LYRICS-BASED MUSIC GENRE CLASSIFICATION

Mike Fan

W266 Spring 2021

Background

LYRICS-BASED MUSIC GENRE CLASSIFICATION USING A HIERARCHICAL ATTENTION NETWORK

Alexandros Tsaptsinos
ICME, Stanford University, USA
alextsap@stanford.edu

ABSTRACT

Music genre classification, especially using lyrics alone, remains a challenging topic in Music Information Retrieval. In this study we apply recurrent neural network models to classify a large dataset of intact song lyrics. As lyrics exhibit a hierarchical layer structure—in which words combine to form lines, lines form segments, and segments form a complete song—we adapt a hierarchical attention network (HAN) to exploit these layers and in addition learn the importance of the words, lines, and segments. We test the model over a 117-genre dataset and a reduced 20-genre dataset. Experimental results show that the HAN outperforms both non-neural models and simpler neural models, whilst also classifying over a higher number of genres than previous research. Through the learning process we can also visualise which words or lines in a song the model believes are important to classifying the genre. As a result the HAN provides insights, from a computational perspective, into lyrical structure and language features that differentiate musical genres.

1. INTRODUCTION

Automatic classification of music is an important and well-researched task in Music Information Retrieval (MIR) [25]. Previous work on this topic has focused primarily on classifying mood [13], genre [21], annotations [27], and artist [9]. Typically one or a combination of audio, lyrical, symbolic, and cultural data is used in machine learning algorithms for these tasks [23].

Genre classification using lyrics presents itself as a natural language processing (NLP) problem. In NLP the aim is to assign meaning and labels to text; here this equates to a genre classification of the lyrical text. Traditional approaches in text classification have utilised n-gram models and algorithms such as Support Vector Machines (SVM), k-Nearest Neighbour (k-NN), and Naïve Bayes (NB).

In recent years the use of deep learning methods such as recurrent neural networks (RNNs) or convolutional neural networks (CNNs) has produced superior results and represent an exciting breakthrough in NLP [16, 17]. Whilst linear and kernel models rely on good hand-selected features, these deep learning architectures circumvent this by letting models learn important features themselves.

Deep learning has in recent years been utilised in several MIR research topics including live score following [7], music instrument recognition [20], and automatic tagging [3]. In many cases, these approaches have led to significant improvements in performance. For example, Kum et al. [18] utilise multi-column deep neural networks to extract melody on vocal segments while Southall et al. [34] approach automatic drum transcription using bidirectional recurrent neural networks.

Neural methods have further been utilised for the genre classification task on audio and symbolic data. Sigtia and Dixon [31] use the hidden states of a neural network as features for song on which a Random Forest classifier was built, reporting an accuracy of 83% among 10 genres. Costa et al. [6] compare the performance of CNNs in genre classification through spectrograms with respect to results obtained through hand-selected features and SVMs. Jeong and Lee [14] learn temporal features in audio using a deep neural network and apply this to genre classification. However, not much research has looked into the performance of these deep learning methods with respect to the genre classification task on lyrics. Here, we attempt to remedy this situation by extending deep learning approaches to text classification to the particular case of lyrics.

