Music Recommendation via Social Media Content Analysis

Abstract

Depression is classified as a leading cause of disability, affecting over 264 million people worldwide. Research has shown that symptoms are reflected by affected individuals in their social media posts. Our aim is to use Twitter data to analyze the sentiment of posts as indicating sadness or happiness using deep learning techniques. We take this a step further by utilizing unsupervised deep learning measures to recommend a song that fits the mood of the sentiment. We use four different deep learning methods for sentiment analysis and the DEC implementation for clustering. The results are discussed.

1. Introduction/Background/Motivation

Depression is classified as a leading cause of disability. The 264 million people of the global population suffering from its symptoms and 800,000 annual suicide cases resulting from it is visible proof of the need for aiding measures [1]. In this regard, we aim to utilize music therapy with the implementation of deep learning technology to contribute to the healing of depressed individuals. Our main focus of this utilization lies within the relation of depression diagnosis and massive social networking tendency in this information era.

Recent research suggests that early depression often depicts behaviors of expressing depressing messages on SNS postings. While detecting mental disorders is a complex task, it is nevertheless important to catch early signs of mental illnesses. Because of the growing popularity of social media, users' posts reflect their personal lives on many levels. It is therefore fruitful to analyze social media posts to catch early signs of mental illnesses.

There have been many approaches to analyzing social media posts. Islam et al. [2] used supervised, 'shallow' learning methods on Facebook posts. The data was tokenized using a psycholinguistic features vocabulary package. The results indicate good performance for these 'shallow' models to classify a post as indicating depression, achieving accuracy scores between 60% and 80%.

Paul et al. [3] analyzed Reddit posts for signs of anorexia and depression using shallow and deep learning methods. The data was represented using a bag of words method and fed into four 'shallow' learning methods. The fifth model they used was an RNN classifier using Fasttext embeddings, which are pretrained word vectors on 600 billion tokens generated from a corpus on Wikipedia. The results indicate that the 'shallow' learning methods perform better than using RNN with Fasttext.

Orabi et al. [4] performed an analysis of Twitter users' posts using deep learning methods. In their study, they attempted to label each user as depressed, suffering from PTSD, or neither. The data was preprocessed using NLTK tokenization and Skip-gram Word2Vec embedding. They compared three flavors of 1D CNN, and an RNN. After hyperparameter tuning, their results show that a variation of CNN with max pooling layer performed the best in identifying mental disorders.

Kim et al. [6] used data collected from mental health related subreddits from Reddit. The posts from each subreddit were labeled as positive for that mental illness and negative for the others. The data was cleaned and preprocessed using NLTK tokenization and pretrained Word2Vec embedding was used. They tested CNN and XGBoost models on whether they could correctly identify which mental illness the post indicated toward. The results show a much better performance across the board for CNN using an embedding layer and a max pooling layer.

We build on these previous studies by also utilizing an unsupervised deep learning method to suggest a fitting choice of music for each post based on its sentiment. Music therapy is a well-established method in treating various mental discomforts. A meta-analysis of the effects of music treatment for depression patients showed a significant improvement in subjects' condition after music therapy [7]. As such, we believed it would be beneficial for the user to be recommended a certain song that fits the mood of the post they made to ease mental discomfort. The main benefit of this experiment could be expanded such that there has not been much work done on combining sentiment analysis of social media tweets with music recommendation. As such, our approach to combining these two involved looking for a way to classify music into the appropriate moods. For this end, we used deep embedded clustering (DEC) [8]. This method uses deep neural networks to simultaneously learn feature representations and cluster assignments.

For our experiments, we used the sentiment140 dataset from Kaggle [9], which consists of 1.6 million tweets pulled using the Twitter API. The purpose of this dataset is to train and build better models for sentiment analysis of large volumes of text. We used the Spotify dataset from Kaggle [10], which was pulled using the Spotify API. This dataset consists of 600,000 tracks with various information including valence, tempo, loudness, danceability, and energy.

2. Approach

2.1. Dataset

Each tweet in the sentiment140 dataset is annotated as either positive or negative. Because the tweets are raw, we cleaned the dataset first by removing nonalphanumeric characters (@, “, ‘, - …). We then used the NLTK library to tokenize and lemmatize each tweet. Lemmatization is essentially grouping variant forms of the same word. We tried removing stopwords, or very common words in a language we, but we found we got worse results than simply keeping them. We also tried removing a limited selection of stopwords that would preserve the sentiment of the tweets, but it was better to use the entire tweet. We used 30% of the entire dataset, split evenly between negative and positive tweets.

1st row	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,9,273,75,3,576,37,162,9,2,19,3,104]
2nd row	[0,0,0,0,0,0,0,739,12286,115,2,9,3113,20,9,167,3,34,102,8,30,9,99,2,1077,8,28,313,191,3278]
3rd row	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2422,359,72775]
4th row	[0,0,0,0,0,0,0,0,0,0,187,20,5,11134,24,1,23,360,8,1,23,16,634,86,1,212,464,61,1603,3852]
5th row	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,50]

Figure 1. Sample Data Sequence

We used 20% of each class as validation data. To prepare the dataset for training, we converted each tweet to a numerical sequence using Keras’ built-in method. We then padded each sequence to a uniform length of 30. In the end, our training and testing datasets had shapes (384000, 30) and (96000, 30) respectively. The total vocabulary size was 125,276.

