Exploratory Analysis of Multi-Task Learning in Song Genre Classification Using CNN and BERT Architectures

Submission By: 21112254, 16008937, 20175911, 21180859

Abstract

This study explored the application of Multi-Task Learning (MTL) in the domain of song genre classification. Two hard parameter sharing architectures based on CNN and BERT models were employed for this purpose. Further, both architectures were trained with a range of auxiliary tasks concerning song characteristics such as romance, violence, and sadness to assess the impact of their addition on the target task of genre classification. It was observed that the predictive performance of the individual genres changed disparately depending on the auxiliary tasks introduced. In sum, the introduction of auxiliary tasks improved the performance of the CNN-MTL architecture while having negligible impact on the BERT-MTL architecture.

1 Introduction

The advent of music streaming apps has dramatically changed the way artists distribute their media and the way listeners consume it. Online streaming services allow musicians from all over the world to reach their target audience. Earlier curtailed by CDs and hefty records, users can now discover local and international artists at the click of a button. Due to these reasons, music streaming apps have attracted a lot of traction over the past years. Spotify alone has approximately 60,000 songs uploaded daily on the app (Ingham, 2021). Amongst this vast volume of media, artists struggle in acquiring their target audience and listeners grapple with finding tracks suitable to their preferences. An automatic song genre classifier can be used here to group together homogeneous songs to narrow down user search, improve recommendations and streamline retrieval processes.

There are several approaches to genre classification of songs. Classification can be done by analysing audio signal, lyrics or tags created by music specialists, e.g. Million Songs Dataset (Bertin-Mahieux et al., 2011).

Audio signal classification is usually done by analysing the audio spectrogram image (Yang et al., 2017; Bahuleyan, 2018). This approach allows to account for all musical instruments present in a song, which is beneficial since different genres tend to have specific instruments in them. However, the use of audio signals requires large storage space and complicated pre-processing to convert audio signals into spectrograms. Further, existing literature suggests that genre classification based on lyrics analysis performs better than based on audio signal only (Tao Li, 2004; Mayer and Rauber, 2011). Classification accuracy of upto 90% has been achieved by Schreiber (2015) using song tags and metadata from the Million Song Dataset. However, in order to use this model in a real-life production environment, each new song must be first tagged by a specialist which makes this approach very cost-inefficient.

Genre classification using song lyrics is more cost and resource efficient compared to other data input types because data mining and pre-processing tasks are considerably easier to implement and automate for text than other forms. For example, lyrics can be easily found on the internet in the form of user-generated content on several websites. The storage space is also not an issue, as a dataset with 28,000 song lyrics has a size of only 26 MB. This rules out the need for a complicated data storage system. These benefits of using song lyrics instead of other forms of input data have been the motivation behind this study.

The traditional approach to a classification task is to have separate neural networks designed and trained to minimize the error of single tasks independently. However, there is a different approach proposed by Caruana (1997) called Multi-Task Learning (MTL). MTL provides an opportunity

for an improvement in performance by introducing *auxiliary* tasks alongside a *target* task, given that tasks share complementary information or act as regularizers. It also has an additional benefit of reducing the memory requirements and training time because network layers are shared across tasks and part of weights only need to be calculated once.

Therefore, the objective of this study is to implement and compare MTL frameworks that combine auxiliary tasks with the target task of genre classification. These auxiliary tasks include determining different characteristics of a song such as danceability, sadness, or violence of the lyrics. Using this framework, we aim to asses the impact of different combinations of auxiliary tasks on the predictive power of the individual genres.

2 Related Works

There are several existing studies which employ the single task approach in classifying music genres. High performances have been observed particularly in works involving Machine Learning techniques. To elaborate, Kumar et al. (2018) used gradient boosted decision trees to achieve an accuracy of 71.84% when using lyrics to predict 4 genres. Oramas et al. (2016) have semantically enriched album reviews by utilizing an SVM classifier between 13 different genres. A different textual approach was demonstrated by Choi et al. (2014) where they combined song lyrics with user interpretations of music, and trained a KNN classifier to predict the subjects of songs such as war or religion. Furthermore, substantial developments utilizing Deep Learning techniques have been made in the domain of song genre classification using lyrics. Kumar et al. (2018) showed that by using a series of fully connected layers and ReLU activation functions, an accuracy of 74% can be achieved. We find further advancements in the similar field of Emotion Recognition. Ahmad et al. (2020) introduced the attention-based C-BiLSTM architecture which combines both CNNs and LSTMs to classify poetry text into emotional states. Here the hybrid model outperforms models based solely on CNNs and LSTMS.

Existing studoes also reflect the use of BERT (Devlin et al., 2018) and transformers in similar classification tasks. Kelly et al. (2021) compared a series of BERTs aimed at classifying the emotional state of songs using a range of features including lyrics, with precision reaching 0.59.

There is only limited work exploring an MTL approach to the domain of genre classification. Pandeya et al. (2021) undertook a multi-learning and multi-modal approach to classifying songs based on genre and emotions. Their architecture successfully yielded a high classification score while also lowering computational complexity. Boonyanit and Dahl (2021) tackled this task by employing unidirectional and bidirectional LSTM models and achieved an accuracy of 68%. Collobert and Weston (2008) used a MTL framework with CNN encoder and classical NLP features such as Part-Of-Speech Tagging, Named Entity Recognition, Semantic Role Labeling , etc. as tasks that were trained jointly. This approach enabled them to achieve state-of-the-art performance. These works act as primary sources of inspiration for our study.

