Song Lyric Analysis and Classification By Artist

Capstone Proposal
Udacity Machine Learning Nanodegree

Greg Mogavero
June 9, 2017

DOMAIN BACKGROUND

Machine learning is commonly used to perform natural language processing (NLP), a "field of computer science, artificial intelligence and computational linguistics concerned with the interactions between computers and human (natural) languages, and, in particular, concerned with programming computers to fruitfully process large natural language corpora [1]." One particular use case for NLP is analyzing song lyrics to classify them by artist. A successful machine learning algorithm would not only have to take into account the songwriter's lexicon, but also pick up on the unique subtleties of the artist's style in order to differentiate artists who write about similar topics.

Efforts have already been made by the machine learning community to classify songs based on their lyrics by genre and artist. Sadovsky and Chen [2] used Maxent and SVM classifiers to accomplish this task fairly successfully. Using no acoustic information whatsoever, they were able to achieve 70-80% artist classification accuracy. Because they used Bag of Words for feature selection, they remark that they might have been able to achieve higher accuracy if they did some sort of semantic analysis.

PROBLEM STATEMENT

Given song lyrics as input and the corresponding artists as labels, I will attempt to build a supervised learner that can classify new songs by artist. Input will be truncated or padded during preprocessing so that each sample is a fixed size.

DATASETS AND INPUTS

For this project, I will be using a dataset comprised of 57,650 song lyrics scraped from LyricsFreak by Sergey Kuznetsov [3]. The file itself is a .csv and has four columns: artist, song, link, and text. For the purposes of this project, I will only be using the text column as input and the artist column as labels. Furthermore, I will not be using the entire dataset. Rather, I will handpick artists with enough song samples in order to get meaningful results.

It should be noted that anyone can submit lyrics to LyricsFreak, which means that the data is most likely not curated and could have discrepancies in formatting among songs by the same artist, as well as errors. I will attempt to alleviate some of these issues in preprocessing.

SOLUTION STATEMENT

I will attempt to solve the problem of classifying song lyrics by artist by training a Long Short Term Memory Neural Network. These models, when trained on text data, can "remember" information they have seen in the past. I hypothesize that the ability to learn this contextual information will help the learner distinguish songs by different artists who have similar lexicons, but different styles.

BENCHMARK MODEL

I will use a model that makes predictions by random guessing as my benchmark. Therefore, the expected accuracy score of the benchmark is $1/L$, where L is the number of class labels assuming a uniform sample distribution. Additionally, I will compare my results with those obtained by Sadovsky and Chen, although this comparison must be taken with a grain of salt because I will not be using the same dataset.

EVALUATION METRICS

I will be using accuracy as the evaluation metric for this project. The accuracy score is represented by the following formula:

$$accuracy\ score = \frac{number\ of\ correct\ predictions}{total\ number\ of\ test\ samples}$$

PROJECT DESIGN

I will build this project in a Jupyter notebook, using the Keras framework for the machine learning model itself.

Since I will not be using the entire dataset, I will first choose 10 artists with a large repertoire of songs. Then, I will for the most part be following Jason Brownlee's Keras implementation of an LSTM for sequence classification from his blog [4]. For the preprocessing step, I first need to transform the text data into numerical data. Following a code example provided by TensorFlow [5], I will replace each word with an integer representing its frequency rank among all words in the dataset. I will restrict the dataset to the 5000 most frequent words, and all other words will be zeroed out. Then I will pad and truncate each sample to a fixed length of 500 words. Finally, I will use an embedding layer to vectorize each sample.

For the model itself, I will use a Keras LSTM layer, followed by a Dense layer. After evaluation, I will try to improve the model's performance by adding Dropout layers and tuning hyperparameters. I will also try adding a convolutional and max pooling layer before the LSTM layer like in Brownlee's article.

Results will be compared against the benchmark model and the accuracy scores reported by Sadovsky and Chen's models.

For further analysis of the model, I will test it on different subsets of the data; for example, using only two artists, and choosing a group of artists in the same genre.

REFERENCES

[1] "Natural language processing," [Online]. Available: https://en.wikipedia.org/wiki/Natural_language_processing.

[2] A. Sadovsky and X. Chen, "Song Genre and Artist Classification via Supervised Learning from Lyrics," 2006. [Online]. Available: https://nlp.stanford.edu/courses/cs224n/2006/fp/sadovsky-x1n9-1-224n_final_report.pdf.

[3] "55000+ Song Lyrics | Kaggle," [Online]. Available: https://www.kaggle.com/mousehead/songlyrics.

[4] J. Brownlee, "Sequence Classification with LSTM Recurrent Neural Networks in Python with Keras," [Online]. Available: http://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-python-keras/.

[5] [Online]. Available: https://github.com/tensorflow/tensorflow/blob/r1.1/tensorflow/examples/tutorials/word2vec/word2vec_basic.py.