Lyric-Based Music Genre Classification

by

Junru Yang
B.A.Honors in Management, Nanjing University of Posts and Telecommunications, 2014

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Lyric-Based Music Genre Classification

by

Junru Yang
B.A.Honors in Management, Nanjing University of Posts and Telecommunications, 2014

Supervisory Committee

Dr. Kui Wu, Co-Supervisor
(Department of Computer Science)

Dr. George Tzanetakis, Co-Supervisor
(Department of Computer Science)

Supervisory Committee

Dr. Kui Wu, Co-Supervisor
(Department of Computer Science)

Dr. George Tzanetakis, Co-Supervisor
(Department of Computer Science)

ABSTRACT

As people have access to increasingly large music data, music classification becomes critical in music industry. In particular, automatic genre classification is an important feature in music classification and has attracted much attention in recent years. In this project report, we present our preliminary study on lyric-based music genre classification, which uses two n-gram features to analyze lyrics of a song and infers its genre. We use simple techniques to extract and clean the collected data. We perform two experiments: the first generates ten top words for each of the seven music genres under consideration, and the second classifies the test data to the seven music genres. We test the accuracy of different classifiers, including naïve bayes, linear regression, K-nearest neighbour, decision trees, and sequential minimal optimization (SMO). In addition, we build a website to show the results of music genre inference. Users can also use the website to check songs that contain a specific top word.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

Dr. Kui Wu, who spent countless hours to guide me and improve the writing of this project.

Dr. George Tzanetakis, who came up with the main and original idea for this report.

My parents, who always be supportive and love me, whatever happens.

It's not that I'm so smart, it's just that I stay with problems longer.
Albert Einstein

DEDICATION

I dedicate this project to my peers in the Department of Computer Science who have always supported and encouraged me.

Chapter 1

Introduction

Music always plays an important role in people's life. Coupled with different cultures, different kinds of music formed, evolved, and finally stabilized in several representative genres, such as classical music, pop music, rock music, and Hip hop. In the era of big data, people are faced with a huge amount of music resources and thus the difficulty in organizing and retrieving music data. To solve the problem, music classification and recommendation systems are developed to help people quickly discover music that they would like to listen. Generally, music recommendation systems need to learn users' preferences of music genres for making appropriate recommendations. For example, the system would recommend a list of rock music if a specific user has listened to rock music a lot. In practice, however, many pieces of music have not been classified, and thus we need a way to automatically classify the music into the right genre.

In this project, we mainly focus on the genre classification of songs. A song consists of two main components: instrumental accompaniment and vocals [16]. The vocals mainly include pitch, gender of singer, and lyrics. Extensive work has been done on music genre classification based on acoustic features of a song, e.g., the instrumental accompaniment, the pitch and the rhythm of the song. Nevertheless, little attention has been paid to song classification based on a song's lyrics, which only include non-acoustic features. This project explores the potential of classifying a song's genre based on its lyrics.

Our main idea is to extract the information from a song's lyrics and identify features that help music genre classification. In particular, we consider the frequency of words and identify those words that appear more frequently in a specific music genre. This intuition is based on our observation that different music genres usually uses different words. For instance, country songs usually include words such as “baby”, “boy”, “way”, and Hip hop may include words like “suckers,” “y’all,” “yo,” and “ ain’t”.

The analysis of lyrics relies on natural language processing (NLP) techniques [2]. Based on data mining, NLP allows computers to understand human languages. In this report, we will use the concept of n-gram in NLP. With n-gram, features can be effectively selected and applied in various machine learning algorithms.

1.1 Structure of the Report

The rest of the project report is organized as follows.

Chapter 1 introduces the current situation of music classification and the problem that the report is solving.

Chapter 2 summarizes existing ideas and approaches in the area.

Chapter 3 gives the procedure for data collection and data cleansing.

Chapter 4 proposes the features that are used for later music genre classification.

Chapter 5 presents our experiments and the results of feature analysis.

Chapter 6 contains how we show the results by building a website to help users easily use our system.

Chapter 7 concludes the project.

Chapter 8 proposes future research.

