Towards Empathetic Music Generation

Hu Yidi
e0202948@u.nus.edu
National University of Singapore
Singapore, Singapore

Poh Lin Wei
e0325923@u.nus.edu
National University of Singapore
Singapore, Singapore

Ye Chenchen
e0261968@u.nus.edu
National University of Singapore
Singapore, Singapore

ABSTRACT

Recently, research in automatic music generation systems seems to have gained traction. Unfortunately, it appears that many of these models are not easily controllable: users are unable to interact with the system to influence the generator's output. Given that there are many possible applications for music generators that can compose pieces that express a user-specified emotion, this trend is regrettable. In an effort to address this shortcoming, we explored several ways in which a music generator can be conditioned to generate pieces when given an emotion label. In our work, we looked at different long-short-term-memory (LSTM) -based models that are trained to generate pieces from four different emotion categories, based on the valence-arousal (VA) model, and compared their performance. Moreover, to evaluate these models' performance we trained a classifier that can identify the emotion expressed by the music piece it is given. Human evaluation of our models was also conducted.

CCS CONCEPTS

• **Applied computing** → **Sound and music computing**.

KEYWORDS

music generation, music classification, affective computing

ACM Reference Format:
Hu Yidi, Poh Lin Wei, and Ye Chenchen. 2021. Towards Empathetic Music Generation. In *N.A.* ACM, New York, NY, USA, 8 pages. https://doi.org/N.A

1 INTRODUCTION

In the last few years, research in music generation systems has gained traction. For instance, in 2016, Google announced Magenta which is a research project that explores the use of machine learning in creating music and art [1, 2]; in 2017, the singer Taryn Southern released an album which was produced with the help of automated music generation systems [12]; and in 2019, OpenAI announced MuseNet which can compose 4-minute musical pieces in different styles [3]. Despite these advancements, relatively lesser attention has been paid to developing systems that allow listeners to control the emotions expressed in the generated music pieces.

CS4347, April, 2021, National University of Singapore

ACM ISBN N.A…$N.A
https://doi.org/N.A

This is regrettable as such systems have useful applications, like in music therapy and/or music medicine for individuals with depression. Specifically, under certain circumstances, the use of new and unfamiliar generated music during therapy sessions may be preferred since it can provide the client with a new perspective [13]; and avoid triggering unpleasant memories [9, 20]. With that said, merely being able to generate music may not suffice in some situations. Instead, it will be more desirable if the user could generate music which expresses a specific emotion. The generated music may then be used to alter the listener's mood [15]; or as a medium through which he or she can explore and experience the specified emotion [7]. Given these considerations, attempts to develop systems that can generate music, with emotional content which matches that selected by the listener, are undoubtedly constructive.

In view of this and the relative success of conditional neural network models in other domains [17, 26], we trained several emotion-conditioned LSTM-based music generators, and compared their relative performance. In particular, we experimented with four different LSTM-based architectures and found that our *MultiInput* architecture, as described in subsection 3.2, performed best in terms of predicting the next note to generate, when given a primer and an emotion label. Since it is desirable to have a model that does not only generate music that expresses the user-specified emotion but also one that generates music of reasonable quality, we decided to solely evaluate this model. This was done using a music-emotion classifier and by conducting a small-scaled user study.

Through this study, in contrast to the observations made in other generation tasks [26], we found that simply conditioning a music generator with an emotion label might not suffice, if we want the model to be capable of changing the emotion expressed by the primer it is given. Nevertheless, we unexpectedly discovered that conditioning the model did not degrade the quality of the music generated even though emotion can be highly subjective and thus introduce 'noise' into the input. On the contrary, based on the results of a subjective listening test, our conditioned model seems to generate pieces that are more pleasant than those generated by a music generator, with a very similar architecture, that is not conditioned on emotion.

2 RELATED WORK

2.1 Generation with Handcrafted Rules

Before deep-learning techniques became a popular choice for generation tasks, researchers have explored the use of handcrafted rules to develop such a system [18, 19, 24]. However, such approaches tend to lead to less-than-satisfactory results. For example, it is difficult, even impossible, to enumerate all the rules such that the generated piece is not only satisfactory but also diverse, especially given the complex nature of emotions.

2.2 Generation with Hidden Markov Chain

An alternative approach to the rule-based approaches described above is to allow the system to learn and determine the musical features and elements that can express a given emotion.

Monteith et al. [21] adopted such an approach to generate music that expresses six emotions: anger, fear, joy, love, sadness or surprise. Specifically, they used two Hidden Markov Models to generate the rhythm and pitches separately. Then, these generated phrases are used as input into one of the six neural networks, each of which is trained to determine whether the input expresses a specific emotion. The generated phrase will then only be used when the classifier predicts that it does indeed express the desired emotion. These phrases are then further processed to generate the final composition. The effectiveness of this approach is indicated by a listening test conducted by Monteith et al. A major drawback of this work is that on average, multiple pieces will have to be generated before suitable piece is accepted.