3 Method

3.1 Dataset

The dataset used for this study is 'Music Dataset: Lyrics and Metadata from 1950 to 2019' (Moura et al., 2020). It comprises of 28,372 songs, their corresponding lyrics and 29 additional features including genre and song characteristics. These genres are Pop, Country, Blues, Jazz, Reggae, Rock and Hip Hop.

3.2 Auxilary Tasks

Choosing suitable auxiliary tasks is highly dependent not only on the target task, but on the nature of the available data (Ruder, 2017). In practice, a common quality looked for in an auxiliary task is for it to be supplementary to the target task.There is no universal notion for what it means for a task to be supplementary in this context, with numerous works proposing different definitions (Caruana, 1997; Ben-David and Schuller, 2003; Xue et al., 2007).

The auxiliary task in this study complements the target task of genre classification by predicting the values of different characteristics of a song such as danceability, sadness and violence based on its lyrics. These characteristics are unique to the dataset used. Below we provide brief descriptions of the features used for the auxiliary tasks:

Violence. Measures the references to physical force intended to damage. The values are 0 for no violence and 1 for completely violent.

Danceability. Measures where the song has a rhythm and style that people can dance to. The

values are 0 for being un-danceable and 1 for completely danceable.

Romantic. Measures the feeling of love and excitement for life in the song. The values are 0 for the void of love and 1 being completely about love.

Sadness. Measures the feeling of sorrow and unhappiness in the song. The values are 0 for being happy and 1 for being completely sad.

All of the above tasks are regression tasks and take values in the range of 0 to 1.

3.3 Class Imbalances

Before performing pre-processing steps, the class frequencies of different genres were examined. Anand et al. (1993) showed that training deep learning models with class imbalances leads to a reduction in performance of classification tasks. Thus, class imbalances were corrected for optimal model performance.

This was done by firstly, eliminating all data points associated with Hip-Hop and Reggae, because these genres had significantly less songs. Further, first few instances of data belonging to the largest classes, i.e. Pop and Country were eliminated to avoid over-representation of certain genres and yield comparable classes. Figures 1 and 2 depict the class sizes before and after these modifications. The resultant balanced dataset had 19,864 data points.

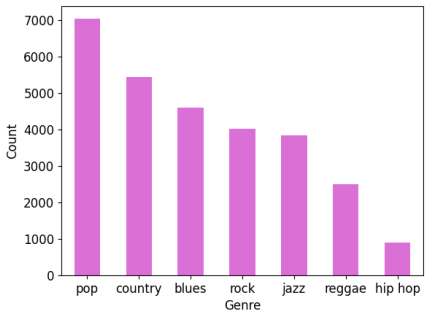


Figure 1: Unbalanced Classes (Raw Data)

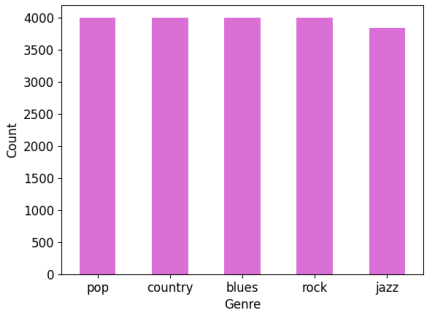


Figure 2: Balanced Classes

3.4 Models

Multi-Task architectures are classically split into hard or soft parameter sharing techniques. Hard parameter sharing networks are generally achieved by sharing a set of hidden layers between all tasks, then branching out into a series of task-specific head networks (Kendall et al., 2018; Chen et al., 2018). In the shared hidden layers, *encoder*, a lower dimensional structure amongst all the tasks is learnt. The features are then passed to task-specific *decoders* where the individual objectives are considered.

In this project, two multi-task models for genre classification were explored. Both models have a similar structure whereby they have a hard-parameter sharing encoder with independent task-specific decoders. Hard parameter sharing was employed as it has been found to reduce the risk of over-fitting, resulting in better generalisation and performance (Baxter, 1997).

CNN and BERT architectures were chosen for the encoder part of the MTL formulations. Below are the details for both models and their high-level architecture visualisations are represented by Figures 3a and 3b.

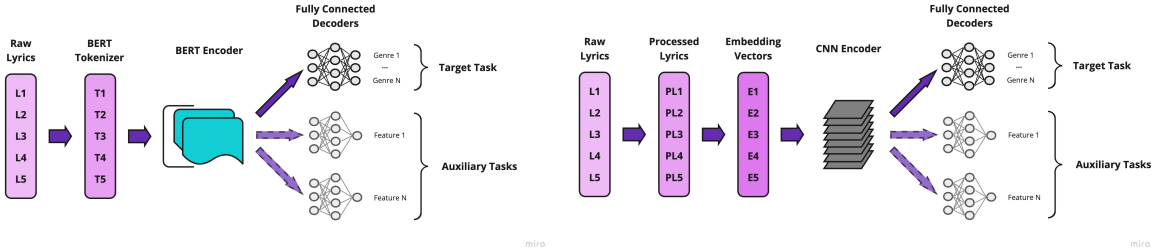
3.4.1 CNN-MTL

The first model utilises a Convolutional Neural Network (CNN) architecture. CNNs are commonly used in multiclassification tasks involving multi-dimensional vectors as input. Since song genre classification is a task of similar nature, the use of CNNs is pragmatic, as it is supported by existing literature (Pelchat and Gelowitz, 2020; Kamtue et al., 2019).

