

Brief Recap:

As a brief recap of our project so far, we had introduced a model that uses a pretrained model of KoELECTRA. We performed k-fold cross validation and our base model seemed to have an accuracy of 57%. We realized the discrepancy of the cross validation accuracy and training accuracy to be due to a lack of a test set. We plan on implementing the test set and finishing up our overall model by the following week.

Progress Explored:

This week we realized the importance of developing our auxiliary components since model completion and fine-tuning can be conducted much more smoothly with a working application. We decided to utilize the Flask framework and performed initial setup. We reached near completion of the system, but ran into a minor issue. Next we acquired song data and procured our own Music Dataset using a web crawler. Finally, we completed the initial setup of our music recommendation system.

Flask

For ease of use, we decided to make a web based application. For this task we utilized Flask, a Python web application framework. We utilized EC2 and RDS architecture in our system and completed the main codebase of the system. As stated before, we completed the initial setup of our application, but ran into issues of implementing our model due to bugs in saving a checkpoint model.

Flask Codebase

The codebase logic is as follows:

1. Obtain the user's name, phone number, and current mood.
2. Based on the mood text, send a query to the RDS (Relational Database Service) to retrieve a list of songs with the same label as the analyzed main emotion.
3. Using cosine similarity, compare the softmax probabilities of the song list and the user's sentiment analysis text to recommend the song with the highest score.
4. Record whether the user is satisfied with the recommended song (to be used as a future performance metric).

Music Dataset:

To obtain music data for the eventual development of our music recommendation system, we utilized a web crawler on Melon to gather Melon Playlist song information. The crawler gathered the title, lyrics, and label of each song. The label of each song corresponds to the appropriate Melon Playlist. For example, if the playlist is a collection of happy songs, all of the songs in that playlist would be labeled as happiness. We also decided to not use this dataset as an augmentation of our training dataset and rather for the sole purpose of being a reference for our recommendation system. More on this will be explained in the following slides.

Music Recommendation

We performed some simple preprocessing measures on the initial dataset. We excluded songs that contained English words that we did not present in the dataset our model was trained on.

We also treated 4 lines of lyrics, with each line separated by a forward slash, as one sentence. The model then calculates the probability value on a per sentence basis. This means that each sentence is evaluated and the probability value for each of the seven emotions are the result. We then average the sum of the probability values of all sentences and use that value as the emotion probability values of one song.

Music Recommendation cont.

The results are saved into a new csv file for the music recommendation system to use. This image shows a part of how the preprocessed dataset is formatted.

Music Recommendation cont.

Our initial music recommendation system compares the probability values of a given text and the songs, referring to the preprocessed music dataset. The following image shows our codebase for the initial version of our recommendation system.

Music Recommendation Results

We ran the system with the following example sentences, 오늘 날씨 너무 좋네, 너무 무섭다, 상사 때문에 화나, and 슬픈 일을 겪었어. The results show the cosine similarity of the inputted text and of the most similar song, the index number of the song, and then the song name. To reiterate, the current implementation of our system compares, using cosine similarity, the probability values of each seven emotions of the given text and of all the songs in the preprocessed music dataset.

Model Results?

We have not made any significant changes to the model since last week's progress report due to placing importance on completing the initial setup of the other components of our project.

Future Endeavors

The most important aspect is completing our model since development of the other components has now more-or-less been completed. Another important task is to implement the model into our Flask-based web application. We believe this will be a simple task as all we need to do is run a saved, checkpoint model. A third task we believe is crucial for our project is the manual verification of the results of our recommendations. We recognize that there are significant nuances in each individual emotion. Our current recommendation system tries to address this issue by using the cosine similarity of the seven probability values for each the inputted text and the recommended song. We also recognize that the current iteration of our music dataset might be a bit too small at only 640 songs. We plan on taking the appropriate steps to address this potential issue.