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-

sults show the network produces higher accuracies in the lyrical classification task than previous research and from the attention learned by the network we can observe which words are indicative of different genres.

The remainder of the paper is structured as follows. In Section 2 we describe our methods, including the dataset and a description of the HAN. In Section 3 we provide results and visualisations from our experiments. We conclude with a discussion in Section 4.

2. METHODS

2.1 Dataset

Research involving song lyrics has historically suffered from copyright issues. Consequently most previous literature has utilised count-based bag-of-words lyrics. In this format, structure and word order are lost, and it has been shown that utilising intact lyrics reveals superior results in classification tasks [11, 32].

Seeking an intact lyrics corpus for the present study, we obtained a collection of lyrics through a signed research agreement with LyricFind [1]. This corpus has been used in the past to study novelty [10] and influence [1] in lyrics. The complete set contained 1,039,151 song lyrics in JSON format, as well as basic metadata including artist(s) and track name. As the corpus provided no genre information, we aggregated it ourselves using the iTunes Search API [2], extracting the value for the `primaryGenreName` key as baseline truth. Several different sources were not used for consistency reasons with iTunes found to be the largest, easily accessible source with reasonable genre tags. This unfortunately still greatly reduced the size of the dataset due to the sparse iTunes database. We then further removed any songs that were linked with a genre tag of 'Music Video', leaving a dataset comprising 244 genres. As this dataset had a very long tail of sparse genres, we further filter the dataset via two methods. Firstly we remove any genres with less than 50 instances, giving a dataset of size 495,188 lyrics and 117 genres. Secondly we retain only the top 20 genres, giving a dataset of 449,458 lyrics. We note also that the dataset originally contained various versions of the same lyrics, due to the prevalence of cover songs; we retain only one of these versions chosen at random. The song lyrics are split into lines and segments which we tokenised using the `nltk` package [3] in Python. We split the dataset into a rough split of 80% for training, 10% for validation, and 10% for testing. All preprocessing was done via Python with the neural networks built using Tensorflow [4].

2.2 Hierarchical Attention Networks

The structure of the model follows that of Yang et al. [37]. Each layer is run through a bidirectional gated recurrent unit (GRU) with attention applied to the output. The attention weights are used to create a vector via a weighted sum which is then passed as the input to the next layer. A representation of the architecture for the example song of 'Happy Birthday' can be seen in Figure 1, where the layers are applied at the word, line, and song level. We briefly step through the various components of the model.

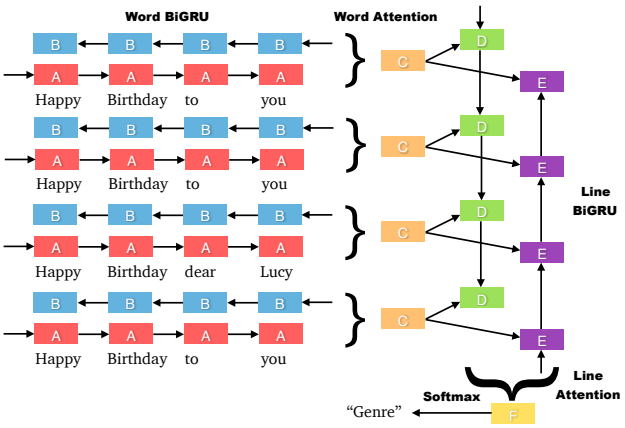

Figure 1: Representation of the HAN architecture; boxes represent vectors. A and B vectors represent the hidden states for the forward and backward pass of the GRU at the word level, respectively. The line vectors C are then obtained from these hidden states via the attention mechanism. The D and E vectors represent the forward and backward pass of the GRU at the line level, respectively. The song vector F is then obtained from these hidden states via the attention mechanism. Finally classification is performed via the softmax activation function.

2.2.1 *Word Embeddings*

An important idea in NLP is the use of dense vectors to represent words. A successful methodology proposes that similar words have similar context and thus vectors can be learned through their context, such as in the word2vec model [26]. Pennington et al. [29] propose the GloVe method which combines global matrix factorisation and local context window methods to produce word vectors that outperform previous word2vec and SVM based models.

