Predicting Artist by the Lyrics

Ruiwen Zhang & Yeqing Liu

Abstract

Authorship detection is an essential part of the Natural Language Process, it helps us to identify the authors of texts. In this project, authorship detection was applied to predict the artists of lyrics. We dealt with this problem by implementing different classification machine learning algorithms to different features that were vectorized by TF-IDF, Bag of words and distribution of function words in each song lyrics where the function words are from the essential word list of 176. Among all these models we utilized, the Support Vector Machine (SVM) classifier on TF-IDF gained the best accuracy of 79%, and Logistic Regression achieved the highest accuracy of 68.4% on bag of words and Naive Bayes performed the highest accuracy of 46% on distribution of function words.

Objective

In this paper, we are interested in the prediction of performers based on song lyrics. Predicting performers of lyrics is similar to authorship attribution. Applications of authorship attribution range from resolving discussions about disputed authorship to forensic linguistics. In this project, we define authorship distribution as the automatic identification of the author of a text on the basis of linguistic features of the text. The one focus in authorship attribution is feature selection (vectorization method). The other focus lies on the classification techniques to be applied. We used TF-IDF, Bag of Words and function words as the feature sets. The machine learning methods we implemented are: Logistic Regression, SVM, Random Forest and Naive Bayes. We applied each feature to above four different learning methods separately and evaluated each model performance by four metrics: accuracy, precision, recall and F1 score.

Dataset

We used 57650 song lyrics data from Kaggle. There are four columns within the dataset: Artist, Song name, Link to a webpage with the song and Lyrics of the song. The song lyrics are from 643 different artists. There is a considerable disparity in the number of songs for each artist. The maximum number of songs for the artist is 191, the minimum number of songs for the artist is only 1. Thus, our dataset is imbalanced. In the paper *Dealing with Data Imbalance in Text Classification,* Cristian (2019) introduced the Synthetic Minority Over-sampling Technique (SMOT) to deal with data imbalance. The algorithm takes each minority class sample and introduces synthetic samples along the line joining the current instance and some of its k nearest neighbours from the same class. However, we only classify four artists with the most number of songs in this project since the ability and capacity of our computers is limited and it would take a long time if we predict all 643 artists.

Background of Our Approach

The most important two stages of detecting authorship are feature selection and machine learning methods. The features we selected need to have high predictive value for the categories to be learned. The

learning methods we chose need to be efficient and are used to learn to categorize new documents by using the features selected in the first stage. We did some research about this topic, and found many relative works. TF-IDF is a common vectorization method. However, a paper (Shahzad Qaiser, 2018) mentioned that TF-IDF is not able to identify words from the same stem and this would lead to a low accuracy of using TF-IDF. To solve this problem, we may use some stemming algorithm to remove commoner morphological before applying TF-IDF.

The author of 'Comparing Frequency- and Style-Based Features for Twitter Author Identification' introduced another vectorization method, bag of word, which performs well in authorship detection. This paper also described how SVM and Naive Bayes work for classification. Luyckx and Daelemans (2005) indicated that studies on authorship attribution have shown that common function words such as 'of' can be an effective marker of author style. Function words may characterize an author's writing so effectively because they are not entirely under his or her control. They also introduced that Part-Of-Speech as one feature tends to work well for authorship classification. In this study, authors implemented SVM and Maxent learning methods to predict the artists. Although many other papers also implemented some clustering algorithms, we would not use them since they are unsupervised machine learning algorithms, which is not available for evaluation.

Methodology

Before modelling, the text should be preprocessed. Firstly, we tokenized the text into single words by applying regular expression and Snowball stemmer, which runs faster than the traditional Porter stemmer. Snowball stemmer works even if the stem is not identical to the morphological root of the word. Stemming eliminates the weakness of the TF-IDF. Therefore, we could implement vectorization methods to these tokens directly.

For vectorization, we considered three ways: TF-IDF, Bag of Words and distribution of function words in each song lyrics. TF-IDF measures the weight of each token to show the importance within the document according to frequency of the word.

$$(TF - idf)_{t,d} = TF_{t,d} * idf_t$$

, where $TF_{t,d} = \frac{the\ number\ of\ terms\ in\ the\ doc}{total\ number\ of\ words\ in\ the\ doc}$

represents the term frequency and

$$idf_t = \log\left(\frac{the\ number\ of\ docs}{the\ number\ of\ terms\ in\ all\ documents+1}\right)$$

represents the inverse document frequency. For example, if the frequency of a word is low in the corpus, but high in a specific document, then this word would gain a higher weight in this document. Applying TF-IDF to lyrics helps us to detect what representative words an artist usually use.

Bag of words is a way of extracting features from the text for use in machine learning algorithms. In this approach, we use the tokenized words for each observation and find out the frequency of each token. Each sentence is treated as a document and makes a list of all words with frequency from all the documents excluding the punctuation. Then we keep a count of the total occurrences of most frequently used words and count the frequency of these words in each document. In general, it's a collection of words to represent a sentence with word count and mostly disregarding the order in which they appear.

Function words are words that have little lexical meaning or have ambiguous meaning and express grammatical relationships among other words within a sentence, or specify the attitude or mood of

the speaker. They signal the structural relationships that words have to one another and are the glue that holds sentences together. Thus, they form important elements in the structures of sentences.

