Predicting Music Emotion with Social Media Discourse

By
Aidan Beery

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Associate)

Presented August 18, 2022
Commencement June 2023

AN ABSTRACT OF THE THESIS OF

Aidan Beery for the degree of Honors Baccalaureate of Science in Computer Science presented on August 18, 2022. Title:
Predicting Music Emotion with Social Media Discourse

Abstract approved:

Patrick Donnelly

Predicting the average affect of a piece of music is a task which has been of recent interest in the field of music information retrieval. We investigate the use of sentiment analysis on online social media conversations to predict a song's valence and arousal. Using four music emotion datasets - DEAM, AMG1608, Deezer, and PmEmo, we create a corpus of social media commentary surrounding the songs contained in these datasets by extracting comments from YouTube, Twitter, and Reddit. Two learning approaches are compared − one bag-of-words model using dictionaries of affective terms to extract emotive features, and a DistilBERT transformer model fine-tuned on our social media discourse to perform direct comment-level valence and arousal prediction. We find that transformer models are better suited to the task of predicting music emotion directly from social media conversations.

Key Words: Music Information Retrieval, Music Emotion Recognition, Natural Language Processing, Transfer Learning

Corresponding e-mail address: beerya@oregonstate.edu

Predicting Music Emotion with Social Media Discourse

By
Aidan Beery

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Associate)

Presented August 18, 2022
Commencement June 2023

Honors Baccalaureate of Science in Computer Science project of Aidan Beery presented on August 18, 2022.

APPROVED:

Patrick Donnelly, Mentor, representing School of Electrical Engineering and Computer Science

---

Jeremy Shaw, Committee Member, representing Department of Mathematics

---

Patrick Ball, Committee Member, representing College of Science

---

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University Honors College. My signature below authorizes release of my project to any reader upon request.

Aidan Beery, Author

Contents

1 Introduction

Music emotion recognition is the application of computational methods to the understanding of emotions elicited in a listener by a given piece of music. Determining the cultural average response of an audience to a song is of interest to the music information retrieval community. Historically, determining musical affect has relied on manual surveys using crowdsourced annotation platforms such as Amazon Mechanical Turk [9]. Emotion annotation surveys are often time-consuming and cost-prohibitive due to the large number of unique annotators required to yield inter-annotator accuracy between ratings [33].

These factors have created generated interest in the automatic detection of music affect. Previous attempts have relied on analysis of acoustic features derived from song audio, affective term extraction from lyrics [58], and combined lyric-audio feature spaces to determine music mood categories [27]. Many of these methods aim to predict music mood at a discrete, categorical level, as opposed to directly determining continuous valence and arousal emotion dimensions as defined by Russell's circumplex model [1] [47] [55].

In recent years, there have been efforts to predict music emotion targets from audio and lyric features using deep learning approaches. The fusion model presented in a study from Deezer applies a combined LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network) model, with raw lyrics and audio mel-spectrograms as features in a combined approach to predict continuous emotion values directly [13]. Transformer models have recently proven effective for natural language understanding and emotion recognition [14]. When applied to music emotion classification, models of this architecture have been able to categorize songs into mood categories with accuracy [1].

The task of automatic emotion recognition is also of interest in the domain of social media sentiment analysis. Historically, social media emotion extraction has relied on a bag-of-words approach, cross-referencing text with a dictionary of unigrams with manually annotated affective ratings [42] [46]. However, this approach generally failed to account for sentiment negation, as it lacked the ability to understand the context a given word was used in. Because of their ability to encode the information from previous tokens in the processing of a given word, transformer models have been found to be effective at social media mood categorization [10].

The use of lyric-based features for music mood prediction inspires an investigation into alternative text inputs for this learning task. The prevalence of sentiment analysis tools for social media commentary indicates that the conversations we have online

[1] Valence is defined as an axis measuring positivity, and arousal is a measure of energy.

contain inherent semantic meaning, which is believed to indicate a user's emotion. We hypothesize that the social media discussions surrounding a song contain semantic information, which an intelligent agent could use for predicting a song's affective qualities.

We present a comparison of two models for predicting continuous music emotion labels from social media discourse. Our first experiment uses a more conventional bag-of-words approach − analyzing the individual terms contained within a comment to estimate the average sentiment of the commentary surrounding a given song. From this, we generate a set of features summarizing the average valence, arousal, and affective norms of the comments by use of a series of word affective dictionaries, and compare a series of models for predicting song valence and arousal from these values. For comparison, we also use a transformer model to predict music valence and arousal labels directly from raw discussion data.

To our knowledge, these experiments are the first attempt to predict music emotion values directly from social media conversations. We produce a dataset of social media discourse based on the songs contained in a series of music emotion datasets to aide the comparison of sentiment analysis models to manual affective labeling. A feature extraction approach for generating a statistical summary of comments in reference to a specific song is defined. Two learning approaches are compared, one of which trains ensemble models on the aforementioned feature space, and the other compares two popular pre-trained natural language processing transformer models − DistilBERT and RoBERTa − for direct comment-to-music-emotion-label prediction without the need for feature engineering.

A song's affective qualities are unique, as the interpretation of a piece of music will vary between cultures and from individual to individual. The emotions a song elicits will vary between listeners and will be dependent on demographic factors. In this work, we focus on estimating a cultural average affective response given conversations from social media platforms.

2 Emotion Modeling

We begin with a brief overview of the models used to represent emotion across music information retrieval and computational linguistics research. The labels used in modern computational emotion understanding literature generally rely on one of three affective models: Plutchik's eight basic emotions [44], Ekman's six basic emotions [17], or the Valence/Arousal/Dominance dimensional model described by Osgood and Russell [47] (see Figure 1). These models are referenced frequently in the analysis of affect in language [36] and music [13], and provide a framework to allow researchers to discuss an approximation of human emotion.

Author	Type	Dimensions
Plutchik [44]	Basic Emotions	Joy, Sadness, Anger, Fear, Trust, Disgust, Surprise, Anticipation
Ekman [17]	Basic Emotions	Anger, Disgust, Happiness, Sadness, Fear, Surprise
Russell [47]	Continuous Dimensions	Valence, Arousal, Dominance

Table 1: A summary of the three emotion models frequently used in computational emotion understanding experiments

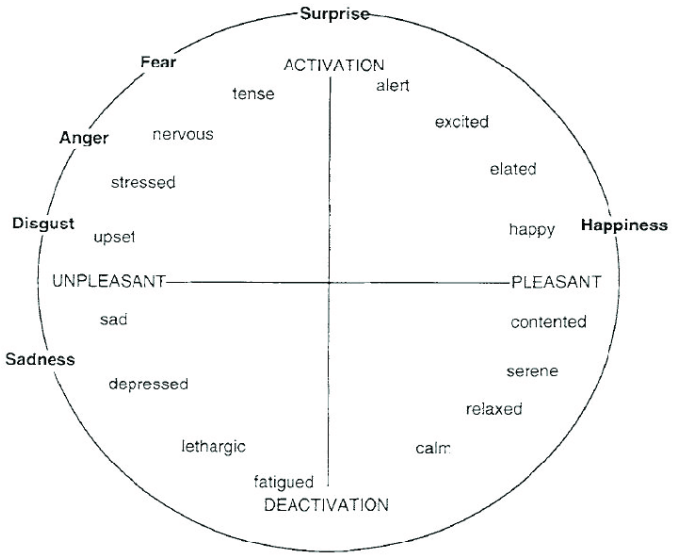


Figure 1: Russell's circumplex model of emotion, with Ekman's six basic emotions mapped to the space [17] (from [47])

In his book *Theories of Emotion* [44], Plutchik seeks to capture the basic human experience by describing a fundamental set of basic emotions. First, he establishes

a definition of a basic emotion as a set of cognitive responses to external stimuli, which motivates an individual to respond in some way. This cognitive framework enables an evolutionary motivation for the experience of human emotion. A natural justification for these cognitive responses would be that they motivate an organism to evaluate its environment and make decisions which correspond with positive emotions and minimize negative ones. Plutchik postulates that the human state of mind can be modeled by a series of four opposing pairs of basic emotion, each of which can vary in affect or intensity. The composition of these emotions at varying intensity can theoretically encompass any range of cognitive experiences.

Plutchik's model lists four antecedent emotive pairs: *joy-sadness*, *trust-disgust*, *fear-anger*, and *surprise-anticipation*. For each of these eight basic emotions, a set of corresponding behaviors and traits are laid out. For example, the behavioral language of sadness would be to cry, and an individual who experienced sadness would have the trait of being gloomy [44].

Ekman's model of six basic emotions seeks to refine the model laid out by Plutchik by first constraining the definition of a basic emotion [17]. Instead of focusing on a pscyhoevolutionary approach to emotion and defining any emotion by the composition of a series of basic emotions, Ekman instead claims that basic emotions must be a set of distinct signals which are brief in nature such that they do not define an individual's traits, and instead focus on their responses to current ongoing stimuli.

