Music Generation based on Emotions

Sungpyo Cho
Department of Computer Science
Korea University
korra0501@gmail.com

Taelim Hwang
Department of Computer Science
Korea University
ghkdxofla@gmail.com

Abstract

Our framework is divided into two models - the text-based sentiment classification model and the music generation model. The sentiment classification model is implemented with Word2Vec embedding and Bi-LSTM architecture. The model classifies given text into two sentiment categories, which is positive and negative. After classification, the music is generated by model which uses RNN-based Google Magenta. Our approach is expected to generate music which reflects emotion and sentiment.

1 Introduction

Music can be associated with human emotions according to each genre. For example, gloomy music is related to negative emotions, and calm music is related to positive emotions. Human feelings are also contained in sentences. Articles containing emotional words, such as depressed or happy, indicate the author's emotional state.

In our paper we extract emotional state of the sentence through sentiment analysis, and generate the music suitable for the emotion. Music composition is a challenging task because of its genuine abstractness and complexity. Therefore, we aim to generate music that is appropriate according to the mood of the situation. For this purpose, our model utilizes word embedding, Bi-LSTM structure and Google Magenta model to build optimized models to create natural and appropriate music for human.

2 Differences from previous studies

In the previous study, melodies were created by learning the melody of music through deep learning or by combining sample bits. On the other hand, research on creating music with different moods based on emotions has not showed significant progress. There were studies which tried to extract emotional states by recognizing human expressions and produce related melodies. However, there were some limitations regarding to the definition of emotion, as well as experiment results. Extraction of emotions and the creation of appropriate melodies are still in a slow study. Our model aims to produce music only with the emotion of the sentence. It is expected to successfully convert literature such as novels and poems which are rich in human emotions to sounds and produce musics which are tailored to specific emotions.

3 Model architecture

Recently, in natural language processing(NLP) research, deep learning technology is applied for NLP and it shows good performance in practice. The development of word-embedding technology such as Word2Vec[1], which connects words to arbitrary dimensions, improves the performance of deep learning architecture for NLP.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

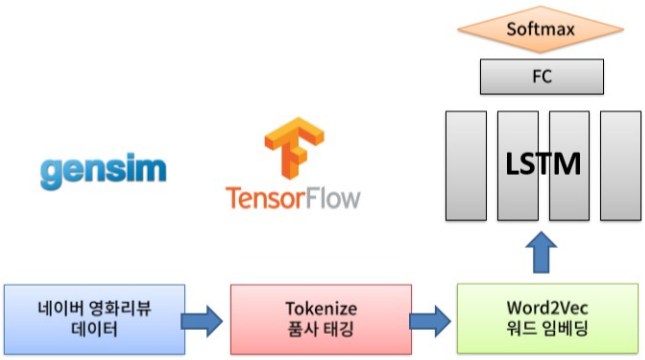

Figure 1: Architecture of our sentiment analysis model

Our sentiment analysis model is as shown above. After the text data is received, tagging, tokenizing and word embedding is performed by konlpy and Word2Vec model. The output of the word embedding then enters the input of the Bidirectional LSTM, and passes through fully connected layer. Finally through softmax layer, it outputs whether the sentence is positive or negative. In other words, this model takes the Korean text as an input and outputs positive or negative flag of emotion.

There are several ways to handle the embedding values, of which we used the Bi-LSTM architecture. Bi-LSTM is a structure that processes the input string bidirectionally and sequentially through LSTM to understand the nature of the input token appearing in the context. In particular, by adding forget gate in the basic structure of RNN, LSTM solves the problem of vanishing gradient that can occur when processing long document-level input data.

In this paper, we used Korean natural language processing library konlpy[2] in the process of document tokenization. Especially konlpy's Okt module is used because Okt module is more efficient and faster than other modules such as Kkma. Then, the embedding was performed using Word2Vec of gensim library.