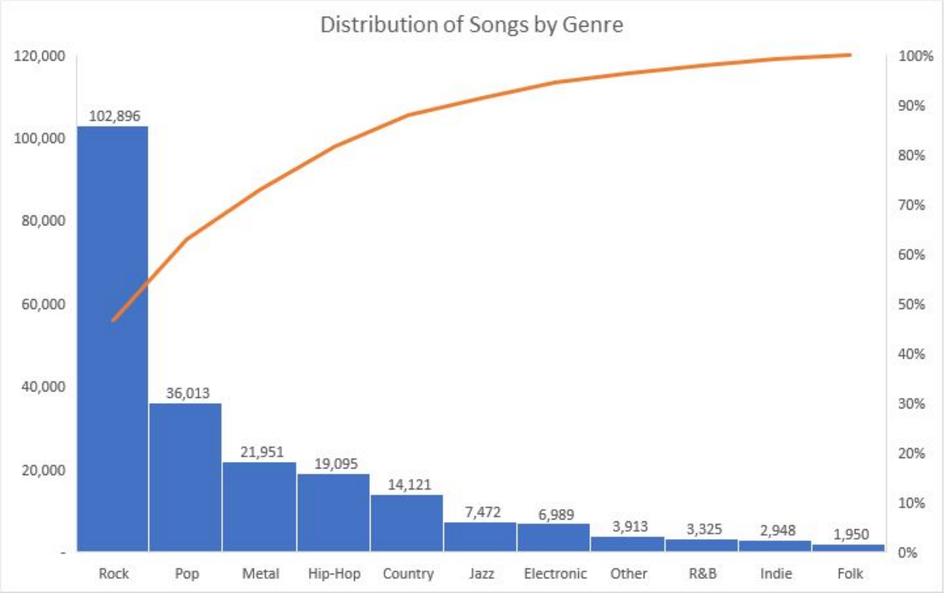
Hierarchical methods attempt to use some sort of structure of the data to improve the models and have previously been utilised in vision classification tasks [30]. Yang et al. [37] propose a hierarchical attention network (HAN) for the task of document classification. Since documents often contain structure whereby words form to create sentences, sentences to paragraphs, etc. they introduce this knowledge to the model, resulting in superior classification results. It is evident that songs and, in particular, lyrics similarly contain a hierarchical composition: Words combine to form lines, lines combine to form segments, and segments combine to form the whole song. A segment of a song is a verse, chorus, bridge, etc. of a song and typically comprises several lines. The hierarchical nature of songs has been previously exploited in genre classification tasks with Du et al. [8] utilising hierarchical analysis of spectrograms to help classify genre.

Here, we propose application of an HAN for genre classification of intact lyrics. We train such a network, allowing it to apply attention to words, lines, and segments. Re-

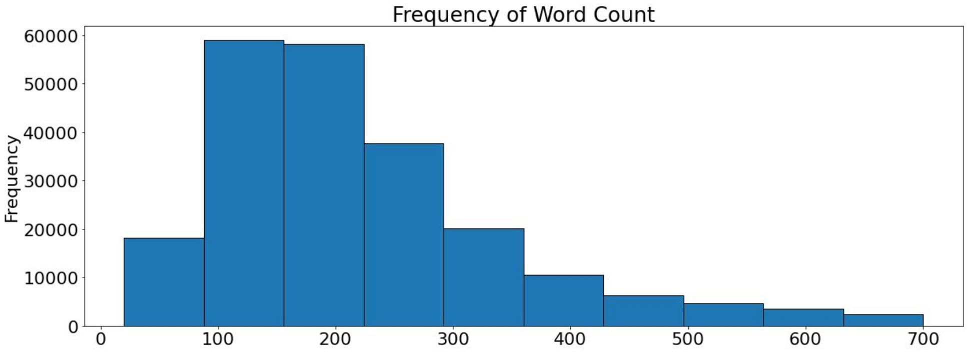
- Previous work focused primarily on classifying song sentiment, genre, and artists (Music Information Retrieval)
- A combination of lyrical, acoustic and symbolic features
- Traditional approaches: n-gram, SVM, kNN, and Naive Bayes
- Neural methods: RNN, Bi-LSTM, HAN, Bi-GRU
- Accuracy between 34 - 53%
- Minimal literature on transformer based models