For embedding, we used global vectors for word representations (GloVe) [11]. We used the 6 billion token version, which is trained on Wikipedia corpus and Gigaword 5. GloVe is a powerful tool in that it takes into account cosine similarity between word vectors, which provides semantic similarity of the corresponding words, and it tries to capture a deeper relationship between words than a singular number. For our purposes, we used GloVe pretrained word vectors for our embedding layer.

For the music recommendation, we used Spotify song dataset pulled using the Spotify API [10]. The dataset consists of 600,000 tracks with various numerical identifiers of each, which include valence, tempo, loudness, danceability, and energy. Since the tracks dataset do not have any label attached to it, we tried to cluster the dataset based on these numerical identifiers to see if there are logical groupings that we can use to tie them back to sentiments. To prepare the dataset for clustering, we first checked for any null values to make sure that only non-null values are included in the clustering process. We also scaled all columns features so all values would be between 0 and 1 to avoid any unintentional bias towards features with underlying large values.

2.2. Models

We compared the performance of various different deep learning models on sentiment analysis of the tweet dataset: CNN with max pooling, LSTM-CNN mixed model, transformer with 8 attention heads, vanillaRNN, LSTM, LSTM tuned and a plain MLP model.

For the first three models mentioned, we used the Adam optimizer, which uses momentum and scales the learning rate by using the squared gradient. Other optimizers including RMSProp and SGD gave us worse performance where Adam is able to navigate difficult gradient surfaces more effectively. We used a batch size of 512 and 20 epochs. While smaller batch sizes generally result in better representational power, we saw no significant difference. 20 epochs were chosen since it struck a good balance between overfitting and not giving the models enough time to learn.

We also used early stopping measures which stops training when the validation loss stagnates. In addition, we reduced the learning rate whenever the validation loss plateaus, which indicates that we are likely in a local minimum.

For the loss functions, all models used the binary cross-entropy as a loss function. It is because there are two binary classes presented on our data (sentiment whether it is a sad post or happy post). Moreover, there are some advantages of using this loss function. First, Binary cross-entropy function is used because it is equal to fitting the model using maximum likelihood estimation. Second, we don’t really need to think about the cost function that it might have.

Our CNN model utilized two blocks of 1D convolutional layers with a 1D max pooling layer and a global max pooling layer after the second block. Global max pooling is essentially the same as max pooling layer except applied to the entire feature map. We included two dense linear layers and a dropout layer with 0.2 probability. While typically used with 2D image data, 1D CNNs have the advantage of being able to learn from the entire sequential data directly.

The mixed LSTM-CNN model utilized a convolutional block with a 2 1D layers and a bidirectional LSTM with 16 units. Bidirectionality trains two LSTMs on the same input forwards and backwards, providing more context to the network and improving performance. The CNN block acts as feature extractors while the LSTM part supports sequence prediction.

Our transformer model is based on the common implementation with an encoder-decoder pair each with multi-head attention layers based on Vaswani et al. [13]. Attention is particularly powerful in that it helps the model focus on the most important parts of the input sentences. We used 8 attention heads and a feedforward dimension size of 512. We also utilized dropout layers with high chance (0.7) to decrease overfitting.

We took this experiment further and implemented recurrent neural networks. Recurrent neural network is chosen since it has it’s advantages on processing different sequences of data. Our vanillaRNN model uses SimpleRNN layer with units 4 and a dense layer. SimpleRNN is basically a fully connected RNN where the output is fed back to the input. Simple architecture of RNN is applied to see the network reacts with the data.

RNN approach has few known problems such as gradient vanishing problem, training difficulty, and difficulties in processing a long sequence. To improve on our vanilaRNN, LSTM approach is implemented. LSTM architecture uses hidden state and cell state to improve the performances of RNN. Like vanilaRNN, a single LSTM layer with a dense layer is inserted to see the performance improvement.

Furthermore, recurrent neural network is improved by inserting different hyperparameter tunings. This was because our vanillaRNN and LSTM model did not have the acceptable accuracy rates. And the results of those models were understandable since we just applied the basic default rates.

Tunings were done by adding more layers, dropout layers, learning rates, activations, optimizers, and momentums. For consistency, same number of epochs and batch size is used.

For our MLP model, we are still using the GloVe matrix as our embedding matrix, with 0.4 dropout rate. The data is then feed into a 2 fully connected hidden layer and an output layer using a sigmoid function to force the output to be between 0 and 1. We also used Adam optimizer as it tends to work more efficiently when encountering potential issues such as saddle points. After multiple fine-tuning, we found that 512 nodes on each hidden layer with ReLU activation in between, a batch size of 1024, and epochs of 20 seems to work best with this MLP model to give the model enough time to learn without overfitting.

