Automatic Song Tagging

Zach Zhang , Binqian Zeng , Lingshan Gao

DS1003 Spring 2017

Abstract

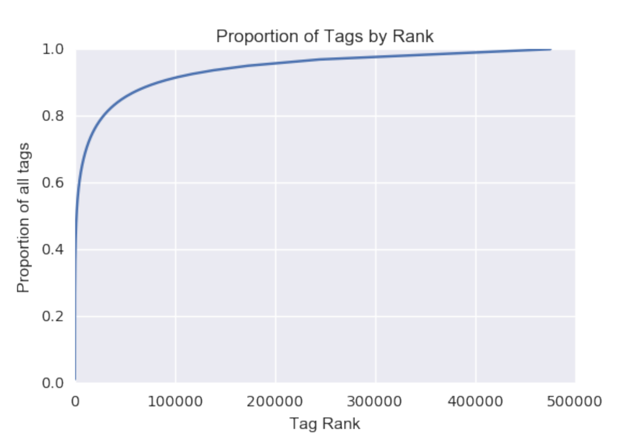
Song tagging is the task of assigning labels to songs, (Blues, Rock, Pop ...). This problem is important for music services for search as well as recommendation. We have created a model for automatically tagging new songs based on song lyrics. We found that using deep learning, we can effectively capture the variability in text data.

Data

For this project we augment the Last.fm song dataset with

Last.fm

- 505,216 songs with one or more associated tags
- 475,080 unique tags (many infrequent)
- Constrain project to top 100



Lyrics

- 180,000 song lyrics collected using PyLyrics API
- 20,000 most frequent words used as features
- Exclude stop words and words that appear in more than 90 percent of documents

Jimi Hendrix - Hear My Train A Comin'
Well, I wait around the train station Waitin' for that train Waitin' for the train, yeah Take me home, yeah From this lonesome place Well, now a while lotta people put me down a lotta changes My girl had called me a disgrace...
Tags - Rock, Blues, Psychedelic Rock, '60s', Guitar...

Multi-Label Classification

Each song can have multiple tags so the problem is a multi-label classification problem. In MLL, we learn a mapping $f : X \Rightarrow [0,1]^{|Y|}$ These problems are typically handled in one of the two ways.

1. **Problem Transformation** - We adjust the data to to work within a standard classification framework. One method of doing this is binary relevance. This technique treats the presence/absence of each label as a separate binary classification problem.
2. **Algorithm Adaptation** - We adjust the algorithm to directly support multi-label classification. For example, using a neural network we can directly predict the binary label vector.

Features

We tested several different strategies for representing text.

Bag of Words

1. Presence or absence of a word or n-gram as individual features
2. Ignores word order

Word Embedding

1. Words mapped to vectors in continuous space
2. Captures semantic relationship between words
3. Input is a sequence of word vectors

[1] Sebastian Ruder, http://sebastianruder.com/word-embeddings-1/

Baseline

To establish a baseline, we train several binary relevance classifiers using Logistic Regression, Naive Bayes, and SVM. We conducted hyper-parameter grid search to establish best models.

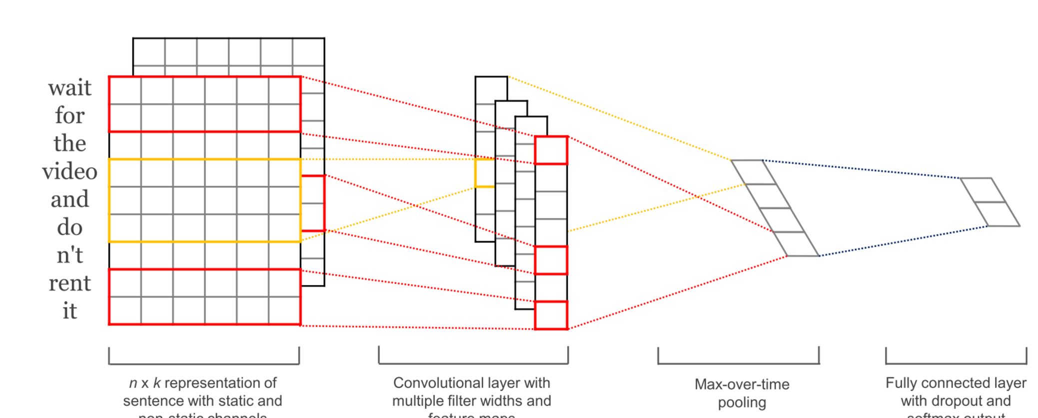
However, we found that these simple models are unable to capture the richness of language which motivated us to test deep learning techniques.

Deep Learning

Deep learning models use hidden variables to build rich representations of data. Deep methods have achieved state of the art results in computer vision as well as NLP problems. We tested two different deep models: Constitutional Neural Network and Gated Recurrent Network.