The Ekman model of emotion lists six basic emotions: *anger*, *disgust*, *happiness*, *sadness*, *fear*, and *surprise*. These overlap with Plutchik's model, with the notable exception of trust and anticipation. Ekman argues that these two emotions are encompassed in the six others, and they do not represent a unique signal of rapid onset in response to a stimuli, and therefore are not basic emotions.

The aforementioned models rely on a categorical approach to emotion understanding. Plutchik and Ekman share the same concept in their frameworks of a core set of basic emotions to describe any given mood. Thus, mood is thought to be broken into a series of discrete values. However, this approach has natural limitations [47], as it restricts the granularity of emotional experience which can be described by a given model. This is especially true for the Ekman model, which makes no claims of the composibility of basic emotions.

In contrast, continuous emotion models treat emotion as a set of continuous values in a space, described by a series of axes which describe some antecedent pair of cognitive experiences. The valence-arousal-dominance model, also known as the pleasure-arousal-dominance model, is the most notable example of such a framework [38] [47]. The three dimensions of emotion were initially proposed by Osgood as three

independent, continuous values which could be composed to describe any emotion [38]. Russell improved on this concept by introducing the circumplex model of emotion and connecting the three dimensions into a three-axis emotion space [47].

Russell's paper made two arguments. One, that the concepts of valence (positivity-negativity), arousal (excitedness-sleepiness), and dominance (in control - out of control) are connected and can be treated as three axes in a space instead of three independent dimensions. Secondly, he claims that the basic emotions laid out by Plutchik and Ekman can be represented in the two-dimensional plane described by the valence and arousal axes.

The valence-arousal-dominance model of emotion is the continuous emotion model we see most frequently in our affective wordlists [6][56][33]. In the case of music emotion recognition studies, dominance is often not included [9][2]. Despite examples in word emotion studies indicating that dominance is strongly correlated to valence, and thus unnecessary to measure independently [56], recent music emotion recognition experiments have considered a lack of dominance annotation a limitation of the current body of work [61].

Recent studies have demonstrated that these two approaches of emotion modeling are inherently connected and inputs can be mapped from one space to one another. Park et al. describes the use of a transformer model for learning valence, arousal, and dominance labels from mood categories on a large corpus of English text [41]. RoBERTa, a pre-trained language understanding transformer model [30] (see 3.2), is fine-tuned on English text corpora manually annotated for valence-arousal and mood labels. These mood categories were derived from the basic emotions described by Plutchik and Ekman. The study demonstrated Pearson's correlation as high as 0.7 when predicting valence from the continuous valence-arousal-dominance labels predicted from categorical mood features, indicating that these two methods of emotion modeling and recognition are related in the case of English text emotion prediction.

3 Computational Models

A variety of computational modeling techniques have demonstrated the potential to be effective at emotion recognition and sentiment analysis tasks [10] [40]. Machine learning algorithms leverage large training datasets to learn decision boundaries to enable decision-making on previously unseen samples. Recent developments in deep learning have yielded larger, more complex models suited to natural language processing and understanding tasks [14]. Understanding the fundamental function of these methods assists in interpreting experiment results in music information extraction, so we briefly investigate a selection of machine learning approaches.

3.1 Machine Learning Models

The five machine learning techniques we assess for our music emotion prediction task are part of a category of supervised learning algorithms. These models rely on an existing set of labeled data in order to make predictions about the labels of new data. In problems which would otherwise require human annotation, supervised learning algorithms allow for automated decision-making using a subset of annotated samples, potentially reducing the cost and overhead of data labelling.

An early example of such a model is the k-nearest neighbors algorithm, proposed by Fix and Hodges in 1951 [18]. The k-nearest neighbors algorithm was originally created for the purpose of binary classification problems, where a sample belongs to one of two classes and a model must predict which of the two categories a sample belongs to. It does so by computing the distance between k feature vectors and an input vector $\vec{i}$, and assigning $\vec{i}$ the label which occurs most frequently in its k neighboring samples. This principle was later extended to regression problems by taking the mean of each neighboring sample's label to assign a label to $\vec{i}$ [3]. This non-parametric approach allows for complex relationships between variables to be explored without any assumed structure of the input space. However, this makes the model sensitive to the local structure of the data, and weak against class imbalance.

Support vector machines are a category of model which generates decision boundaries in high-dimensional spaces in order to label samples. The original implementation of the support vector machine, proposed by Cortes and Vapnik, describes an algorithm for binary classification by identifying a hyperplane with the maximum separation between two classes. The original implementation of the support vector classifier was only capable of drawing linear decision boundaries, however later work used kernel functions to project inputs into a higher dimensional space. If this kernel function was nonlinear, it would enable the model to create linear hyperplanes in the transformed space which represent non-linear decision boundaries in the input space.

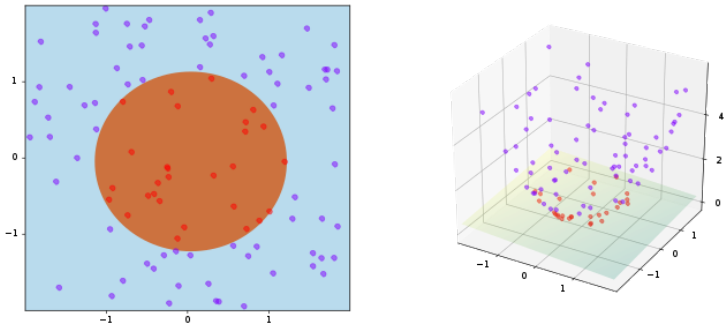


Figure 2: An example of drawing a non-linear hyperplane in a 2-D input space by projecting it into a higher dimensional space where a linear decision boundary can be more clearly defined, often referred to as the kernel trick [2]

The principle of support vector machines was later extended to regression by generalizing the optimization problem for identifying a hyperplane as a decision boundary to one which approximates a function with an error no greater than ϵ on the training data [16].

Ensemble methods in machine learning have enabled the combination of predictions from weak learners to be used to build a robust decision system. Random decision forests provide a method for reducing the risk of overfit found in decision trees. A decision tree builds a sequence of decision boundaries based on the automatic identification of conditions within the feature space which delineate one class of samples from another. This model is often represented as nodes descending in a tree, where each intermediary node is a "split", representing one of these boundary conditions. As a decision tree grows deeper and integrates more splits into its hierarchy, it risks overfit, reducing the model's ability to generalize to new samples. Random forests seek to mitigate this by applying bagging to an ensemble of shallow decision trees — training each tree on a random subset of the dataset, and taking the mean of each tree's outputs to yield a label. Furthermore, at each split in a given decision tree, a random subset of features is selected. By using bagging and random feature sampling, random forests generalize by decreasing sensitivity to noise in the training set [21].

Adaptive Boosting (AdaBoost) is another form of ensemble model. Unlike a random forest model, AdaBoost is a procedure generalizable to any weak classifier. Boosting involves the process of iteratively training a series of weak learners, selecting inputs for the next learner based on the errors of the previous learner. During the

[2] https://commons.wikimedia.org/wiki/File:Kernel_trick_idea.svg

training of an AdaBoost model, each sample is assigned a weight, and samples which are misclassified at a training step are given a greater weight, causing future weak learners to focus on these samples [19]. The sum of each weak model's prediction is taken to generate a prediction. Though originally proposed to solve binary classification problems, the principle of AdaBoost was extended to regression by Solomatine and Shrestha using the AdaBoost.RT algorithm [49], where sample weights are updated according to the error of each naive regressor.

Another method for building ensemble models of weak estimators relies on applying gradient descent to the aforementioned boosting procedure. Instead of assigning weights to misclassified samples, each subsequent learner in a Gradient Boosting Trees model trains on the residuals of the previous learner by calculating the negative gradients of the loss function at each step [20]. Though this method of bagging decision trees was applied with success, it suffered from an inability to scale to datasets of high dimensionality and large sizes due to the computational complexity of calculating each gradient. Ke et al. contributes LightGBM, an efficient implementation of Gradient Boosting Trees which relies on sampling only from data instances which have a gradient greater than some pre-defined threshold and using feature bundling to reduce the dimensionality of the data instances [24].

3.2 Transformers

Because of the increasing relevance of transformer models for NLP and emotion recognition [10] [1], we consider it important to establish a foundational definition of this novel deep learning architecture. Transformers are based off the principle of self-attention. They originate from previous models which used recurrent architectures and convolution operators for time series analysis. Often, these models used attention operators for encoding inputs and decoding outputs. Recurrence depended inherently on the ability to define a hidden state based on information from previous hidden states, allowing a given token or input to understand the context provided by some derivative of a prior input state.