2.3 Generation with Deep Learning Techniques

Very recently, Tan et al. developed Music Fadernets [25], a music generator that allows users to steer the level of arousal expressed in the piece of music. This system is based on a variational autoencoder (VAE). While it is impressive that listeners have found that their model is effective, it does not allow its users to steer the level of valence, another critical dimension in the emotion space as described by the valence-arousal (VA) model. Moreover, VAEs are notoriously difficult to train [14]. Clearly, in the event that time is extremely limited for a study, adopting such an approach may not be wise.

Around the same time, Ferreira et al. [11] looked at the use of LSTM models for the generation of music with a user-specified sentiment. Their work is largely informed by Radford et al.'s multiplicative long short-term memory (mLSTM) model [23] which has been found to be effective at generating reviews when given a positive-sentiment or negative-sentiment label. In the same spirit, Ferreira et al. studied how mLSTM could be used to generate pieces that express positive and negative emotions. However, in our study, we hope to generate pieces from four – as opposed to two – categories.

Lastly, as an extension, Ferreira et al. developed Bardo composer [10]. This system is more complex than those previously discussed as it seeks to generate music pieces for games such that these pieces are aligned with the emotion expressed in the users' speech. Nevertheless, the relevant component of the system, which is responsible for generating music based on a given emotion, is a language model that is guided by, what the authors term, Stochastic Bi-Objective Beam Search (SBBS). With this method, they found that their system is generally effective at composing pieces that convey emotions from four different categories. However, as search is involved during generation, the system might not be as quick in creating a new piece. In contrast, in our system, no search is involved during the generation of music as the model is trained to generate pieces that express the given emotion label.

3 PROPOSED METHODS

In this study, we explored different LSTM-based architectures for the generation of music with a user-specified emotion label. Moreover, as we were aware of the time and resource constraints we had, we decided to evaluate the effectiveness of our music generator's pieces using a music-emotion classifier. (This evaluation was then complimented with a small-scaled user study.) Since we needed such a classifier for our evaluation, in this section, we elaborate on both the generators and classifiers we explored. Furthermore, we briefly discuss about our chosen emotion representation.

3.1 Emotion Representation

There are two commonly used emotion representations: categorical representation and dimensional representation. Presently, there is no consensus on which is better, and each has its advantages and disadvantages [29].

We chose the dimension approach, because, as discovered by Cespedes-Guevara and Eerola [5], perceived emotion in music is nuanced and varied. Hence, the use of a continuous representation of emotion is likely to be more suitable. Besides, dimensional representation strips away the ambiguity present in the emotion categories [28]. To be specific, we chose the VA model as shown in figure 1.

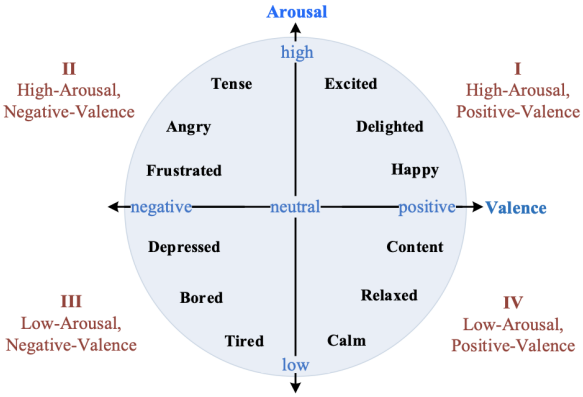

Figure 1: Valence-Arousal Representation of Emotions by Yu et al. [30]

While we have chosen to use the VA-model representation, we were unfortunately unable to make full use of its expressive nature due to the limitations of our dataset. In particular, as far as we are aware, the only relevant dataet only considers the four emotion categories, as demarcated by the axis of the VA model. For more details of the dataset used, refer to subsection 4.1.

3.2 Generator

Like Ferreira et al. [10], we formulated the problem of generating music as a language modelling task. In particular, using the music representation suggested by Ferreira et al. [11], we were able to represent each musical note as a token $x_i \in X$, where X is the set

of all tokens or the *vocabulary* V. From this perspective, a piece of music is a sequence $x = (x_1, ..., x_n)$.

As such, we usually want to train a model that is capable of learning the probability distribution

$$p(x) = \prod_{i=1}^{n} p(x_i|x_{<i})$$

In fact, since we want the model to generate pieces based on a given emotion e, we need to incorporate that in the model as well. This gives us

$$p(x|e) = \prod_{i=1}^{n} p(x_i|x_{<i}, e)$$

In this case, $e \in \{e_{0,0}, e_{0,1}, e_{1,0}, e_{1,1}\}$, where $e_{0,j}$ indicates that the arousal is low, $e_{1,j}$ indicates that the arousal is high, $e_{i,0}$ indicates that the valence is low and $e_{i,1}$ indicates that the valence is high.