Here we take as our vocabulary the top 30,000 most frequent words from the whole LyricFind corpus, including those from songs we did not match with a genre. We train 100-dimensional GloVe embeddings for these words using methods obtained from the GloVe website [5]. Previous research has shown that retraining these word vectors over the extrinsic task at hand can improve results if the dataset is large enough [5]. In a preliminary genre classification task we found that retraining these word embeddings did improve accuracy, and so we let our model learn superior embeddings to those provided by GloVe [29].

2.2.2 *Gated Recurrent Units*

Introduced by Chung et al. [4], GRUs are a form of gating mechanism in RNNs designed to help overcome the struggle to capture long-term dependencies in RNNs. This is achieved by the introduction of intermediate states between the hidden states in the RNN. An update gate z_t is

[1] `http://lyricfind.com/`
[2] `http://apple.co/1qHOryr`
[3] `http://www.nltk.org/`
[4] `https://www.tensorflow.org/`
[5] `http://nlp.stanford.edu/projects/glove/`

introduced to help determine how important the previous hidden state is to the next hidden state. A reset gate r_t is introduced to help determine how important the previous hidden state is in the creation of the next memory. The hidden state is h_t, whilst new memory is computed and stored in $\tilde{h}_t$. Mathematically we describe the process as

$$z_t = \text{sigmoid}\left(W_z x_t + U_z h_{t-1} + b_z\right) \quad (1)$$

$$r_t = \text{sigmoid}\left(W_r x_t + U_r h_{t-1} + b_r\right) \quad (2)$$

$$\tilde{h}_t = \tanh\left(W_h x_t + r_t \circ U_h h_{t-1} + b_h\right) \quad (3)$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t, \quad (4)$$

where x_t is the word vector input at time-step t, $\circ$ is the Hadamard product, and sigmoid is the sigmoid activation function. W_z, U_z, W_r, U_r, W_h, and U_h are weight matrices randomly initialised and to be learned by the model along with the b_z, b_r, and b_h bias terms. Bias terms were not included in the original model by Chung et al. [4], however have been included here as in Jozefowicz et al. [15].

2.2.3 Hierarchical Attention

Attention was first proposed by Bahdanau et al. [2] with respect to neural machine translation to allow the model to learn which words were more important in the translation objective. Along the lines of that study, we would like our model to learn which words are important in classifying genre and then apply more weight to these words. Similarly, we can apply attention again on lines or segments to let the model learn which lines or segments are more important in classification.

Given input vectors h_i for $i = 1, \ldots, n$ the attention mechanism can be formulated as

$$u_i = \tanh\left(W_a h_i + b_a\right) \quad (5)$$

$$\alpha_i = \frac{\exp(u_i^T u_a)}{\sum_{k=1}^{n} \exp(u_k^T u_a)} \quad (6)$$

$$s = \sum_{i=1}^{n} \alpha_i h_i, \quad (7)$$

where s is the output vector passed to the next layer consisting of the weighted sum of the current layers vectors. Parameters W_a, b_a, and u_a are learned by the model after random initialisation.

One layer of the network takes in vectors $x_1, \ldots, x_n$, applies a bidirectional GRU to find a forward hidden state $\overrightarrow{h}_j$ and a backward hidden state $\overleftarrow{h}_j$, and then uses the attention mechanism to form a weighted sum of these hidden states to output as the representation. Letting GRU indicate the output of a GRU and ATT represent the output from an attention mechanism, one layer is formulated as

$$\overrightarrow{h}_j = \overrightarrow{GRU}(x_j), \quad (8)$$

$$\overleftarrow{h}_j = \overleftarrow{GRU}(x_j), \quad (9)$$

$$h_j = [\overrightarrow{h}_j; \overleftarrow{h}_j], \quad (10)$$

$$s = ATT(h_1, \ldots, h_L). \quad (11)$$

Our HAN consists of two layers, one at the word level, and one at the line/segment level. Consider a song of L lines or segments s_j, each consisting of n_j words w_{ij}. Let E be the pre-trained word embedding matrix. Letting LAY represent the dimension reduction operation of a layer in the network as in Eqns 8–11 the whole HAN can be formulated for $i = 1, \ldots, n_j$ and $j = 1, \ldots, L$ as

