, we utilize two tables- Genre, MXM from MSD website, in combination with Musixmatch API and Lyricswikia API to get the data we want. Suppose we want to develop a classifier that can classify the song into three genre: Country, Rap and Jazz. The detailed process is as follows (all the table operation was done by SQLite):



We finally got the table "Lyrics" that contain title, artist, genre and lyrics, and this is our desired dataset. Since there are lots of songs in the original dataset that are not English songs or have no Lyrics in Lyricwikia, the final song data of each genre is far less than the original dataset (<10%). We run 6000 dataset for each genre and got 400 songs in "Lyrics" table for each genre.

Approach

To develop a classifier that can distinguish music genre, we prepare three datasets with three different music genres- Country, Rap and Jazz. We generated 400 song data for each genre, and train the classifier with the dataset. We use three different datasets to train three models – one use (Country, Rap) dataset, another use (Country, Jazz) as dataset, and the other use (Country, Rap, Jazz) dataset. As for attributes, we perform text mining technique- TF-IDF on the lyrics to extract the features (the important keywords) of each song. TF-IDF here is used to weigh a keyword in a song, and assign the importance to that keywords based on the frequency it appears in that song. It can also filter out the keywords that appear frequently but in fact has low relevance to that song by finding out the frequency of the keywords' appearance on the other songs. After obtaining TF-IDF matrix of all the song lyrics, we tried several different classifiers such as Naïve Bayes, Logistic Regression, MultiLayer Perceptron (MLP) using Sklearn and DNN using Keras to fit the model. For the validation method, we use 10-fold cross validation.

Detailed implementation:

Steps	Techniques	Detail
Data preprocessing	Lyricswikia API, MusixmatchAPI, SQLite	Get three different .csv files- country.csv, rap.csv and jazz.csv, each contains 400 songs for that specific genre. Each song contains title, artist, lyrics and genre
Dataset for training the model	SQLite, numpy	Combine tables and got three datasets for training three different models- (Country, Rap), (Country, Jazz), (Country, Rap, Jazz) and randomly shuffle the order of the dataset to make it evenly distributed
Split training and testing data	Sklearn, KFold	Set K=10 to perform 10-fold cross validation, it splits the dataset into 10 portion, each time select 1 portion as test dataset and the other 9 portion as training dataset. Repeat this process for 10 times
Text feature extraction- TF-IDF	Sklearn, CountVectorizer, TfidfTransformer	Use ConutVectorizer in combination with TfidfTransformer to get TFIDF matrix. Each row vector represent a song.
Try different classifier to train the model	Sklearn, Keras	We tried on several different classifiers to fit the training data, including MultinomialNB, LogisticRegression, SGDClassifier, MLPClassifier in Sklearn and DNN structure we build in Keras.

Result

	LO	NB	SGD	MLP	DNN
(Country, Rap)	92.25%	92%	91.75%	92.86%	94.25%
(Country, Jazz)	74.75%	69.8%	72%	73.5%	75.12%
(Rap, Jazz)	93.83%	84.83%	93%	92.8%	94.75%
(Country, Rap, Jazz)	77.4%	73.35%	77.25%	76%	79.91%

Naïve Bayes classifier

Naïve Bayes classifier is a simple and efficient linear classifier. The probabilistic model of naive Bayes classifiers is based on Bayes' theorem, and it assumes that every word is independent given the class variable. Naïve Bayes can perform well under this assumption especially for small sample sizes. However, in our case, every word in the song lyrics can't be independent to each other. Each word in a sentence is relevant to each other to form a meaningful sentence. Therefore, due to the violation of the independent assumption, NB gives the relatively low accuracy compare to other classifiers. We also noticed that NB got particularly bad result on the dataset (Jazz, Rap) and (Jazz, Country), compared to the results of other classifiers. We printed out the error prediction of the test data and find out that for the (Jazz, Rap) dataset, almost all the error occurs due to the fact that Jazz songs are mislabeled to Rap. As for the (Jazz, Country) dataset, almost all the error occurs due to the fact that Jazz songs are mislabeled to Country. This phenomenon didn't happen with any other classifiers. The reason might because that the vocabulary volume in Jazz is much less than that in Rap or Country. Therefore, lots of word in Rap or Country don't appear in Jazz song and this result in low probability product. This may be the reason that why Jazz songs are always misclassified to other genres.

