FROM MUSIC TO SEMANTICS: AUTOMATICALLY GENERATING TIME-VARYING SEMANTIC TAGS FROM MUSIC

Tong Shan
University of Rochester
tshan@ur.rochester.edu

ABSTRACT

Music auto-tagging is a specific task in music information retrieval that has been developed for several years. However, very few studies have developed a model that could predict time-varying semantic tags. In this project, a convolutional neural network model (CNN) and a CNN combined with a Long Short-Term Memory (LSTM) model were proposed to predict emotional semantic tags. The CAL500exp dataset was used as the input and Mel-spectrograms were extracted as features. The mean average accuracy for tags showed that for some of the intense labels the models could predict accuracy over 0.8, but for most of the tags, the models cannot predict well. Improvement in preprocessing and network architecture could be done in the future work.

1. INTRODUCTION

Music tagging is a *music information retrieval (MIR)* task that gives music descriptive tags based on music content and its metadata. Music was once tagged manually by album listeners. Since digital music has been more and more popular, music applications like Spotify and Apple Music are now developing music recommendation systems to their users. However, the recommendation system now is mainly based on *collaborative filtering* [4], which is a method of making automatic predictions about the interests of a user by collecting preferences information from many users. The problem exists along with this method is that it's only applicable when usage data is available, which means it's difficult to recommend a new song or unpopular song [8]. Recently, machine learning and deep learning techniques have been used widely, some content-based music recommendation algorithms have been implemented that predict latent factor based on the music itself for recommendation [6, 8, 11].

Users are also likely to use music tags to explore new songs. Those tags are from metadata (e.g. artist, album, year, etc.) and semantic tags such as genre (e.g. jazz, classical...), instrument (e.g. piano, strings...), mood (e.g. sad, angry, arousing...), etc. Since *Deep Neural Networks (DNNs)* is now popular in both the research and industry fields for solving complicated problems, DNNs are also being used for *music auto-tagging* to predict music tags from latent features of music. Such auto-tagging algorithm models could facilitate text-based music retrieval [1].

Emotion	Instrument	Vocals
Angry/Aggressive	Acoustic Guitar	Breathy
Calming/Soothing	Drum Machine	Duet
Cheerful/Festive	Bass	High-pitched
...	...	...

Table 1. Examples of tags in CAL500exp dataset.

However, all of the methods could generate tags for a whole piece of music, which doesn't make sense since most music has time-varying semantic representations, especially for a symphony or a movie original sound track. The instrument, emotion, and even vocal artist could change over time. It has been shown that only track-level is tagging is not enough since different segment of music tent to have different tags [5]. Thus, time-varying auto-tagging is in need.

This project aimed to use *Convolutional Neural Network (CNN)* and *Long Short-Term Memory (LSTM) cell* to build a model to predict time-varying tags for music.

2. DATASET

The dataset I used is **CAL500exp** which is introduced by Wang et al [10]. The data is adapted from **CAL500** dataset [7], which is widely used in MIR field, especially in music auto-tagging. **CAL500** contains 500 songs all from unique artists. Each song is labeled with 174 expert-defined tags covering 8 semantic categories. But the tags are derived for track-level.

The **CAL500exp** expanded the **CAL500** dataset by including time-varying tags. Each song in the dataset was processed by Foote and Cooper's segmentation algorithm [3]. They then used k-medoids clustering to merge the similar segments in each track. Finally, each track was cut down to variable-length (3–16 second) segments, on average 6.4 segments per track.

Each segment was tagged with 67 binary semantic labels including emotion, genre, instrument, instrument solo, vocal style, song characteristic. Examples are shown in table 1.

All the labels are shown in ".csv" files for every track.

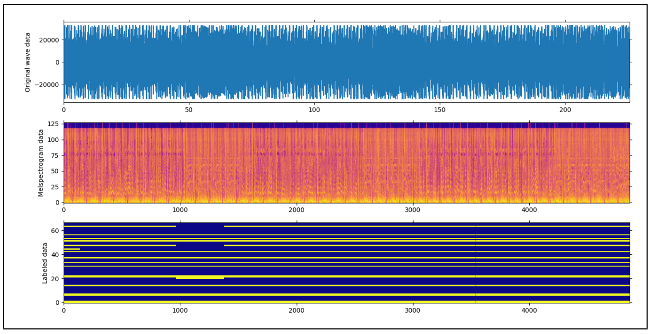

Figure 1. Example of a pre-processed track. The figure on the top is the waveform of the track. The figure in the middle is the Mel-spectrograms. The figure at the bottom is the 67 labels corresponding to each frame.

3. PREPROCESSING

3.1 Feature Extraction

STFT, Mel-spectrogram and MFCC are the most popular features in MIR that has both time and frequency representation. Mel-spectrograms provide an efficient and perceptually relevant representation compared to the other two [2]. In this project Mel-spectrograms were used as the feature that feed in the neural network model.

The sampling frequency for every track is 20500 Hz. Mel-spectrograms were computed with function in "librosa" library using 2048 window size and 1024 hop size and 128 bins of frequency.