Preprocessing: First some pre-processing steps were performed to extract unigrams from the lyrics. Pre-processing helps discard low-level information and build a collection of words which contributes significantly to the semantics of a song. This allows the extraction of useful knowledge from a pool of vocabulary while also limiting the extent of input data to make modelling feasible. The following pre-processing steps were involved.

Removing Punctuation: All punctuation was excluded as it does not contribute significantly to the meaning of lyrics.

Removing Non-English/Non-ASCII Characters: The considered dataset only constitutes of songs written predominantly in English language. Hence, the rare Non-English and Non-ASCII characters



(a) BERT-MTL Model

(b) CNN-MTL Model

Figure 3: High-Level Architecture

observed were discarded.

Removing Numbers: As mentioned above, the lyrics mainly included English words and the occurrence of numbers was very rare. As numbers did not contribute meaningfully to the semantics of a song, they were removed.

Conversion to Lower Case: We aim for our framework to be case-insensitive for accurate weighting of terms. For example, 'LOVE' and 'love' deliver the same meaning in a song and shall be weighted equally.

Tokenization: The data was segmented into unigram terms called tokens.

Removing Stop-Words: Stop-words are 'filler' terms and are present in abundance throughout any text. Omitting them did not change the meaning of a song.

Lemmatization: Lastly, inflected forms of terms which delivered the same meaning were grouped together by Lemmatization. For example, 'changes' and 'changing' were lemmatized to 'change'.

After performing all preprocessing steps, a vocabulary was built by mapping each unique token to an integer.

Encoder: The encoder is divided in two sections.

Feature Representation: Each token was transformed into an embedding vector. Using these dense and low-dimensional vectors, as opposed to one-hot-encoding or other sparse representations, increases generalization power and decreases computational cost (Goldberg, 2017). This was implemented in the model through PyTorch `nn.Embedding` class. The embedding was learnt jointly with the neural network for the specific task. This approach requires a lot of training data and has the possibility of being slow, but performs well as the embedding is specific to the task.

Feature Extraction: Given the learnt representation, the n-gram features were extracted. This was implemented in our model through three parallel convolutions using the PyTorch `nn.Conv1d` class. CNN is a commonly used method to extract semantic features from the sentences before feeding them to a fully connected layer that performs classification. The parallel CNN architecture allowed to simultaneously extract information about 2, 4 and 6 grams, and therefore provided a classifier with both specific and general information about each lyric. ReLU (Agarap, 2018) was used as the non-linear activation function of the convolutional operations as it has been shown to increase the performance and to speed up the training (Rani and Kumar, 2019). Each convoluted output was then transformed via a max operation in order to record only the strongest semantic signals in the sentence (Severyn and Moschitti, 2015). Finally it was passed through a dropout layer that turns off individual neurons in the network to improve the training efficiency.

The detailed architecture of the CNN encoder is depicted in Figure 4.

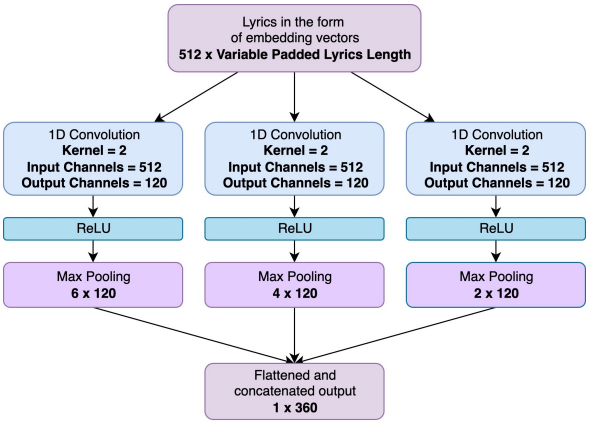


Figure 4: CNN Architecture

3.4.2 BERT-MTL

This utilises Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018).

BERT is used to generate a language model and uses an encoder part of Transformer architecture. The detailed description of this architecture is described in Vaswani et al. (2017). When training BERT, 15% of the input words in each sequence are replaced with a [MASK] token, this is called Masked Learning Model (Masked LM). The Masked ML loss function only depends on the prediction of the masked words. The other part of the training is Next Sentence Prediction (NSP). All the sentences are split in pairs and the model attempts to predict the next sentence. In 50% of the cases, the pair has two subsequent sentences and in the other 50% of the cases, a random sentence from the corpus is chosen as the second sentence. Masked LM and NSP are trained together and the combined loss function of both of the methods is minimized. After the BERT model is trained, it can be fine-tuned for a specific task. In order to perform a genre classification, a fully connected layer is added on top of the Transformer output.

Preprocessing: The transformers library has a tokenizer `BertTokenizer`. As BERT needs information about sentences, no punctuation was removed. So the input to `BertTokenizer` was unprocessed text.

Encoder: A pre-trained model `'bert-base-cased'` in `BertModel` class from the transformers library was used.

3.4.3 Decoders

From the encoders, the networks branches into the task-specific decoder networks. Each decoder or *head* consists of a series of fully connected layers. The output size of the final layer depends on the corresponding task. For classification tasks, the output dimension was the number of classes. The output of the regression tasks was a single numerical value.

