Music Genre Prediction using Song Lyrics

Manu Agarwal
University of Texas at Austin
2317 Speedway, Austin, TX 78712
UTEID: ma53767
manuagarwal@utexas.edu

Kunal Lad
University of Texas at Austin
2317 Speedway, Austin, TX 78712
UTEID: kl28697
kunal.lad@utexas.edu

Abstract

People's perception of music is influenced by many different characteristics of music. A song's audio is arguably the most important piece of information contained in a song. Not surprisingly, most of the work on music analysis focuses on the audible part of a piece of music. However, there are other important aspects to a song such as lyrics, context and cultural spirit. In this paper, we explore the importance of lyrics for predicting the genre of a song. We present a Deep Learning Framework to classify songs into different genres based on songs' lyrics. We use word2vec to extract features from the lyrics of a song which are then fed into an LSTM (Long Short Term Memory) network that outputs the genre of a song.

Introduction

Music is close to many people's lives. With the availability of multiple songs just a click away, the need for intelligent tools for browsing music databases has risen. Music recommendation systems can help users find songs as per their preferences. Such systems require automatic music analysis, for instance automatic genre prediction, artist prediction etc.Further, automatic music analysis helps in musicology research, for instance, placing a music in its sociocultural context.

The most prevalent approach in music information retrieval is to analyze the audio features of a song which are low level features represented computed from audio transform or transcriptions of the music. However, recent approaches [21, 15, 16, 10] have shown that lyrics can play a crucial role in such analysis. Song lyrics exhibit distinct properties different from traditional text documents; many lyrics are composed of rhyming verses while some others use slang language; some might use a few parts-of-speech more often than others while some others might differ in complexity of the usage of language.

Another aspect that makes usage of lyrics a handy tool is the ease of obtaining and processing lyrics as opposed to audio data. Lyrics not only add semantic content, they can also serve as a proxy for melodic and rhythmic properties contained in the audio signals. Psychological study [2] also shows that our brain processes audio and lyrics of a song independently, hence the usage of song lyrics and audio features for music analysis is complementary to each other.

In this paper, we explore the task of identifying the genre of a song based on the lyrics of a song. A song's genre is a categorical description of the style of music it is, for instance pop, rock, classical, metal etc. We use word2vec to extract features from the lyrics of a song. We then pass these features onto a Recurrent Neural Network which maps them to a genre.

Related Work

Music Information Retrieval is a pretty diverse area with research on a variety of topics including genre prediction, visualization of music collections, identifying the publication time of a song or marking the best and worst songs from a given list according to a set of predefined preferences. Many of these domains have been heavily driven by audio analysis. [4] and [24] focus on the task of automatic genre classification based on audio features alone.

A number of works have tried to exploit lyrics-based features for music analysis. The task of predicting music genre from lyrics is different from a conventional text classification problem in that the shallow features such as bag-of-words and part-of-speech tags alone might not work well on this problem. Hence, there is a need to augment the feature set with more specific ones such as meter and rhyme properties. Most studies on lyrics-based features have focused on rather simple features, such as tf-idf weighted bag of words [20, 13, 11], BoW features enriched by synonymy and hypernymy [23], BoW features along with Part-of-speech tag distributions [15, 14].