Logistic Regression

Logistic regression can make predictions of P(Y|X) directly based on our training data. In contrast to Naïve Bayes, it is a discriminative classifier that need not to hold under the assumption that features are independent to each other given the class variable. Also, it uses gradient descent technique to solve the problem. As a result, LO gives a better accuracy in our result.

SGD Classifier (SVM)
SGD classifier implement linear classifier with stochastic gradient descent (SGD). In our case, we use default- a linear support vector machine (SVM) as the classifier.

.
Multilayer Perceptron

A Multilayer Perceptron consists of an input layer, at least one hidden layer and also an output layer. It includes non-linear activations for each layer. MLP is trained using backpropagation, where the weights updated iteratively. It is capable of learning nonlinear models well if we tune the parameters properly.

We tried out using MLPClassifier in Sklearn and Sequential Model in Keras to build different DNN models and compared the two result.

MLPClassifier:

```
##MLP
clf = MLPClassifier(solver='lbfgs', activation='tanh',alpha=1e-3,early_stopping=True,max_iter=500,hidden_layer_sizes=(10,10,10), random_state=1)
clf.fit(X_train_tfidf, train_label)
```

We use add three hidden layers with tanh being an activation function for each layer. Each hidden layer unit is set to 10. The result is slightly better than LO classifier.

DNN in Keras:

```
clf = Sequential()
clf.add(Dense(64, input_dim=dim, init='glorot_uniform', activation='tanh'))
clf.add(Dropout(0.5))
clf.add(Dense(64, activation='tanh'))
clf.add(Dropout(0.5))
clf.add(Dense(64, activation='tanh'))
clf.add(Dropout(0.5))
clf.add(Dense(3, activation='softmax'))
rms = Adamax(lr=0.002, beta_1=0.9, beta_2=0.999, epsilon=None, decay=0.0)
clf.compile(loss='categorical_crossentropy', optimizer=rms, metrics=['accuracy'])
clf.fit(X_train_tfidf, train_labels, nb_epoch=20, batch_size=128)
score = clf.evaluate(X_new_tfidf, test_labels, batch_size=128)
```

The neural network we built using sequential model in Keras also consists of three hidden layers with activation function tanh for each hidden layer. Different from the previous one, we apply softmax activation function on the output layer, which makes the accuracy better than the previous one. MLP for multi-class softmax classification so far gives the best result among all the classifiers.

Discussion

For each training dataset, i.e. (Country, Rap), (Country, Jazz), (Rap, Jazz) and (Country, Rap, Jazz), we extracted the top 50 frequently-appeared words of each genre. The score of each word is obtained by adding up the TFIDF value of the word that appeared in each song in that specific genre. For each training dataset, we compared their 50 frequently-appeared words, extracting the words in common and the words that are different in each genre.

Country(400) v.s. Rap(400):
Words that are different in two genres (extracted from top 50 frequently-appeared words

Word	Weight
love	148.18638680975113
ll	109.19121896548684
ve	103.59636076686999
her	90.92472133156646
there	89.2225643214167
re	88.0417635070363
heart	73.75767485687243
one	61.83424436671834
never	60.38523062912831
baby	56.01393104600553
have	25.742092870953563
his	19.59351211654215
night	19.484561938305344
let	12.806003780536994
him	12.506472652153464
at	6.532536724408759
little	6.219113586316871