To cluster the music dataset, we used an unsupervised Deep Embedding Clustering (DEC) [8]. The DEC model consists of an autoencoder and clustering layer. The structure and flow of the model can be seen in Figure 2 below. Instead of clustering directly in the original data space, we first encoded the data with non-linear mapping and learnable parameters to allow the model to embed the data in the latent feature space. While the dimension of our encoded data might not be much smaller than the original datasets, we were hoping to get a better representation of the data sets through this method.

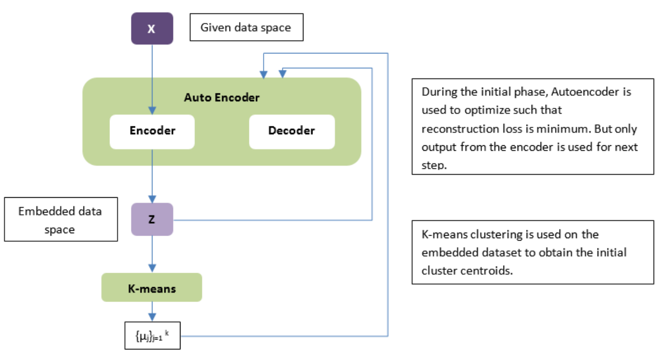

Figure 2. DEC general flow [12]

To determine the initial number of clusters, we used the elbow method to plot the cluster inertia. Based on the inertia graphs, we tried both 3 and 4 cluster sets to see which one would work better for our purpose. We first initialized the cluster centers using k-means methods. Then, we set the k-means cluster centers as the initial weights for the clustering layer of DEC. The model is then trained iteratively to refine the clusters with the help of an auxiliary target distribution using the KL divergence as the loss function. The auxiliary target distribution that we used here is the squared value of the soft assignments divided by soft cluster frequencies [8]. We set a maximum iteration of 1,000 and the training is stopped if the change in the label between the current and the previous iterations is less than 0.1% of the datasets.

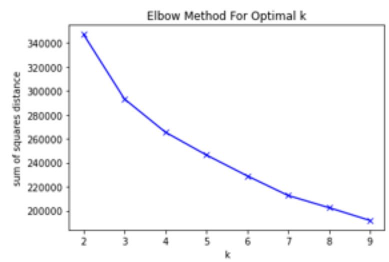

Figure 3. Elbow Method to Determine Initial Number of Clusters

We also utilized sentence embeddings models to embed and vectorize our song titles and tweets. We used a pre-trained Sentence Bert model to embed our sentences that is based on the Siamese BERT-Networks (SBERT) [14]. Using a cross-encoder, BERT has set a state-of-the-art performance for sentence pairs and classification. SBERT adds a pooling operation to the output of BERT and uses siamese and triplet networks to fine tune the model further to produce sentence embeddings that are semantically meaningful and can be compared. We would use these embeddings as part of our song recommendation by calculating the cosine similarities between the song titles and the input tweet.

3. Experiment and Results

3.1 Model Results

Model	Valid. Acc.	Valid. Loss
CNN	0.8054	0.4213
LSTM-CNN	0.8002	0.4335
Transformer	0.7933	0.4556
RNN	0.5186	0.6920
LSTM-RNN	0.5331	0.6889
LSTM Tuned	0.7923	0.4495
MLP	0.5453	0.6816

Table 1. Comparison of Sentiment Analysis Accuracy Between Models

3.2 Results Explanation

Our CNN model achieved the best validation accuracy and loss across all the different models we implemented as seen in Table 1. The learning curves for the CNN model can be seen in Figure 4 below. In general, the validation loss is decreasing give or take some stochasticity as indicated by the bumps, which might indicate that at these epochs the model might have been trending toward overfitting. However, with the learning rate reduction schedule, we can see a general downward trend afterward, which shows that the model is learning effectively.

The reason why our CNN model shows the best performance might be because we first pad the training sequences to fixed length. In this case, CNN is able to pick up on neighboring information more effectively in some cases than using recurrent approaches.

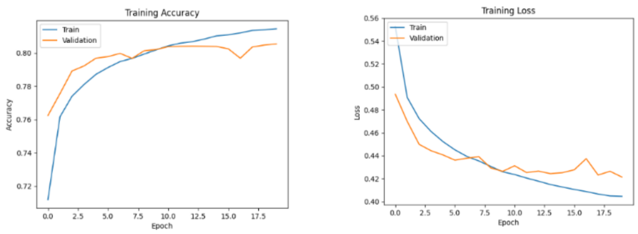

Figure 4. CNN Accuracy and Loss

Our combined LSTM-CNN model showed a similar performance to CNN, just to a lower accuracy. This makes sense since we are using CNN to extract the features and as such, these two models are likely operating on similar features. As for the sequential part, using LSTM here yielded a worse score than just sticking with CNN layers. As is, the model skirts the line of overfitting. Increasing the complexity of the model any further would have pushed it into overfitting territory.

We tried more complex models, increasing the number of 1D convolutional layers, tried traditional RNN cells instead of LSTM. Despite our best efforts, this was the best performance we could achieve without overfitting on the training data.