The basic premise of the transformer model architecture is to build a model from exclusively self-attention operators to provide a similar contextualization effect as recurrent structures by embedding the information of the previous tokens in the current token. Self-attention layers process a sequence of n inputs, returning a sequence of n outputs as a function of the computed attention scores of each input. Each token's attention score considers information from the hidden layer state representing every other input. From this, each token can assign a weight to each other token's key, representing the relationship between that prior token and the current one. These weights are then updated during training, allowing the model to gain an understand-

ing of how past information affects the current input [53].

This architecture has been found to be especially useful in natural language processing tasks. Sentences can be encoded as a series of tokens, where each ID represents a word. Word embeddings allow for entire sentences to be represented as a single feature vector, where unique numeric values are mapped to each word based on the lexical distance between terms. This word vector representation allows free-form text inputs and outputs to operate seamlessly with transformer models.

An example of one such model is the Bi-directional Encoder Representations from Transformers (BERT) [14]. It represents one of the most popular frameworks for developing transformer models for natural language understanding. Inspired by the Cloze task, it uses a masked language model to pre-train on a large unlabeled corpus of English text. The BERT team pre-trained its model on a large dataset of English literature, as well as the entirety of English Wikipedia. During pre-training, the model takes a sentence as an input, hiding one token from itself, and attempts to predict the value which should be in the masked location. This allows the model to be fit to predicting the next probable token from any given sequence. For example, it can complete a sentence by predicting what the next word may be. It can also train on question and answering tasks by predicting the likely response to an input question in what is referred to as next-sentence prediction.

The bi-directional nature of BERT's implementation of the transformer model allows for each token to have both look-ahead and look-behind capability, embedding the context of other words in the sentence based on their attention scores [14]. In the context of a sentence, this means BERT's self-attention heads understand the context both from words before the current token, as well as those appearing after it. This, alongside pre-training on a massive English text dataset, make it a powerful model for developing language understanding models. These pre-trained weights can be updated during a fine-tuning process, where BERT is trained on the specific downstream prediction task. Because of the embedded language understanding developed as a result of the pre-training process, fine-tuning often requires far fewer epochs than traditional feed-forward neural networks.

RoBERTa promises an improvement over the pre-training approach used in the original BERT model by increasing the size of the training dataset by an order of magnitude [30]. Consisting of 125 million parameters compared to BERT's 110 million, RoBERTa uses the same architecture design as its predecessor. The main difference between the models is in the pre-training corpora and the pre-training methods. BERT pre-trains on a 16 gigabyte English text dataset derived from Wikipedia and BooksCorpus, while RoBERTa uses 160 gigabytes of text, including the BERT corpora as well as a series of articles scraped from online blog and news platforms. RoBERTa

does not use next-sentence prediction during its pre-training, unlike BERT. Because of the significantly larger pre-training dataset, RoBERTa is able to exceed BERT performance on many benchmark NLP tasks. However, the additional compute resources necessary to train RoBERTa are significant, as Meta's team pre-trained the model on 1024 Nvidia v100 GPUs for 1 day.

Though pre-training allows BERT-like models to be fine-tuned in relatively few epochs, training can still require immense compute resources, especially in the case of many NLP tasks where datasets are often very large. Deep learning approaches have historically benefited from large training sets for complex tasks, so reducing the size of inputs may not be ideal. DistilBERT aims to offset the computational cost of training transformer models on large datasets by reducing the size of the model, and therefore improving training and inference times significantly [48].

DistilBERT uses a similar pre-training process and data source for its pre-training, however it applies knowledge distillation to compress the overall parameters of the model from 110 million to only 66 million. Knowledge distillation relies on a student-teacher model, where the "student" model is trained to minimize loss between its probabilities and the outputs of the larger "teacher" model. This allows BERT's existing robust performance to be leveraged to train a model with half as many layers [48]. The HuggingFaces team reports that DistilBERT retains 97% of the performance of its predecessor, while also reducing inference time by up to 39%. The DistilBERT model was pre-trained on 8 v100 GPUs, offering a significant reduction in compute resources relative to RoBERTa.

The original self-attention mechanism described in [53] scales quadratically with sequence length, meaning that the use of transformer models on long form text inputs is not practical. As a result, most popular pre-trained transformer models are restricted to input lengths of 512 tokens, including BERT and its aforementioned derivatives. `xl-net` attempts to extend the performance of this architecture to be compatible with longer form inputs [60]. By pre-training on larger sequences, this model natively supports inputs of up to 1024 tokens. Long-distance attention relationships are formed thanks to an intermediary recurrence layer connected to the base `xl-net` attention heads, learning from all hidden layer states instead just the last one, as is the case in BERT-like models. Longformer, another transformer model intended for long-form text inputs, provides a linear approximation of the self-attention mechanism, allowing for documents of unrestricted length to be used as inputs, at the cost of performance in popular NLP benchmark tasks [4].

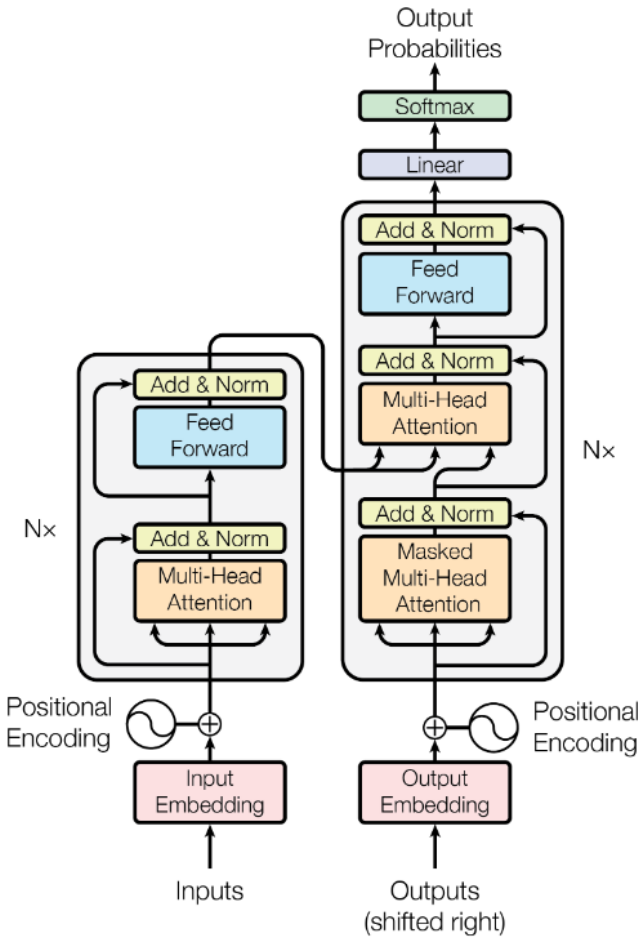


Figure 3: Architecture of a transformer model (from [53])

4 Background: Emotion Extraction from Text

The recent use of sentiment analysis for music mood classification on lyrics [22] [8] and Last.FM tags [5] [13] indicates that natural language processing is becoming an increasingly important component of modern music information retrieval research. The use of Last.FM for music emotion recognition tasks specifically motivates an investigation into the methods for affective modeling of social media conversations. First, we analyze a series of word affective dictionaries which are frequently used for computational emotion recognition research both in music and social media contexts. Then, we review the application of these wordlists as well as other techniques for social media sentiment analysis.

4.1 Word-Emotion Association

Computational linguistics researchers have made multiple efforts to create high quality affective dictionaries for the association of certain emotions to certain words. Many of the language models used for sentiment analysis of English text depend on a ground truth for individual words and their associated affective dimensions. Because of the relevance of sentiment analysis across various domains [40], a variety of affective dictionaries have been created to assist in the development of natural language understanding systems. Many approaches are taken to the annotation and psychological modelling of words [50] [51], however we focus on datasets which adhere to either Russell's model of emotion, a subset of the basic emotion dimensions, or positive/negative sentiment.

4.1.1 Annotation Crowdsourcing

To produce accurate emotion labels for a dataset, whether it be one of words or of songs, researchers rely on conducting surveys to estimate an affective average based on the response of many individuals. Traditionally, these studies would be conducted in a university environment, leveraging students as a relatively easy means of gaining a large volume of annotators [6] [61]. However, these environments are constrained in available sample size. Furthermore, there is a risk that the demographics of the university setting may introduce bias into the dataset. As a result, recent approaches in music emotion annotation [9] and affective dictionary generation [56] have relied on online crowdsourcing platforms to conduct surveys at much broader scale.

One such platform is Amazon Mechanical Turk, a web platform which enables users to be compensated for online manual processing work [3]. Each user is presented

[3] https://www.mturk.com/

with a Human Intelligence Task (HIT), usually consisting of a series of questions regarding a particular sample. The user will be paid for completing this HIT based on the price set by the institution conducting the survey, usually no more than a few cents USD. CrowdFlower is another such survey platform which pays annotators by task completed, and both of these platforms are used in the creation of the following affective dictionaries and music emotion datasets [4].