Table 1. Words in Rap but not in Country

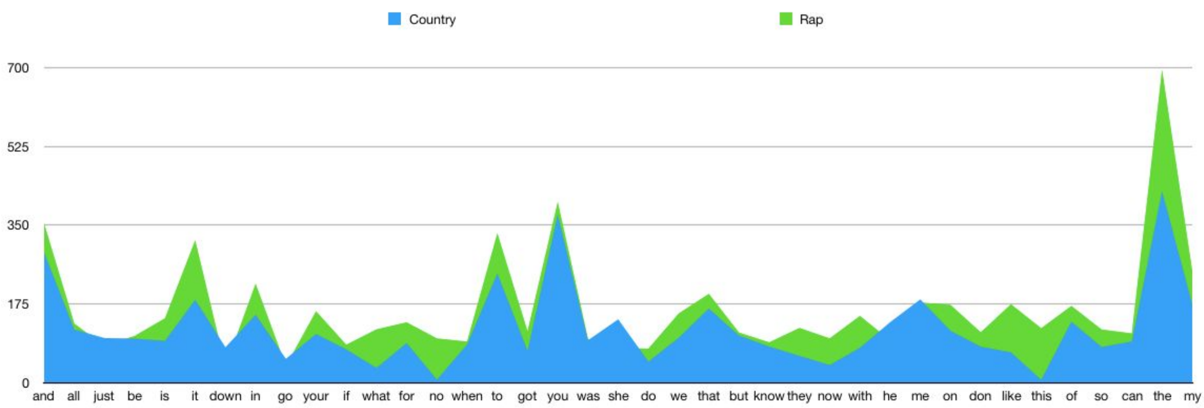
Word	Score
up	138.64090844666387
get	136.82263460137708
shit	109.6613512948264
out	103.94763445287998
nigga	103.8471902570351
ya	100.06615587391431
cause	100.03145048781535
ain	92.0940102266006
yo	84.81673536871588
from	83.15222590014862
man	77.72232404233284
who	43.99864265601583
niggas	14.58650840247656
back	14.235837357099893

Table 2. Words in Country but not in Rap

Word	Country	Rap
and	289.14068945323453	353.1535048293455
all	117.74298695064724	130.98067044383512
just	98.79617017747853	80.83566831999714
be	97.5460554339445	103.8677742359170
is	92.612503734932	142.894942120811
it	183.85729285353506	316.2191621944241
down	78.30566139409939	36.133326866282246
in	150.70994642561269	219.6741926539968
go	52.596782974482785	36.94436325202456
your	107.89128656805339	158.49124868301448
if	73.64243613238948	84.16622926182676
what	32.488074883218395	118.2824214854368
for	87.90736867573375	133.8803848597959
no	6.277274474849595	98.00643402781077
when	84.54763262492044	91.14806004658239
:	:	:

Table 3. Words appeared in both genre

(extracted from top 50 frequently-appeared words)



Country, Jazz

Word	Weight
was	107.14970067416536
her	94.43502580056942
down	91.55243394667892
they	81.83215919148101
one	79.19000690273407
up	76.4083075816162
never	57.94681567471146
at	49.810729502539104
let	36.06231630319406
out	35.03810084763374
well	7.120772256388163

Table 4. Words in Country but not in Jazz

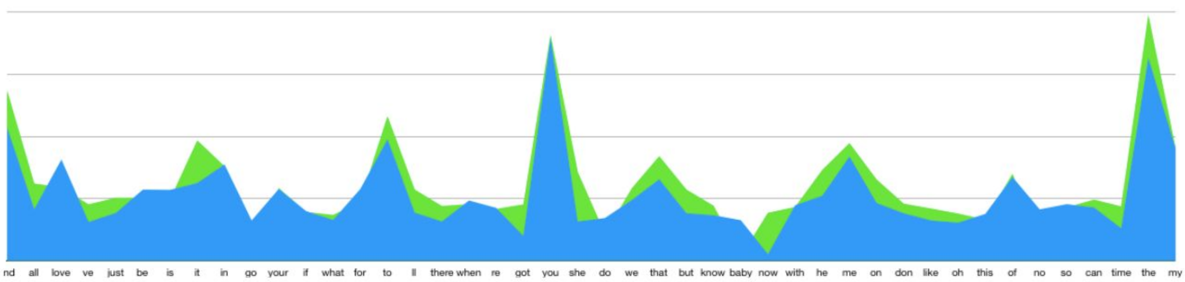
Word	Weight
will	79.18049036568033
day	68.59694368635932
as	67.09766889442285
heart	65.56124753394118
are	65.1742806625068
come	28.261310550043625
would	22.824150717656217
have	11.32333119813438
from	5.604019439366577