Chapter 2

Related Work

With the popularity of data mining, text mining techniques have been implemented in music classification for a long time. There is quite a lot existing work on text mining and classification, including genre detection [14], authorship attribution [24], text analysis on poetry [23], and text analysis on lyrics [7].

In the early stages of development, music classification was mainly based on acoustic features. Audio-based music retrieval has made great success in the past, e.g., classification with signal processing techniques in [8] and [28]. Lyric-based music classification, however, was not considered effective. For instance, McKay et al. [17] even reported that lyric data performed poorly in music classification.

In recent years, lyric-based music genre prediction has attracted attention, especially after the invention of Stanford's natural language processing (NLP) techniques. Some research has combined lyrics and acoustic features to classify music genres, leading to more accurate results [10]. Lustrek [29] used function words (prepositions, pronouns, articles), specific words in genre, vocabulary richness, and sentence complexity in lyric-based song classification. He also used decision trees, naïve Bayes, discriminant analysis, regression, neural networks, nearest neighbours, and clustering. Peng et al. [19], on the other hand, focused on the model study. They described the use of upper-level n-grams model. Another approach is reported by Fell and Caroline [7], which combines n-gram model and different features of a song content, such as vocabulary, style, semantics, orientation towards the world (i.e., "whether the song mainly recounts past experience or present/future ones" [7]), and song structure. Their experiments showed the classification accuracy between 49% and 53% [18].

Recently, many interesting algorithms and models have been proposed in the field of text mining. Tsaptsinos [27] used a hierarchical attention network to classify music genre. The method replicates the structure of lyrics and enables learning the sections, lines or words that play an important role in music genres. Similarly, Du et al. [6] focused on the hierarchical nature of songs.

Deep learning is also a popular approach to song classification. According to Sigtia and Dixon [22], random forest classifier using the hidden states of a neural network as latent features for songs can achieve an accuracy of 84% over 10 genres in their study. Another method using temporal convolutional neural networks is described by Zhang et al.[31]. Surprisingly, their result achieved an accuracy up to 95%.

So far, most studies on lyric-based classification use rather simple features [12], for example, bag-of-words. Scott and Matwin enriched the features by synonymy and hypernymy information [21]. Mayer et al. [16] included part of speech (POS) tag distributions, simple text statistics, and simple rhyme features [11].

Chapter 3

Data Processing

Our research is based on lyrics. We collect the lyric data and manually label the data. After that, we split the data into two datasets, one for training and the other for testing.

3.1 Data Collection

Song lyrics are usually shorter in length than normal sentences, and they use a relatively limited vocabulary. Therefore, the most important characteristic is the selection of words in a song. Therefore, the most important characteristic is the words in a song. We used data from the Million Song Dataset (MSD) [1]. MSD is a free-available collection of data with metadata and audio features for one million contemporary popular songs. It also includes links to other related datasets, such as musiXmatch and Last.fm, that contain more information.

The musiXmatch is partnered with MSD to bring a large collection of song lyrics for academic research. All of these lyrics are directly associated with MSD tracks. In more detail, musiXmatch provides lyrics for $237,662$ songs, and each of them is described by word-counts of the top $5,000$ stemmed terms (i.e., the most frequent words in all the lyrics) across the set. Also, the lyrics are in a bag-of-words format after the application of a stemming algorithm. [20]

The other linked dataset, Last.fm, contains tags for over $900,000$ songs, as well as pre-computed song-level similarity [25]. The categories are obtained using the social tags found in this dataset, following the approach proposed in [13].

We integrate the above three dataset for this project. We then clean the combined dataset by removing irrelevant information.

3.2 Data Pre-processing

Although the musiXmatch and Last.fm have already included the data we need, we still need to manually process the data into a form that is directly usable for our project.

According to musiXmatch's website [1], there are two tables in the lyrics dataset: "words" and "lyrics." The "words" table only has one column ***'word'***, where words are ordered according to their popularity. Thus the ROWID of a word represents its corresponding popularity. The "lyrics" table contains 5 columns: ***'track_id'***, ***'mxm_tid'***, ***'word'***, ***'count'***, ***'is_test'***.