Exploratory Data Analysis

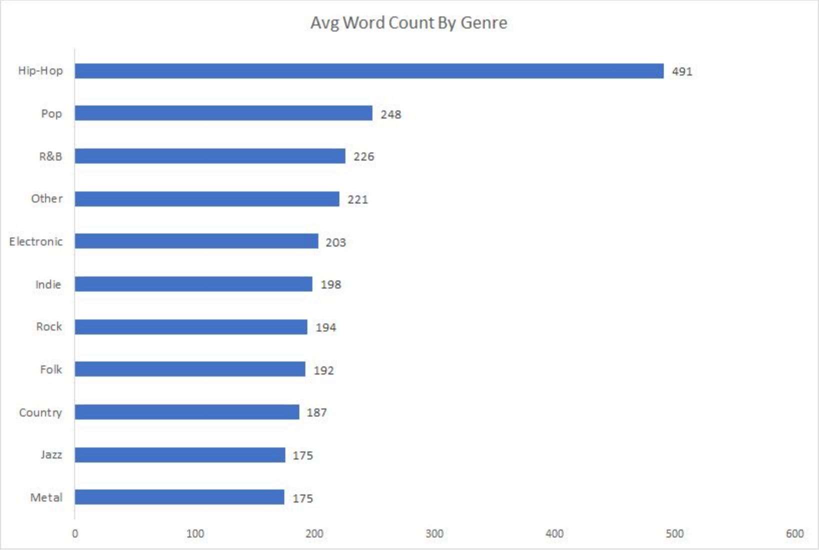
- MetroLyrics: 1 million+ songs and 16,000+ unique artists
- Lyrics in part contributed by music fans
- No governance on standardized formatting; extensive data cleaning
- 223,000 samples into 11 genres; song title, artist, genre, and lyrics
- Highly imbalanced