There exists concerns about the ability to yield high-quality annotations from these platforms. Conducting a traditional survey is expensive and limits the scope of annotations. However, platforms like Amazon Mechanical Turk are prone to abuse or low-quality annotations. Because "Turkers" are paid by the task completed, not by time spent, users have a financial interest in completing tasks as quickly as possible, regardless of quality. To compensate for this, surveys must be designed in a way which both encourages accurate ratings and discards inconsistent responses. This rejection system would benefit from being an automated task, as paying annotators to manually verify crowdsourced ratings requires additional human effort.

To ensure high quality labels in their datasets, Mohammad and Turney implement a robust method for acquiring and validating crowdsourced annotations in the creation of their lists of affective terms [37]. Users are presented with a calibration problem, asking them to identify one of four words which is closest in meaning to the target word. This enables filtering of both users aiming to abuse the survey system, and those who genuinely are not familiar with the target word and therefore would not provide a meaningful rating, even if in good faith. Users are then asked if a word exhibits an emotion or not, for all eight emotions. They are also asked to rate the word's positivity and negativity.

4.1.2 Affective Dictionaries

	EmoVAD	EmoLex	EmoAff	MPQA	eANEW
# Words	20007	14181	4192	6886	13915
# Features	3	10	4	1	64
Feature Type	V/A/D	Affect	Affect	+/-	V/A/D

Table 2: Summary of the affective wordlists used for feature extraction

One of the most influential works in valence/arousal/dominance lexicon development was the creation of the Affective Norms for English Words (ANEW) dataset (1999)

[4]CrowdFlower was acquired by Appen in 2019 https://appen.com/resources/

[6]. 1,106 affective terms were selected based on prior work in the International Affective Picture System. The words were manually rated for valence, arousal, and dominance by a group of undergraduate psychology students from the University of Florida. For this dataset, Bradley and Lang developed the Self-Assessment Manikin, a framework for conducting VAD annotation experiments. This allows the valence, arousal, and dominance to be represented graphically in a way which is generally understandable by an untrained subject. Many future emotion annotation experiments base their work around the model and experiment procedure designed by Bradley and Lang [61].

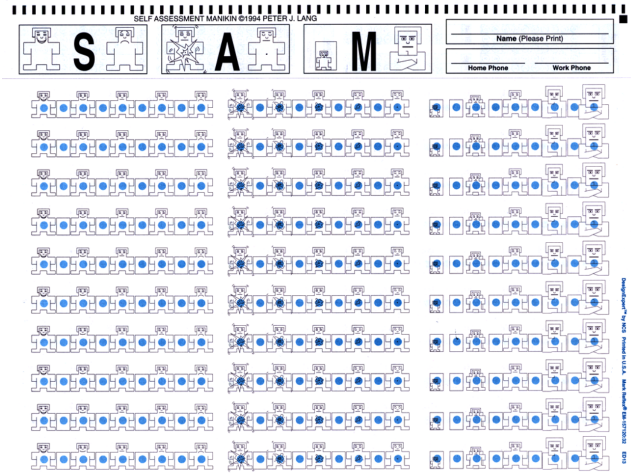


Figure 4: A Self-Assessment Manikin worksheet for rating affective terms (from [6])

The ANEW dataset established a benchmark for word affective dictionaries. However, due to annotations being gathered from students over a fixed timeframe, the dataset was constrained in size. By limiting the number of affective terms in a lexicon, sentiment extraction methods are constrained by the quantity of understood words. This may lead to semantically relevant information being thrown out in a bag-of-words model, where the language model may only identify words already existing in its dictionary (or semantically similar words [32]). Warriner et al. set out to expand on ANEW using modern annotation methods to allow for surveys of larger scope [56].

The Extended ANEW Lexicon (2013) provides valence, arousal, and dominance labels for 13,915 English words, as a superset of the existing terms chosen in ANEW [56]. The procedure defined by Bradley and Lang was repeated, including the use of the Self-Assessment Manikin. However, the survey was conducted using Amazon Mechanical Turk, and annotators were only asked to rate one dimension at a time (valence, arousal, dominance) instead of being presented with all three axes. 1,085,998 annotations from 1,827 annotators were acquired, and annotations which had a cor-

relation of less than 0.10 with other user's labels were discarded.

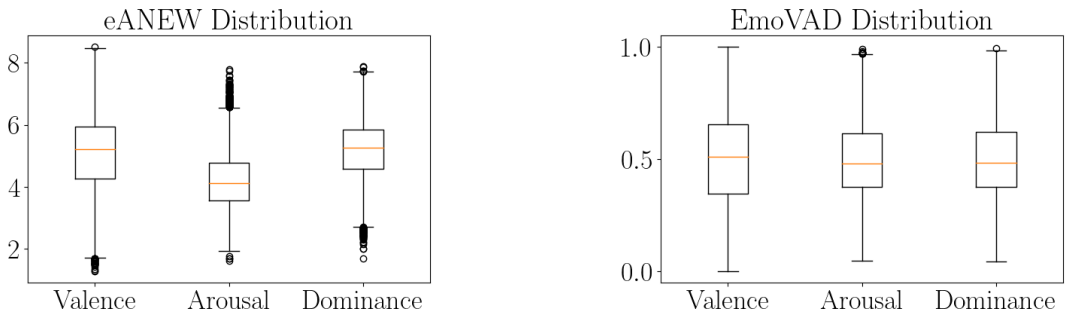


Figure 5: Affective label distributions of EmoVAD and eANEW

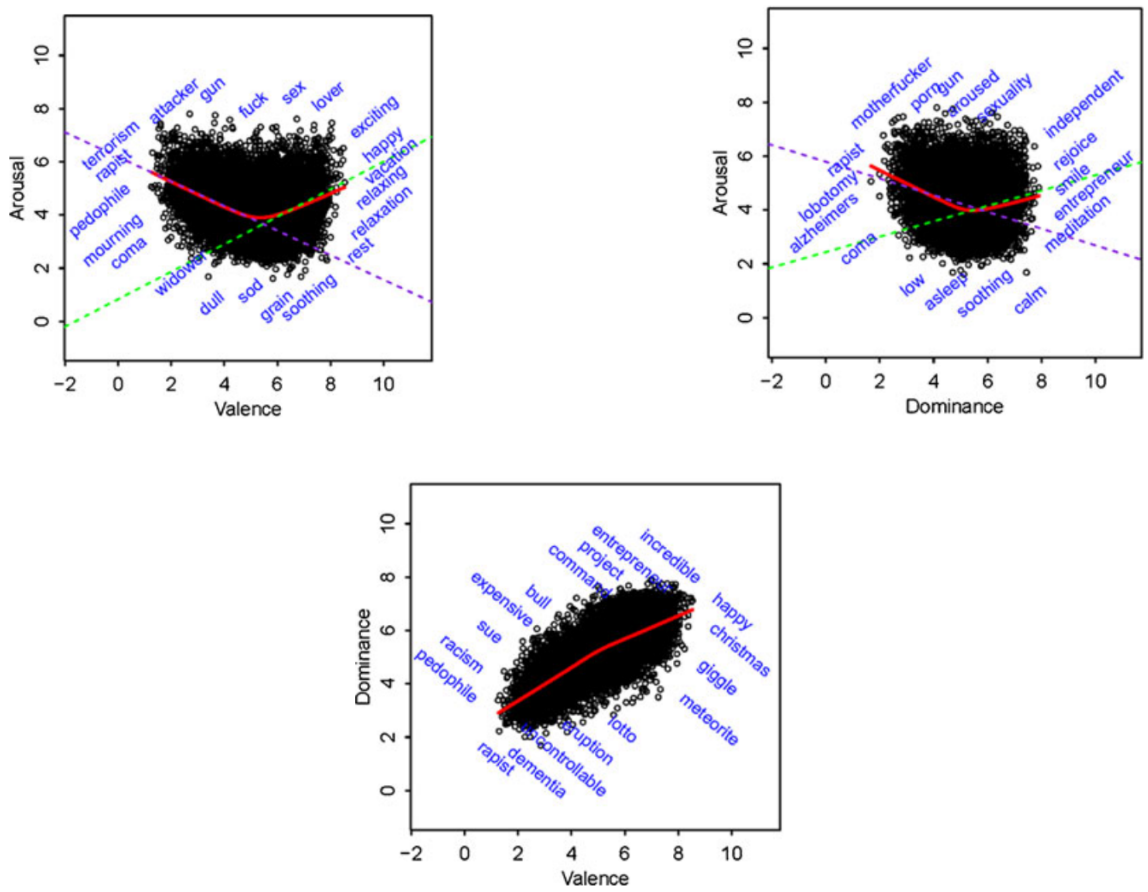


Figure 6: Relationships between each affective axis for Extended ANEW (from [56])

Each word is presented in the dataset with a series of 64 features, representing

a statistical summary of the annotation responses received for that word. Words are annotated on a scale of $[0, 10]$ and not explicitly centered, as seen in Figure 5 [56]. Along with contributing an extension of ANEW, Warriner et al. note a strong correlation between valence and dominance labels, unlike the U-shaped relationships existing between the other dimensions. This finding calls into question the relevance of dominance for the purposes of sentiment analysis for emotion understanding.