Some works have also tried to exploit the information

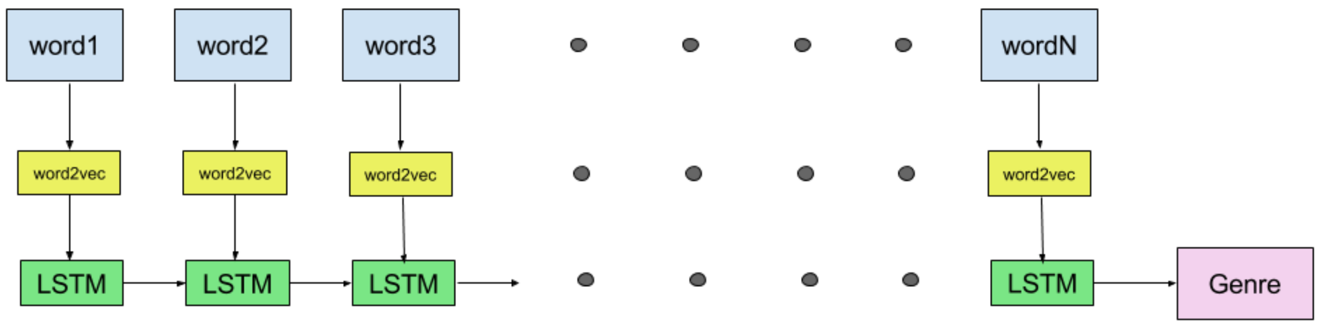

Figure 1. Overview of our approach

contained in simple text statistics (average word length, distribution of digits and punctuation marks etc). [10] have tried using function word distributions in addition to conventional feature sets. [5] focus on rhyme features to analyse rap lyrics.

Many recent works have tried using Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for text classification [27, 7, 26, 22] use Convolutional Neural Networks (CNNs) with Rectified Linear Units (ReLUs) to build a latent factor model using music audio for music recommendation.

In this paper, we propose to use a special type of Recurrent Neural Network - Long Short Term Memory (LSTM) modules to classify a piece of music into it's appropriate genre.

Dataset

We use the Million Song Dataset (MSD) [1] for this project. It is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. However, it does not contain songs' lyrics due to copyright issues. Hence, we used meta data from the Million Song dataset to scrape the web for extracting songs' lyrics. We used Lyricfy API to obtain lyrics of 8671 songs spanning 10 genres. Figure 2 shows the distribution of songs across different genres in the training set. Figure 3 shows the distribution for test set. It provides 30% of song lyrics for every song we query the API for. We also tried https://www.azlyrics.com and http://www.musixmatch.com to gather full lyrics for songs. However, their server would block our IP after a few requests. Hence, we proceeded with the dataset consisting of 30% lyrics for each song.

To remove inconsistency and noise from the data, we employed certain measures to clean the data - we removed unnecessary punctuation marks from the lyrics followed by a removal of stop words. We used stopwords from GoogleNews word2vec pretrained model. For training Bag of Words model we removed stop words using nltk English stop words corpus.

Technical Approach

A song is represented as a tuple of words $S = (W_1, W_2, \dots W_n)$. We are given a set of songs along with their genres $H = \{(S_1, G_1), (S_2, G_2), \dots, (S_N, G_N)\}$ as training data. Our task is to predict genres for a test set of songs $T = \{S_1, S_2, \dots, S_M\}$.

We begin by preprocessing song lyrics as explained in the Dataset section. We then extract word2vec features corresponding to the cleaned lyrics of the song. word2vec is a 2-layer neural network that turns text into a numerical form that deep nets can understand. The word2vec representation captures many linguistic regularities not captured by basic POS or other shallow features [17, 19]. Thus, it preserves the vector representation for phrases. Phrases play a far more important role in deciding the genre of a song than mere words. Hence, the intuition is that the word2vec model would give higher level semantic features. We used the continuous Bag-of-Words architecture of word2vec for the purposes of this project.

Convolutional Neural Networks(CNNs) have been used recently for a variety of text classification tasks. However, CNNs suffer from a major problem - they dont have persistence. In this case, we want to model a sequence of words and hence we need to reason about previous words in the song to infer useful information from later ones. This problem is addressed by Recurrent Neural Networks(RNNs) which have loops in them for information to persist.

For the Music Genre classification problem, our design should capture representativeness(vectors should be representative of the sense contained in a word or phrase), diversity(should not be redundant), interestingness (should weigh salient features more heavily than non-salient ones) and importance (should contain important objects that drive semantics). We can capture things such as diversity and interestingness only when we model the genre classification problem as a sequence modeling problem. Hence, using RNNs for this project makes more sense.