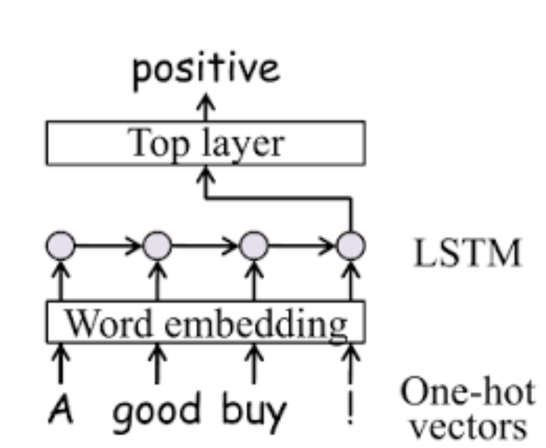
Constitutional Neural Network

- Use multiple Convolutions across word embeddings
- Extract n-gram features in text



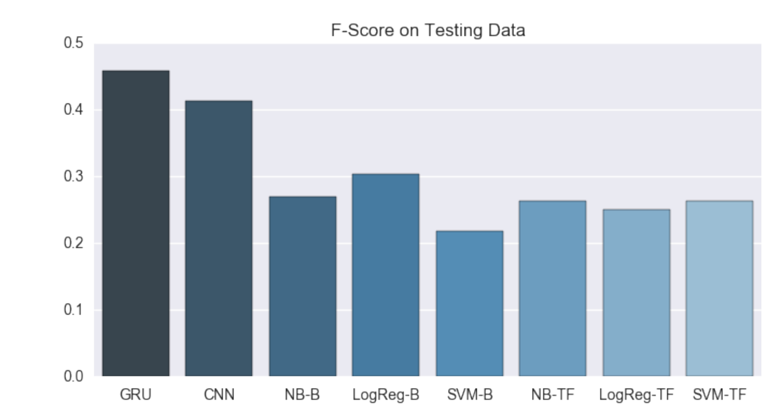
Gated Recurrent Network

- Network maintains internal state that encodes history
- internal state updated at each time step with new input



Results

We trained our models on 80% of the data and tested against the other 20%. All models use a vocabulary size 20,000. Concluded that GRU produced the best accuracy on this task.



Analysis

We analyzed the behavior of our models to gain insight into how they make predictions.

Figure 1: For logistic regression we assessed feature importance by taking the features weight and dividing by the variance. We show a subset of the features for the tag 'Rock' Here.

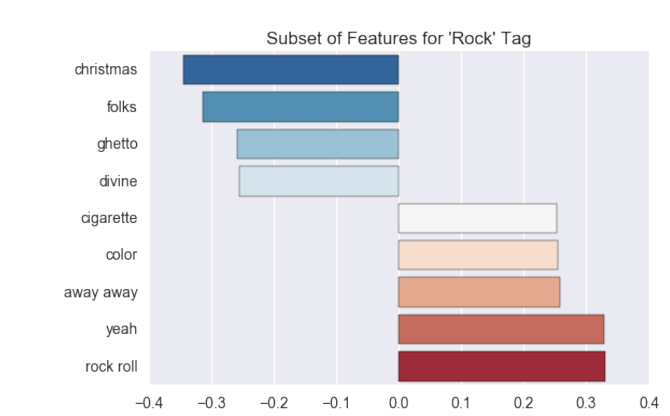
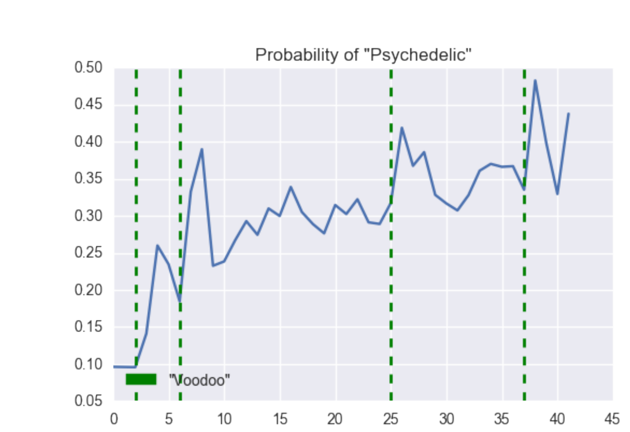


Figure 2: GRU's probability of 'psychedelic' over time for Jimi Hendrix - Voodoo Child. Model has learned that 'voodoo' is evidence of 'psychedelic'



Conclusion

In conclusion, we have created an effective model of automatic song tagging using deep learning. Our GRU model is able to outperform classification baselines by taking into account word order as well as using word embeddings.