3.2 Assigning Labels to Segmentations

The labels are only present for segments. For each track, labels were converted to mapping each frame so that each frame has a corresponding label. An example is shown in Figure 1.

3.3 Long-term Frame Segmentation

In order to predict the time-varying labels, the model needed more temporal data than only one single frame. Therefore, Long-term frame was then aggregated by a window of 128 frames and hop size of 64 frames. Thus, one long-term frame is about 4.7 seconds.

Then the labels were aggregated by computing the mean value within the window and was applied a step function that if mean value is greater than 0.5 assign it as a 1, otherwise assign a 0.

Overall, one track contained around 15 to around 200 long term frames depending on their original length, and each long term frame contains a $128 * 128$ Mel-spectrogram, and one vector of labels.

There were 67 labels that range over different semantic meaning. Considering that different semantic level could have different latent features, but one neural network model could only extract feature that fit for only one or two similar level. Therefore, only emotion labels (18 in total) were used in this project.

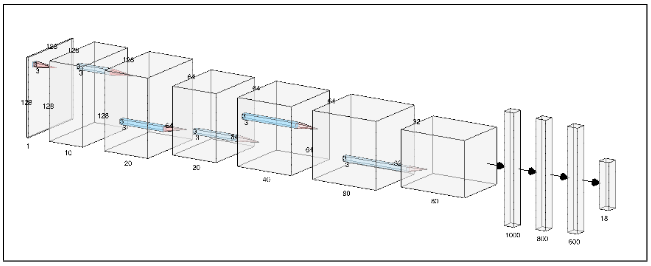

Figure 2. Convolutional Network Architecture.

3.4 Splitting Dataset

The 500 tracks in the **CAL500exp** dataset were split randomly: 400 tracks in train set, 50 in validation set, and 50 in test set.

4. NEURAL NETWORK MODELS

Models in this project were built using "Pytorch". *Dataloader* was used to load the train, valid and test set.

4.1 CNN Only Model

Input shape of the CNN model is $1 * 128 * 128$ (i.e. Depth=1, width=128, height=128) Mel-spectrogram for each long term frame. Then two convolutional layers with kernel size of 3 and 1 padding that converted the depth from 1 to 20 and then 40. Maxpooling layer was used with stride of 2 to shrink the shape of data to $40 * 64 * 64$ then same two convolutional layers were used after the Maxpooling layer that convert the data to $80 * 64 * 64$. Then a final Maxpooling layer convert the data to $80 * 32 * 32$.

After the convolutional layers, the data were fed in fully connected layer with 1000 neurons, 800 neurons and 600 neurons. Finally, the output layer had 18 neurons corresponding to 18 labels.

The architecture of CNN model is shown in Figure 2.

4.2 CNN Combined with LSTM Model

To account for the temporal related information across all the long-term frames in a track, an CNN conbined with LSTM model is constructed.

The architecture of this model is similar to the CNN only model. The only difference happens after the first fully connected layer. One LSTM cell was implemented after the fully connected layer with an output of 500. Then output from the layer before LSTM was concatenated with the output from LSTM layer and fed into the next fully connected layer.

The architecture is shown in Figure 3.

4.3 Hyperparameters

The models were both trained with similar hyperparameters. Batch size of 1 was used in training, since the the length (duration) of each track varied significantly from 10 to 150 long term frames. Thus, for training convenience, only 1 track in a batch was used to avoid information loss.

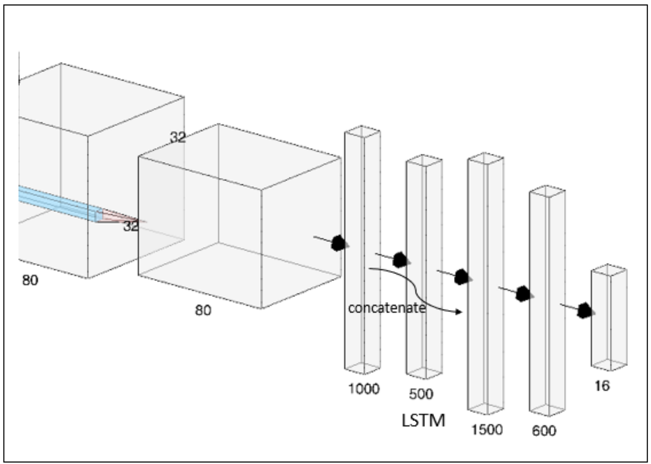

Figure 3. Convolutional combined with LSTM Network Architecture.