To model such a distribution, we trained four different LSTM-based models that are conditioned on emotion and hypothesised that model that does not treat the emotion label as a time-dependent information will outperform the rest. This hypothesis is premised on the fact that we want the entire piece, not merely a segment of it, to express the selected emotion. Therefore, it should not be time dependent.

These models take in a fixed sequence length l of tokens at each time step and seek to predict the next token.

Baseline: Music Generator without Conditioning

Since we hope to have a model that is not only capable of generating music that expresses a particular emotion but also one that generates music of reasonable quality, we decided to train a vanilla music generator. This will allow us to compare the performance of our conditioned models easily. For instance, we will be able to determine whether our choice of conditioning has caused the model to fail to converge optimally, by comparing the conditioned model with the non-conditioned one.

We developed a *NoCond* model with the architecture shown in figure 2. The layers are an embedding layer, a Bidirectional-LSTM layer, 2 LSTM layers, 3 dropout layers (each one is placed after a recurrent neural network layer) and a fully-connected layer. In addition, dim_{embed} refers to the embedding dimension while $units_{rnn}$ refers to the number of units used in the network.

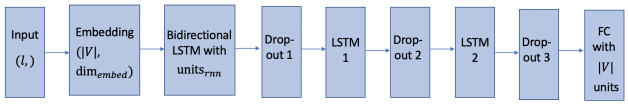

Figure 2: Architecture for *NoCond*

Conditioning by Concatenation, without Embedding

Inspired by Kaliakatsos-Papakostas et al.'s [16] discovery that conditioning a music generator by concatenating rhythmic information enabled them to influence the generated pieces' rhythm, we decided to adopt a similar approach. In particular, in this $Concat_1$ model, for every time step t, we simply concatenated the token x_t and the 1×2 emotion label $e_{i,j}$, which is represented as $[i, j]$, to give $[x_t, i, j]$.

Since off-the-shelf embedding layers often take a one-dimensional input but we have a multi-dimensional input for this model, we had to leave out the embedding layer. The resulting architecture is shown in figure 3. Notice that it is almost identical to that shown in figure 2 in that it has all the layers except the embedding layer.

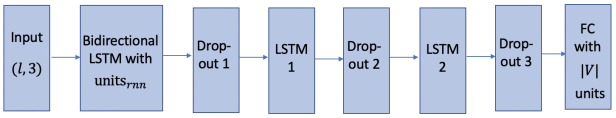

Figure 3: Architecture for $Concat_1$

Conditioning by Concatenation, with Embedding

To use an embedding layer, we decided to flatten the input used for $Concat_1$. With this change, the input to the model at each iteration is $[x_0, i, j, ..., x_t, i, j, x_{t+1}, i, j, ..., x_l, i, j]$. This input is then passed into an embedding layer before it is being reshaped to 'restore' its original representation. Details of this architecture are shown in figure 4.

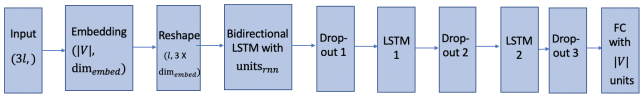

Figure 4: Architecture for $Concat_2$

Conditioning by Vocabulary Expansion

Based on the observation that the same musical note (or more precisely, pitch) can have different harmonic purposes in music and thus convey different emotions, depending on the tonality of the piece, we also attempted exploring conditioning by associating each token with an emotion label. Specifically, x_t in a piece that expresses emotion $e_{i,j}$ is viewed to be distinct from x'_t in a piece that expresses emotion $e_{i',j'}$, where $i \neq i'$ or $j \neq j'$, regardless of whether x_t and x'_t are equal.

Since we have four distinct emotion categories, we needed to quadruple our vocabulary size. However, these changes should only affect the input tokens, not the output tokens, because ultimately, we want to generate the actual notes. In other words, the vocabulary of the output tokens is $|V|$, not $4|V|$.

The architecture of this model *Expanded* is illustrated in figure 5.

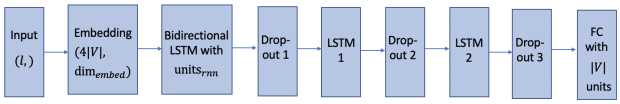

Figure 5: Architecture for *Expanded*

Conditioning with Multiple Inputs

Finally, in an attempt to decouple the emotion label from the sequential model, we decided to separate the note input from the emotion input. The resulting model *MultiInput* is shown in figure

6. Intuitively, this model should be easier to train, and possibly even perform better because in this case, the emotion label is not a time-dependent information and hence should not be processed by a sequential model like the LSTM.

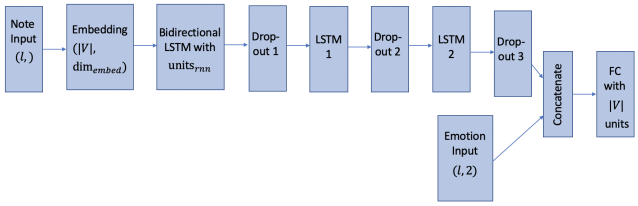

Figure 6: Architecture for *MultiInput*

3.3 Classifier

In this subsection, we discuss on the various music emotion classifiers we explored.