3.5 Loss

In both MTL formulations, the model is optimised using the sum of the individual task losses. For the classification tasks the cross-entropy loss was used (Kohler and Langer, 2020). For the regression tasks the Root-Mean-Square-Error loss was used.

4 Experiment

For each MTL formulation, first the performance of the *baseline* model was assessed. The baseline model is the configuration with only the target task of genre classification. Choosing a baseline network structurally close to the MTL variants ensures a fair comparison when the auxiliary task heads are introduced. Then, for each MTL formulation, the performance of a series of networks is assessed where different auxiliary tasks were introduced against the corresponding baseline model. Lastly, a combination of multiple auxiliary tasks were used and their performance against the baseline and single auxiliary task cases were compared.

4.1 Metrics and Displays

The performance of each model was measured through a number of task dependent metrics. For the classification task, F1 scores were computed using the formula below (Sasaki, 2007):

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)}, \qquad (1)$$

where TP is the count of True Positives, FP is the count of False Positives and FN is the count of False Negatives. A higher F1 score implies that there are less misclassifications. For each genre, the corresponding F1 score is provided as well as the weighted average across all the genres.

The performance of the classifiers was also measured using a confusion matrix. The units of the confusion matrix is the count of test songs. To measure the impact of each auxiliary task a confusion matrix of differences was generated. A confusion matrix of differences reflects the difference between the baseline confusion matrix and the auxiliary task confusion matrix. Red cells show an increase of counts for the specific classification when the auxiliary task is introduced, whereas blue shows a decrease. The optimal confusion matrix of differences is expected to show red on the diagonal where the target is equal to the predicted value and blue elsewhere. For the regression tasks RMSE was used as a measurement score.

4.2 Implementation Details

For both MTL architectures, we ran each configuration 5 times and averaged the performance metrics to produce a single score for that configuration. This was done to ensure that the results represent the mean result of a model and to account for the variability of neural network training. For each run, the data set was randomly split into train and test sets in a 0.8:0.2 ratio.

4.3 CNN-MTL

For this architecture, the model was trained using the AdamW optimizer (Loshchilov and Hutter, 2017) for 10 epochs with a learning rate of 1 and a batch size of 32.

4.4 BERT-MTL

For this architecture, Devlin et al. (2018) recommends the number of epochs to be in the range of 2 to 4. This is due to BERT being already pre-trained and training the model further for a larger number of epochs can cause it to overfit and 'forget' the pre-trained weights. Additionally they recommend using a learning rate in the range of $6 \cdot 10^{-5}$ to $6 \cdot 10^{-6}$. Thus, our model was trained for 3 epochs using an AdamW optimizer with a learning rate of $1 \cdot 10^{-5}$ and a batch size of 32.

5 Results and Discussion

5.1 Baseline

Figures 5 and 6 depict the performance of the baseline BERT-MTL and CNN-MTL classifiers, respectively. A heavier concentration of correct classifications (represented by lighter colours) along the true positive diagonal of the confusion matrices is desirable and can be distinctly seen for both models. Additionally it can be seen that although the BERT-MTL generally classifies well, it regularly gets certain combinations of genre predictions incorrect, such as predicting rock when the genre is pop. For CNN-MTL it can be noticed that although there is a clear diagonal trend, the classifier also struggles with certain genre combinations and tends to over-predict blues. Comparatively, the model with the BERT encoder outperforms the one with the CNN encoder. This result is also reinforced by the F1 scores recorded in Tables 1 and 2 where we see an increase in F1 scores across genres as well as with the overall classifier. The F1 scores show that there are similarities between the models, both perform well with country but struggle with blues. However, Figures 5 and 6 display that these similarities can arise for different reasons. Specifically, in the case of classifying blues, the poor F1 score for the BERT-MTL baseline is due to the model *under-predicting* the genre whilst for the CNN-MTL baseline, it is due to the model *over-predicting* the genre.

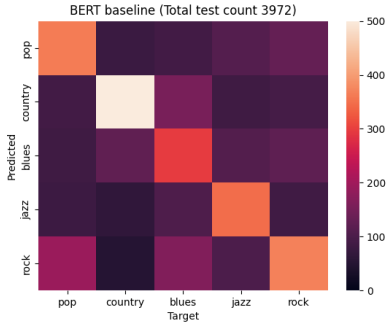


Figure 5: BERT-MTL Baseline Confusion Matrix (Average Test Data Count)

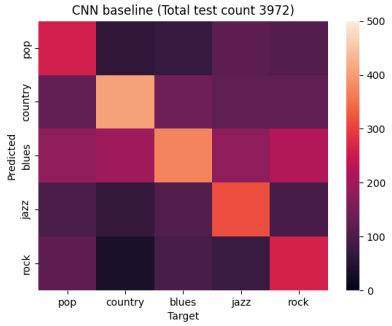


Figure 6: CNN-MTL Baseline Confusion Matrix (Average Test Data Count)

5.2 Single Auxiliary Tasks

For both formulations, the inclusion of an auxiliary task severely impacted their individual performances in genre classification both, positively and negatively. Significant improvements from the baseline performance caused by the addition of an auxiliary task are highlighted in bold in Tables 1 and 2. Here, a significant improvement is quantified as being over one standard deviation away from the baseline performance, where the standard deviations were calculated from the variance over the runs.