$$x_{ij} = Ew_{ij} \quad (12)$$

$$s_j = LAY(x_{1j}, \ldots, x_{n_j j}), \quad (13)$$

$$s = LAY(s_1, \ldots, s_L). \quad (14)$$

Each layer has its own set of GRU weight matrix and bias terms to learn, as well as its own attention weight matrix, bias terms, and relevance vector to learn.

2.2.4 Classification

With the song vector s now obtained, classification is performed by using a final softmax layer

$$p = \text{softmax}\left(W_p s + b_p\right), \quad (15)$$

where intuitively we take the entry of highest magnitude as the prediction for that song.

To train the model we minimise cross-entropy loss over K songs

$$J = -\sum_{k=1}^{K} \log(p_{d_k k}), \quad (16)$$

where d_k is the true genre label for that song.

3. EXPERIMENTS

3.1 Baseline Models

We compare the performance of the HAN against various baseline models.

1. Majority classifier (MC): 'Rock' is the most common genre in our dataset. The MC simply predicts 'Rock'.
2. Logistic regression (LR): A LR run on the average song word vector produced from the GloVe embeddings.
3. Long Short-Term Memory (LSTM): An LSTM, treating the whole song as a single sequence of words and use max-pooling of the hidden states for classification. Fifty hidden units were used in the LSTM and each song had a maximum of 600 words. For full discussion of the LSTM framework see Hochreiter and Schmidhuber [12].
4. Hierarchical network (HN-L): The HN structure in the absence of attention run at the line level. At each layer all of the representations are simply averaged to produce the next layer input.

For LR, LSTM, and HN-L we let the model retrain the word embeddings as it trained.

Model	117 Genres	20 Genres
MC	24.71	27.17
LR	35.21	38.13
LSTM	43.66	49.77
HN-L	45.85	49.09
HAN-L	46.42	49.50
HAN-S	45.05	47.60

Table 1: Genre classification test accuracies for the two datasets (%) using majority classifier (MC), logistic regression (LR), Long Short-Term Model (LSTM), hierarchical network (HN-L), and line- and segment-level HAN (HAN-L, HAN-S).

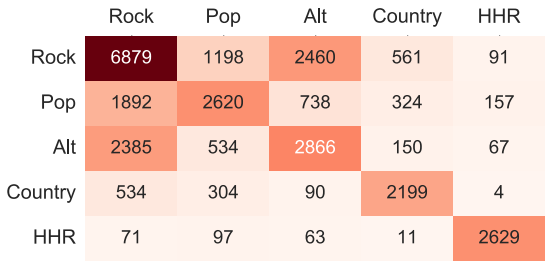

Figure 2: HAN-L confusion matrix for Rock, Pop, Alternative (Alt), Country, and Hip-Hop/Rap (HHR) genres over larger (117-genre) dataset. Rows represent true genre, whilst columns are predicted.

3.2 Model Configuration

The lyrics are padded/truncated to have uniform length. In the line model, each line has a maximum of 10 words and a maximum of 60 lines. In the segment model each segment has a maximum of 60 words and a maximum of 10 segments. Fifty hidden units are utilised in the bidirectional GRUs, whilst one hundred states are output from the attention mechanisms. Before testing the model, hyperparameters were tuned on the validation set. Dropout [35] and gradient clipping [28] were both found to benefit the model. We dropout at each layer with probability $p = 0.5$ and gradients are clipped at a maximum norm of 1 in the backpropogation. We utilise a mini-batch size of 64 and optimise using RMSprop [36] with a learning rate of 0.01. The models were all run until their validation loss did not decrease for 3 successive epochs. In all the HAN models, this occurred between the 5th and 8th epoch.