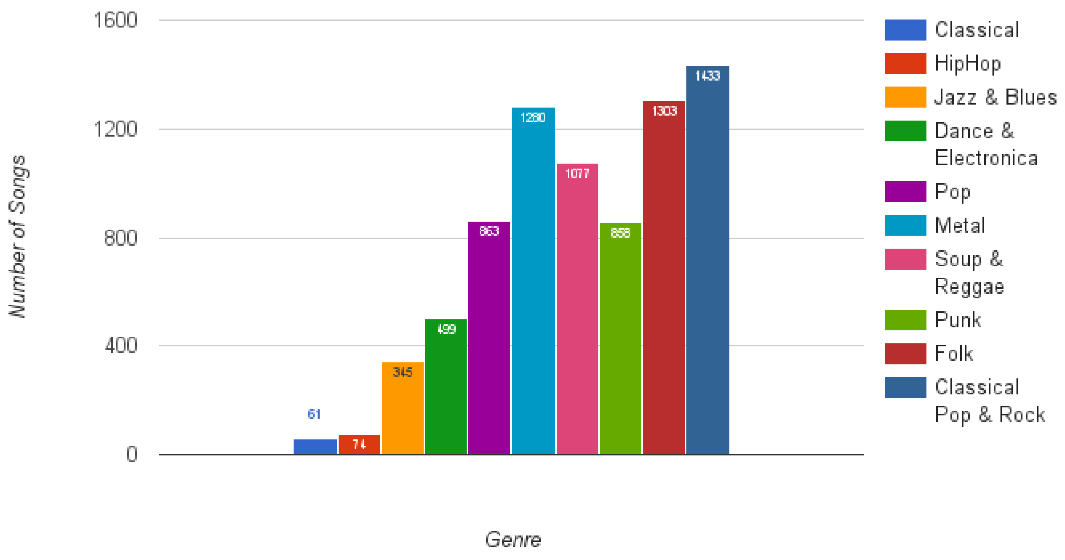

Figure 2. Distribution of songs across different genres in the training set

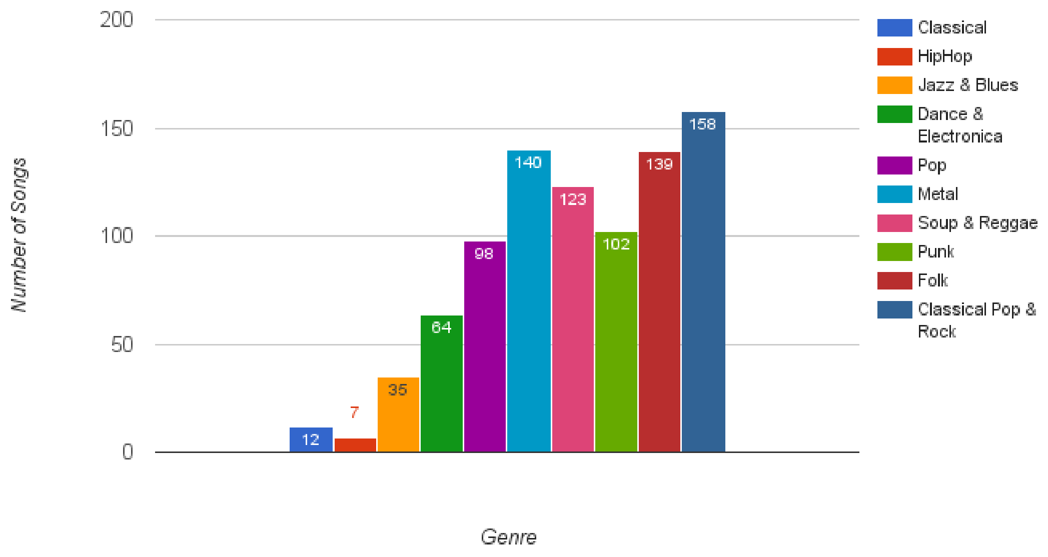

Figure 3. Distribution of songs across different genres in the test set

Now for Music Genre classification, we need to model longterm dependencies as well since two verses long distance apart might be highly correlated and hence need to be modeled jointly to avoid redundancy. Long Short Term Models(LSTMs) are a special kind of RNNs capable of learning such long term dependencies. They were introduced by Hochreiter and Schmidhuber [6]. They have been refined by a lot of people since then and work well on problems that require modeling time as a dimension.