To elaborate, inclusion of the romantic regression task in the BERT-MTL architecture notably improved the model's ability to classify blues. On the other hand, incorporating the violence regression task significantly reduced the model's ability to correctly classify rock. This pattern could indicate a correlation between songs classified as blues and rock and the emotion of romance and violence, respectively and should be subjected to further research. Similarly, the CNN-MTL model predicted pop rather accurately in the presence of the danceability regression task while the inclusion

	Genre					
Network	Pop	Country	Blues	Jazz	Rock	Average
Baseline	0.446	0.563	0.366	0.461	0.426	0.452
Violence	0.445	0.550	0.375	**0.487**	0.392	0.450
Romantic	0.450	0.547	**0.392**	0.437	0.420	0.449
Sadness	0.434	0.550	0.380	0.470	0.412	0.449
Danceability	0.447	0.547	0.354	0.435	0.405	0.437
Violence + Romantic	0.448	0.545	0.373	0.443	0.400	0.442
Sadness + Romantic	0.446	0.530	0.380	0.471	0.414	0.448

Table 1: BERT-MTL Results: F1 Scores

	Genre					
Network	Pop	Country	Blues	Jazz	Rock	Average
Baseline	0.344	0.460	0.369	0.407	0.352	0.386
Violence	0.299	0.471	0.382	0.355	0.365	0.374
Romantic	0.380	0.483	0.376	0.377	0.374	0.398
Sadness	0.363	**0.490**	0.360	0.399	**0.391**	0.401
Danceability	**0.387**	0.481	0.358	0.347	0.387	0.392
Sadness + Danceability	0.357	0.479	0.389	0.375	0.380	0.396
Sadness + Romantic	0.372	0.488	0.395	0.348	0.346	0.390

Table 2: CNN-MTL Results: F1 Scores

of violence caused it to perform poorly in predicting jazz.

Furthermore, the addition of identical auxiliary tasks differently impacted the two formulations. For instance, introduction of the violence regression task successfully enhanced BERT-MTL's performance in classifying jazz as observed by an increase in the F1 score from 0.461 to 0.487 whereas, the same value plunged from 0.407 to 0.355 in the case of CNN-MTL. This is depicted by Figures 7 and 8 which show the differences in classifications between the baseline models and the models with the violence regression task. For CNN-MTL, the reduction in F1 score for jazz can be explained by the auxiliary task causing the model to predict blues more. Visualisations for all auxiliary tasks can be found in Appendix B.

In sum, it was observed that the introduction of any auxiliary task to the BERT-MTL model worsened its overall performance in comparison to the baseline and led to reduced F1 scores. Contrastingly, enhanced performances and increased F1 scores were recorded for the CNN-MTL architecture with auxiliary tasks. This could be an indicator that CNN model is more suitable for the encoder part of MTL architecture than the language model generated by BERT. However, despite the positive

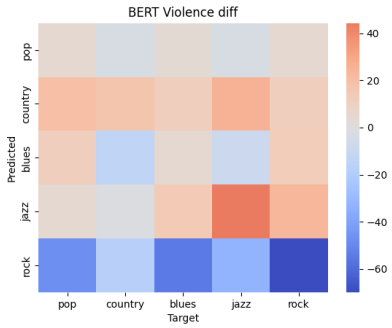


Figure 7: BERT-MTL Confusion Matrix of Differences (Average Test Data Count)

impact of auxiliary tasks on CNN-MTL, it still underperformed against the BERT-MTL model with the exception of blues.

5.3 Multiple Auxiliary Tasks

More often than not, music encompasses a range of emotions rather than a solitary emotive state. Based on this intuition, the introduction of multiple auxiliary tasks was explored to examine if a combination of emotions enhanced the performance of our models. For this purpose, the best performing auxiliary tasks for both architectures were coupled for genre classification.

Specifically, a combination of danceability and

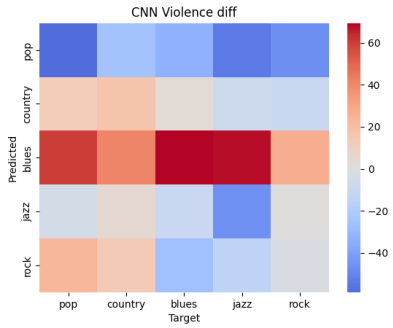


Figure 8: CNN-MTL Confusion Matrix of Differences (Average Test Data Count)

sadness was considered for the CNN-MTL model based on their significant F1 scores noted in the previous section. A model involving both sadness and romantic regression tasks was also assessed as these characteristics yielded the highest overall F1 scores averaged across the genres. The latter highlighted a rather interesting result. In the case of the model with both the sadness and romantic regression tasks, the F1 score for blues increases to 0.395 from 0.376 and 0.360, for single romantic and sadness tasks respectively, which is significantly higher than the baseline. This indicates that certain genres are better represented and classified in the presence of a combination of emotions rather than a single one. However, this result cannot be generalised for all genres. For the same model there is evidence of the converse happening, where for rock, the F1 score decreases to 0.346 from 0.374 and 0.391, for single romantic and sadness tasks respectively, which is considerably below the baseline.

A similar empirical exercise was carried out for the BERT-MTL model. Firstly, a model with both violence and romantic regression tasks was considered as they individually produced the most significant F1 scores over the genres. Further, a model including sadness and romantic regression tasks was assessed for comparison against the similar CNN-MTL case. As per the F1 scores shown in Table 1, the introduction of multiple auxiliary tasks worsened the performance of the BERT architecture across the board.