In the Last.fm dataset, we have tags associated with trackIDs. First of all, since there are lots of tags not related to music genres, we need to identify songs with genre tags from the whole dataset. Here, seven genres are picked up for the study: rock, pop, electronic, jazz, metal, blues, and Hip hop. In this step, we wrote code in Python, and imported SQLite into the Python code to get the wanted "trackID" of each picked genre, which is exactly the same trackID from the musiXmatch dataset. For example, the code below shows how we get all trackIDs for the tag 'rock'.

```
tag = 'rock'
sql = ''SELECT tids.tid FROM tid_tag, tids, tags WHERE tids.ROWID=
    tid_tag.tid AND tid_tag.tag=tags.ROWID AND tags.tag='%s'" %lastfm(
    tag)
res = conn.execute(sql)
data = res.fetchall()
print map(lambda x: x[0],data)
```

After getting all trackIDs in each genre, we added the genre information to the "lyrics" table. Using SQLite queries, we can manage data and compile them to get the desired format. After that, we divided the data into two subsets: training set and testing set. The training set contains 70% of the data we have, while the rest of 30% is for test. Table 3.1 shows the amount of lyric data by music genres. The musiXmatch website reports that musiXmatch dataset includes lyrics for 77% of all MSD tracks [5]. However, in the genres selected, only 37% of the tracks have lyrics information. In some specific music genres, like classical and jazz, the songs only have acoustic information but no lyrics. For other genres, some lyrics might simply be missing for various reasons.

Genre	**Training**	**Testing**
Rock	49,524	21,224
Pop	33,887	14,523
Electronic	19,433	8,328
Jazz	8,442	3,618
Metal	9,600	4,114
Blues	5,732	2,456
Hip hop	8,188	3,509
Total	134,806	57,772

Table 3.1: The number of songs in each music genre, split into training set and testing set

Chapter 4

Features

In the project, we experimented with some advanced features that model different dimensions of a song's lyrics, to analyze and classify songs.

4.1 Bag-of-Words

With bag-of-words, a lyric is represented as the bag of its words. Each word is associated with the frequency it appears in the lyric. For instance, consider the following two sentences:

1. John likes to listen to music. Mary likes music too.

2. John also likes to watch movies.

After converting these two text documents to bag-of-words as a JSON object, we get:

1. $BoW1 = \{"John" : 1, "Likes" : 2, "listen" : 1, "music" : 2, "Mary" : 1, "too" : 1\}$

2. $BoW2 = \{"John" : 1, "also" : 1, "likes" : 1, "watch" : 1, "movies" : 1\}$,

where the order of elements does not matter. In the above example, we apply the frequency with a term weighting scheme [15]: $TFIDF$ (i.e., term frequency $\times$ inverse document frequency). The scheme sets a text file as $\boldsymbol{d}$, a term, or a token, as $\boldsymbol{t}$. The term frequency $tf(t, d)$ represents the number of times that term $\boldsymbol{t}$ appears in the text file $\boldsymbol{d}$. The text file frequency $f(d)$ is denoted by the number of text files in the collection that term $\boldsymbol{t}$ occurs. For the purpose, the process of assigning weights to terms according to their importance for the classification is called *term-weighing.* And the weight $TFIDF$ is computed as:

$$TFIDF(t, d, N) = tf(t, d) \times \ln(\frac{N}{f(d)})$$

where $\boldsymbol{N}$ is the number of text files in the text corpus. The weighting scheme considers a term as important when the term occurs more frequently in a text file, but less frequently in the rest of the file collection.

4.2 Part of Speech (POS)

The past works have shown that POS statistic was a useful feature in text mining. In general, POS explains how a word is used in a sentence. In English, there are nine main word classes of a speech: nouns, pronouns, adjectives, verbs, adverbs, prepositions, conjunctions, articles, and interjections [3]. In Natural Language Processing, these POS can be tagged by Part-Of-Speech Tagger(POS Tagger) [26], which is a piece of software that reads text and assigns parts of speech to each word. Intuitively, a writer's use of different POS can be a subconscious decision determined by the writer's writing style. If artists in a given genre exhibits similar POS style, and artists in different genres have different POS style, then POS style in lyrics could be used as an effective feature in genre classification.