For an input x_t at time step t, the LSTM computes a hidden state h_t and a memory cell state c_t which is an encoding of whatever the cell has seen until time t:

$$
\begin{aligned}
i_t &= \sigma(W_{x_i}x_t + W_{h_i}h_{t_1} + b_i)\\
f_t &= \sigma(W_{x_f}x_t + W_{h_f}h_{t_1} + b_f)\\
o_t &= \sigma(W_{x_o}x_t + W_{h_o}h_{t_1} + b_o)\\
g_t &= \phi(W_{x_g}x_t + W_{h_g}h_{t_1} + b_g)\\
c_t &= f_t \odot c_{t_1} + i_t g_t\\
h_t &= o_t \odot c_t
\end{aligned}
$$

Here ϕ represents the hyperbolic tangent non-linearity, σ is the sigmoidal non-linearity, $\odot$ represents the elementwise product with the gate value, and the weight matrices denoted by W_{ij} and biases b_j are the trained parameters. We combine word2vec model features and LSTM for this task. The basic architecture of our framework is shown in Figure 1.

Implementation Details

Preprocessing and feature extraction was done in python using nltk and GoogleNews word2vec pretrained model. LSTM was implemented in torch using the Sequential module and Fast LSTM implementation from RNN library [9]. Training set consisted of 90 percent of songs from each genre and the remaining 10 percent were used for testing. We used Negative Log Likelihood loss criteria in torch for LSTM training. Hyperparameters were chosen empirically. We started with a learning rate of 0.01 and reduce it by 0.8 after every 1000 iterations. We trained our model for 10000 iterations and took snapshots at intervals of 2000 iterations. Figure 5 shows the results of all trained models. Empirical results suggest that if we train for more than 6000 iterations then model is overfitting and underfitting if we train lesser.

Baselines

We compare the performance of our model against the following baselines:

1. **Majority genre**: A trivial baseline was to compare our method against outputting the majority genre of the test set as the output every time. The distribution of songs across different genres in the training and test sets is shown in Figure 2 and Figure 3 respectively. If we were to output the majority genre from the test set everytime, we would achieve an accuracy of 18.00%.

2. **Bag of words model+Random Forest Classifier**: The second approach we compare our model against is the simple Bag of words model. Each song is represented as a bag of words after removing stop words. We used stopwords from GoogleNews word2vec pretrained model. The feature vector for a song consists of the frequency of each word in the song.

   For classification, we used the Random Forest Classifier. Random Forests operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. We used a Random Forest Classifier with 100 tress for the purposes of our project. For computational feasibility, we restrict the feature size to 100 most frequent words. This model achieves an accuracy of 22.29%.

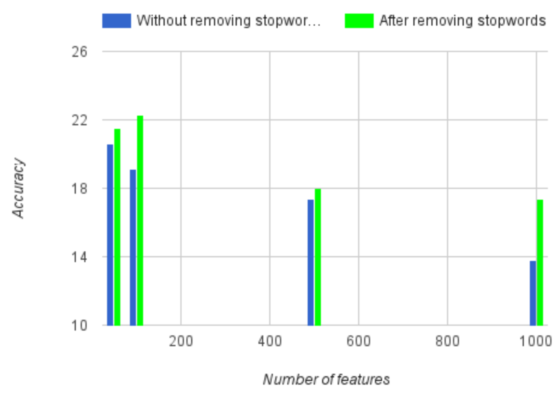

Figure 4. Performance of BoW+Random Forest

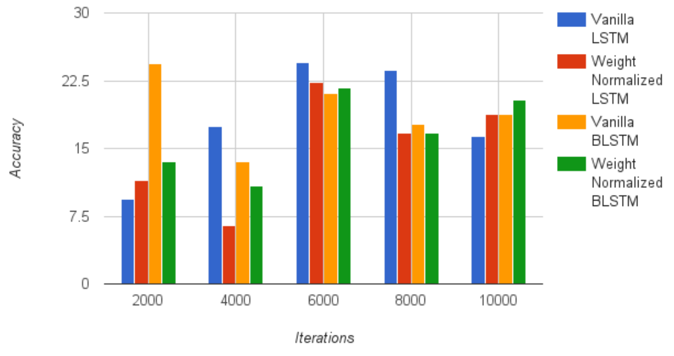

Figure 5. LSTM Accuracy vs Number of Training Iterations

Experiments and Results

This section presents the experiments we performed and the results we obtained.