SVM and MLP

As highlighted by Yang et al.[27], two of the most common approaches to develop a music-emotion classifier involve the user of support vector machines (SVM) and neural networks. Therefore, we decided to explore using SVM and Multilayer Perceptrons (MLP), a classical type of neural network.

To use SVM and/or MLP, we needed to extract relevant music features. We referred to the paper A Music Emotion Recognition Algorithm with Hierarchical SVM Based Classifiers[6] to determine which features might be relevant. Specifically, Chiang et al. proposed using 35 features, including dynamics, rhythm, pitch, and timbre.

With the extracted features, we trained the SVM-based and MLP-based classifiers to recognise emotions labelled according to the valence-arousal model.

LSTM

Long Short Term Memory (LSTM) model and its variation is the state-of-the-art model for the music classification task. For example, Chun et al. [8] used it in a hierarchical way for music genre classification. Thus, we would like to also utilize the sequential nature of music and take the note order into consideration by using the LSTM model.

GPT-2

Generative Pretrained Transformer (GPT) model which was first proposed by Alec et al. [4] in 2018, was largely used in natural language processing tasks. Instead of typical approach where models are trained in a supervised learning way on task-specific datasets, the usual way of adopting GPT model contains two stages: pretraining and finetuning.

In the pretraining stage, the model is usually trained on several large collections without any explicit supervision. In our experiment, to let the model learn information in the music domain, we pretrained the model on ADL-Pinao-Midi dataset [10], which consists of 11,086 piano pieces from different genres.

Then the finetuning stage is a supervised learning stage where the model is tuned towards a specific task. In our music-emotion classification task case, we add a classfication layer on top of the pretrained GPT-2 model and compile them as a new GPT-2 classification model. We then train the model in an end-to-end way on the VGMIDI data set, where the model takes in the encoded music text expression and predicts its emotion label.

4 EXPERIMENTS

In this section, we discuss the details of our experiments. This includes the dataset used, hyper-parameters of the models and implementation details.

4.1 Dataset

The dataset used for our project is the VGMIDI (Ferreira and Whitehead 2019) dataset[11]. VGMIDI is a dataset of 200 MIDI labelled piano pieces (video game soundtracks). Each piece was annotated according to a valence-arousal model of emotion. The valence and arousal values are discrete values in $\{0, 1\}$: if the valence is positive, the value is one; otherwise, it is zero; similarly, if the arousal is high, the value is one; otherwise, it is zero. Since the authors have split the dataset into test and training, we did not further perform splitting of the dataset.

To fully train the classifiers, we follow the data augmentation technique proposed by Lucas [10]. Each original piece is modified on its key, tempo and velocity, and therefore, we are able to generate 108 different pieces from each original piece after augmentation.

4.2 Generator

The generator was trained on Tesla K80 GPU as provided by the free version of Google Colab. The implementation was done in a Jupyter notebook. We primarily used TensorFlow as our machine learning library.

For all models, we followed the approach taken by [22] and set the number of cells $units_{rnn}$ used for each LSTM layer (including the birdirectional layer) to 512. We also fixed the sequence length l to 100 and kept the embedding dimensions dim_{embed} to 256. Moreover, we set the dropout rate for dropout layer 1, 2 and 3 to 0.4, 0.6 and 0.7, respectively. Lastly, we used the Adam optimiser with a learning rate of 0.0003 and the categorical cross entropy loss function, and trained the models for 200 epochs, at which point any further increase in accuracy is minimal.

Due to the resource constraint we faced, to pick the best model, we trained the models on three quarters of the training dataset available. Then, we picked the best performing model, based on how accurately it is able to predict the next token, and trained it on the entire training dataset.

We only selected the model that performed best at predicting the next note correctly because it is important for the model to not only generate pieces that convey a specified emotion but also pieces with reasonable quality. Moreover, since the notes in a music piece are chosen by a composer to express a specific emotion, the ability to correctly predict the next note is also an indication that the model has learnt the conditional probability distribution well.

The selected model is then used to generate 12 pieces, 3 from each emotion category. (We were only able to generate 12 pieces as

the test dataset provided by VGMIDI is rather small and there are very few pieces with arousal and valence values that are both zero.) These were fed then into the best classifier we trained, enabling us to evaluate whether the model is capable of generating pieces with the intended emotion.

To further investigate the efficacy of the best model, we also generated pieces by conditioning the model on emotions that differed from the primer's emotion label. For example, if the primer was obtained from a piece with $e_{0,0}$, we would condition the model with different labels $e_{1,0}$, $e_{0,1}$ and $e_{1,1}$. If the later half of the generated piece indeed expresses the emotion used to condition the model, it will suggest that the model can also be used to alter the emotion conveyed by a piece of music. If our model is capable of altering a piece's emotion, it will be a significant advantage because it that users can use this system to generate pieces that is aligned to their emotions even when they do not have a suitable primer.