The code to train the model and perform the experiments described are made publicly available [6] .

3.3 Results

For both dataset sizes we run the baseline models and the HAN at the line and segment level. Let HAN-L represent running over lines and HAN-S represent running over segments. The test accuracies are seen in Table 1.

From the results we see a trend between model complexity and classification accuracy. The very simple majority classifier performs weakest and is improved upon by the simple logistic regression on average bag-of-words. The neural-based models perform better than both of the simple models. The LSTM model, which takes into account word order and tries to implement a memory of these words, gives performances of 43.66% and 49.77%, outperforming the HAN on the 20-genre dataset. Over the 117-genre dataset the best performing models were the HANs, with a highest accuracy of 46.42% when run over lines. It is observed that for the simpler 20-genre case, the more complex HAN is not required since the simpler LSTM beats it, although the LSTM took almost twice as long to train as the HAN. However for the more challenging 117-genre case, the HAN-L outperforms the LSTM, perhaps picking up on more of the intricacies of rarer genres.

[6] https://github.com/alexTsaptsinos/lyricsHAN

In both cases the HAN run at the line level produced superior results than that run over the segment level, giving a bump of roughly 1.4% and 1.9% in the 117-genre and 20-genre datasets, respectively. The HN-L, which is run at the line level, additionally outperforms the HAN at segment level. This indicates that the model performs better when looking at songs line by line rather than segment by segment. In the HAN-L the model can pick up on many repeated lines or lines of a similar ilk, rather than the few similar segments it attains in the HAN-S, and this may be attributive to the better performance. The network does benefit from the inclusion of attention, with HAN-L classifying with higher accuracies than HN-L. This increase is marginal and requires an increased cost, however allows for the extraction of attention in the visualisations of the following section.

As expected, classifying over the 20-genre dataset has given boosts of roughly 3% and 2.5% in the HAN-L and HAN-S, respectively. It is interesting to note that discarding roughly 10% of the data by only keeping roughly a sixth of the genres has not strengthened the model by much. Given the similarity of recognition performance between the two datasets, even with the simplest of models, it is likely that the extra genres are predominantly noise added to the 20-genre dataset. With the HAN-L outperforming the LSTM over the 117-genre dataset this then indicates that the model is more robust to noise.

The confusion matrix for HAN-L run over the larger dataset for the top 5 genres can be seen in Figure 2. We can see from the matrix that Rock, Pop, and Alternative (Alt) are all commonly confused; the model predicts Rock for Alternative almost as many times as it does Alternative. As the most common genre in the dataset by about 30,000 it is unsurprising to see the model try and predict Rock more often, and it is unclear whether a person would be able to distinguish between the lyrics of these genres. However, we see that both Country and Hip-Hop/Rap (HHR) are more separated. With their distinct lyrical qualities, especially in the case of Hip-Hop/Rap, this is an encouraging result indicating that the model has learned some of the qualities of both these genres.

3.3.1 Attention Visualisation

To help illustrate the attention mechanism, we feed song lyrics into the HAN-L and observe the weights it applies