BoW + Random Forest: We began with trying out the simple Bag of Words model along with the Random Forest Classifier as explained above. We tried out different feature sizes ranging from 50 to 1000. We also tried the model without removing stopwords. The results we obtained for this model are shown in Figure 4.

We can see that the best performance was obtained when we restricted the feature size to 100 after removing stopwords. This shows that not all words are equally important in predicting the genre of a song. In fact, the 100 most frequent words are enough. These 100 words should most likely represent certain distinct characteristics of the song such as rhyme, slang or non-slang usage of language, semantics etc. Also, removing most common words such as 'a','the' in the form of stopwords helps improve accuracy.

Vector Averaging: Next, we tried an approach similar to doc2vec. We took word2vec representations for words in a

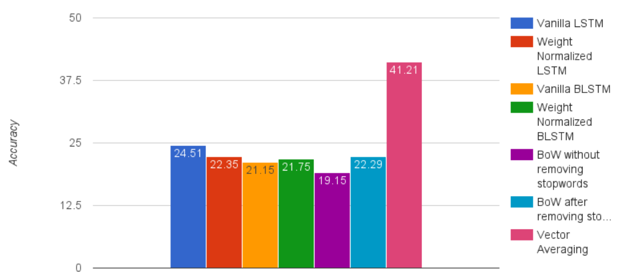

Figure 6. Comparison of all the approaches

song (after removing stopwords from song lyrics) and averaged these feature vectors to obtain a feature vector for the song. The word2vec model gave a 300 dimensional feature vector for every word in the song. we took element-wise average of the representations of words in the song.

This method achieved a significantly high accuracy of 41.21%.

LSTM vs Bidirectional LSTM: This experiment was motivated from superior results obtained for bidirectional models for tasks of Language Modelling (HW1) and Sequence to Sequence Video to Text task [25] (studied as part of Visual Recognition course by Prof Grauman). Thus we expected bidirectional LSTM to perform better. However our results from Figure 6 show that both bidirectional models perform slightly worse than their unidirectional counterparts. This can possibly be due to our dataset being small (contains only 30% of song lyrics) and hence the model parameters for the bidirectional model could not be learnt well.

How does class distribution based re-weighting of training examples affect the models ? We observed that our training set is unbalanced in the sense that classical and hip hop genres have less than 100 examples whereas genres like folk, metal and classical pop and rock have more than 1000 examples. Figure 2 shows the distribution of all the genres in the training set. Thus we re-weighted the importance of the examples so that the model learns well for all classes. Figure 6 shows the comparison of LSTM and BLSTM models with and without class weight normalization. We expected that models with weight normalization would perform better as they do not overfit to high frequency classes and generalize well to other classes. We did get an improvement (although very small 21.75 vs 21.15) for BLSTM model. For the LSTM model the accuracy decreased from 24.51 to 22.35 percent. This probably implies that the genres with fewer songs do not contain songs that are representative of that genre. Hence, weighing them more heavily cost us rather than benefiting us.

Conclusion

In this project, we tried using LSTMs for the task of Music Genre prediction using song lyrics. We used word2vec model trained on GoogleNews to extract features from song lyrics. Even with 30% lyrics for each song, our model achieved decent results on 10 genres. We also tried using Bag-of-words features along with a Random Forest classifier and word2vec features with vector averaging. The use of word2vec features with vector averaging achieved the best results with an accuracy of just over 41%. Our experiments indicate potential for the use of song lyrics for music analysis purposes. It also demonstrates that deep learning can be a handy tool for music analysis. However, our results are still far from state-of-the-art models that achieve an accuracy of close to 50% on prediction amongst 8 genres [3].

Future Work

An interesting extension to our approach would be to cluster word2vec representations of words in songs to obtain semantically related words. This would help us exploit the similarity of words within a cluster. This approach is popularly known as 'vector quantization'. We can then assign each word to it's nearest cluster centre (known as cluster centroid). This shall enable us to convert each song into a bag-of-centroids which can then be passed onto an SVM or a Random Forest Classifier to obtain genre prediction for that song.