The National Research Council Canada (NRC) has released three high-quality crowdsourced word emotion lexicons, contributing the largest set of manually labeled affective terms to date. The work by Mohammad and Turney has had significant impact on the natural language processing community by enabling the development of more robust emotion understanding models. These three datasets rely on Amazon Mechanical Turk and CrowdFlower for publishing large-scale surveys of English-speaking subjects for the annotation of a combined 39,168 terms over valence/arousal/dominance, affective dimensions, and positive/negative sentiment.

The first of these datasets was the NRC Emotion Lexicon (2010), otherwise known as EmoLex (2010). Mohammad and Turney develop a dictionary of 14,182 English words rated by mood category using Plutchik's basic emotions as mood categories (see Table 1). Annotators are asked to describe whether a given word evokes one of the eight given emotions, as well as rating terms for positive and negative sentiment. [36] [37].

The EmoLex dictionary was built from a combination of three sources−the Macquarie Thesaurus, the General Inquirer, and Google's n-gram corpus. Overall, high inter-annotator agreement was achieved, with over 80% of words having five or more annotators in agreement. When a subset of EmoLex was compared to the sentiment ratings from the General Inquirer, it was found that 100% of terms which were identified by GI as having negative sentiment were also associated with one of the four negative emotions in EmoLex, and 88.82% of words which were rated as eliciting positive emotions were rated as having positive sentiment by GI. The dataset consists primarily of words associated with negative sentiment, as seen in Table 3.

Word Counts			
Anger	1247	Anticipation	839
Disgust	1058	Trust	1231
Fear	1476	Surprise	534
Sadness	1191	Joy	689
Negative	3324	Positive	3213

Table 3: Number of terms in EmoLex by their affective category

Further extending this work, the EmoVAD lexicon (2018) builds a dataset of 20,007 words annotated for their valence, arousal, and dominance on a scale of $[0, 1]$ [33]. Unlike EmoLex, which identified discrete emotion categories for each word, EmoVAD aims to place words in the circumplex emotion space. A large affective dictionary was created from the EmoLex, General Inquirer, ANEW, and Warriner et al. datasets − as well as 1000 frequently used hashtags from Twitter. EmoVAD's labels appear to be evenly distributed based on Figure 5, indicating the dataset contains a variety of terms with differing affect.

	EmoVAD		eANEW	
	Avg.	σ	Avg.	σ
Valence	0.50	0.216	5.04	1.28
Arousal	0.50	0.171	4.21	0.90
Dominance	0.50	0.170	5.19	0.94

Table 4: Distribution of word affect valence, arousal, and dominance labels

A best-worst scaling annotation method was implemented in Mohammad and Turney's CrowdFlower survey to help account for the variability in how human annotators interpret the emotion scales [36]. Because of the subjectivity of emotion models, one user might interpret the scaling of the circumplex domain differently than another. For example, a user might think a valence of 0.5 is "modestly happy", where another thinks of it as "generally very happy". However, the relative distance between annotations for a given annotator still reveals the user's opinion. To account for this, the HITs designed by Mohammad present a user with four words at a time, and asks the users to rate which word represents the affective dimension the most, and which word represents it the least. By asking for relative labels instead of asking users to put samples directly into the VAD space, such as in Warriner et al.'s experiment, both the number of annotator errors and the necessary level of prerequisite knowledge are reduced.

A third affective lexicon dubbed NRC-Affect Intensity Lexicon (2018) aims to measure the intensity at which each word elicits a given emotion, as opposed to simply categorizing words based on which emotions they elicit [34]. Affect intensity is measured on a scale of $[0, 1]$. Terms were selected from the hashtags from a body of emotion-annotated tweets, where the hashtag also appeared frequently in the Google n-gram corpus. Best-worst scaling was used once again in a process similar to that used for the EmoVAD survey. Roughly 6000 words were rated from CrowdFlower, with a median of four annotators per affect.

Words for which an affect is not present are given an intensity of 0. This causes the distributions of each affect to be clustered near 0. For the distributions labeled "scaled" in Figure 7, we drop all words with an affect of 0 for each dimension to better understand the properties of each affect. When removing the words for which an affect is not elicited, we observe that the affect labels are normally distributed between 0 and 1.

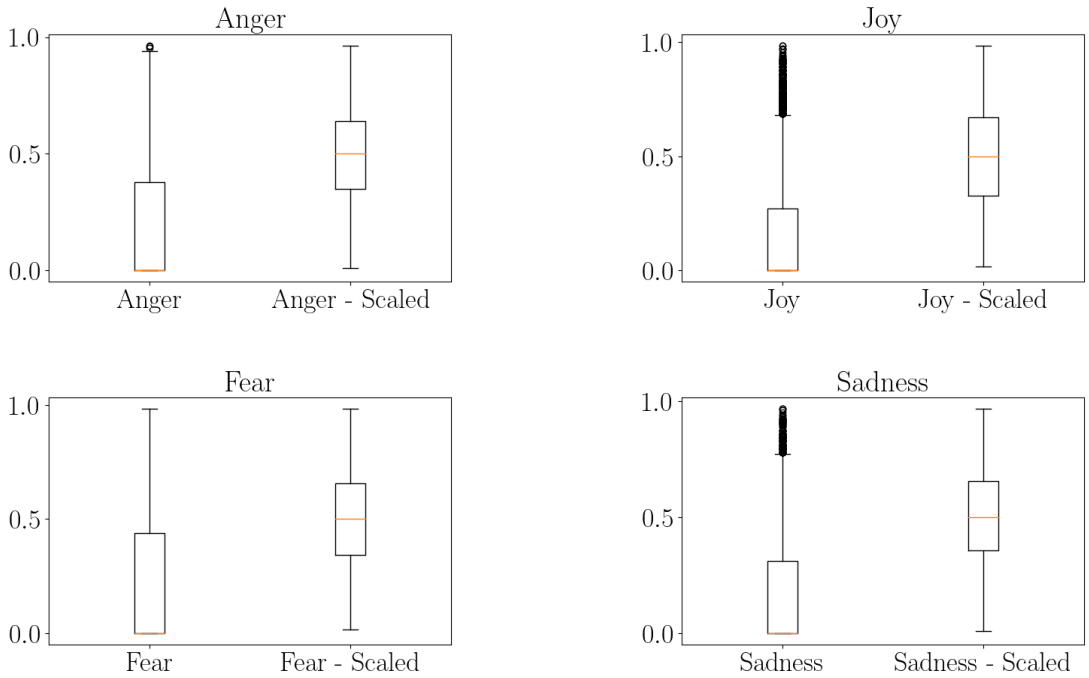


Figure 7: Distribution of affect intensity labels in EmoAff

In addition to the aforementioned affect and valence-arousal dictionaries, we also investigate the Multi-Perception Question and Answering lab's +/-EffectWordNet sentiment dictionary (2014) [11] [5]. This lexicon focuses on classifying terms as positive, negative, or neutral sentiment. Identifying neutral sentiment words is an important contribution of this work, as it can aide in filtering semantically irrelevant words in bag-of-words models. The 8,221-word dictionary is built first from 592 seed words, manually annotated for sentiment and classified into a WordNet lexical unit. From this, the graph nature of WordNet is used to identify words which are semantically similar to the seed words, which are then classified for sentiment in by a semi-supervised SVM.

[5] We refer to this dictionary as the MPQA dataset going forward.

	Positive	Neutral	Negative
Word Count	2298	439	4148

Table 5: Counts of sentiment words from MPQA

4.2 Affective Analysis of Social Media Commentary

Social media has become an ever-present component of our society, as platforms like Meta report billions of unique users each month [6] With this tool for sharing opinions, ideas, and life events being in the hands of individuals around the globe, it presents a unique opportunity to be able to measure public opinion regarding individual topics. As a result, the need for social media sentiment analysis has driven further experiments into understanding human emotion from text corpora.

Though social media sentiment analysis has existed nearly as long as social media platforms themselves, and experiments in the field have adopted a variety of emotion models, feature extraction methods, and prediction techniques [40], we focus specifically on intelligent sentiment analysis systems which aim to predict a text's affective qualities according to Russell's circumplex model of emotion. Preoţiuc-Pietro et al. set out to create a dataset of Facebook posts, annotated for valence and arousal, for the evaluation of machine learning systems aimed at continuous emotion prediction of social media discourse [45].