In the experiments, we defined word classes into nouns, verbs, articles, pronouns, adverbs, and adjectives. We counted the numbers of each word classes. According to Stanford NLP research, POS can also be an indicator of the content type in a song. For instance, frequent use of verbs reveals a song that is about action, and in this case it is probably that the song is more story oriented. If adjective words are used, the song might be more descriptive in purpose. Furthermore, to generate the top words for each music genre, before using POS Tagger, the top words in a song is most likely article words such as "a", "the", "an"; or prepositions such as "in", "of", and "on". Since these words are less informative, we filtered out those words and only kept on the nouns, verbs, adverbs and adjectives.

Chapter 5

Experimental Results

Our evaluation consists of two steps: In the first step, we generated 10 top words for each music genre, and classified music by their genres. In the second step, we used the classical bag-of-words indexing as well as the features introduced in the previous section. We ran the machine learning algorithms in Weka [9] to get the result. Weka includes tools for data pre-processing, classification, regression, clustering, association rules, and visualization. We tested several algorithms in Weka to classify music genres.

5.1 Experiment 1: Top Words of Each Music Genre

We studied seven genres: rock, pop, electronic, jazz, metal, blues, and Hip hop. After gathering the lyrics of each music genre using the tags offered by Last.fm and the corresponding trackIDs, the code below shows how we get word counts for each word in a song.

```
sql = ``SELECT word, count FROM lyrics WHERE track_id='%s'" %my_track
res = conn.execute(sql)
data = res.fetchall()
```

We ordered the words by frequency. A partial result is shown in Table 5.1, which shows the result of top words in rock songs. We can see that the top words are mostly pronouns like “I”, “you”, “me”, or articles like “a”, “the”, which are not informative in identifying music genres. In other words, to get the expected vocabulary, a good solution is to filter out these less informative words and keep only informative nouns, verbs, adjectives and adverbs instead. POS Tagger can help handle this problem. It marks every words with their part of speech as super tags, then cleans the rough result by extracting function POS, which is set to nouns, verbs, adjectives and adverbs (refer to Figure 5.1).

Words	**Count**
the	206,592
I	206,483
you	206,300
and	201,235
love	199,401
a	199,189
baby	187,257
be	187,252
for	186,342
have	174,285
on	132,453
it	131,794
......	......

Table 5.1: The partial result of top words in rock music

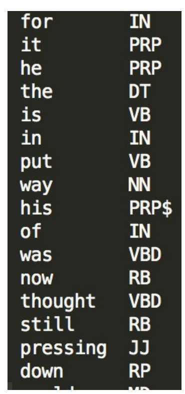

Figure 5.1: Words marked by POS Tagger before filtering

Figure 5.2 to Figure 5.8 reveal the top 20 unigram (i.e., special case of n-gram where $n = 1$) for each music genre. It is clear to see the lyrical differences and similarities. Some music genres pop out lexically, like Hip hop, which uses lots of dominant slang, or metal, which is mainly about death and violence. However, other genres are lexically similar, such as jazz, blues, and pop. There are plenty of reasons for the similarity among these music genres. One element might be jazz is a music genre that developed from roots in blues and ragtime. As we mentioned before, many jazz and blues lack lyrics. Also, pop music usually describes a kind of music that is popular, although it has developed separately from other music genres.

Figure 5.2: Top 20 words in rock music

Figure 5.3: Top 20 words in pop music

Figure 5.4: Top 20 words in electronic music

Figure 5.5: Top 20 words in jazz music

5.2 Experiment 2: Music Genre Classification

After the first experiment, we split the dataset into training set and testing set randomly. Each song in the dataset was paired with a dictionary of lyrics containing the word counts for each word. We used the two features and the training set to train the classifiers in Weka. Then, we ran the classifiers on the test set, without using the genre information, and compare the classification results with the genre tags in the test set. After testing all the classifiers with all features, the following are the result of accuracy (Figure 5.9).

Furthermore, Table 5.2 shows the confusion matrix (i.e., a table that shows the performance of a classifier [30]) of naïve Bayes, which directly offers the number of classified songs and mistakes in each music genre.

Figure 5.6: Top 20 words in metal music

Figure 5.7: Top 20 words in blues music

Figure 5.8: Top 20 words in Hip hop music

We also compared the results of different classifiers. Table 5.3 shows the results. From the result, we conclude that the naïve Bayes method results in the best performance in accuracy.