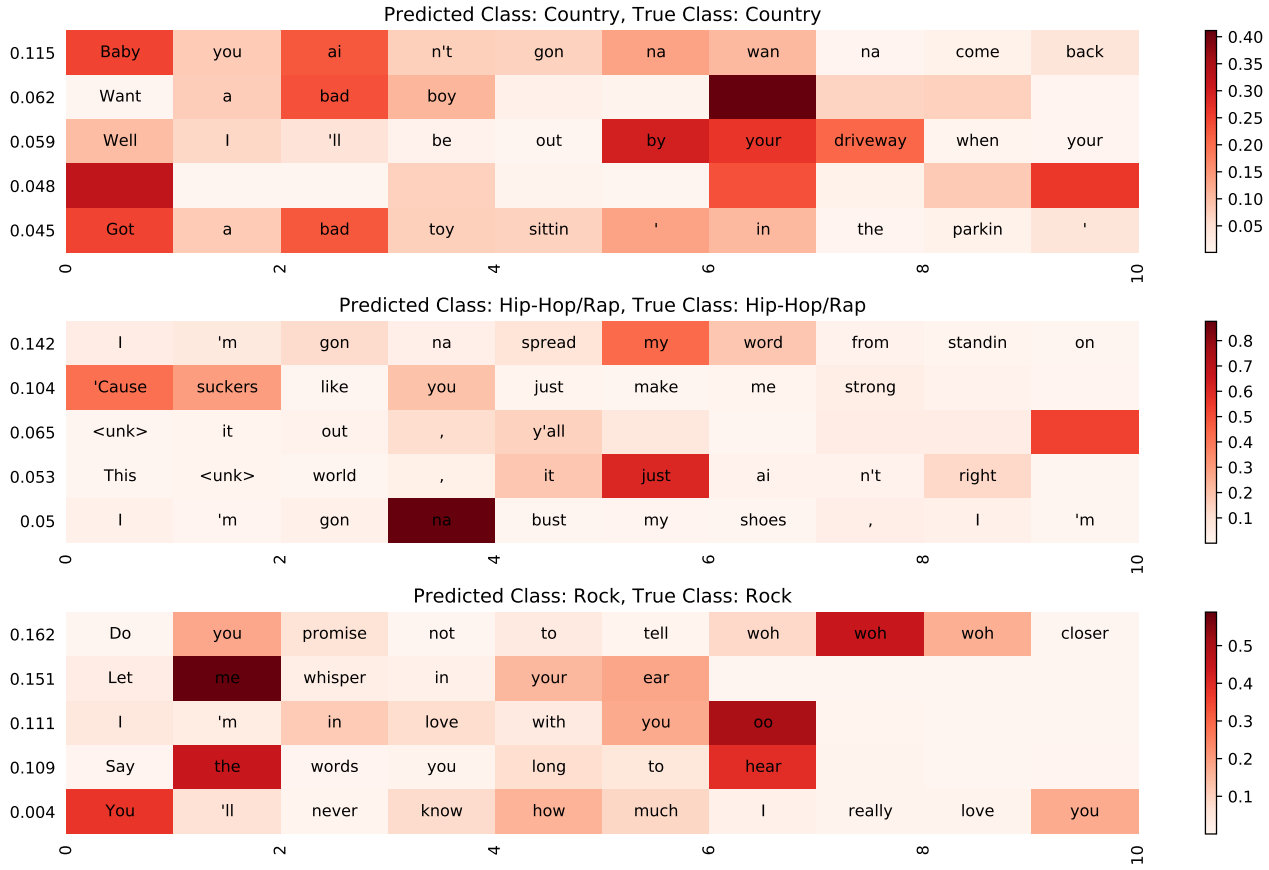

Figure 3: Weights applied by the HAN-L for song lyrics that were correctly classified. Line weights appear to the left of each line and word weights are coloured according to the respective colorbars on the right.

to words and lines. For each song we extract the 5 most heavily weighted lines and a visualisation of their weights and the individual word weights for a few different correctly predicted song lyrics can be seen in Figure 3.

From these visualisations we notice that the model has placed greater weights on words we may associate with a certain genre. For example 'baby' and 'ai' are weighted heavily in the Country song, and the most heavily weighted line in that song is characteristically Country. The model has placed great weight on a blank line, indicating the break between segments; it is unclear whether the model is learning to place importance on how songs are segmented and the number of segments occurring. In the Hip-Hop/Rap song the model places attention on colloquially spelled words 'cause' and 'gonna'. Although not included here, it was observed that for many rap songs swear words and racial terms were heavily weighted. The model picks up the 'woh' and 'oo' in the Rock song and also heavily weights occurrences of second-person determiner 'your' and pronoun 'you'. It was found that for many Rock songs this was the case.