Table 5. Words in Jazz but not in Country

Word	Country	Jazz
and	273.73333420387905	214.79098471161763
all	124.3864829104786	83.4261543221371
love	118.13886277996868	162.95332574240533
ve	90.87606450244705	62.81089467848399
just	101.55515870640247	77.20767857980871
be	101.6373233139430	114.7176500069240
is	98.83423164361413	114.29176387773943
it	193.64361579630116	125.41759221582372
in	152.2361302896926	155.10769033011215
:	:	:

Table 6. Words appeared in both genre

(extracted from top 50 frequently-appeared words)



Rap, Jazz

Word	Weight
up	151.46094085024322
get	141.6425930501449
they	126.46155019383154
got	120.09193728870324
out	110.18145369410887
shit	107.54285106682478
was	106.45041180304449
cause	103.56495321628829
now	102.62311860493514
nigga	101.68907220990715
ya	97.80449386787916
ain	95.22037158397629
man	83.8374693154965
from	83.31366046569245
yo	82.92408888885947
down	80.62182862569203
back	78.33341426207696

Table 7. Words in Rap but not in Jazz

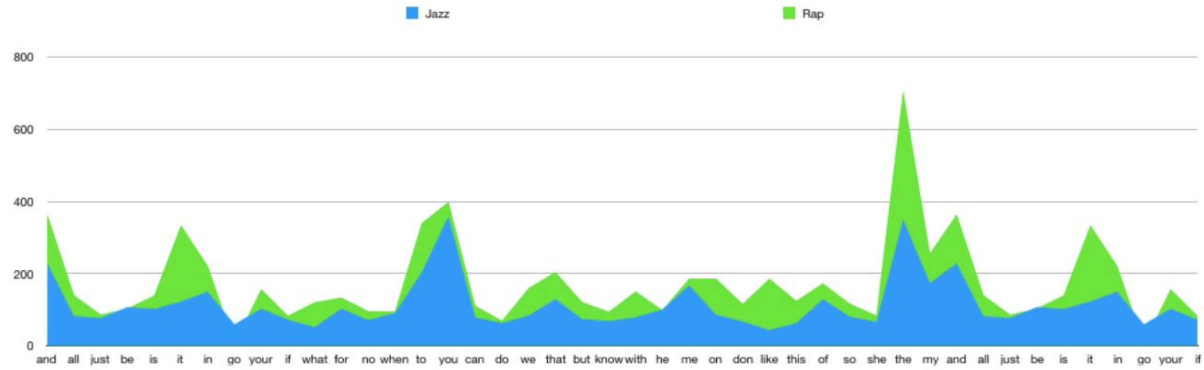
Word	Weight
love	193.7163984638336
re	89.3988008026875
ll	78.24062600052734
heart	77.5749546458581
will	75.9407825055262
day	75.11634026193997
ve	74.20865095053875
baby	68.76472945569917
there	68.28155742499322
oh	67.18269821226178
are	62.41915852755891
as	60.09400373088789
time	51.325677986735336
would	46.36339969050843
night	38.47955212460447
have	33.54381700647547
come	10.39421911045671
eyes	5.1200041911431144

Table 8. Words in Jazz but not in Rap

Word	Rap	Jazz
and	364.0189144626156	229.30015319407818
all	140.52580320925716	82.73309294273062
just	86.1894182547764	77.62594377776911
be	103.72458460081675	107.41506279763638
is	139.65080504695953	102.56240466654873
it	334.0781330820560	123.35149819519394
in	221.4543822278005	150.4855644948178
go	7.415551754385931	59.755642525559374
your	157.10909540027953	103.34194735913013
if	83.39256425217496	73.05063623574641
what	120.76374778811228	52.057248400947586
for	133.6626224642996	103.57185598823318