4.3 Classifier

SVM and MLP

This method composed of the following main procedures: midi files data preparation, feature extraction, classifier construction and comparison.

In the data preparation stage, we implemented codes to traverse the folder structure of the midi files, create a dataframe to hold the contents, then join with the emotion label dataframe. This would facilitate training and the prediction in later steps.

In the feature extraction part, although Chiang et al. [6] proposed a detailed method in which a total of 35 features from dynamic, rhythm, pitch, and timbre of music were generated from music audio recordings, the implementation would not be exactly the same as the reference model performed feature extraction from audio files whereas our project intends to perform training on midi files. Moreover, the timbre features that Chiang et al. proposed were not considered in our case, because our midi files are all piano pieces in which there would not be significant difference in the timbre aspect. In this model, we extracted a total of 21 features related to tempo, pitch and loudness of music from midi files. Tempo features are related to rhythm of the music. Pitch features include calculation of pitch related characteristics of midi files. Loudness features include calculation of loudness related characteristics of midi files.

In order to extract features from midi files, we used the python package called pretty_midi, and implemented functions to extract relevant information from midi files. Tempo, pitch and loudness of notes are information that can be extracted from midi files. For the pitch related features, we included calculation of pitch mean, median, variance and pitch class histogram. By storing pitch of each note in an array, we could calculate the following features:

- *Pitch Mean*: the average pitch for each note.
- *Pitch Median*: the median pitch for each note.
- *Pitch Variance*: the variance of pitch for each note.

The pitch class histogram features were not implemented in the research paper before, but we implemented them as they were found to increase the accuracy of the model in the experiment.

For the loudness features, we included calculation of note loudness mean, variance, range and root-mean-square. By storing loudness information of each note in an array, we could calculate the following features:

- *Loudness Mean*: the average loudness for each note.
- *Loudness Variance*: the variance of pitch for each note.
- *Loudness Range*: the difference between the maximal and minimal Loudness for each note.
- *Loudness Root − Mean − Square*: the root mean square value of Loudness for each note.

These loudness features can reflect the dynamic characteristics of music pieces.

After the feature extraction, we used scikit-learn to fit our data to neural networks with different configurations and an SVM. scikit-learn has a MLP model as well as an SVM model that we could use for supervised learning.

LSTM

The implementation of the LSTM classifer is on a single GeForce RTX 2080 GPU. The LSTM model was trained with a batch size of 8 and dropout rate of 0.25. It contains 4 hidden layers and the dimensions of both embedding layer and hidden layer are set to 1024. We trained the LSTM classifier for 10 epoches using an Adam optimizer with learning rate of 3e-5.

GPT-2

To make a more direct comparison on the classification models' performance, we make the implementation settings for the GPT-2 model as similar as possible to the LSTM model. It is also trained on a single GeForce RTX 2080 GPU with a batch size of 8 and dropout rate of 0.25. We constrained the maximum input sequence length to be 1024 to fit the model size. We use a 4-layer transformer with 8 heads for the multi-head attention layer. In the finetuning stage, similar to the training of LSTM classifier, we also trained the GPT-2 classifier for 10 epoches using an Adam optimizer with learning rate of 3e-5.

5 RESULTS

5.1 Classifier

Table 1 shows the training and testing accuracy for different classifiers on the VGMIDI dataset.

For the LSTM and GPT-2 classifiers, though they were constructed and trained under similar settings, the GPT-2 classifier significantly outperforms the LSTM classifier (and in fact, other models). It shows that the pre-training process benefits the classification model by providing it with sufficient domain knowledge and thus booting its learning ability in specific tasks during fine-tuning. Hence, based on the accuracy result shown in Table 1, we chose the GPT-2 classifier for later evaluation.

On this note, interestingly, we observed that although MLP performed better than SVM in terms of training accuracy, it has a lower accuracy in the tests. This could be because the MLP classifier actually over-fitted the training data.

Classifier Model	Training	Testing
SVM	0.76	0.65
MLP	0.84	0.63
LSTM	0.47	0.45
GPT-2	**0.99**	**0.76**

Table 1: Classification Accuracy of Different Classifiers

Model	Test Accuracy
NoCond	**0.99**
$Concat_2$	0.88
Expanded	0.96
MultiInput	**0.99**

Table 2: Next-Token Generation Accuracy of Different Models

Type of Piece	Classification Accuracy
Aligned	0.625
Differing	0.33

Table 3: Emotion Accuracy of Generated Pieces

5.2 Generator

Table 2 shows the testing accuracy of the various music generators after being trained on a subset of the training dataset for 200 epochs. We intentionally left out the results for $Concat_1$ as the model did not converge optimally. Instead, during training, its validation accuracy hovered around the range of 0.1 and 0.2.