For the music generation, we used Google Magenta model. Among several music generation model, Attention RNN is used. This model is composed of LSTM. In addition to the basic LSTM, this model consists of accessing past information without storing previous information in the RNN cell's state. The model is an encoder-decoder RNN, and the model uses attention to look at all the encoder outputs during each decoder step. This process can be expressed as follows:

$$u_i^t = v^T \tanh(W_1' h_i + W_2' c_t) \tag{1}$$

$$a_i^t = \text{softmax}(u_i^t) \tag{2}$$

$$h_t' = \sum_{i=t-n}^{t-1} a_i^t h_i \tag{3}$$

A softmax is used to normalize these values and create a mask-like vector a_i^t, called the attention mask. The RNN outputs from the previous n steps are then multiplied by these attention mask values and then summed together to get h_t'. For example, let's assume we are on the 4th step of our sequence. the value of n is 3, which means our attention mechanism is only looking at the last 3 steps. For this example, the RNN output vectors will be small 4 length vectors. If the RNN outputs from the first 3 steps are:

$$Step1 : [1.0, 0.0, 0.0, 1.0] \tag{4}$$

$$Step2 : [0.0, 1.0, 0.0, 1.0] \tag{5}$$

$$Step3 : [0.0, 0.0, 0.5, 0.0] \tag{6}$$

calculated attention mask is:

$$a_i^t = [0.7, 0.1, 0.2] \tag{7}$$

Then the previous step would get 20% attention, 2 steps ago would get 10% attention, and 3 steps ago would get 70% attention. So their masked values would be:

$$Step1(70\%) : [0.7, 0.0, 0.0, 0.7] \tag{8}$$

$$Step2(10\%) : [0.0, 0.1, 0.0, 0.1] \tag{9}$$

$$Step3(20\%) : [0.0, 0.0, 0.1, 0.0] \tag{10}$$

And then they'd be summed together to get h'_t:

$$h'_t = [0.7, 0.1, 0.1, 0.8] \tag{11}$$

The h'_t vector is essentially all n previous outputs combined together, but each output contributing a different amount relative to how much attention that step received.

This h'_t vector is then concatenated with the RNN output from the current step and a linear layer is applied to that concatenated vector to create the new output for the current step. Some attention models only apply this h'_t vector to the RNN output, but in our model, this h'_t vector is also applied to the input of the next step. The h'_t vector is concatenated with the next step's input vector. The linear layer is applied to that concatenated vector to create the new input to the RNN cell. This helps attention not only affect the data coming out of the RNN cell, but also the data being fed into the RNN cell.

This h'_t vector, which is a combination of the outputs from the previous n steps, is how attention can directly inject information from those previous steps into the current step's network of calculations, making it easier for the model to learn longer-term dependencies without having to store all that information from those previous steps in the RNN cell's state.

4 Experiment and result

For sentiment analysis, Naver movie review dataset is used. It is composed of 200,000 Korean sentences classified as positive and negative according to movie ratings. For example, a review like "A movie that is not really boring" is classified as positive, and a review like "It is not boring but it is complete trash..." is classified as negative. We split the whole data into 150,000 traning set and 50,000 test set. Of course, considering the size of a huge dataset, it would be better to increase the ratio of traning data. However, 150,000 traning data showed good enough performance with this split.

```
[('공포영화/Noun', 0.6746385097503662),
 ('스릴러/Noun', 0.6706466674804688),
 ('보다/Verb', 0.6374683976173401),
 ('./Punctuation', 0.6208621263504028),
 ('하다/Verb', 0.6119281649589539),
 ('액션/Noun', 0.6097371578216553),
 ('장르/Noun', 0.6062196493148804),
 ('.../Punctuation', 0.6050468683242798),
 ('이/Josa', 0.6049748063087463),
 ('이/Noun', 0.6039228439331055),
 (',/Punctuation', 0.5988036394119263),
 ('의/Josa', 0.5955643057823181),
 ('에/Josa', 0.5953145623207092),
 ('만들다/Verb', 0.5952882170677185),
 ('에서/Josa', 0.5932609438896179),
 ('가/Josa', 0.5932246446609497),
 ('들/Suffix', 0.5929362773895264),
 ('코미디/Noun', 0.5918705463409424),
 ('?/Punctuation', 0.5915305614471436),
 ('것/Noun', 0.5914014577865601)]
```

Figure 2: top 20 similar tokens related to 'horror'

To classify the emotional analysis of the text, we first tokenized 150,000 pieces of movie review data using konlpy. After that, the tagged words are converted into embedding vectors learned by Word2Vec algorithm. As a result, in regards to the word 'horror', words such as 'horror movie', 'thriller' showed the most highest similarity. After completing successful embedding process, Word2Vec model was saved for later use.