A series of 2,895 English-language Facebook status updates are labeled for valence and arousal by two psychologically-trained annotators. Inter-annotator agreement was very strong, achieving a Cohen's Kappa of $\kappa = 0.93$[7]. A variety of bag-of-words modelling approaches were evaluated for valence-arousal prediction. A weighted mean of all words in the text which also existed in the ANEW lexicon was used as a baseline. Similar approaches were taken using the Extended ANEW, MPQA, and NRC Hashtag Sentiment datasets. Preoţiuc-Pietro et al. achieve a maximum arousal correlation of 0.19 using features from Extended ANEW, and maximum valence correlation of 0.41 using the NRC hashtag dataset. These experiments demonstrate the power of word affect lexicons to be used directly to calculate emotion labels in text without the need for machine learning systems. However, this naive statistical model only exhibits weak correlation with expert annotations, indicating a need to investigate other predictive systems to be used in combination with features generated from affective dictionaries.

[6] https://s21.q4cdn.com/399680738/files/doc_financials/2022/q1/Q1-2022_Earnings-Presentation_Final.pdf

[7] Such a high correlation may be attributed to the fact that only two annotators participated in the experiment. Furthermore, we do not know if the annotators belonged to the same demographic, university, research lab, etc.

Pellert et al. took a temporal approach to modeling emotion from Facebook status updates, focusing not just on static labels but how valence and arousal states change over time in response to a stimuli [43]. Their model of emotion dynamics sought to analyze how people returned to a baseline valence/arousal state after an event by performing sentiment analysis on a user's status updates over time. A dataset of 17 million status updates from 114,967 unique users is used to model user affect. For each user, a timeline of status updates is created. The valence and arousal of each post is calculated using a weighted average of the V/A ratings of all words in the comment matching the Extended ANEW dataset [56]. Pellert et al. use the EmoVAD dictionary for cross-validation. No ground truth valence/arousal labels exist for this dataset, so there is no way to validate the consistency of these predictions. However, baseline valence and arousal states were calculated to be 5.88 and 4.13, respectively (on a scale of 0 to 10), which is found to be consistent with other dynamic affect modeling experiments on both online and traditional English text corpora.

SemEval is a natural language processing workshop which announces a series of challenge datasets (referred to as "tasks") annually [35]. SemEval tasks often focus on semantic understanding and sentiment analysis of language. One of the tasks at SemEval 2018 challenged NLP researchers to develop a model for predicting affect on Twitter posts. Given Twitter's short-form post format (240 characters in 2018), traditional sentiment analysis models struggle to extract meaningful information from so few words.

SemEval 2018 Task 1 included predicting affect intensity, emotion classification, and valence-arousal modeling [35]. A dataset of tweets was built by selecting submissions from 2017 and querying those which were close in semantic distance to the mood category descriptors (e.g. for angry, tweets which contained the words "miffed", "annoyed", "irritated", etc.) were selected, and evenly distributed across the eight mood categories. From this, a subset of 1,400 tweets were labeled for mood category, affect intensity, and valence-arousal using crowdsourced annotators from CrowdFlower. Best-worst scaling was used for continuous labels, as described in [33].

A wide variety of feature extraction and mood prediction techniques were used to evaluate this dataset. Of the methods presented by the SemEval 2018 teams, word embeddings and feature spaces built from affective and sentiment lexicons achieved the best performance across all tasks. For emotion lexicons, particularly those from the NRC lab (EmoAff, EmoLex, EmoVAD) were used to generate affective features from unlabeled text. This was the second best performing model, behind approaches relying on word embeddings [35].

More recently, transformer models have been used to great effect across a variety of NLP tasks [14] (see 3.2). Chiorrini et al. uses BERT to predict emotion and sentiment

from a series of Twitter posts [10] The use of pre-trained transformer models would indicate that a smaller set of emotion-labeled training data is necessary to achieve significant predictive performance. In the case of affect prediction, where developing high-quality annotations for datasets is cost-prohibitive and time-consuming, this could prove useful for a variety of emotion recognition tasks.

Chiorrini et al. aggregate a set of 1,600,000 tweets for this affect prediction task. These were labeled through an unsupervised approach, using emojis to determine mood and sentiment [10]. 430 manually annotated tweets are also included in the dataset. Tweets are first pre-processed, removing URLs, retweets, and user mentions. A variety of BERT derivatives of various parameter sizes were tested on classifying text as positive, negative, or neutral sentiment. They also evaluate model performance on mood classification across four categories: happiness, anger, sadness, and fear. BERT-base-cased was found to have the most robust performance in both tasks, indicating that capitalization is relevant for social media mood classification. The authors report 90% accuracy for mood categorization and 92% for sentiment prediction. Given the relatively small set of manually annotated data, this indicates that transformer models have potential to be effective at predicting emotion states from social media text.

5 Music Emotion Recognition

Music emotion recognition has been of particular interest in recent years in the music information extraction domain, as music recommender algorithms continue to optimize user discovery experiences on large music streaming services with ever-growing catalogs [13]. Automated systems for estimating the affect of a piece of music can aide in the automatic categorization of songs in large databases. We investigate four music emotion datasets created to assist the development of music emotion recognition systems, and then detail some of the approaches used to estimate musical affect.

5.1 Music Emotion Datasets

The work to develop a music emotion recognition system has created a need for high-quality datasets of songs, manually rated for their affective qualities. Surveys conducted using large numbers of human annotators aim to establish a baseline for the emotion elicited in the average listener for a given song. Because emotion is highly subjective, these surveys require many ratings for a given sample to yield statistically significant results. We focus on datasets which rate songs for continuous valence and arousal values, as it has been proven that valence and arousal ratings can be mapped to mood categories [8] [55].

AMG1608 is one such dataset providing valence and arousal rated samples of music. The goal of AMG1608 was to surpass previous datasets in both number of songs and number of annotators per song by leveraging crowdsourced annotation platforms [9]. Previous music annotation datasets were limited in scope, as the logistics and expense of conducting annotation surveys in traditional settings was prohibitive to large-scale labeling projects.

Songs described as "Western Contemporary" were selected from the All Music Guide. To create a dataset of songs evenly distribute across the valence and arousal space, Chen et al. first generate synthetic valence and arousal values for all songs in this genre. Last.FM, a popular music catalog and social media platform used by music enthusiasts for storing song metadata, allows users to add tags to songs. These user-annotated keywords are related to the song, and often consist of genre or mood descriptors. Synthetic emotion labels were generated from these tags using the process presented in [5].

These values were used to select a 1,608 song subset of AMG which were evenly distributed across the circumplex model. Annotators were presented with thirteen 30-second audio samples, where the subject was asked to listen to a given sample and plot the song's valence and arousal on an interactive circumplex graph. To verify the

quality of an individual HIT, one song was duplicated per annotator. If an annotator failed to rate the duplicate samples similarly, the ratings were not included in the data.

Figure 8: The annotation interface used for AMG1608 (from [9])

Overall, 665 annotators participated in the survey, with 15 to 32 annotations per song. The majority of ratings fell within the first quadrant of the valence/arousal space, despite the even distribution of songs selected from the synthetically generated valence/arousal labels.

One major shortcoming of many music emotion datasets is their lack of accompanying audio information. Failing to provide audio with a song list restricts the development of novel acoustic-feature extraction methods and testing audio-based emotion recognition models. Furthermore, only providing annotators with a 30-second clip of a song may impact their perception of the song, and lead to emotion ratings based on a limited or inaccurate understanding of the piece. However, the copyrighted nature of the majority of songs prevents researchers from including full songs in their surveys, much less publishing them freely as a part of their dataset. The MediaEval Database for Emotion Analysis in Music (DEAM) was created to address these flaws in existing music emotion dataset generation [2].

DEAM consists of 1,803 songs selected from royalty-free music platforms such as *freemusicarchive*, *jamendo*, and *medleyDB*. Valence and arousal annotations were gathered from Amazon Mechanical Turk following similar procedure as that outlined in the AMG1608 dataset. However, as opposed to presenting an annotator with a sample and then asking for a single valence/arousal label, DEAM gathers continuous

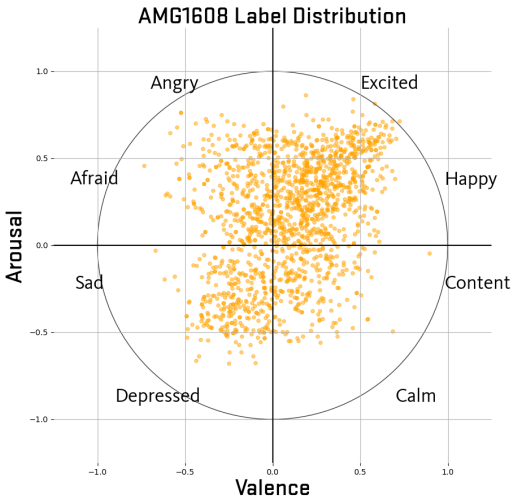
annotations every second for a 45-second excerpt of a sample[8]. By taking the average of these continuous ratings, a single valence/arousal label is produced and published with the dataset. Full song audio is also provided, as well as a set of pre-computed acoustic features.