Similar to the single auxiliary task case, it was noticed that CNN model has benefited more from MTL parameter sharing, than BERT model. A form of inheritance can also be observed whereby the models under/over-predict genres in a similar fashion to their individual single task counterparts. For instance, sadness and danceability, characteristics that performed well overall compared to the baseline, also resulted in a better overall performance when combined together.

6 Conclusion and Limitations

Our empirical exercise suggests that the inclusion of auxiliary tasks can improve the performance of the target task of genre classification. For certain auxiliary tasks, we see evidence of inductive transfer where the shared information enhances the predictive performance. However, a negative transfer was also incurred in some cases. It is unclear what specific qualities of the lyrics and auxiliary tasks cause this transfer to be inductive or negative in nature and should be subjected to further investigation.

Furthermore, the study showcases that the impact of an MTL formulation is not the same over different language encoders. While the model with the BERT architecture outperformed the one with the CNN architecture across the board, considerable enhancements were seen in the latter with the addition of auxiliary tasks. This might be due to the relatively small dataset used in this study whereas, BERT usually demands a greater volume of training data. It is also acknowledged that our study uses a rather simple CNN structure and thus, we aim to explore if the same observation stands true for more complex CNN architectures in future.

Lastly, our results also reaffirm that certain genres can be better predicted by a combination of song characteristics. Naturally, a song genre encompasses a variety of sentiments in which certain emotions might be dominant. Even in our study, some cases of multiple auxiliary tasks showed instances of inherited characteristics from their single task counterparts. In future advancements, we hope to investigate this further by varying the impact of individual auxiliary tasks. This can be achieved by introducing weights for the task-specific losses. There are numerous existing weighting strategies depending on the desired output. For instance, a constant weighting strategy can be consulted to explore the impact of disparate auxiliary tasks, or a random loss weighting strategy can be considered to boost performance (Lin et al., 2021).

References

Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Shakeel Ahmad, Muhammad Zubair Asghar, Fahad Mazaed Alotaibi, and Sherafzal Khan. 2020. Classification of poetry text into the emotional states using deep learning technique. *IEEE Access*, 8:73865–73878.

Rangachari Anand, Kishan G Mehrotra, Chilukuri K Mohan, and Sanjay Ranka. 1993. An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 4(6):962–969.

Hareesh Bahuleyan. 2018. Music genre classification using machine learning techniques. *CoRR*, abs/1804.01149.

Jonathan Baxter. 1997. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, 28(1):7–39.

Shai Ben-David and Reba Schuller. 2003. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines*, pages 567–580, Berlin, Heidelberg. Springer Berlin Heidelberg.

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. 2011. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

Anna Boonyanit and Andrea Dahl. 2021. Music genre classification using song lyrics.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28:41–75.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR.

Kahyun Choi, Jin Ha Lee, and J Stephen Downie. 2014. What is this song about anyway?: Automatic classification of subject using user interpretations and lyrics. In *IEEE/ACM Joint Conference on Digital Libraries*, pages 453–454. IEEE.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, 10(1):90–92.

Tim Ingham. 2021. Over 60,000 tracks are now uploaded to spotify every day. that's nearly one per second.

Kawisorn Kamtue, Kasina Euchukanonchai, Dittaya Wanvarie, and Naruemon Pratanwanich. 2019. Lukthung classification using neural networks on lyrics and audios. In *2019 23rd International Computer Science and Engineering Conference (ICSEC)*, pages 269–274.

Markelle Kelly, Kai Malloy, Mathias Moelgaard, Rohan Mannem, and André Rösti. 2021. Cs 274c project: An exploration of bert for song classification and recommendation.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491.

Michael Kohler and Sophie Langer. 2020. Statistical theory for image classification using deep convolutional neural networks with cross-entropy loss. *arXiv preprint arXiv:2011.13602*.

Akshi Kumar, Arjun Rajpal, and Dushyant Rathore. 2018. Genre classification using word embeddings and deep learning. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2142–2146.

Baijiong Lin, Feiyang Ye, and Yu Zhang. 2021. A closer look at loss weighting in multi-task learning. *arXiv preprint arXiv:2111.10603*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Rudolf Mayer and Andreas Rauber. 2011. Music genre classification by ensembles of audio and lyrics features. pages 675–680.

Luan Moura, Emanuel Fontelles, Vinicius Sampaio, and Mardônio França. 2020. Music dataset: Lyrics and metadata from 1950 to 2019.

Sergio Oramas, Luis Espinosa-Anke, Aonghus Lawlor, et al. 2016. Exploring customer reviews for music genre classification and evolutionary studies. In *The 17th International Society for Music Information Retrieval Conference (ISMIR 2016), New York City, United States of America, 7-11 August 2016*.

Yagya Raj Pandeya, Jie You, Bhuwan Bhattarai, and Joonwhoan Lee. 2021. Multi-modal, multi-task and multi-label for music genre classification and emotion regression. In *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1042–1045.

Nikki Pelchat and Craig M. Gelowitz. 2020. Neural network music genre classification. *Canadian Journal of Electrical and Computer Engineering*, 43(3):170–173.

Sujata Rani and Parteek Kumar. 2019. Deep learning based sentiment analysis using convolution neural network. *Arabian Journal for Science and Engineering*, 44(4):3305–3314.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Yutaka Sasaki. 2007. The truth of the f-measure.