5.2.1 Feature Analysis

We performed a more detailed analysis on effectiveness of each of our features. Table 5.4 and Figure 5.10 summarize the performance and contribution of each features in our experiment.

Bag-of-Words As we expected before, the feature bag-of-words played the most important role in the classification (66.2%), which was proved by its high performance alone. It is reasonable since bag-of-words has the most lexical and semantic information.

We noticed that the contribution of bag-of-words was different in the different classifiers. The feature performed better in Bayes algorithms, compared with other

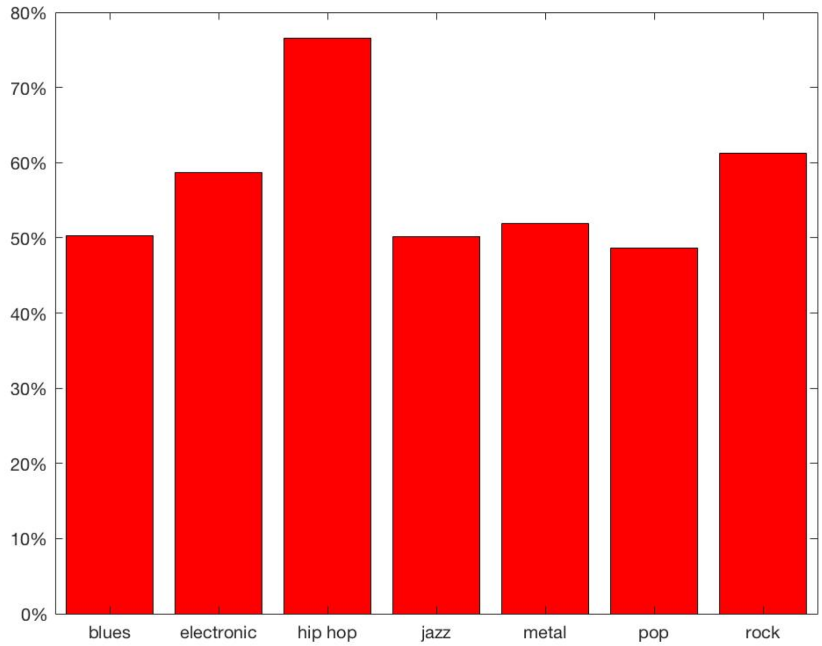

Figure 5.9: Accuracy of naïve Bayes classifier

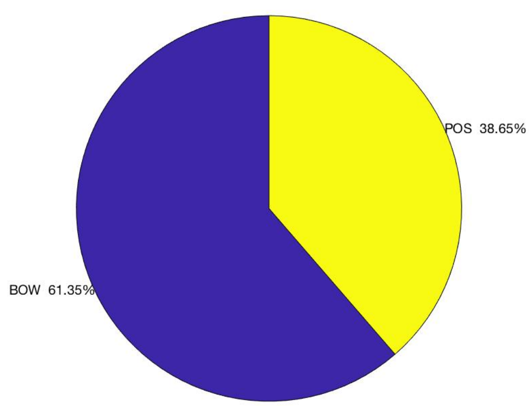

Figure 5.10: Feature contributions in naïve Bayes

classifiers, such like k-Nearest Neighbors.

Rock	Pop	Electronic	Jazz	Metal	Blues	Hip hop
12,980	3,829	1,010	842	576	1193	794
3,728	7071	268	1,077	318	1644	417
683	829	4889	432	784	201	151
894	341	200	1816	107	201	59
943	273	29	511	2135	171	52
281	463	94	76	185	1235	122
153	215	22	147	53	239	2680

Table 5.2: Confusion matrix of naïve Bayes.

Classifiers	**Accuracy** (%)
Naïve Bayes	65.71
Linear Regression	61.25
K-nearest Neighbour	63.83
Decision Trees	50.32
SMO	50.53
ZeroR	49.76

Table 5.3: The accuracy of different classifiers.