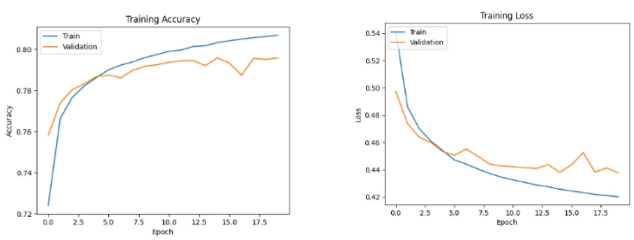

Figure 5. LSTM-CNN Accuracy and Loss

The training accuracy and loss curves for the transformer model can be seen in the following figure. We initially had difficulties with the model overfitting to the training data, but with high dropout chance, we were able to achieve a good performance. One of the reasons for the underwhelming performance of the transformer model might have been that we used a subsample of the dataset where the transformer might have benefitted from using a larger sample.

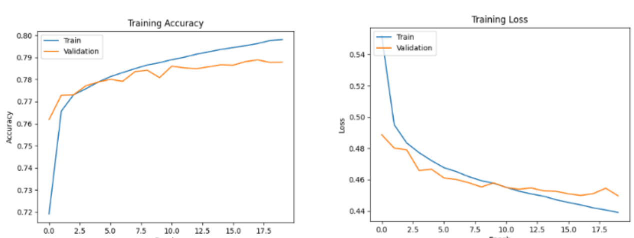

Figure 6. Transformer Accuracy and Loss

For vanillaRNN model that we implemented, as expected, there was a problem of Long-term dependencies as length of the sequence grows. Also, it was not able to solve the gradient vanishing problem. After running the simple recurrent neural network, we had training loss of 0.6920, training accuracy of 0.5188, validation loss of 0.6920 and validation accuracy of 0.5186. Figure 7 below shows the accuracy and loss graph of the network.

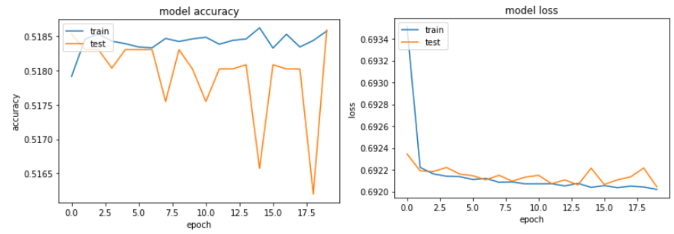

Figure 7. vanillaRNN model training, validation accuracy, loss

To solve the problems of vanillaRNN, LSTM layer is used instead. The result got better and stable compared to vanillaRNN model. After 20 epochs, training loss was 0.6890, training accuracy of 0.5319, validation loss of 0.6889 and validation accuracy of 0.5331. the losses stayed around the same range with vanillaRNN but there was a slight increase in the accuracy. Figure 8 shows the training and validation loss and accuracy.

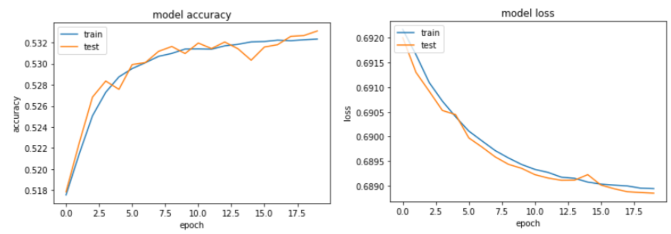

Figure 8. LSTM model training, validation accuracy, loss

As shown on Figure 5, validation and training accuracy and loss grows together and decreased together. There was no overfitting or underfitting issues. Problem was the accuracy. Validation accuracy of 0.5331 was still very low for us to confirm that the experiment is successful. Therefore, we started tuning the hyperparameters to see if we can make different results.

We inserted several different layers to improve the overall performances. First, embedding layer is implemented. Word embedding layer turns words in text into a dense vector. Compared to one-hot vector we learned, dense vector is processed to a low dimension and a real number. Second, we increased the depth of LSTM model. Two same LSTM layers were inserted. Stacking LSTM layer enables the model to get deeper by using more hidden layers. This can also add levels of abstractions of input observations as time flows. Moreover, 3 dropout layers were added after each LSTM layers to prevent the neural network from overfitting. Using dropout layers is a technology to ignore some portions of neurons during the training phase which is chosen randomly. Using different optimizers also resulted to have a different accuracy value.

As a result, we could achieve validation accuracy of 0.7923, validation loss of 0.4499, training accuracy of 0.8018, training loss of 0.4431. Figure 9 shows the final accuracy and loss of our experiment of LSTM hyperparameter tuning. The results were stable and there was no either overfitting or underfitting.

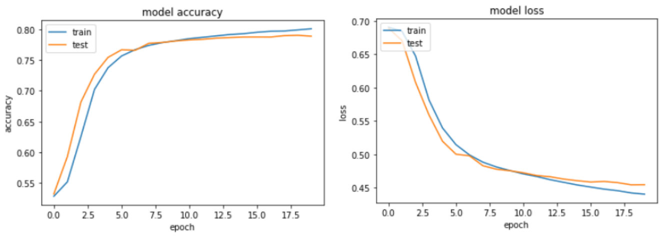

Figure 9. LSTM tuned model training, validation accuracy and loss

Our MLP model did not do well in classifying these tweets. After a couple of fine tuning, it was only able to achieve around 54.5% of accuracy as seen in Figure 10 below. This is also expected since MLP model usually is not able to remember the sequential patterns of the tweets language as it requires a huge number of parameters that grows as the sequence length grows given the structure of its fully connected layers.