Another thing which could be tried is using the doc2vec model that comes along with the standard word2vec toolkit. The doc2vec model shall help us convert every song into a feature vector directly rather than we trying to average feature vectors of each word to obtain a vector for the song. This shall possibly help generate better feature vectors for songs.

We used the GoogleNews word2vec pretrained model. Another point of improvement would be to use word2vec model trained on a dataset consisting of song lyrics. As song structure is different from normal text, using a model trained on song lyrics should give better results.

One can also try out Glove (Global Vectors for Word Representation) model instead of word2vec to see how it performs.

References

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

[2] M. Besson, F. Fata, I. Peretz, A. Bonnel, and J. Requin. Singing in the brain: Independence of lyrics and tunes. 1998. Psychological Science, 9(6):494498.

[3] S. C. Fell M. Lyrics-based analysis and classification of music. 2014. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers [Internet]. Dublin, Ireland: Dublin City University and Association for Computational Linguistics; 2014. p. 62031. Available from: http://www.aclweb.org/anthology/C14-1059.

[4] J. Foote. An overview of audio information retrieval. *Multimedia Syst.*, 7(1):2–10, Jan. 1999.

[5] H. Hirjee and D. G. Brown. Using automated rhyme detection to characterize rhyming style in rap music. 2010. Empirical Musicology Review, 5(4):121145.

[6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[7] S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification, 2015.

[8] N. Leonard. Github library for rnns. https://github.com/Element-Research/rnn.

[9] N. Léonard, S. Waghmare, and Y. Wang. rnn: Recurrent library for torch. *arXiv preprint arXiv:1511.07889*, 2015.

[10] T. Li and M. Ogihara. Music artist style identification by semi-supervised learning from both lyrics and content. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, pages 364–367, New York, NY, USA, 2004. ACM.

[11] B. Logan, A. Kositsky, and P. Moreno. Semantic analysis of song lyrics. In *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, volume 2, pages 827–830 Vol.2, June 2004.

[12] M. Lustrek. Overview of automatic genre identification, 2007. Technical Report IJS-DP-9735, Jozef Stefan Institute, Department of Intelligent Systems, Jamova 39, 1000 Ljubljana, Slovenia, January.

[13] Mahedero, J. P. G., A. MartÍnez, P. Cano, M. Koppenberger, and F. Gouyon. Natural language processing of lyrics. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, pages 475–478, New York, NY, USA, 2005. ACM.

[14] R. Mayer, R. Neumayer, and A. Rauber. Rhyme and style features for musical genre classification by song lyrics.

[15] R. Mayer, R. Neumayer, and A. Rauber. Combination of audio and lyrics features for genre classification in digital audio collections. In *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, pages 159–168, New York, NY, USA, 2008. ACM.

[16] R. Mayer and A. Rauber. Music genre classification by ensembles of audio and lyrics features. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 675–680, Miami (Florida), USA, October 24-28 2011. `http://ismir2011.ismir.net/papers/PS6-4.pdf`.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[19] T. Mikolov, W. tau Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics, May 2013.

[20] R. Neumayer and A. Rauber. Integration of text and audio features for genre classification in music information retrieval (accepted for publication. In *in Proceedings of the 29th European Conference on Information Retrieval (ECIR07*, pages 724–727, 2007.

[21] R. Neumayer and A. Rauber. Multi-modal music information retrieval: Visualisation and evaluation of clusterings by both audio and lyrics. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, RIAO '07, pages 70–89, Paris, France, France, 2007. Le Centre De Hautes Etudes Internationales D'Informatique Documentaire.

[22] A. Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2643–2651. 2013.

[23] S. Scott and S. Matwin. Text classification using wordnet hypernyms. In *Use of wordnet in Natural Language Processing Systems: Proceedings of the conference, Pages 38-44. Association for Computational Linguistics*, pages 45–52, 1998.

[24] G. Tzanetakis and P. Cook. Marsyas: A framework for audio analysis. *Org. Sound*, 4(3):169–175, Dec. 1999.

[25] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence – video to text. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4534–4542, Dec 2015.

[26] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626, 2015.

[27] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau. A C-LSTM neural network for text classification. *CoRR*, abs/1511.08630, 2015.