From this table, we observe that the way in which conditioning is performed for *Expand* and *MultiInput* did not cause the music generator to perform more poorly in terms of its ability to generate the next token. In fact, *MultiInput* has a test accuracy that is very close to that of *NoCond*, when rounded off to 3 decimal places, if not identical, when rounded off to 2 decimal places.

It is not shown in this table but the results for *MultiInput* and *NoCond*, when trained on the full training dataset, remained at 0.99.

Table 3 shows the accuracy of the generated pieces' emotions when compared to the given emotion label. The row "Aligned" means the target emotion label taken into the generator is aligned with the music primer's emotion; whereas the row "Differing" means the target emotion label is different from the music primer's emotion. The accuracy of the emotions expressed is evaluated by the GPT-2 classifier which we have trained.

6 ANALYSIS AND DISCUSSION

In this section, we discuss the results we have obtained and provide some reasons for them.

6.1 Next-Token Prediction Accuracy

It was heartening that we managed to find a way to condition the model with emotion labels without causing significant degradation in the model's performance. This highlights that the emotion label is informative. Initially, we were concerned that the introduction of emotion label could result be treated as 'noise'. However, as observed from the results in table 2, this is fortunately not the case. In hindsight, this could be because the dataset we used only contains game music. In particular, it is observed that listeners tend to be better able to come to a consensus about the emotions expressed by game music, when compared to other types of music [11]. As a result, there will not be as large a standard deviation in the musical characteristics that is associated to a particular emotion. Therefore, the models are less likely to view the additional information as noise, and even find them useful. For example, when training the models multiple times, we observed that there were several occasions when *MultiInput* performed better than *NoCond* in terms of validation accuracy. Nevertheless, the eventual models we chose, based on the validation accuracy, happened to give the same test accuracy.

In addition, through these experiments, we observed that $Concat_1$ performed the worst among all the models. In fact, its performance was so disappointing that we stopped training it after 150 epochs because its validation loss was no longer decreasing but stagnant or increasing. This is in stark contrast to $Concat_2$ which is even capable of achieving a test accuracy of 0.88. Given that $Concat_1$ and $Concat_2$ largely differs in the fact that $Concat_2$ has an embedding layer while $Concat_1$ does not, it hints that the input we have is extremely sparse. Hence, the use of an embedding layer is essential to training an efficacious model.

Moreover, the fact that *Expanded*'s performance is rather high could lend some credence to our hypothesis that identical tokens with different emotions should be viewed differently. In this case, it seems unlikely that the embedding layer has negated the impact of expanding the vocabulary size because if it did, we would expect its performance to be much closer to *NoCond* when it converges. Nonetheless, we have to confess that the approach of increasing the vocabulary size may not be feasible if more emotion categories have to be considered.

Lastly, given that *MultiInput* performed the best among all models, it is likely that our proposition that, in this context, emotion labels should not be treated as a time-dependent information.

6.2 Emotion Accuracy

Based on the sample we used, the pieces with aligned emotion performed better than those that differed from the primer's emotion label. This is despite the fact that we have segmented the later half of the piece. This could possibly be caused by our model's inability to alter the emotion of a piece. Alternatively, it could be caused by the fact that the generated piece is too short for any changes in musical characteristics to be significant. Unfortunately, we are not able to reliably determine whether this is the case by lengthening the generated pieces because with our approach of greedily sampling from the learnt distribution, we are unable to generate longer pieces without encountering significant performance degradation.

It is regrettable that these results indicate that our models might not have learnt the emotion classes well. Nevertheless, we will like to bring up that the results obtained here may not be indicative as the classifiers, while impressive, still have room for improvement. Despite knowing this, we chose to use the classifiers for evaluation

Music	Average Quality Score
Conditioned	3.58
Non-conditioned	2.67
Non-generated	4.00

Table 4: Quality Score of Different Types of Music

Generation Setting	Arousal	Valence	Combined
Aligned	1.00	0.542	0.542
Differing	0.722	0.778	0.667
Overall	0.881	0.643	0.595

Table 5: Seperated VA Accuracy with Different Settings

because it is the most convenient approach to gathering data, especially when live interviews are difficult to conduct these days and conducting such a survey over teleconferences is not ideal.

7 HUMAN EVALUATION

In this section, we analyse the results we got from a user study. We engaged 6 participants, all of whom are in their twenties, and two of whom play a music instrument. We were unfortunately unable to recruit more participants due to the limited time and constraints we faced when trying to conduct interviews.

The primary objective of this survey is to answer the following questions: (1) How is the quality of the generated pieces in comparison with non-generated pieces? (2) Can the generated music express the emotion correctly even when the emotion label is different from the emotion of the music primers? (3) Is the generated music's expression ability different in terms of Valence and Arousal?

Indeed, without this study, we would not have been able to answer the first question beyond looking at the loss function; as for the second and the third question, we would have had to rely on our classifier which is understandably never a good replacement for human listeners.