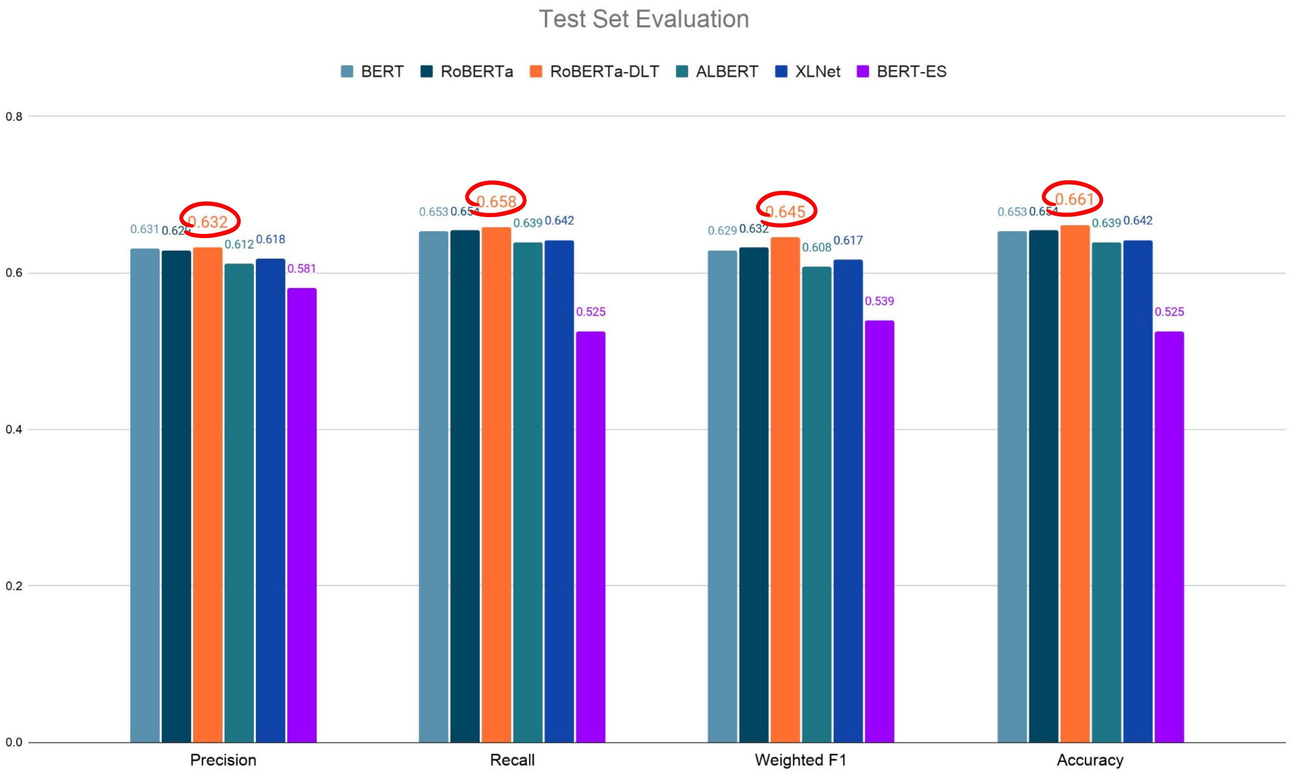
Exploratory Data Analysis



- Wide range of word frequencies
- Hip-Hop’s average word count >= 2x as long as other genres
- Impact model performance due to max sequence length of SOTA models
- 80% train, 10% dev, 10% test



Experiments - Precision, Recall, F1, Accuracy

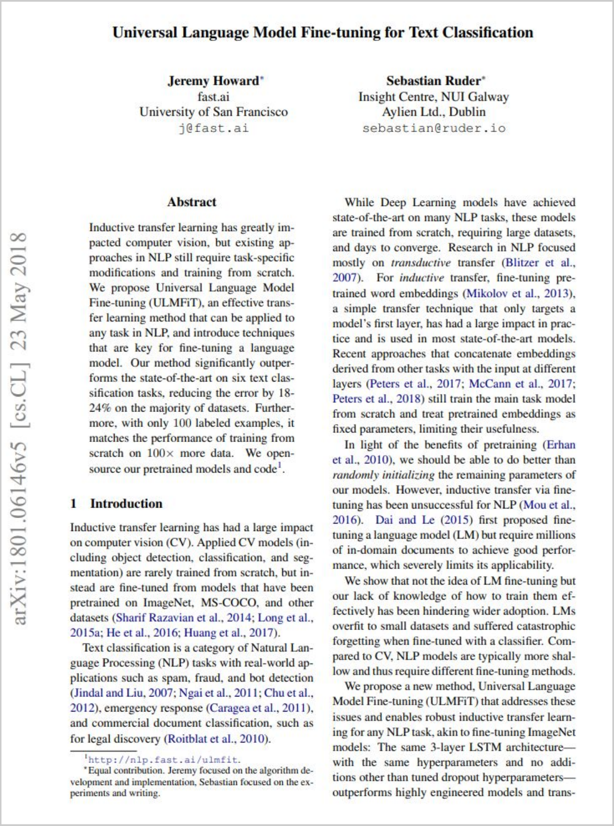


- BERT-ES: Even training samples per genre/class
- RoBERTa-DLT: Discriminative layer training
- F1 is the north star metric due to imbalance dataset

Experiments - Per Genre Accuracy by Model

Genre	BERT	BERT-ES	RoBERTa	RoBERTa-DLT	ALBERT	XLNet
Pop	43.94%	47.31%	45.44%	**49.87%**	40.87%	43.22%
Hip-Hop	80.97%	80.26%	82.11%	**83.05%**	81.68%	80.87%
Rock	83.73%	49.31%	81.86%	**82.49%**	83.82%	82.25%
Metal	60.68%	62.18%	62.23%	**63.16%**	61.27%	60.68%
Other	4.83%	**21.11%**	3.31%	4.42%	1.53%	1.02%
Country	55.98%	63.39%	61.57%	**62.90%**	49.47%	56.48%
Jazz	35.29%	37.04%	36.90%	**37.82%**	32.75%	35.03%
Electronic	22.33%	**36.70%**	23.90%	23.76%	11.24%	18.78%
Folk	15.82%	**28.06%**	15.31%	16.43%	12.25%	16.84%
R&B	11.41%	**30.63%**	12.01%	12.27%	3.90%	6.31%
Indie	0.34%	**16.27%**	0.00%	0.86%	0.00%	0.00%
Overall	65.30%	52.53%	65.40%	**66.10%**	63.90%	64.20%