Table 9. Words appeared in both genre

(extracted from top 50 frequently-appeared words)



Word	Weight
get	148.35842514471813
shit	119.29504521799177
out	110.69555482547455
nigga	110.05444771848332
cause	107.26593253697804
ya	105.9157468786450
ain	99.12437670581944
yo	91.5201159037967
from	86.30602282309583
man	83.00993977721059
back	69.35333454246359
niggas	23.12342308812009
who	15.158102950984883

Table 10. Words in Country but not in Rap and Jazz

Word	Weight
get	148.35842514471813
shit	119.29504521799177
out	110.69555482547455
nigga	110.05444771848332
cause	107.26593253697804
ya	105.9157468786450
ain	99.12437670581944
yo	91.5201159037967
from	86.30602282309583
man	83.00993977721059
back	69.35333454246359
niggas	23.12342308812009
who	15.158102950984883

Table 10. Words in Rap but not in Country and Jazz

Word	Weight
her	93.9264720867841
one	59.030926160015014
never	61.840571161186766
let	33.3934401787819
his	13.734498865483683
at	13.38223373596807

Table 11. Words in Country but not in Rap and Jazz

Word	Weight
will	78.8881896396295
day	72.0360662202920
are	63.2648204495559
as	62.26248776779351
would	39.613766301251204
come	21.37516418929471
night	20.6753109544507
eyes	5.080934167200515

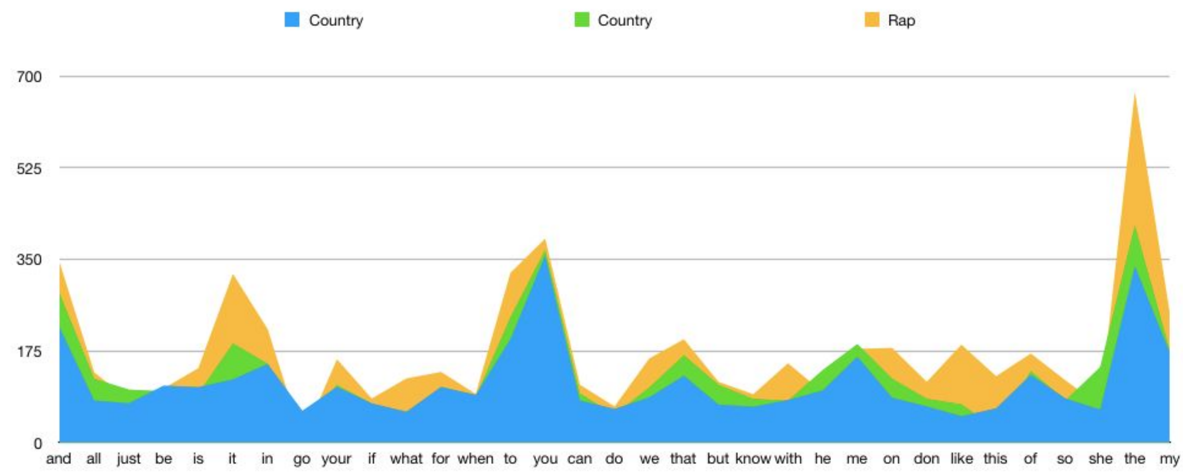
Table 12. Words in Jazz but not in Country and Rap

Word	Weight
her	93.9264720867841
one	59.030926160015014
never	61.840571161186766
let	33.3934401787819
his	13.734498865483683
at	13.38223373596807

