Playlist Generation using Emotion Recognition and Semantic Textual Similarity

Team member 1
Daniel King

Team member 2
Thomas Hayter

1 Background and Motivation

The music industry is a huge industry, with a market cap of around $25.9 billion(McCain, 2023), making any music-related applications potentially hugely profitable. For companies such as Spotify, they need to leverage technology in any way they can to try and keep users - with one of the main ways of doing this being the generation of playlists for subscribers to listen to.

As such, we propose a novel system for playlist generation, based on selecting songs with similar semantic and emotional meanings to an input sentence or song. We believe this would be useful not only to music companies such as Spotify, who need the functionality to keep users from their competitors, but also to small artists who rely heavily on such playlist-generating systems to have their music discovered and listened to by users across the world. Furthermore, for people who listen to music, it gives them a way to discover and listen to new music based on what kind of mood they are in, what kind of music they want to listen to and songs that they already know and like. We believe this is a challenging task as it combines multiple NLU tasks into one complete system, and we will have to make recommendations on a very small sample of user input data.

2 Problem Statement

The NLU tasks invloved in this system are Emotion Recognition and Semantic Textual Similarity. The input to the system will be either a song or a string of text that will be used to generate a playlist of songs (the output).

We will first train an Emotion Recognition model using datasets of tweets(Gupta, 2021)(Pandey, 2022). This model will be used to classify song lyrics into different emotion classes. We will input a dataset of song lyrics(Shah, 2021) into this model and produce a dataset (keyed by emotions) that stores all songs exhibiting that emotion.

Then, the user will input a song or a string of text, which will be passed into the Emotion Recognition model to identify the emotions that the user wants a playlist to be generated for. A list of all the songs that have at least one matching emotion will be selected from the song lyric emotion dataset. After this, we will compare the Semantic Textual Similarity of the input string/song lyrics with each of the songs chosen previously, and rank them based on similarity. We will experiment using this(sen) sentence transformer model to produce the similarity values. A playlist of k-length will then be generated based on the most similar songs.

3 Related Work

As we are exploring a completely new task, we will research into the two separate parts of the task separately - Semantic Textual Similarity (STS) and Emotion Recognition (ER).

Regarding STS, Sanborn et al. (Sanborn, 2015) experimented with both Recurrent Neural Networks and Recursive Neural Networks to measure the semantic similarity of short, individual sentences. They constructed word vectors using GloVe (Pennington et al., 2014) and found that the Recurrent Neural Network model was the better of the two, and substantially outperformed any baseline models. There has also been much success in STS through the use of pre-trained language models such as BERT (Devlin et al., 2019), but these models have hundreds of millions of parameters which can lead to issues with fine-tuning them for specific tasks. As such, Wang et al. (Wang et al., 2020) propose a method to compress large Transformer based models, called self-attention distillation. With regards to STS, this allows for smaller transformer networks to be tuned more easily to specific tasks - in the case of our project, the task of STS between song lyrics.

Pre-trained models such as BERT have also seen

much success in text ER, and Edmonds et al. (Edmonds and Sedoc, 2021) experimented with BERT models for multi-emotion classification over song lyrics. It is interesting to note that they found that BERT models trained on much smaller song datasets managed to achieve marginally better performance than those trained on typical large social media or dialog datasets - showing that such models fail to fully generalise across all types of text data. Due to the novelty of our project, combined with the lack of substantial, publically available song lyric emotion datasets, we will be training our own models on large tweet emotion datasets for ER, and fine tuning existing compressed transformer networks to obtain the best performance possible for STS.

4 Datasets and Evaluation Resources

The datasets we plan to use are datasets of tweets(Gupta, 2021)(Pandey, 2022), annotated with the emotions they contain, to train the Emotion Recognition model. These will be all English tweets, as we are only considering English songs. It will also be a large dataset as we want the emotion recognition to be as accurate as possible. The labels for this dataset will be the range of emotions we can capture, such as 'joy', 'sandess', 'happiness' etc. This dataset will be split into training and testing data, so that we can evaluate the emotion recognition model.

We are also using a dataset of song lyrics(Shah, 2021) which will be classified by emotion to create a new song lyric emotion dataset. The songs dataset should not be too large, as we have to compare each song semantically whenever the system is run. However, we still want it to be large enough that there is a comprehensive range of emotions throughout the songs, and the songs found are still very similar.

As we are tuning a pre-trained STS model, we will use the song lyric emotion dataset to evaluate the performance of the model where we assume that songs with similar emotion classes will have more similar semantic meaning. By passing pairs of songs where we expect one song to be more similar to the input than another, we will evaluate the performance of the model by calculating the percentage of the time the system gives a higher similarity score to the expected more similar song. While this is not a perfect way to evaluate the model, due to the lack of available datasets we would otherwise have to evaluate manually.

With regards to the playlist generation itself, the only way to evaluate this is by user feedback and manual testing.

5 Proposed Activities

Activity	Any comments	Duration	Lead
Pre-process data from datasets		1 week	D.K.
Build emotion recognition model		3 days	T.H.
Train, tune and evaluate emotion recognition model		1 week	T.H.
Predict emotions in songs	Create dataset of results	2 days	D.K
Finetune semantic textual similarity model		1 week	D.K
Build simple UI and combine models to generate playlists		1 week	T.H.

References

Sentence-transformers/all-minilm-l6-v2 · hugging face.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Darren Edmonds and João Sedoc. 2021. Multi-emotion classification for song lyrics. In *Proceedings of the*

Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 221–235, Online. Association for Computational Linguistics.

Pashupati Gupta. 2021. Emotion detection from text.

Abby McCain. 2023. 30 harmonious music industry statistics [2023]: Facts, trends, and sales.

Parul Pandey. 2022. Emotion dataset for emotion recognition tasks.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Adrian Sanborn. 2015. Deep learning for semantic similarity.

Deep Shah. 2021. Song lyrics dataset.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.