Previous datasets selected songs from online hobbyist music datasets or royalty-free sources. In an effort to aide music emotion recognition experiments specifically for pop music, the PmEmo dataset takes a new approach to sample selection [61]. 1000 songs are selected from the Billboard Top 100, iTunes Top 100, and U.K. Top 40. After removing duplicates, the dataset consists of 794 songs considered to be popular between 2016 and 2017. 457 annotators rated the songs for valence and arousal as according to the Self-Assessment Manikin annotation interface described in [6]. 366 of these annotators were Chinese university students, potentially introducing a bias into these ratings due to a poor representative sample. However, unlike AMG1608 and DEAM which rely on crowdsourced annotations from Amazon Mechanical Turk, the PmEmo annotators participated in a survey conducted in a lab environment. Annotators rated the song on a per-second basis, as well as providing an overall emotion rating at the end of the 30-second clip. Electrodermal activity was also recorded, and was published in the dataset alongside continuous labels, discrete labels, and audio from the 30-second chorus sections presented to the annotators.

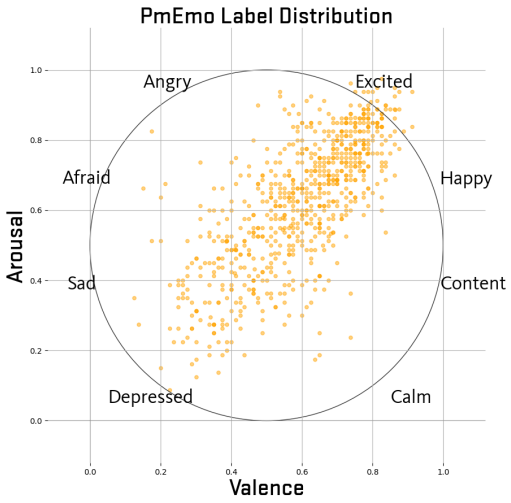
Researchers from Deezer make available a dataset of 18,644 songs selected from the Million Song Dataset[9] with synthetically generated valence and arousal labels [13]. Instead of relying on manual annotation, labels are estimated by tags from Last.FM in a process similar to that described in [5]. By averaging the known valence and arousal values of any tags which match the affective terms in Extended ANEW, the Deezer team is able to provide a music emotion dataset at previously unseen scale. It is essential to consider that these labels are not comparable to real human emotion annotations, as no manual verification is provided. However, the large size of Deezer's dataset relative to existing works make it a valuable tool in developing deep learning models for emotion recognition, which historically benefit from very large training sets.

[8]Continuous, in this context, refers to annotations given over a time interval, as opposed to a single valence and arousal label given for the entire sample.

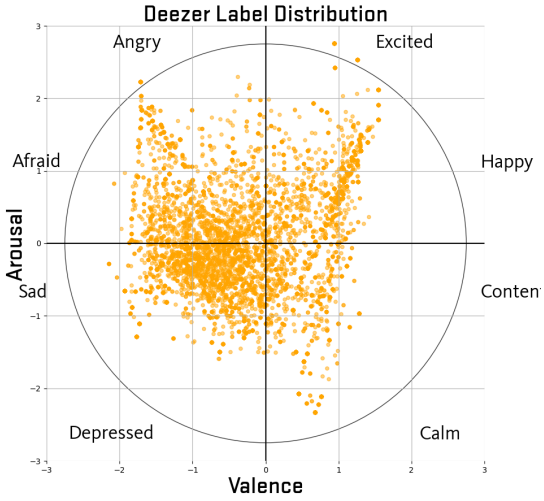
[9]http://millionsongdataset.com/



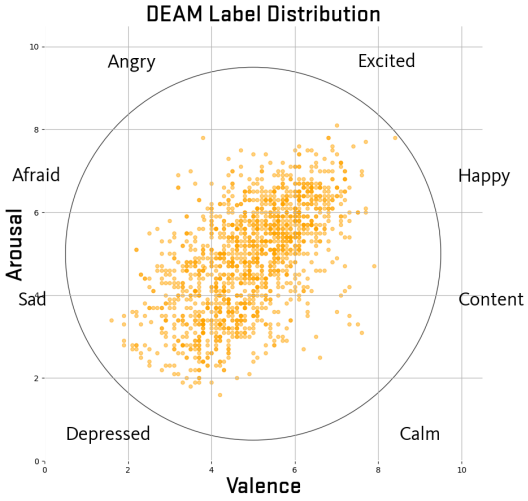
(a) AMG1608



(b) PmEmo



(c) Deezer



(d) DEAM

Figure 9: Circumplex graphs emotion labels for four music emotion datasets

5.2 Acoustic Features for Emotion Understanding

Historically, approaches for automatic music emotion recognition have relied on learning information from acoustic features derived from the raw audio of a song. Lu et al. designs an automatic mood classifier on a sample of 250 classical pieces, with mood labels manually annotated by domain experts [31]. Music mood classification was performed across four categories: *contentment*, *depression*, *exuberance*, and *anxiety*; loosely based on the quadrants of Russell's circumplex model.

Lu et al. identified acoustic features to translate the audio waveform into a symbolic representation for machine understanding [31]. Mode, intensity, timbre, and rhythm were identified as key details in a composition which determined music mood. The signal is divided into frames of 32 ms, and split into seven frequency bands using a Fast Fourier Transform (FFT). The authors use the spectrum sum and deviation to model intensity in a given frame. Spectral shape features represent brightness and spectral flux, both of which have been identified to represent variance in mood categorization. MFCCs have been used frequently in speech processing [29], and as such are also often used in music information extraction to represent timbre. Lu et al. contest this, instead opting to use octave-based spectral contrast for timbre representation. Future literature demonstrates the continued use of MFCCs for feature engineering in music information retrieval [23] [39].

The strength, regularity, and tempo of the rhythm for a given sample are identified as being emotionally relevant for music emotion recognition. To model this, each frame is divided into octaves after applying an FFT, and the amplitude envelope and onset sequence are calculated for each subframe. An onset curve representing overall rhythm is created using autocorrelation. Average tempo is derived from this autocorrelation curve. A Gaussian mixture model is used to predict one of four music mood categories based on the aforementioned acoustic feature space. A maximum classification accuracy of 86.3% is achieved on the test set using a combination of rhythm and timbre features.

However, limitations exist with the approach suggested by Lu et al. Clips which did not have inter-annotator agreement between three expert annotators were thrown out of the dataset. This potentially artificially inflates accuracy results by only including clearly defined samples. Furthermore, this dataset was constrained specifically to samples of classical music, and does not demonstrate generalization across other musical genres. Finally, the ground truth labels required expert annotation, which may be prohibitively expensive to demonstrate at scale.

To address some of these concerns, Wu and Jeng present a system for automatically determining affect across a dataset of 200 30-second song clips sourced from film soundtracks[57]. Instead of relying on expert annotators to form a cohesive opinion

on a track, the authors use manual ratings sourced from an online crowdsourced survey. Each sample had an average of 28.2 annotators, and the label most frequently assigned to a given sample is used to assign a track one of eight given mood categories: *sublime*, *sad*, *touching*, *easy*, *light*, *happy*, *exciting*, and *grand*.

Unlike Lu et al. who present their own acoustic feature extraction and engineering method, Wu and Jeng leverage four existing music feature extraction frameworks, yielding a total of 88 features. We note MARSYAS [52] and PsySound [28] as being of particular interest here, as they are used in future experiments [22]. In order to reduce the dimensionality of their audio feature set, the authors train an SVM on all features, and then greedily remove the feature with the worst F_1 score[10]. An SVM trained on the 29 best features achieves a cosine similarity of 0.73 between predicted and user-annotated labels across the eight mood categories.

Recent work in systems for determining affect directly from acoustic features has focused on explainability of emotion prediction systems. Chowdhury et al. approaches this problem by modifying the feature space such that it is composed of values which could be understood by a knowledgeable listener [12]. Referred to as mid-level features, this approach focuses on describing a piece of music as a series of values which represent perceptual concepts such as tonal stability, articulation, and rhythm. These features are usually connected to concepts derived from music theory, and are a more coarse-grained view of the signal than traditional acoustic feature sets. Chowdhury et al. applies a Convolutional Neural Network (CNN) to predict valence, arousal (denoted as energy), and affective values from a set of 110 movie soundtracks. They demonstrate average correlations of roughly 0.71, within 5% of a similar model predicting movie soundtrack affect on a more traditional set of acoustic features. Feature-importance analysis on these mid-level features can provide an explanation for model outputs, allowing a recommender system to provide context for mood-classification based suggestions.