In addition some visualisations of lyrics that were incorrectly classified by the HAN-L can be seen in Figure 4. We observe the model predicting Country for a Pop song, applying weights to 'sin' and 'strong' which could be characteristic of Country songs. The dataset contains songs with foreign language lyrics. Here we observe a song with Spanish lyrics classed as Pop Latino by the model whilst iTunes deems it Pop. This seems like a fair mistake for the model to have made since it has evidently recognised the Spanish language. The model also incorrectly classifies the Hip-Hop/Rap song as Pop. In the 5 most heavily weighted lines we do not spot any instances of language that indicate a Hip-Hop/Rap song and we hypothesise that the genericness of the lyrics has led the model to predict Pop.

4. DISCUSSION

Genre is an inherently ambiguous construct, but one that plays a major role in categorising musical works [24, 33]. From one standpoint, genre classification by lyrics will always be inherently flawed by vague genre boundaries and many genres borrowing lyrics and styles from one another. Previous research has shown that lyrical data performs weakest in genre classification compared to other forms of data [23]. As a consequence, this problem is not as well researched and preference has been given to other methods.

SVMs, k-NN, and NB have been heavily used in previous lyrical classification research. In addition very rarely has research looked into classifying more than between 10 genres despite the prevalence of clearly many more genres. Fell and Sporleder classify among 8 genres using n-grams along with other hand-selected features to help represent vocabulary, style, structure, and semantics [11].