Since this project is a classification problem, we would implement logistics regression, SVM, random forest, and Naive Bayes. We will apply these machine learning algorithms to both TF-IDF, Bag of words and function words. To improve the performance of each model, we plan to utilize cross validation on each algorithm. Before getting results, our hypothesis is that SVM and Naive Bayes would give better accuracy than other models. Compared with logistic regression, SVM works better for high dimension data (many features). Random forest is an ensemble bagging algorithm, it applies bootstrap method to avoid overfitting. Naive Bayes classifier predicts the class by computing conditional probabilities according to the Naive Bayes theorem:

$P(class|w) = \frac{P(w|class)*P(class)}{P(w)}$, where

$P(W_i|C) = \frac{count(w_i,c)+1}{\sum count(w_i,c)+the\ number\ of\ words\ in all\ docs}$.

The prime Naive Bayes is the independence between features. Obviously, features of this dataset obey this rule.

Results

We applied logistic regression, SVM, random forest and Naive Bayes to TF-IDF, Bag of words and function words, and printed out their accuracy, precision, recall and F1-score, respectively.

For TF-IDF, SVM shows the best result for TF-IDF with a 79% accuracy, while random forest gives the lowest accuracy, which is 52.6%. Logistic regression also performs not badly, its metrics are very close to SVM's.

	Accuracy	Precision	Recall	F1-score
Logistic regression	77.6%	80.8%	77.6%	0.782
SVM	**79.0%**	**79.4%**	**79.0%**	**0.791**
Random forest	52.6%	56.3%	52.6%	0.518
Naive Bayes	68.4%	75.5%	68.4%	0.684

Fig1. Evaluation values of TF-IDF

	Accuracy	Precision	Recall	F1-score
Logistic regression	**68.4%**	**71.0%**	**68.4%**	**0.692**
SVM	63.1%	67.6%	63.1%	0.639
Random forest	53.9%	57.4%	53.9%	0.544
Naive Bayes	67.1%	66.0%	67.1%	0.659

Fig2. Evaluation values of BOW

Logistic regression makes the best performance evaluated by all four metrics, with 68.4% of accuracy, 71.0% of precision,

68.4% of recall and 0.692 of f1-score, while random forest had the worst performance among the four models.

	Accuracy	Precision	Recall	F1-score
Logistic regression	44.7%	48.7%	44.7%	0.454
SVM	40.7%	47.6%	40.7%	0.417
Random forest	35.5%	38.3%	35.5%	0.346
Naive Bayes	**46.0%**	**49.1%**	**46.0%**	**0.469**

Fig3. Evaluation values of Function Words

According to the evaluation metrics from Bag of words and function words, neither of them works as well as TF-IDF. All of these four algorithms displayed extremely low accuracy on function words method.

Overall, TF-IDF produces the best results among three vectorization methods, and the accuracy of SVM for TF-IDF archives 79%. However, this result is still not good enough. To figure out the reason, we printed out the confusion matrix of SVM for TF-IDF. As the table shows below, the most wrongly predicted songs are from Gordon Lightfoot, which indicates that there are 3 songs by Gordon Lightfoot were predicted as Donna Summer's, incorrectly. (The value is shown in the following confusion matrix in bold type)

	Donna Summer	Gordon Lightfoot	Bob Dylan	George Strait
Donna Summer	15	1	2	0
Gordon Lightfoot	**3**	20	2	1
Bob Dylan	1	2	9	2
George Strait	0	0	2	16

Fig4. Confusion matrix of SVM for TF-IDF

By this, we also computed the TF-IDF weights of Donna Summer and Gordon Lightfoot. The top 10 words with highest TF-IDF weights of Donna Summer and Gordon Lightfoot are very similar to each other, this shows that lyrics from Gordon Lightfoot are easily predicted as Donna Summer's.

Discussion

Though some lyrics may not be produced by the performers themselves, we believe that every singer has their own styles of performance, as well as their lyrics. For example, the lyrics for a rapper is definitely different from the lyrics for a blue singer. Therefore, the question of 'who wrote the lyrics' would not be a problem anymore, and we could only focus on 'who performed the song'.

For future work, we plan to consider more advanced methods of tokenization. Current tokenization method could remove stopwords and get stemming, but it could not deal with words, such as 'we'll' and 'don't'. In addition, we would try more algorithms to model features. As we hypothesized before, the result indicates

that the SVM model for TF-IDF gives the best prediction. Random forest works badly for either TF-IDF or Bag of words, and it is like a black-box algorithm, we cannot track what features the model selects. While SVM finalized an accuracy close to 80%, the result is still not satisfactory enough. Hence, we may select other more sophisticated deep learning algorithms, such as LSTM and CNN to our dataset (Nils Schaetti, 2017). More researches will be also done on how the performance of different models would be if we classify song genres first and then predict the artists within the same genre. Even though the artists in a single song genre will probably have more similar styles than between genres, there are still distinct patterns between different artists. Narrowing down the scope of song genres would probably help us to extract the essential features in the song lyrics from different artists.

Reference

1. Shahzad Qaiser, 2018, *Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents*
2. Rachel M. Green, *Comparing Frequency- and Style-Based Features for Twitter Author Identification*
3. Cristian Padurariu, 2019, *Dealing with Data Imbalance in Text Classification*
4. K Luyckx, W Daeleman, 2005, *Shallow Text Analysis and Machine Learning for Authorship Attribution*
5. Nils Schaetti, 2017, *UniNE at CLEF 2017: TF-IDF and Deep-Learning for Author Profiling*
6. Adam Sadovsky, Xing Chen, *Song Genre and Artist Classification via Supervised Learning from Lyrics*