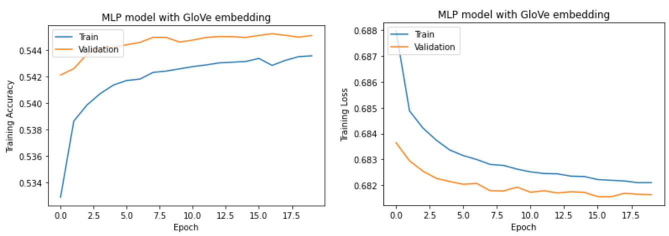

Figure 10. MLP Accuracy and Loss

For all 7 models discussed above, we measured the success based on validation accuracy with low amount of validation losses. By looking at table 1, we could conclude that our CNN model had the best results. But there are other models that were successful as well since the values are really close to our highest model, such as LSTM, LSTM-CNN, and Transformer models.

For DEC results, we eventually settled on using 3 clusters as it would provide the grouping that is better suited for our purposes. The clustering results divide our tracks dataset into 3 groupings based on energy, acousticness, and valence (musical positiveness), and the average values of each cluster are summarized in the table below.

Cluster	Energy	Acousticness	Valence
0	0.564876	0.422996	0.548184
1	0.349182	0.763728	0.471570
2	0.712973	0.166453	0.638067

Table 2. Tracks/Songs Clustering Results

Based on this table, we can see that there are 3 general clusters within the tracks dataset. Songs in Cluster 1 tend to have lower energy, high acousticness and lower valence; songs in Cluster 2 tend to have higher energy, lower acousticness, and more positive (higher valence); and songs in Cluster 0 are those in-between. Based on these results, we decided to pair the lower energy and valence songs in Cluster 1 with the negative sentiment, and higher energy and valence songs in Cluster 2 with positive sentiment, with the assumptions that people who are feeling down tends to prefer listening to slower tempo songs. After plotting the clusters' histogram, we further subset the data from these 2 clusters to only include songs with energy and valence lower than 0.6 for Cluster 1, and higher than 0.6 for Cluster 2 for better representations and reduce skewness.

To give song recommendations, we first pass through the sample input tweet into one of the models to predict the sentiments. Based on the sentiment output, we will match the input tweet to the respective cluster. We then vectorize the sample tweet and calculate the cosine similarities between the tweet and the song titles' vector embeddings and recommend songs within the highest 5 cosine similarity scores with the tweets.

4. Conclusions & Discussion

It was surprising to find that CNN performed the best out of all the models we used. Since we were processing text data, we expected the transformer model to give us the best performance followed by the LSTM model. This might have been due to how we were preprocessing the dataset to be padded to the same length sequences. This allowed CNN to be able to extract and focus on the most significant features since the input data was more like a row of an image instead of variable sized sequential data.

In addition, based on the structure of the tweets in our dataset, CNN might have performed slightly better because of similar features across tweets. For example, many negative tweets contained the word 'no' or 'don't', which the CNN model looks for across different examples. As expected, though, the CNN model ran the fastest compared to all other models.

The dataset itself consisted of tweets with many different typing errors (e.g., 'dived' instead of 'dove'), grammar mistakes (e.g., run-on sentences), abbreviations (e.g., 'idk' for 'I don't know'), and combinations of numbers in words (e.g., 'some1'). Since we are not able to manually comb through the data and fix typos, these issues combine to make some inputs particularly confusing for the models. For example, when vectorizing words, 'some1' and 'someone' would be separate vectors instead of being combined. While meaning the same thing, we now have two words instead of one.

We also have to be mindful of how different people express themselves differently. This in turn leads to a question of how reliable the provided labels are. For example, the following tweet is labeled negative in the dataset: "just leaving the parking lot of work!". To some, this may be taken to be a positive sentiment instead of negative. Since sentiment for sentences such as these is largely a subjective classification, this calls into question the labels of such 'fringe' examples where the sentiment is not immediately clear unlike the following example: "im sad now". To address this to some degree, we used cosine similarity between vector embeddings of the tweets and the song titles to give a closer recommendation to the sentiment of the tweet.

Using our best model, CNN, we sampled 20 tweets for testing and pass it through the CNN model to predict its sentiment. We then pair up each tweet with its song cluster based on the predicted sentiment and recommend songs whose titles have the highest 5 cosine similarities with the tweets. Some of the results are shown in Table 3 (see Appendix A for all other results). As we can see here, the recommended songs tend to have similar semantics with song titles (e.g., tweets talking about family day is recommended songs about family). The model also tends to recommend same songs with different style (i.e., different tempo, acousticness, etc.) or different artists if the song titles are similar or the same. And sometimes it will pick up an attribute (e.g., places like Brazil) from the tweets and then recommend songs whose title relate to it (e.g., Sao Paulo). This is also expected since titles and tweets with similar semantics will be embedded as similar vectors and results in higher cosine similarities computed between them as well.