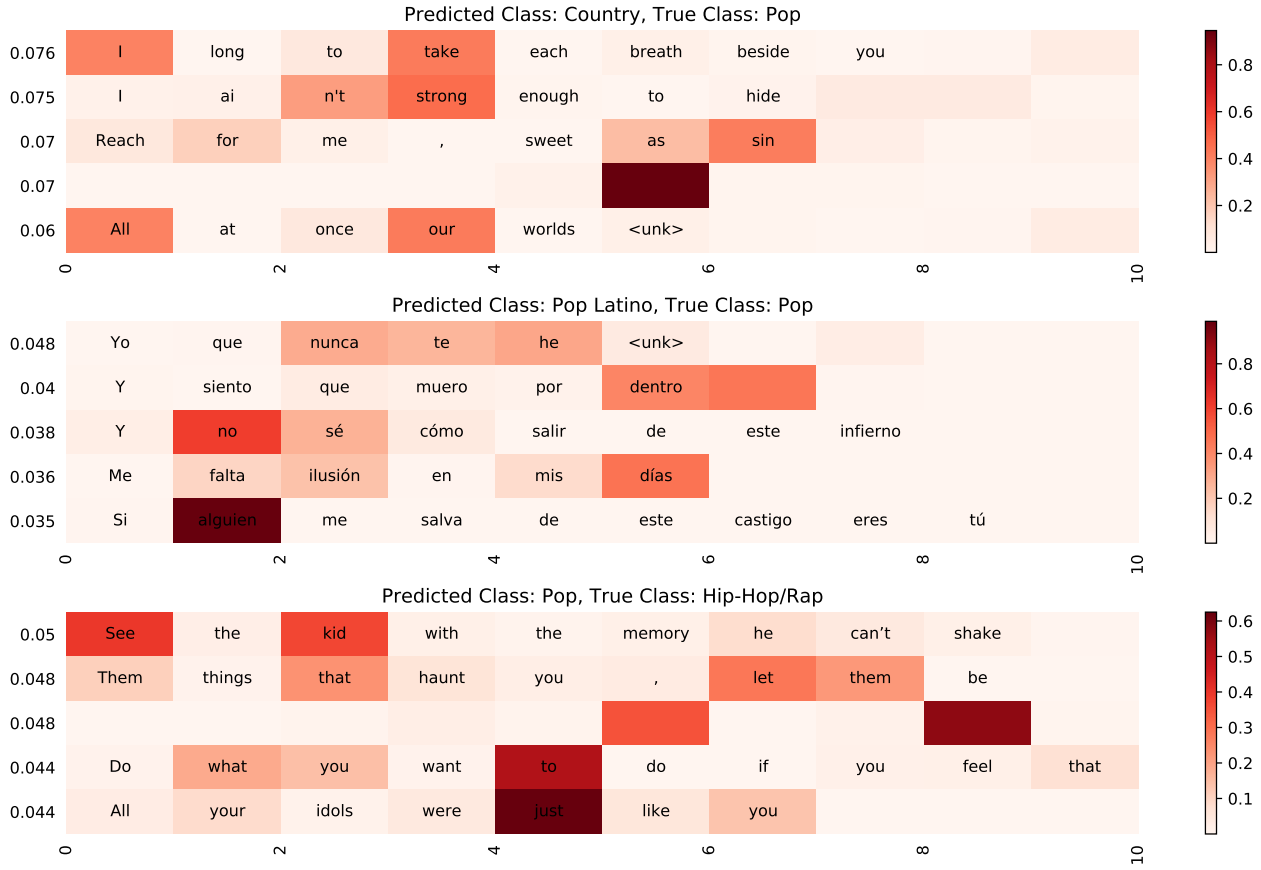

Figure 4: Weights applied by the HAN-L for song lyrics that were incorrectly classified. Line weights appear to the left of each line and word weights are coloured according to the respective colorbars on the right.

Ying et al. make use of POS tags and classify among 10 genres using SVMs, k-NN, NB with a highest accuracy of 39.94% [38]. McKay et al. utilise hand-selected features to produce classification accuracies of 69% among 5 genres and 43% among 10 genres [23].

In this paper we have shown that an HAN and other neural-based methods can improve on the genre classification accuracy. In large part this model has beaten all previously reported lyrical-only genre classification model accuracies, except for the classification among 5 genres. Whilst having been trained on different datasets the jump in classification accuracies achieved by the HAN and LSTM across the 20-genre datasets compared to previous research indicate that neural structures are clearly beneficial. However, with very similar results between the neural structures it is still unclear what the optimal neural structure may be and there is certainly room for further experimentation. We have shown that the HAN works better with layers at the word, line, and song level rather than word, segment, and song level. One known issue of the present dataset is that iTunes attributes genres by artist, not by track; this is a problem for artists whose work may cover multiple genres and is something that should be addressed in the future. A larger issue concerns the accuracy of the iTunes genre labels more generally, especially for the larger 117-genre dataset which naturally includes more subjective and vague genre definitions.

Visualisations of the weights the HAN applies to words and lines were produced to help see what the model was learning. In a good amount of cases, words and lines were heavily weighted that were cohesive with the song genre; however, this was not always the case. We note that in general the model tended to let one word dominate a single line with the greatest weight. However this was not as apparent across lines, with weights among lines more evenly spread. With a large amount of foreign-language lyrics also present in the dataset, an idea for further research is to build a classifier that identifies language, and from there classifies by genre. Any such research would be inhibited, however, by the lack of such a rich dataset to train on.

To produce a state-of-the-art classifier it is evident that the classifier must take into account more than just the lyrical content of the song. Mayer et al. combine audio and lyrical data to produce a highest accuracy of 63.50% within 10 genres via SVMs [21]. Mayer and Rauber then use a cartesian ensemble of lyric and audio features to gain a highest accuracy of 74.08% within 10 genres [22]. Further research could look into employing this hierarchical attention model to the audio and symbolic data, and combining with the lyrics to build a stronger classifier. Employment of the HAN in the task of mood classification via sentiment analysis is another possible area of research. In addition the HAN could be extended to include both a layer at the line and segment level, or even at the character level, to explore performance.

5. ACKNOWLEDGEMENTS

Many thanks to Will Mills and Mohamed Moutadayne from LyricFind for providing access to the data, and the ISMIR reviewers for their helpful comments.

6. REFERENCES

[1] Jack Atherton and Blair Kaneshiro. I said it first: Topological analysis of lyrical influence networks. In *ISMIR*, pages 654–660, 2016.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. *arXiv preprint arXiv:1606.00298*, 2016.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[6] Y. MG Costa, L. S. Oliveira, and C. N. Silla. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied Soft Computing*, 52:28–38, 2017.

[7] Matthias Dorfer, Andreas Arzt, Sebastian Böck, Amaury Durand, and Gerhard Widmer. Live score following on sheet music images. *arXiv preprint arXiv:1612.05076*, 2016.