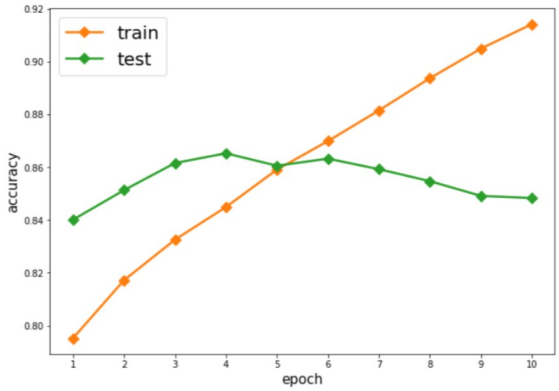

Figure 3: train and test accuracy graph

We then trained classifiers with the Bi-LSTM model. As a result, the accuracy of the train data continuously increased, while the accuracy of the test data showed decrease. As the number of learning times increased, the accuracy tended to increase in the range of 78 93 percent. The accuracy of test data was highest at epoch 4, so early stopping was performed in epoch 4 and the model was saved. Korean text sentiment analysis was performed with this saved model. The model showed high accuracy of 87.6 percent.

```
문장을 입력하세요:친구랑 보고왔는데 꽤 만족했습니다
긍정입니다
문장을 입력하세요:여친이 돈아깝다고 화냈네요;;
부정입니다
```

Figure 4: Sentiment analysis of our model

Using the results of the emotional analysis (negative, positive), the model is trained through distinct datasets to generate different models. For positive emotions, classical musics of major scale and fast tempo are used. For negative emotions, minor scale and slow tempo music were used. The dataset was categoried by listening. Before learning with each dataset, we created a model using a major scale midi file and a monotonic scale midi file for simple testing. Each learned result produced new major-oriented musics and minor scale musics.

Model learning and music generation are performed in command prompt. Traning and generation is expected to done in Python language, but magenta currently only explains how to train with the commands of command prompt. Furthermore, the model is based on Tensorflow, so it takes less time to learn using GPU.

5 Conclusion

We combined the sentiment analysis model and music generation model to create music appropriate to the sentiment of the input text. First, the Bi-LSTM model used for emotional analysis showed the model is suitable for sentiment analysis and shows high performance. Especially, our model showed 87% accuracy of sentiment classification for Naver Korean movie review data. The implemented model has better performance than other models, but it has a disadvantage of relatively long training time. Furthermore, out music generation model takes positive and negative emotion as an input, and generated quite appropriate melodies. Our model is expected to provide emotional stability through customized music based on user's emotional state.

6 Future works

Currently, our model produces music by classifying the sentiment of input text into two categories: positive and negative. The problem is that it fails to categorize various sensibilities such as surprise,

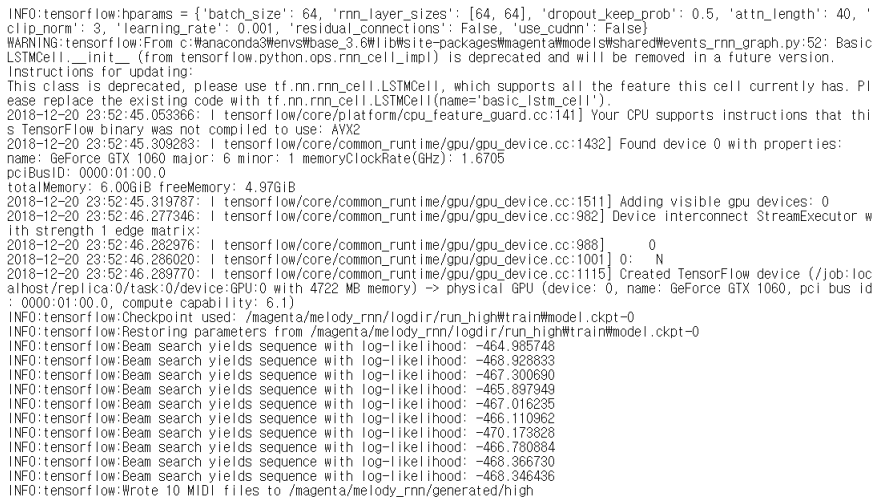

Figure 5: Result of melody RNN generation

anger, and sadness in detail. This is due to the inherent limitations of the emotional analysis dataset. Therefore, it is deemed necessary to have data or algorithms that can analyze and classify more diverse sensitivities.

In addition, the problem with music generation models is that they only accept positive and negative flags of music generation models as input. Therefore, it seems necessary to create a more advanced model that reflects the intensity of emotion, not just positive and negative.

References

[1] Le, Q., Mikolov, T. (2014, January). Distributed representations of sentences and documents. In International Conference on Machine Learning (pp. 1188-1196).

[2] Park, E. L., Cho, S. (2014, October). KoNLPy: Korean natural language processing in Python. In Proceedings of the 26th Annual Conference on Human Cognitive Language Technology (pp. 133-136).