Tweet	Detected Sentiment	Recommended Songs
i have learn one of the very sad facts of life i be allergic to murray my cat	Sad	1. So Sad (to Watch Good Love Go Bad) - 2008 Remaster by ['Emmylou Harris'] 2. So Sad (To Watch Good Love Go Bad) [feat. Connie Smith] by ['John Prine', 'Connie Smith'] 3. Mad Cat Sadie by ['Drahdiwaberl'] 4. So Sad (To Watch Good Love Go Bad) by ['The Everly Brothers'] 5. So Sad (To Watch Good Love Go Bad) by ['The Everly Brothers']
good morning how r u today lol i want bac to brasil but i can not	Happy	1. Alguém No Seu Lugar - Live In Sao Paulo / 2010 by ['Jorge & Mateus'] 2. Chove Chove - Live In Sao Paulo / 2010 by ['Jorge & Mateus'] 3. Esquece O Medo E Vem - Live In Sao Paulo / 2010 by ['Jorge & Mateus'] 4. Tempo Ao Tempo - Live In Sao Paulo / 2010 by ['Jorge & Mateus'] 5. Me Namora (feat. Edu Ribeiro) - Natiruts Reggae Brasil - Ao Vivo by ['Natiruts', 'Edu Ribeiro']
spend a family day	Happy	1. Family Song by ['Gen Hoshino'] 2. Family Tree by ['Ramz'] 3. Family Tree by ['H2O'] 4. In My Family by ['Sparks'] 5. Vacation Time by ['Part Time Musicians']

Table 3. Sample Song Recommendations

4.1. Limitations

Due to limited computational power, we had to resort to using a small subsample of the entire training data. Perhaps we would have gotten better results if we could use the entire dataset without the runtime becoming prohibitively long. In addition, training our models for more epochs could have yielded better results.

Another limitation was not being able to manually sift through the tweets to readjust labels that would have been better served as the opposite. For certain examples like the one given before (“just leaving the parking lot of work!”), perhaps it would have been better to relabel this as a positive sentiment class. This might have improved the performance of our models since sentiments would have been more consistent with what we expected.

4.2. Future Works

In the future, researchers might like to test their models on the entire dataset instead of a subsample. Given more computational resources, it might be better to try a larger embedding dimension and input sequence size.

In addition, testing more complex models might be a good idea to explore. The models presented in this report toe the line to overfitting closely, but given different batch sizes and embedding dimensions, more complex models might show better performance.

Future researchers might try adjusting the dataset or use entirely different preprocessing techniques to overcome the fringe cases as defined before. As mentioned in previous research, it might even be beneficial to use psycholinguistic vocabulary packages to preprocess the text.

Finally, a different embedding strategy might also be interesting to explore. We relied on GloVe in our report, but Fasttext and Word2Vec might also give good performance.

For songs recommendation, we rely on the cosine similarities to recommend songs after pairing the tweets with a high energy / low energy song cluster. This means that we mainly depend on the similar meaning or semantics of the tweets and the title, and the effectiveness of the clusters based on the available numerical values. For future work, if we can also label the song titles in a more relevant manner similar to the tweets’ sentiment, we might be able to match song recommendations better.

In addition, right now we do not have a feedback loop on how good the song recommendation is. Given enough resources, future works can potentially incorporate a feedback loop for songs recommendation (how good or bad the recommendation is) from users’ perspectives and add a layer of learnable parameters that can help fine tune the songs recommendation and the assigned sentiments /clusters further.

5. References

[1] Word Health Organization. Pulled July 2021. https://www.who.int/news-room/fact-sheets/detail/depression.

[2] Islam, R., Kabir., M., & Ahmed, A. "Depression Detection from Social Network Data using Maching Learning Techniques." 2018.

[3] Paul, S., Kalyani, J., & Basu, T. "Early Detection of Signs of Anorexia and Depression Over Social Media using Effective Machine Learning Frameworks" 2018.

[4] Orabi, A., Buddhitha, P., & Inkpen, D. "Deep Learning for Depression Detection of Twitter Users." 2018.

[5] DeLancey, E., Simms, J., Mahdianpari, M., & Brisco, B. "Comparing Deep Learning and Shallow Learning for Large-Scale Wetland Classification." 2019.

[6] Kim, J., Lee, J., & Han, J. "A Deep Learning Model for Detecting Mental Illness from User Contect on Social Media." 2020.

[7] Maratos, A., Gold, C., Wang, X., & Crawford., M. "Music Therapy for Depression." 2008.

[8] Xie, J., Ross G., & Ali, F. "Unsupervised deep embedding for clustering analysis." International conference on machine learning. PMLR, 2016.

[9] Go, A., Bhayani, R., & Huang., L. "Twitter Sentiment Calssifiction Using distant Supervision". 2009.

[10] Ay, Y. "Spotify Dataset 1922 and 2021." Pulled from https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks

[11] Pennington, J., Socher, R., & Manning, C. "GloVe: Global Vectors for Word Representation." 2014.