Quality

Participants were asked to rate the quality of each piece on a scale of 1 to 5 (5 being natural, 1 being terrible). The result of this is shown in Table 4. Participants recognised the non-generated music pieces sound most natural among those pieces that we provided to them. The music pieces generated with condition has an average quality score of 3.58 which is close to the score of non-generated music pieces. However, the non-conditioned music pieces only have an average score of 2.67 which is significantly lower than the other two types of music pieces. This result shows that the music generated with conditioned model sounds more natural than the non-conditioned music.

There has also been an observation that some participants rated the high-arousal and negative-valence music pieces (tensional emotion) as low quality because of their high occurrence of repetitive notes. Although the occurrence of repetitive notes can serve to express the tensional emotion, participants often found it not pleasant to hear and thus rated it with low quality score.

Classification Accuracy

Table 5 shows a more detailed analysis for the human classification accuracy under different settings. The "Aligned" and "Differing" in the table has the same meaning as in Table 3, where "Aligned" means the target emotion is the same with the primer emotion, and "Differing" means the opposite.

If we conduct a row-wise comparison, it is interesting to find that the human classification accuracy for the "Differing" case is actually better than the "Aligned" case, which is the opposite in the GPT-2 classification. One possible reason could be that the GPT-2 classifier fails to capture the minor differences when the pieces are generated from the same primer but with different emotion labels, whereas people have better ability to notice the changes.

In another pointer of view, if we conduct a column-wise comparison, we could find that generally, participants have a better performance in correctly classifying the arousal value then the valence value. We obtain an accuracy of 0.881 for Arousal, but only an accuracy of 0.643 for Valence.

This difference in Valence and Arousal classification accuracy is possibly caused by the difference in how music pieces express them. From our observation, in the arousal dimension, the generated music pieces for high arousal tend to have faster tempo, whereas pieces for low arousal tend to be slower and contains pauses between some notes. Then in the valence dimension, the generated music pieces for high valence tend to be in major tonality, whereas pieces for low valence tend to be in minor tonality. Naturally, the difference in tempo is easier to be distinguished by non-professional listeners. However, it might not be straightforward for them to tell whether a piece is in minor or major tone. Therefore, accordingly, this might lead to the result that participants have a higher classification accuracy for Arousal than Valence.

8 FUTURE WORK AND CONCLUSION

In this paper, we looked at different LSTM-based models that are trained to generate pieces from four different emotion categories, based on the valence-arousal (VA) model, and compared their performance. We found that *MultiInput* performed the best in terms of learning the conditional probability distribution. Based on the classification results, the classifier is able to generate pieces that express the target emotion when the primer also follows that emotion. However, when the emotion label provided differs from the primer's, it appears that the proposed architectures are insufficient to enable the model to generate pieces with the user-specified emotion.This is despite the fact that we have segmented the generated piece to remove the primer before performing the classification. However, the human evaluation suggests that the proposed generator might be able to generate pieces with the user-specified emotion. Unfortunately, the sample size of participants is too small (due to the time and resource limitation under the covid situation) for us to draw a conclusion.

9 REFERENCES

REFERENCES

[1] [n.d.]. Google Has Set Up an AI Group Called 'Magenta' to See If Computers Can Produce Original Art and Music. https://www.businessinsider.com/google-