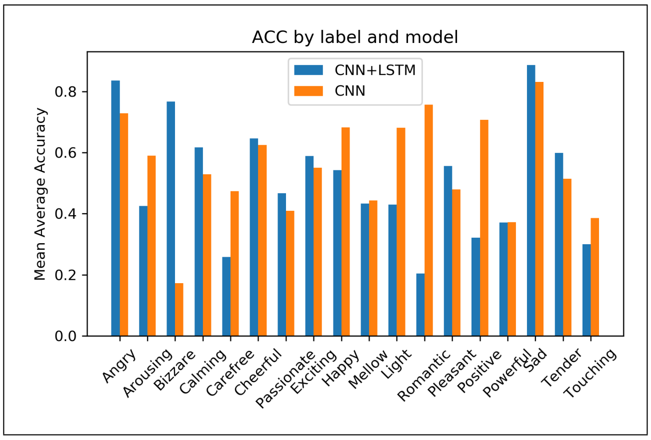

Figure 4. Mean average accuracy for CNN only and CNN combined with LSTM models.

After every convolutional layer, a ReLU activation was followed. Sigmoid activation is used for output layer.

The optimizer used was *Adam* with learning rate of $1 * 10^-5$.

Since this is a multi-label classification task, loss function *Multi Label Margin Loss* was used. The loss function was defined as

$$Loss(y', y) = \sum_{ij} \frac{(\max(0, 1 - y'[y[j]] - y'[i]))}{y'.size[0]}$$

where y' is the predicted value and y is the target.

For each track, loss was computed for each frame and summed. Then average loss was computed as the epoch loss.

For CNN combined with LSTM model, initial random state was used for the LSTM hidden state.

5. EVALUATION

Mean average accuracy (MAA) is computed for every label. MAA is defined as

$$MAA_l = \frac{1}{N} \sum_{N} \frac{TP_{nl} + TN_{nl}}{TP_{nl} + TN_{nl} + FP_{nl} + FN_{nl}}$$

where N is the total number of tracks, n the the index of track and l is the index of label. (TP: true positive, TN: true negative, FN: false negative, FP: false ositive).

The result of the two models are shown in Figure 4.

6. DISCUSSION

The accuracy is higher in intense emotions such as "Angry" and "Sad". However the accuracy of most of the labels are just around a coin toss or even worse.

There are some possible reasons:

- Feature used in this project is the pure Mel-spectrogram, however, studies have shown that log-scale Mel-spectrogram outperform pure Mel-spectrogram [2]. This make sense that the human ear can perceive log-scale magnitude rather than physical energy.

- Normalization of the data might be needed for each dimension of Mel-spectrogram features to keep the data normalized since normalized data is easy for a DNN to perform classification.

- In this project, only 4 convolutional layers were used. It might be not enough to extract the latent factors. Deeper network could be used to make a better performance. Another architecture could be integrating attentive DNN model with CNN and LSTM as proposed by Wang et al that take both Mel-spectrogram and waveform [9].

By fixing the preprocessing and the architecture of the model mentioned above, the result could be improved.

There are some interesting results such as for the tag "Bizarre" CNN combined with LSTM model outperformed much more than CNN only model. However. for tags like "Romantic" CNN only model outperformed much more than LSTM model. There is not much explanation so far, but it would be helpful to explore what feature does CNN layers extracted to represent emotions.

Furthermore, the time-varying auto-tagging technique is not only useful in the field of MIR and music recommendation, it also paves a way for understanding the semantic meaning of music that related to human brain processing of music and the relationship between music and speech. For example, if we have time-varying tag for emotion change in music, we could learn the changing brain pattern related to the emotion changing and thus learn how human brain react to music emotion.

7. CONCLUSION

In this project, a CNN model and a CNN combined with LSTM cell model were proposed to predict time-varying semantic tags from Mel-spectrogram feature. However, the accuracy of both model were not very good for most of the tags. More works needs to be done in the future to improve the model.

8. REFERENCES

[1] Thierry Bertin-Mahieux, Douglas Eck, and Michael Mandel. Automatic tagging of audio: The state-of-the-art. In *Machine audition: Principles, algorithms and systems*, pages 334–352. IGI Global, 2011.

[2] Keunwoo Choi. *Deep Neural Networks for Music Tagging*. PhD thesis, Queen Mary University of London, 2018.

[3] Jonathan T Foote and Matthew L Cooper. Media segmentation using self-similarity decomposition. In *Storage and Retrieval for Media Databases 2003*, volume 5021, pages 167–175. International Society for Optics and Photonics, 2003.

[4] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, 2008.

[5] Winter Mason and Siddharth Suri. Conducting behavioral research on amazon's mechanical turk. *Behavior research methods*, 44(1):1–23, 2012.

[6] J. Su, H. Yeh, P. S. Yu, and V. S. Tseng. Music recommendation using content and context information mining. *IEEE Intelligent Systems*, 25(1):16–26, 2010.

[7] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):467–476, February 2008.

[8] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems 26*, pages 2643–2651, 2013.

[9] Qianqian Wang, Feng Su, and Yuyang Wang. A hierarchical attentive deep neural network model for semantic music annotation integrating multiple music representations. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, ICMR '19, pages 150–158, New York, NY, USA, 2019. ACM.

[10] S. Wang, J. Wang, Y. Yang, and H. Wang. Towards time-varying music auto-tagging based on cal500 expansion. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2014.

[11] Xinxi Wang and Ye Wang. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 627–636, New York, NY, USA, 2014. ACM.