Hendrik Schreiber. 2015. Improving genre annotations for the million song dataset. In *ISMIR*, pages 241–247.

Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 959–962.

Mitsunori Ogihara Tao Li, Chengliang Zhang. 2004. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15):2429–37.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

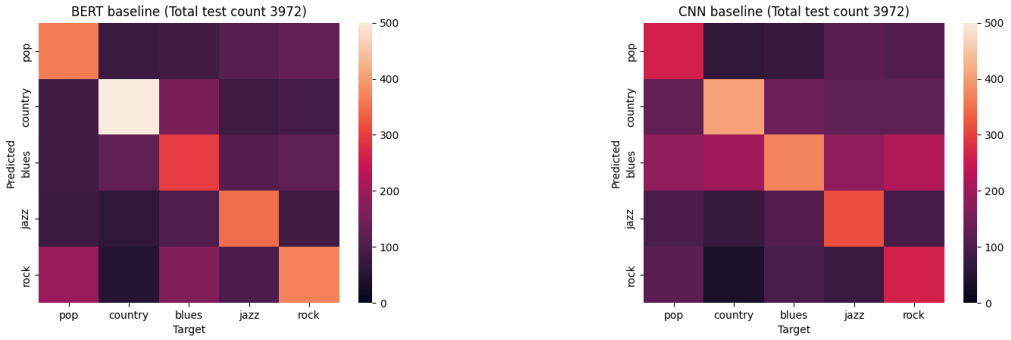
Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. 2007. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(1).

Li-Chia Yang, Szu-Yu Chou, Jen-Yu Liu, Yi-Hsuan Yang, and Yi-An Chen. 2017. Revisiting the problem of audio-based hit song prediction using convolutional neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 621–625.

A
CNN encoder parameters

We use three parallel 1D convolution filters with kernels 2, 4 and 6. Input channels number is the same as the number of the embedding dimensions and is equal to 300. The number of output channels for each convolution is equal to 120. 1D max pooling is applied to each convolution and all three outputs are concatenated to produce the flattened vector of size 360.

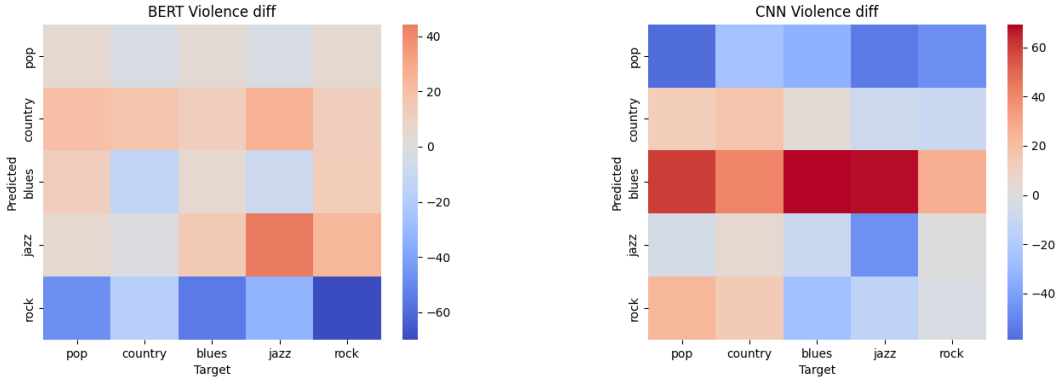
B
Additional results for different task configurations