Table 13. Words in Country but not in Rap and Jazz

Word	Rap	Country	Country
and	343.7602166566702	284.8950132269766	220.1778024181699
all	134.0709183924124	121.88500751763993	80.42207113489505
just	81.98771201224858	101.31033462969589	75.13115227872251
be	103.35492402855628	98.22248352097336	108.96689087448233
is	142.11594506616248	93.19228358141845	105.91281159277561
it	322.1241453748660	189.4322665648935	120.89196967998026
in	216.70216279889354	150.49472679411076	150.15346866272844
go	7.26083322587724	40.90168848878729	60.86680873494032
your	159.10113212872108	109.87021009583151	106.44370468269602
if	84.14409703029045	74.45670191492712	74.96367343976041
what	122.5093879981456	46.96280599699476	58.89086859604816
for	134.99373123070518	89.84652399157457	106.39840482591585
when	92.864588862561	87.12898564800227	91.23625883655653
to	324.3066532584284	239.51932442564632	197.9208080874944
you	389.51037417123274	367.4420076039436	357.2902055056127

Table 14. Words appeared in the three genre (extracted from top 50 frequently-appeared words)



From the score we obtained for each word, we observed that the top frequently appeared words (word with highest scores) in each dataset are mostly words in common (words that appeared in each genre).
However, from the histogram of common words, we can see that in general, the frequently appeared common words have similar score in each genre, and the histogram follows pretty similar trend. This indicates that these frequently appeared common words are not the main features for each genre and therefore are not the main reason for classifiers to perform different classifications.

As for frequently appeared words that are not in common, we first take a look at (Country, Rap) dataset. The scores of words in Country but not in Rap, and the scores of words in Rap but not in Country are both high. Top score 138 for "up" in Rap and top score 148 for "love" in Country. We think that the high scores (refer to high TFIDF weight) of words not in common in each genre makes (Country, Rap) dataset easily distinguishable. Therefore result in the relatively high accuracy when classifying this dataset.

For (Country, Jazz) dataset, we also observe frequently appeared words that are not in common for each genre.The top scores of words in Country but not in Jazz is 107, and the top scores of words in Jazz but not in Country is 79. Compared to the highest score in the dataset (>200), the scores are relatively low. We think that this could be one of the reason that makes the classifiers harder to classify this dataset. The accuracy for classifying this dataset is relatively low.

For (Rap, Jazz) dataset, we again observe frequently appeared words that are not in common for each genre.The top scores of words in Rap but not in Jazz is 151 for the word "up", and

the top scores of words in Jazz but not in Rap is 193 for the word "love". Again, both scores are high. This makes the dataset easily distinguishable and results in relatively high accuracy when classifying this dataset.

For (Country, Rap, Jazz) dataset, the top scores of words in Rap but not in Jazz and Country is 148 for the word "get", the top scores of words in Country but not in Rap and Jazz is 93.9 for the word "her", and the top scores of words in Jazz but not in Rap and Country is 78.8 for the word "will". The top scores of words in Rap but not in Jazz and Country is relatively high. We printed out the error classification of test data and find out most error predictions occur due to the classifier misclassify Country music to Jazz genre or Jazz music to Country. This agree with our expectation, that Rap is the most distinctive type in this dataset and is the easiest distinguishable one.

Conclusion and Future work

In our project, we spend lots of time on data preparation. We are glad that with 400 data for each genre, we got pretty good results on classifying two different genres. For classifying three different genres, we can classify ~80% of song correctly with DNN. It is also quite interesting to extract features for each genre and got a brief idea of how lyrics of different genre type in common or different to each other. There are more works that we want to continue to do:

- Develop a classifier that can distinguish more genres. Now we have developed a genre classification system that can classify lyrics into Country, Rap and Jazz. In the future, we would like to add more genres such as Pop, Rock, Punk…etc. This could be quite challenging since some genres are not easily distinguishable if we only use TFIDF to get features. We might need to find additional features to complete the task.
- Since now we use TFIDF to extract the feature of the lyrics, we don’t consider the meaning of the sentence, or the meaning of sequence of the words. This could be one of the reason that we can’t achieve very high accuracy when classifying lyrics into three genres. In the future, we would like to try LSTM or RNN, to fit our model and improve our accuracy.