has-set-up-an-ai-group-called-magenta-original-art-music-2016-5?IR=T
[2] [n.d.]. Make Music and Art Using Machine Learning. https://magenta.tensorflow.org/
[3] [n.d.]. MuseNet. https://openai.com/blog/musenet/
[4] Rewon Child David Luan Dario Amodei Ilya Sutskever Alec Radford, Jeffrey Wu. 2018. Language Models are Unsupervised Multitask Learners. (2018).
[5] Julian Cespedes-Guevara and Tuomas Eerola. 2018. Music Communicates Affects, Not Basic Emotions – A Constructionist Account of Attribution of Emotional Meanings to Music. *Frontiers in Psychology* 9 (2018), 215. https://doi.org/10.3389/fpsyg.2018.00215
[6] Wei-Chun Chiang, Jeen-Shing Wang, and Yu-Liang Hsu. 2014. A Music Emotion Recognition Algorithm with Hierarchical SVM Based Classifiers. (2014), 1249–1252.
[7] Tan-Chyuan Chin. 2019. *Measuring adolescents' emotional responses to music: Approaches, challenges, and opportunities.* Oxford University Press, 39–52. https://doi.org/10.1093/oso/9780198808992.003.0004
[8] Ying Kin Yu Zhiliang Zeng Kin Hong Wong Chun Pui Tang, Ka Long Chui. 2018. Music Genre classification using a hierarchical Long Shot Term Memory (LSTM) model. (2018).
[9] Kerstin Denecke. 2017. A Mobile System for Music Anamnesis and Receptive Music Therapy in the Personal Home. *Studies in health technology and informatics* 245 (2017), 54–58.
[10] Lucas N Ferreira, Levi HS Lelis, and Jim Whitehead. 2020. Computer-Generated Music for Tabletop Role-Playing Games. (2020).
[11] Lucas N. Ferreira and Jim Whitehead. 2019. Learning to Generate Music with Sentiment. (2019).
[12] Dom Galeon. [n.d.]. The World's First Album Composed and Produced by an AI Has Been Unveiled. https://futurism.com/the-worlds-first-album-composed-and-produced-by-an-ai-has-been-unveiled
[13] Susan C. Gardstrom and James Hiller. 2010. Song Discussion as Music Psychotherapy. *Music Therapy Perspectives* 28, 2 (11 2010), 147–156. https://doi.org/10.1093/mtp/28.2.147 arXiv:https://academic.oup.com/mtp/article-pdf/28/2/147/6834409/28-2-147.pdf
[14] Ian Goodfellow. 2017. NIPS 2016 Tutorial: Generative Adversarial Networks. arXiv:1701.00160 [cs.LG]
[15] Annie Heiderscheit and Amy Madson. 2015. Use of the Iso Principle as a Central Method in Mood Management: A Music Psychotherapy Clinical Case Study. *Music Therapy Perspectives* 33, 1 (02 2015), 45–52. https://doi.org/10.1093/mtp/miu042 arXiv:https://academic.oup.com/mtp/article-pdf/33/1/45/5115924/miu042.pdf
[16] Maximos Kaliakatsos-Papakostas, Aggelos Gkiokas, and Vassilis Katsouros. 2018. Interactive Control of Explicit Musical Features in Generative LSTM-Based Systems. In *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion* (Wrexham, United Kingdom) *(AM'18)*. Association for Computing Machinery, New York, NY, USA, Article 29, 7 pages. https://doi.org/10.1145/3243274.3243296
[17] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. *CoRR* abs/1909.05858 (2019). arXiv:1909.05858 http://arxiv.org/abs/1909.05858
[18] Roberto Legaspi, Yuya Hashimoto, Koichi Moriyama, Satoshi Kurihara, and Masayuki Numao. 2007. Music Compositional Intelligence with an Affective Flavor. In *Proceedings of the 12th International Conference on Intelligent User Interfaces* (Honolulu, Hawaii, USA) *(IUI '07)*. Association for Computing Machinery, New York, NY, USA, 216–224. https://doi.org/10.1145/1216295.1216335
[19] Pedro Lucas, Efraín Astudillo, and Enrique Peláez. 2016. *Human–Machine Musical Composition in Real-Time Based on Emotions Through a Fuzzy Logic Approach.* Springer International Publishing, 143–159. https://doi.org/10.1007/978-3-319-44735-3_8
[20] B. Medcalf. 2017. Exploring the music therapist's use of mindfulness informed techniques in practice. *Australian Journal of Music Therapy* (2017), 47–66. https://www.austmta.org.au/journal/article/exploring-music-therapist%E2%80%99s-use-mindfulness-informed-techniques-practice-0
[21] Kristine Monteith, Tony R. Martinez, and Dan Ventura. 2010. Automatic Generation of Music for Inducing Emotive Response. In *ICCC*.
[22] Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. 2018. This time with feeling: learning expressive musical performance. *Neural Computing and Applications* 32, 4 (Nov 2018), 955–967. https://doi.org/10.1007/s00521-018-3758-9
[23] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to Generate Reviews and Discovering Sentiment. arXiv:1704.01444 [cs.LG]
[24] D. Riecken. 1998. Wolfgang: "emotions" plus goals enable learning. In *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, Vol. 2. 1119–1120 vol.2.
[25] Hao Hao Tan and Dorien Herremans. 2020. Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling. arXiv:2007.15474 [eess.AS]
[26] Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. 2016. Conditional Image Generation with PixelCNN Decoders. *CoRR* abs/1606.05328 (2016). arXiv:1606.05328 http://arxiv.org/abs/1606.05328
[27] Xinyu Yang, Yizhuo Dong, and Juan Li. 2018. Review of Data Features-based Music Emotion Recognition Methods. *Multimedia Syst.* 24, 4 (July 2018), 365–389. https://doi.org/10.1007/s00530-017-0559-4
[28] Y. Yang, Y. Lin, Y. Su, and H. H. Chen. 2008. A Regression Approach to Music Emotion Recognition. *IEEE Transactions on Audio, Speech, and Language Processing* 16, 2 (Feb 2008), 448–457. https://doi.org/10.1109/TASL.2007.911513
[29] Yi-Hsuan Yang and Homer H. Chen. 2012. Machine Recognition of Music Emotion: A Review. *ACM Trans. Intell. Syst. Technol.* 3, 3, Article 40 (May 2012), 30 pages. https://doi.org/10.1145/2168752.2168754
[30] Liang-Chih Yu, Lung-Hao Lee, Shuai Hao, Jin Wang, Yunchao He, Jun Hu, K. Robert Lai, and Xuejie Zhang. 2016. Building Chinese Affective Resources in Valence-Arousal Dimensions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 540–545. https://doi.org/10.18653/v1/N16-1066