(a) Baseline Confusion Matrix: BERT

(b) Baseline Confusion Matrix: CNN

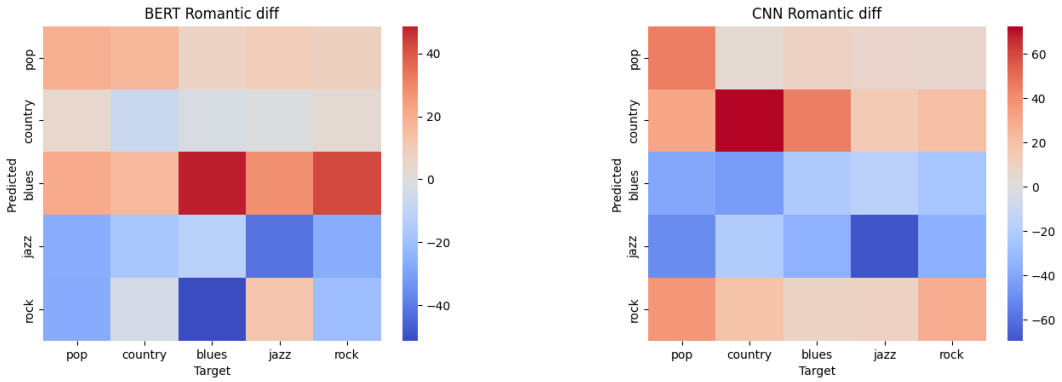
Figure 9: Baseline confusion matrices



(a) BERT-MTL

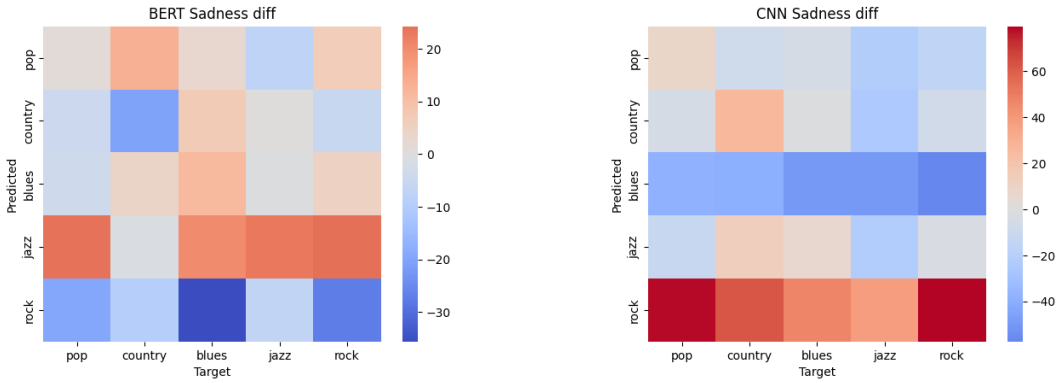
(b) CNN-MTL

Figure 10: Violence confusion matrices difference



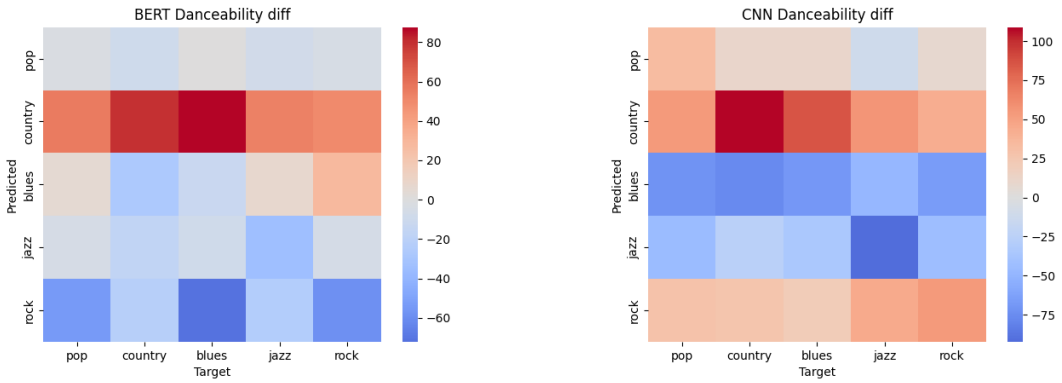
(a) BERT-MTL (b) CNN-MTL

Figure 11: Romantic confusion matrices difference



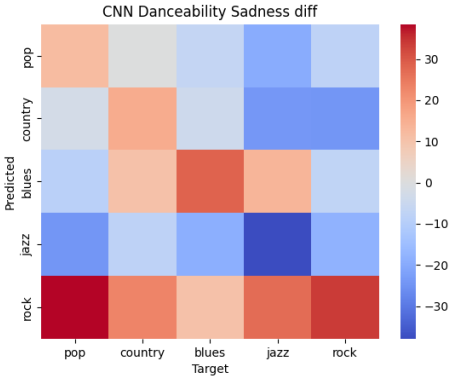
(a) BERT-MTL (b) CNN-MTL

Figure 12: Sadness confusion matrices difference

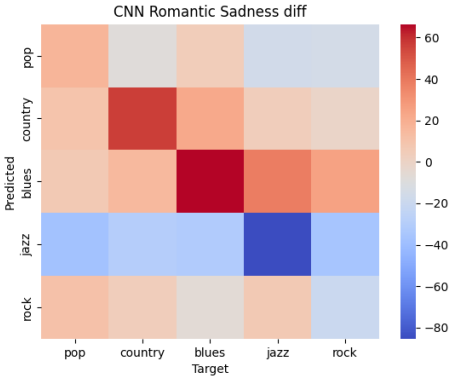


(a) BERT-MTL (b) CNN-MTL

Figure 13: Danceability confusion matrices difference

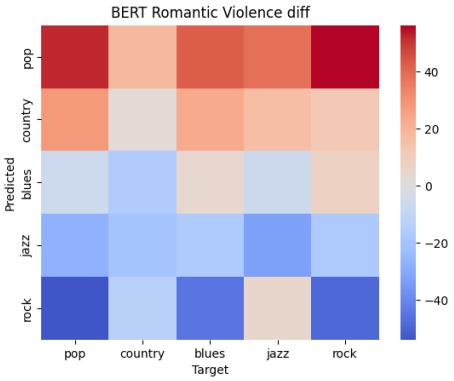


(a) Auxilary tasks danceability and sadness

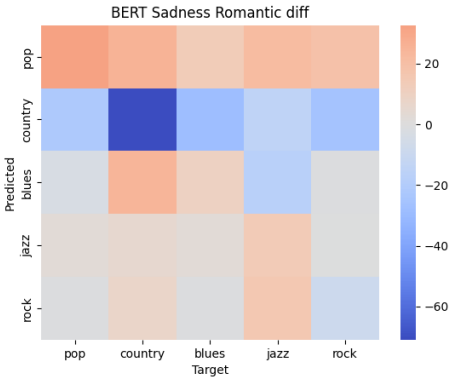


(b) Auxilary tasks romantic and sadness

Figure 14: CNN-MTL models with a combination of auxiliary tasks



(a) Auxilary tasks danceability and sadness



(b) Auxilary tasks romantic and sadness

Figure 15: BERT-MTL models with a combination of auxiliary tasks