[12] Deepthi. "Unsupervised Deep Embedding for Clustering Analysis – A Summary." 2019. https://medium.com/@dptmn200/unsupervised-deep-embedding-for-clustering-analysis-a-summary-f6e5f2dce94f

[13] Vaswani, A., Shazeer, N., Parmar, N., & Uszkoreit, J. "Attention is All You Need." 2017.

[14] Reimers, Nils, and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." arXiv preprint arXiv:1908.10084 (2019).

6. Appendix A

Result from 20 sample test tweets, with its predicted sentiments and recommended songs:

Tweet	Detected Sentiment	Recommended Songs
135 and swim tomorrow hope it s water polo night twhorld	Sad	Swimming Across Your Sea - Strong - 2010 by ['Roli Frei'] There'll Be Another Spring - Live In Miami, FL/1959 / Remastered 2002 by ['Peggy Lee', 'George Shearing', 'Ron McMaster'] Porgy and Bess: Act III, Scene 2: There's a Boat That's Leavin' Soon for New York by ['George Gershwin', 'Lehman Engel', 'Lehman Engel Orchestra'] There's a Boat That's Leaving Soon for New York - Mono Version by ['Miles Davis'] Sangre en el Pool Party by ['Crudo Means Raw', 'Juan Gabriel Quintero Tanga']
epic bad dream officially freak out to get my wisdom teeth out on the 8th	Happy	In the Bad, Bad Old Days (Before You Loved Me) by ['The Foundations'] You're Breaking Me Up (And I'm Wasting Away) - Mono; 2009 Remaster by ['Cliff Bennett & The Rebel Rousers'] I'm Afraid of Americans - Nine Inch Nails V1 Mix by ['David Bowie'] We Hate It When Our Friends Become Successful - 2014 Remaster by ['Morrissey'] She's Got It Bad - 2008 Remaster by ['Simply Red']
no dillo day damn you sickness and rain	Happy	Fool in the Rain - Remaster by ['Led Zeppelin'] Fool in the Rain - Remaster by ['Led Zeppelin'] Please Don't Stop The Rain by ['James Morrison'] Mar De Rosas (I Never Promised You A) (Rose Garden) by ['The Fevers'] En voi ostaa rakkauttasi - Can't Buy Me Love by ['Eddy and The Lightnings']
i have learn one of the very sad facts of life i be allergic to murray my cat	Sad	So Sad (to Watch Good Love Go Bad) - 2008 Remaster by ['Emmylou Harris'] So Sad (To Watch Good Love Go Bad) [feat. Connie Smith] by ['John Prine', 'Connie Smith'] Mad Cat Sadie by ['Drahdiwaberl'] So Sad (To Watch Good Love Go Bad) by ['The Everly Brothers']

		So Sad (To Watch Good Love Go Bad) by ['The Everly Brothers']
good morning how r u today lol i want bac to brasil but i can not	Happy	Alguém No Seu Lugar - Live In Sao Paulo / 2010 by ['Jorge & Mateus'] Chove Chove - Live In Sao Paulo / 2010 by ['Jorge & Mateus'] Esquece O Medo E Vem - Live In Sao Paulo / 2010 by ['Jorge & Mateus'] Tempo Ao Tempo - Live In Sao Paulo / 2010 by ['Jorge & Mateus'] Me Namora (feat. Edu Ribeiro) - Natiruts Reggae Brasil - Ao Vivo by ['Natiruts', 'Edu Ribeiro']
i like the fresh breeze of night but today be so hot amp humid here	Sad	That Warm Summer Night - Remastered by ['Ricky Nelson'] I Love the Rain the Most by ['Joe Purdy'] Even the Nights Are Better by ['Air Supply'] Even the Nights Are Better by ['Air Supply'] Even The Nights Are Better by ['Air Supply']
yea	Happy	Oh, Yeh by ['Jimmy Castor'] Uhh Ahh by ['Boyz II Men'] Ooh! by ['Palmy'] Yeh-Yeh by ['Mongo Santamaria'] Yeh Yeh by ['Matt Bianco']
sorry i be in traffic not really move	Happy	This Town Ain't Big Enough For Both Of Us by ['Sparks'] This Town Ain't Big Enough For Both Of Us by ['Sparks'] This Town Ain't Big Enough For Both Of Us by ['Sparks'] The Road Less Travelled by ['Graeme Connors'] Traffic Jam by ['Kaya Brüel']
i seriously think some thing s wrong with my eye now really eye doctor please	Happy	Why You Wanna See My Bad Side by ['Smokey Robinson'] What's A Matter Baby (Is It Hurting You) by ['Timi Yuro'] Bad Medicine by ['Bon Jovi'] Bad Medicine by ['Bon Jovi'] Why Am I Treat So Bad by ['James Brown']
this guy have lose it i can only shake my head in wonder	Sad	I'm A Fool To Care by ['Ringo Starr'] I'm A Fool To Care by ['Les Paul', 'Mary Ford'] I'm A Fool To Want You by ['Oscar Peterson Trio'] What A Fool I Was by ['Percy Mayfield'] Esther's Awful Makeup/The Man That Got Away - Previously Unreleased Version by ['Judy Garland']
u can do it not that i m against former prof but it s always nice 2 win a bet eh	Happy	Va, Weurd Wakker Want We Moete Gon Doppen (I Don't Love You But I Think I Like You) by ['De Strangers'] It's Only Rock 'n' Roll (But I Like It) - Remastered by ['The Rolling Stones']