Part-of-Speech POS performed surprisingly well when used alone (Table 5.4). It scored an over 63% accuracy (see Table 5.5 in Hip hop) in almost all classifiers we used. The result shows that POS is a strong indicator of style so it can make significant distinctions in data. Moreover, POS may perform better in one particular genre than in others. For example, Hip hop has a very distinctive use of POS, while rock have more variation in their style. However, in general, POS has performed well in all the classifiers, and it is possible that the more the data, the better the POS performs.

Features	**Accuracy** (%)
Bag-of-words	79.91
Part-of-speech	63.34

Table 5.4: The performance for two features in naïve Bayes

Rock	**Pop**	**Electronic**	**Jazz**	**Metal**	**Blues**	**Hip hop**
4133	2970	2560	1212	513	152	307
2781	3313	1260	825	931	372	1129
1219	893	3726	318	346	237	201
231	297	203	575	163	79	178
382	209	128	82	572	70	28
124	186	39	284	93	201	97
221	184	77	50	22	23	1039

Table 5.5: The confusion matrix for POS in each genre using partial testing set.

Chapter 6

A Web Application

We implemented a web application to allow users to use our song classification system easily. We built a web service via which users can find all the songs with the words in the lyrics. We also display the results showing how lyrics predict music genres.

6.1 The Platform

We build our web service using Wix. Wix is a cloud-based web development platform. Wix is built on a freemium business model. It is a convenient tool which allows users to create HTML5 websites. However, users have to purchase packages in order to connect their sites to their own domains, add e-commerce capabilities, or buy extra data storage and bandwidth.

With the blank template provided by Wix, we uploaded tables and figures to show the classification result (Figure 6.1 to Figure 6.2). The site menu includes home page, result page, services page, and contact page. The home page briefly introduces the project, including two pictures that show our collected data and all genres in music. The result page exhibits what we have achieved from the research. Basically, the page shows charts and tables that we discussed above in an interactive way. The service page is a function page that links top words in each music genre and the songs. More details of the service page will be disclosed in next subsection. Last but not least, the contact page includes the contact information of the project.

Figure 6.1: A screen shot of the home page

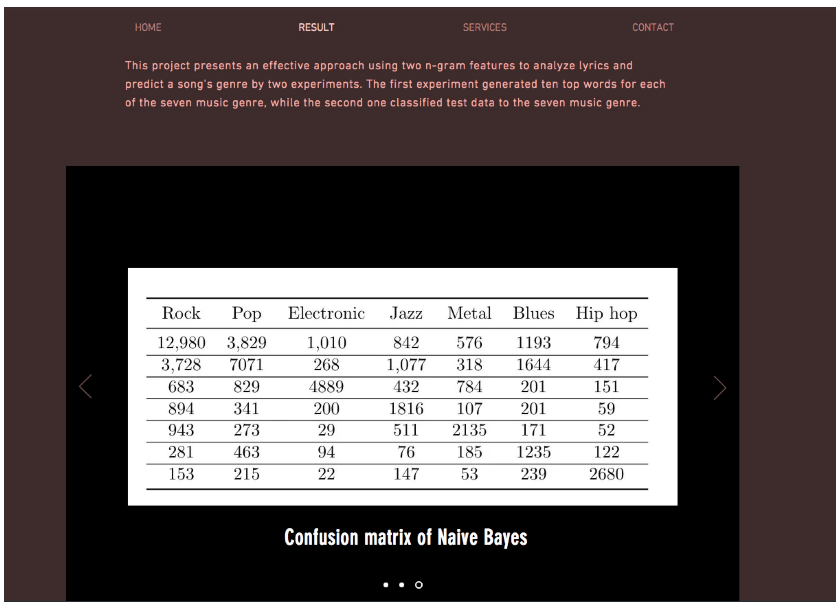

Figure 6.2: A screen shot of the result page: an exhibition of experiments results

6.2 Technical Details behind the Service Page

As we mentioned before, the biggest challenge in the web service is to manage and query data. Wix provides wix code and wix-data API to help users build their database.