Discriminative Layer Training (ULMFit)



Universal Language Model Fine-tuning for Text Classification

Jeremy Howard*
fast.ai
University of San Francisco
j@fast.ai

Sebastian Ruder*
Insight Centre, NUI Galway
Aylien Ltd., Dublin
sebastian@ruder.io

arXiv:1801.06146v5 [cs.CL] 23 May 2018

Abstract

Inductive transfer learning has greatly impacted computer vision, but existing approaches in NLP still require task-specific modifications and training from scratch. We propose Universal Language Model Fine-tuning (ULMFiT), an effective transfer learning method that can be applied to any task in NLP, and introduce techniques that are key for fine-tuning a language model. Our method significantly outperforms the state-of-the-art on six text classification tasks, reducing the error by 18-24% on the majority of datasets. Furthermore, with only 100 labeled examples, it matches the performance of training from scratch on 100× more data. We open-source our pretrained models and code[1].

1 Introduction

Inductive transfer learning has had a large impact on computer vision (CV). Applied CV models (including object detection, classification, and segmentation) are rarely trained from scratch, but instead are fine-tuned from models that have been pretrained on ImageNet, MS-COCO, and other datasets (Sharif Razavian et al., 2014; Long et al., 2015a; He et al., 2016; Huang et al., 2017).

Text classification is a category of Natural Language Processing (NLP) tasks with real-world applications such as spam, fraud, and bot detection (Jindal and Liu, 2007; Ngai et al., 2011; Chu et al., 2012), emergency response (Caragea et al., 2011), and commercial document classification, such as for legal discovery (Roitblat et al., 2010).

While Deep Learning models have achieved state-of-the-art on many NLP tasks, these models are trained from scratch, requiring large datasets, and days to converge. Research in NLP focused mostly on *transductive* transfer (Blitzer et al., 2007). For *inductive* transfer, fine-tuning pretrained word embeddings (Mikolov et al., 2013), a simple transfer technique that only targets a model's first layer, has had a large impact in practice and is used in most state-of-the-art models. Recent approaches that concatenate embeddings derived from other tasks with the input at different layers (Peters et al., 2017; McCann et al., 2017; Peters et al., 2018) still train the main task model from scratch and treat pretrained embeddings as fixed parameters, limiting their usefulness.

In light of the benefits of pretraining (Erhan et al., 2010), we should be able to do better than *randomly initializing* the remaining parameters of our models. However, inductive transfer via fine-tuning has been unsuccessful for NLP (Mou et al., 2016). Dai and Le (2015) first proposed fine-tuning a language model (LM) but require millions of in-domain documents to achieve good performance, which severely limits its applicability.

We show that not the idea of LM fine-tuning but our lack of knowledge of how to train them effectively has been hindering wider adoption. LMs overfit to small datasets and suffered catastrophic forgetting when fine-tuned with a classifier. Compared to CV, NLP models are typically more shallow and thus require different fine-tuning methods.

We propose a new method, Universal Language Model Fine-tuning (ULMFiT) that addresses these issues and enables robust inductive transfer learning for any NLP task, akin to fine-tuning ImageNet models: The same 3-layer LSTM architecture—with the same hyperparameters and no additions other than tuned dropout hyperparameters—outperforms highly engineered models and trans-

[1] http://nlp.fast.ai/ulmfit.