[8] W. Du, H. Lin, J. Sun, B. Yu, and H. Yang. A new hierarchical method for music genre classification. In *CISP-BMEI*, pages 1033–1037. IEEE, 2016.

[9] Hamid Eghbal-Zadeh, Markus Schedl, and Gerhard Widmer. Timbral modeling for music artist recognition using i-vectors. In *EUSIPCO*, pages 1286–1290. IEEE, 2015.

[10] R. J. Ellis, Fang J. Xing, Z., and Y. Wang. Quantifying lexical novelty in song lyrics. In *ISMIR*, pages 694–700, 2015.

[11] M. Fell and C. Sporleder. Lyrics-based analysis and classification of music. In *COLING*, volume 2014, pages 620–631, 2014.

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

[13] X. Hu and J. S. Downie. Improving mood classification in music digital libraries by combining lyrics and audio. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 159–168. ACM, 2010.

[14] Il-Young Jeong and Kyogu Lee. Learning temporal features using a deep neural network and its application to music genre classification. In *ISMIR*, pages 434–440, 2016.

[15] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *ICML*, pages 2342–2350, 2015.

[16] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[17] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[18] Sangeun Kum, Changheun Oh, and Juhan Nam. Melody extraction on vocal segments using multi-column deep neural networks. In *ISMIR*, pages 819–825, 2016.

[19] T. LH Li, A. B. Chan, and A. Chun. Automatic musical pattern feature extraction using convolutional neural network. In *Proc. Int. Conf. Data Mining and Applications*, 2010.

[20] Vincent Lostanlen and Carmine-Emanuele Cella. Deep convolutional networks on the pitch spiral for musical instrument recognition. *arXiv preprint arXiv:1605.06644*, 2016.

[21] 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*, pages 159–168. ACM, 2008.

[22] R. Mayer and A. Rauber. Musical genre classification by ensembles of audio and lyrics features. In *ISMIR*, pages 675–680, 2011.

[23] C. McKay, J. A. Burgoyne, J. Hockman, J. BL Smith, G. Vigliensoni, and I. Fujinaga. Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features. In *ISMIR*, pages 213–218, 2010.

[24] C. McKay and I. Fujinaga. Musical genre classification: Is it worth pursuing and how can it be improved? In *ISMIR*, pages 101–106, 2006.

[25] M. McKinney and J. Breebaart. Feature for audio and music classification. In *ISMIR*, pages 151–158, 2003.

[26] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[27] J. Nam, J. Herrera, M. Slaney, and J. O. Smith. Learning sparse feature representations for music annotation and retrieval. In *ISMIR*, pages 565–570, 2012.

[28] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. *ICML*, 28:1310–1318, 2013.

[29] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[30] P. H. Seo, Z. Lin, S. Cohen, X. Shen, and B. Han. Progressive attention networks for visual attribute prediction. *arXiv preprint arXiv:1606.02393*, 2016.

[31] S. Sigtia and S. Dixon. Improved music feature learning with deep neural networks. In *ICASSP*, pages 6959–6963. IEEE, 2014.

[32] A. Smith, C. Zee, and A. Uitdenbogerd. In your eyes: Identifying clichés in song lyrics. In *Australasian Language Technology Workshop*, pages 88–96, 2012.

[33] M. Sordo, O. Celma, M. Blech, and E. Guaus. The quest for musical genres: Do the experts and the wisdom of crowds agree? In *ISMIR*, pages 255–260, 2008.

[34] Carl Southall, Ryan Stables, and Jason Hockman. Automatic drum transcription using bi-directional recurrent neural networks. In *ISMIR*, pages 591–597, 2016.

[35] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[36] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 2012.

[37] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *NAACL-HLT*, pages 1480–1489, 2016.

[38] T. C. Ying, S. Doraisamy, and L. N. Abdullah. Genre and mood classification using lyric features. In *International Conference on Information Retrieval & Knowledge Management*, pages 260–263. IEEE, 2012.