		Only the Loot Can Make Me Happy by ['R. Kelly'] Love is Gonna Save Us (Original) - Benny Benassi Presents The Biz by ['Benny Benassi', 'The Biz'] I Like Winning Better (2012) by ['Bruce Gaddy']
goood morning	Happy	When the Morning Comes by ['kalapana'] Morning Light by ['The Cats'] Morning Light by ['E-Type'] Morning Light by ['Novelbright'] Good Morning-Call by ['Kyoko Koizumi']
i ll be back in akureyri in approximately 7 days and 182 hours happy happy happy	Sad	I Say You Say I ♥ You - Happy New Year Acoustics! IN 九段教会 2018.01.27 by ['moumoon'] “Best of 2016 Medley: Stressed Out / 7 Years / Work / Treat You Better / Can't Stop the Feeling / Closer / 24k” by ['Anthem Lights'] Best of 2016 Medley: Stressed Out / 7 Years / Work / Treat You Better / Can't Stop the Feeling / Closer / 24k by ['Anthem Lights'] トモダチ/コイビト - Happy New Year Acoustics! IN 九段教会 2018.01.27 by ['moumoon'] うたをうたおう - Happy New Year Acoustics! IN 九段教会 2018.01.27 by ['moumoon']
black forest cake for my birthday	Sad	Shadows And Tall Trees / Saturday Matinee - Remastered 2008 by ['U2'] Christmas Oratorio, BWV 248 / Pt. One - For The First Day Of Christmas: No.8 Aria: "Großer Herr, o starker König" by ['Johann Sebastian Bach', 'Franz Crass', 'Münchener Bach-Orchester', 'Karl Richter'] I'll Be Home for Christmas (with Percy Faith & His Orchestra) by ['Johnny Mathis', 'Percy Faith & His Orchestra'] A Charlie Brown Thanksgiving by ['George Winston'] Passover Seder Festival: A Passover Service: Medley Of Traditional Songs (Sung After the Festive Meal): Vayhi Bachatsi Haloyloh; Ki Lo Noeh-Ki Lo Yoeh; Adir Hu; Echod Mi Yodea - Voice by ['Sholom Secunda', 'Richard Tucker']
hey girl	Sad	Hey Girl by ['Norma Tanega'] That Girl by ['Marques Houston'] That Girl by ['Glenn Frey'] this girl by ['Elijah Who'] See That Girl by ['The Righteous Brothers']
cavan get toby back today i miss him	Sad	I Miss You (feat. Teddy Pendergrass) by ['Harold Melvin & The Blue Notes', 'Teddy Pendergrass'] I Miss You (feat. Teddy Pendergrass) by ['Harold Melvin & The Blue Notes', 'Teddy Pendergrass'] I Miss You, Pt. 1 (feat. Teddy Pendergrass) by ['Harold Melvin & The Blue Notes', 'Teddy Pendergrass']

		To Know Him Is to Love Him - 2015 Remaster by ['Dolly Parton', 'Linda Ronstadt', 'Emmylou Harris'] It Might Be You - Theme from Everyday I Love You by ['Michael Pangilinan']
i ll definitely have to give it a shoot might pick that up tomorrow weekend be for relax after all	Sad	It Could All Be Gone Tomorrow by ['Nessy'] Laid-back Backdrops for New Years Resolutions by ['Hotel Lobby Music'] Relaxing Moods for New Years Resolutions by ['Office Work Music'] see u tomorrow - stripped by ['Arash Buana'] Teach Me Tonight by ['Blossom Dearie']
pop it lock it polka dot it countrify then hip hop it put your hawk in the sky move side to side jump to the leave stick it glide	Sad	The Blackbird (Air, Set Dance And Reel) by ['The Bothy Band'] My Lady's a Wild Flying Dove by ['Tom Paxton'] The Firebird: Scherzo - Dance of the Princesses by ['Igor Stravinsky', 'New York Philharmonic'] The Firebird: II Entry and Dance of the Firebird by ['Igor Stravinsky', 'Walter Straram Concerts Orchestra'] Punch and the Child, Op. 49: Scene I: Overture - Child's Dance - Storm - Punch and the Child by ['Richard Arnell', 'Royal Philharmonic Orchestra', 'Sir Thomas Beecham']
spend a family day	Happy	Family Song by ['Gen Hoshino'] Family Tree by ['Ramz'] Family Tree by ['H2O'] In My Family by ['Sparks'] Vacation Time by ['Part Time Musicians']
your tweet be just include in the longest poem in the world	Happy	Remember in 2006 When It Was Fashionable to Have the Longest Song Names You Could Think Of, Literally Like a Short Story by ['Next Year'] The Longest Line by ['NOFX'] 旅人よ ～The Longest Journey by ['Bakufu Slump'] Greatest Story Ever Told by ['Bob Weir'] The Horniest Single in the World by ['E-Rotic']