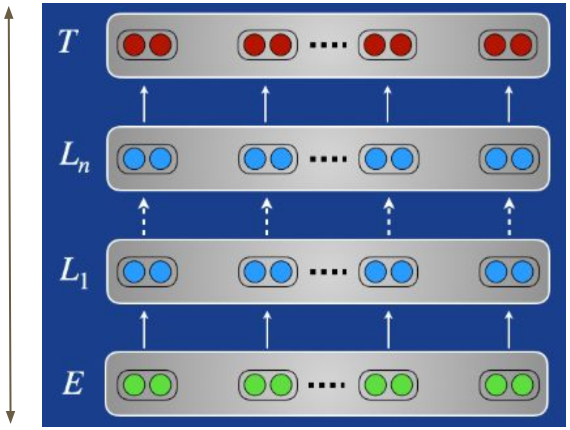
*Equal contribution. Jeremy focused on the algorithm development and implementation, Sebastian focused on the experiments and writing.

- Introduced 3 ideas during learning stage:
  1. Discriminative fine-tuning
  2. Gradual unfreezing
  3. Slanted triangular learning rates
- Different layers capture different types of information, so fine-tune to different extents
- Unfreeze each layer gradually and train with different learning rates
- High learning rate at starting stage for increased learning and low learning rates for fine tuning at later stages

Discriminative Layer Training (ULMFit) Illustrations

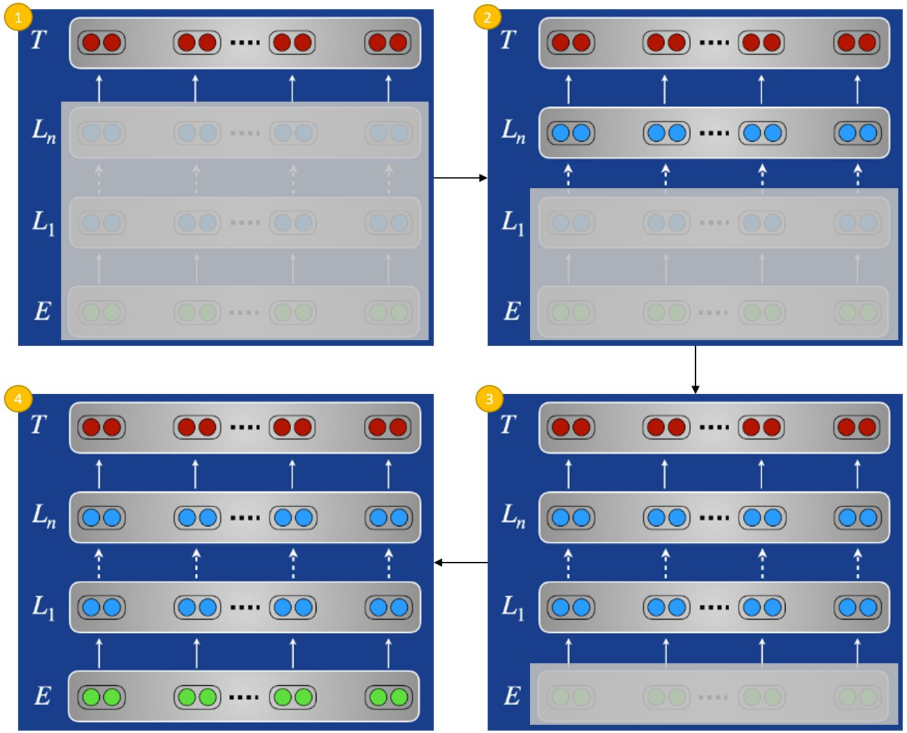
Discriminative fine-tuning with varying learning rates

Higher LR



Lower LR

Gradual layer unfreezing



Learnings and Future Work

1. Length and repetitive nature of lyrics is challenging; may not add incremental information gain

2. Genre imbalance is a fact of life; less Folk song artists than Hip-Pop or Rock

3. Music industry heuristics can degrade classification performance (i.e. Indie)

4. Augment lyrics with audio/acoustic features to improve model performance

5. Add artist as additional feature; artists tend to produce the same genre of music overtime

6. Experiment with larger model variants with increased GPU power (limited to Tesla P100-PCIE-16GB from Google Colab Pro)

7. Reduce train samples of popular genres by marginal amount while increase underrepresented genres; danger of overfitting and longer training time