Database in Wix is made up of collections. Each collection can be thought of as a table of data, like a spreadsheet. And there are a sandbox version and a live version of the data, and as such it requires users to edit their data twice in both

Content Manager for sandbox version and Database App for live version. Collections are created using the site structure tool in the sidebar. Once we created the data collections, the next step was to import collection data using wix-data API. Since the API requires data in JSON format, we need extra processing before importing the data. We used an online tool [4] to convert the CSV data to JSON format. The data use the "field key" from the data collection we just created, in order to identify which fields need the data source. Furthermore, we write code using wix-data API to import our data. In the service page, we listed the top 10 words of each music genre, and made the top words as text buttons, so that they could be linked to the songs whose lyrics contain the same word. When the user clicks a word, the corresponding top 12 songs will be displayed. In addition, the user can view those songs with their lyrics by clicking the 'view lyrics' button, which will lead to a new page. The new page shows a table that contains titles and lyrics of the 12 songs, which are grabbed from our database. Figure 6.3 shows the screen-shot of the top 12 song names after clicking on the top word "love".

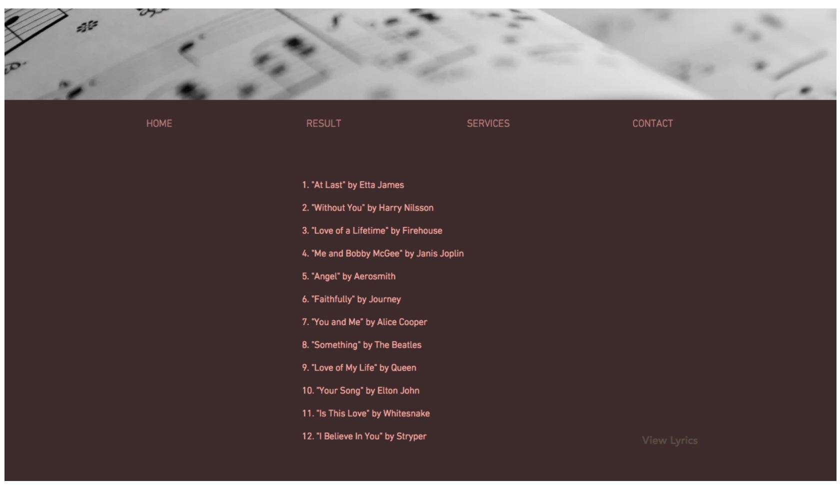

Figure 6.3: The top 12 songs with the word "love"

Chapter 7

Conclusion

In this project, we showed how lyrics-based statistical features could be employed to classify different music genres. Our experiments show interesting and promising results. We generated top 20 words of seven music genres and used a limited feature set derived from song lyrics and definitely no acoustic elements to classify over 65% of songs correctly. In particular, we tested and analyzed the performance of two features: bag-of-words and part-of-speech. Also, we compared several classification algorithms in Weka, including naïve Bayes, linear regression, k-nearest neighbour, decision trees, and SMO. Our results show that it showed the naïve Bayes is the most accurate classifier. Finally, we built a web service to allow users to easily use our song classification system.

To summarize, lyrics-based music mining is still in its infancy, and as such our project would benefit the music retrieval community by providing a basic building block for more sophisticated music genre predication systems.

Chapter 8

Future Work

The project could be further extended in various ways:

- Add more training data. Although we have tried hard to collect as much data as we could, the lyrics source still needs further expansion. During the experiments, we found that some music genres lack enough training data compared to other genres. We expect that with more training data available, certain features such as POS may lead to better results.

- Add more features. In this project, we only considered two features in classification. There might be other features that can be used to improve the accuracy of our classifiers. For instance, some research used the length of a sentence in lyrics as a feature, while some used the title of the song.

- Combine other models or algorithms. This project used n-gram model and classifiers in Weka for the study. If we introduce other model or new classification algorithms, we may obtain better results.

Bibliography

[1] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[2] Gobinda G Chowdhury. Natural language processing. *Annual Review of Information Science and Technology*, 37(1):51–89, 2003.

[3] Wikipedia contributors. Part of speech — wikipedia, the free encyclopedia, 2018. [Online; accessed 3-April-2018].

[4] CSVJSON. Csvjson, 2018. [Online; accessed 3-April-2018].

[5] Danny Diekroeger. Can song lyrics predict genre? [Online; accessed in March 2018].

[6] Wei Du, Hu Lin, Jianwei Sun, Bo Yu, and Haibo Yang. A new hierarchical method for music genre classification. In *Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), International Congress on*, pages 1033–1037. IEEE, 2016.

[7] Michael Fell and Caroline Sporleder. Lyrics-based analysis and classification of music. In *Coling*, 2014.

[8] Jonathan Foote. An overview of audio information retrieval. *Multimedia Systems*, 7(1):2–10, 1999.

[9] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[10] Yajie Hu and Mitsunori Ogihara. Genre classification for million song dataset using confidence-based classifiers combination. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 1083–1084, New York, NY, USA, 2012. ACM.

[11] Fang Jiakun. *Discourse Analysis of Lyric and Lyric-based Classification of Music*. PhD thesis, National University of Singapore, 2016.

[12] Seonhoon Kim, Daesik Kim, and Bongwon Suh. Music genre classification using multimodal deep learning. In *Proceedings of HCI Korea*, HCIK '16, pages 389–395, South Korea, 2016. Hanbit Media, Inc.

[13] Florian Kleedorfer, Peter Knees, and Tim Pohle. Oh oh oh whoah! towards automatic topic detection in song lyrics. In *Ismir*, pages 287–292, 2008.

[14] Mitja Lustrek. Overview of automatic genre identification 1. Technical report, Jozef Stefan Institute, Department of Intelligent Systems, Jamova 39, 1000 Ljubljana, Slovenia, 01 2007.

[15] Rudolf Mayer, Robert Neumayer, and Andreas Rauber. Rhyme and style features for musical genre classification by song lyrics. In *Ismir*, pages 337–342, 2008.

[16] Rudolf Mayer and Andreas Rauber. Musical genre classification by ensembles of audio and lyrics features. In *Proceedings of International Conference on Music Information Retrieval*, pages 675–680, 2011.

[17] Cory McKay, John Ashley Burgoyne, Jason Hockman, Jordan BL Smith, Gabriel Vigliensoni, and Ichiro Fujinaga. Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features. In *ISMIR*, pages 213–218, 2010.

[18] Hasan Oğul and Başar Kırmacı. Lyrics Mining for Music Meta-Data Estimation. In Lazaros Iliadis and Ilias Maglogiannis, editors, *12th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI)*, volume AICT-475 of *Artificial Intelligence Applications and Innovations*, pages 528–539, Thessaloniki, Greece, September 2016. Part 10: Mining Humanistic Data Workshop (MHDW).

[19] Fuchun Peng, Dale Schuurmans, and Shaojun Wang. Language and task independent text categorization with simple language models. In *In Proc. of HLT-NAACL 03*, pages 110–117, 2003.

[20] Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[21] Sam Scott and Stan Matwin. Text classification using wordnet hypernyms. In *Use of Wordnet in Natural Language Processing Systems: Proceedings of the Conference, Pages 3844. Association for Computational Linguistics*, pages 45–52, 1998.

[22] Siddharth Sigtia and Simon Dixon. Improved music feature learning with deep neural networks. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6959–6963, 2014.

[23] Dean Keith Simonton. Lexical choices and aesthetic success: A computer content analysis of 154 shakespeare sonnets. *Computers and the Humanities*, 24(4):251–264, Aug 1990.

[24] Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the Association for Information Science and Technology*, 60(3):538–556, 2009.

[25] Bhavika Tekwani. Music mood classification using the million song dataset, 2016. [Online; accessed in April, 2018].

[26] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[27] Alexandros Tsaptsinos. Lyrics-based music genre classification using a hierarchical attention network. *CoRR*, abs/1707.04678, 2017.

[28] George Tzanetakis and Perry Cook. Marsyas: A framework for audio analysis. *Organised Sound*, 4(3):169–175, 2000.

[29] Vedrana Vidulin, Mitja Luštrek, and Matjaž Gams. Training a genre classifier for automatic classification of web pages. *Journal of computing and information technology*, 15(4):305–311, 2007.

[30] Wikipedia contributors. Confusion matrix — Wikipedia, the free encyclopedia, 2018. [Online; accessed 23-April-2018].

[31] Xiang Zhang and Yann LeCun. Text understanding from scratch, 2015. cite arxiv:1502.01710.