5.3 Integrating Lyrics into Affective Models

Experiments using exclusively acoustic information to predict music emotion have not yet proven entirely effective. Specifically, regarding the use of low-level audio features, the so-called semantic gap limits the ability of raw acoustic information to explain the human perception of a song and their subsequent emotional response. It is possible that that music emotion prediction systems must be augmented with additional data sources in order to improve emotion recognition performance in any

[10]It was not clarified in the paper, but we believe the authors trained an SVM with each feature independently, and then ranked each feature according to the F_1 score of the predictions from the resulting single-feature model.

meaningful capacity [59]. In an effort to build more robust music mood classification systems, different feature spaces have been explored to either replace or complement acoustic features. Yang and Lee set out to develop a system to aide the task of manual emotion annotation. Their theory was that annotator fatigue could lead to inconsistent labels in the development of music emotion datasets. They set out to develop a music emotion prediction tool to guide annotators by providing context for the predicted mood category of a song to potentially reduce annotator fatigue and yield more robust datasets [58].

Yang and Lee compare three approaches for music mood prediction - acoustic features, lyric analysis, and a combined approach (referred to as a "fusion model"). To extract affective features from song lyrics, the authors use the General Inquirer framework for sentiment analysis on English text. The General Inquirer is a natural language processing tool consisting of a word affective dictionary, expert-annotated by psychologists, and a manually created ruleset for word disambiguation and counting in order to produce a set of features representing the emotive qualities of a given text [50]. While the General Inquirer does not yield direct valence/arousal/dominance values, the authors believe G.I. features can be used to predict these dimensions.

Compared against volunteer-annotated mood categories of 145 songs described as Alternative Rock, an SVM trained on acoustic features achieved correlations up to 0.90. Adding the General Inquirer feature vectors extracted from the song lyrics into the feature domain improved classification accuracy by 2.1%. Despite the limited sample size, both of annotators and songs, evidence exists that incorporating text-based features into music emotion prediction models can improve predictive performance.

The evaluation of lyrics as an input for automatic music mood classification systems is continued in Laurier et al.'s work [27]. Using 1000 pop-genre songs, Laurier et al. introduces a system for automatically generating a ground truth from Last.FM. By comparing the similarity of these tags to one of four mood category descriptors (*angry*, *happy*, *sad*, *relaxed*) using WordNet [32][11], songs are categorized by mood. These specific moods were chosen to be representative of the four quadrants of Russell's circumplex model [47], and this quadrant classification task is seen frequently in emotion recognition experiments. To validate the automatic labeling, human annotators reviewed the Last.FM mood tags for relevance to the song. 71.3% of automatically labeled songs were manually verified.

[11] https://wordnet.princeton.edu/

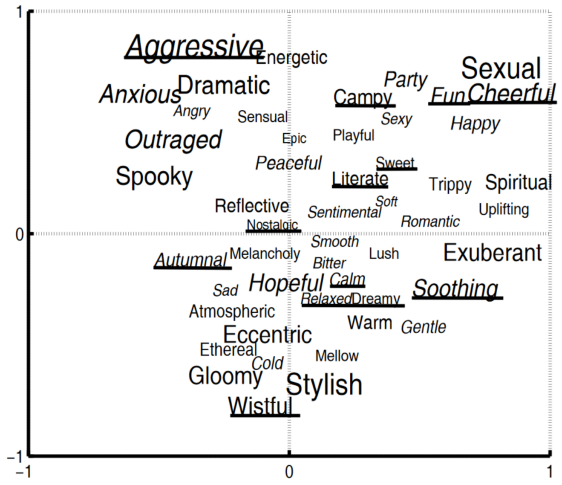


Figure 10: Last.FM tags automatically mapped to valence/arousal values (from [55])

Using the aforementioned low-level acoustic feature sets - MFCCs, spectral centroids, tonal and temporal descriptors, and onset rate - a baseline quadrant classification rate of 89.8% is achieved. Various approaches were tested for lyrics-only classification models. A k-Nearest Neighbor model was trained, using Lucene search as a distance metric between various sets of lyrics, which achieved a maximum average accuracy of 62.5%. Latent Semantic Analysis was tested to reduce the relatively high dimensionality of using entire lyric documents as a feature space, yielding 61.3% accuracy. Finally, the 100 most discriminant terms for each mood category were determined by measuring the frequency at which a word appeared in a given class of document. An SVM was trained on the frequency at which these top 100 words appeared, and achieved 80% classification accuracy. This semi-supervised approach to music-relevant affective term identification and lyric sentiment extraction demonstrates the ability for unigram analysis to accurately classify music mood. By combining this language model difference approach with acoustic features, a separate SVM model was able to achieve 92.4% average accuracy, with a maximum accuracy of 98.3% for songs labeled as angry.

The decision to use Last.FM tags for automatically generating mood labels came from earlier experiments in the use of social media data for mood and theme classification [5]. Bischoff et al. uses the All Music Guide as a ground truth for mood and theme labels. The information from *AllMusic* is manually curated by a group of knowledgeable hobbyists, as opposed to Last.FM's open community nature, and provides a good source of comparison for the use of Last.FM tags for mood classification. A subset of the *AllMusic* database was chosen, consisting of 178 mood categories and 73 theme categories. These 178 moods are then reduced to five: *passionate*, *cheerful*, *brooding*, *silly*, and *aggressive*. By only selecting songs which were tagged on *AllMu-*

sic as matching exactly these descriptors or one of their manually chosen synonyms, the original dataset selection was reduced from 5,770 songs to 1,992. Accuracy of 53.7% is achieved using a Naive-Bays model trained only on Last.FM tags, indicating a weak but present relationship between Last.FM tags and *AllMusic* mood labels.

Hu and Downie set out to establish a framework for feature extraction and mood prediction from song lyrics by first comparing various approaches to lyric sentiment analysis [23], and then identifying cases in which the performance of lyric-only models exceeded those of acoustic feature models [22]. The first of a series of music emotion papers from these authors demonstrates the use of a bag-of-words model for sentiment extraction on lyrics [23]. By using the wordlist used for General Inquirer [50], as well as ANEW [6], individual words are rated for their affective value. From these individual words, a list of summary statistics is generated based on the distribution of valence, arousal, and dominance ratings of the words in a given text. Text stylistic properties were also extracted - including word frequencies, use of interjections, and punctuation. A dataset of 5,296 songs were labeled using Last.FM tagging, WordNet distance comparison to mood category descriptors, and manual verification by two human annotators, over a total of 18 mood categories. Hu and Downie choose to use MARSYAS for acoustic feature extraction.

The best lyric-only model combined stylistic text features with an ANEW-based bag-of-words approach, achieving a 63.7% classification accuracy. No model which used General Inquirer features outperformed the ANEW bag-of-words model. It is important to acknowledge that while this accuracy may not seem like an immediate improvement over the General Inquirer fusion model used in [58], Hu and Downie's experiment is classifying over a much larger database, with many more mood categories than those used in prior works. Similarly, their audio-only accuracy of 57.9% is significantly lower than that seen of prior experiments, but is reasonable when considering the increased complexity of the prediction task. A fusion model achieves the experiment's best performance, with an accuracy of 67.5%, by combining features from all aforementioned lyric processing methods as well as the acoustic features generated by MARSYAS.

This was one of the first experiments in which lyric-only models outperformed acoustic features for mood classification. Hu and Downie further investigate this in [22]. The previous 18 mood categories are mapped to the Russell circumplex model, demonstrating the compatibility of their work with existing emotion classification systems in the domain. The performance of a series of lyric-based feature sets − including content word counts, General Inquirer automated analysis, bag-of-words using the General Inquirer lexicon, bag-of-words using ANEW, and text stylistic features − as well as MARSYAS audio features, were analyzed for each class. Audio features were found to significantly outperform any lyric based features for identifi-

cation of songs categorized as calm. In the case of songs labeled romantic, angry, cheerful, aggressive, anxious, hopeful, or exciting, models trained on lyric features were found to outperform their audio-based counterparts, and are within 5% accuracy in all other mood classes. Hu and Downie's work established a precedent for the use of natural language processing and sentiment analysis for the purpose of music emotion recognition.

As well as contributing a dataset of synthetically labeled songs (see 5.1), researchers from Deezer evaluate a series of traditional and deep learning approaches for music emotion prediction from a variety of sources [13]. They compare three approaches, one learning directly from lyrics using a 1-D convolution and Long Short-Term Memory model, one using an audio mel-spectrogram for its feature space and using a similar CNN + LTSM model, and a fusion model combining the intermediate outputs of the aforementioned models, without applying an LTSM layer to the audio-based input. Many other deep learning based models were compared for lyric-only analysis, however none achieved noteworthy performance. No deep learning approach to lyric analysis was able to outperform the baseline feature-engineering approach described in Hu and Downie's work [23] [22]. However, the CNN + LTSM model was able to outperform traditional acoustic feature engineering models. Best performance was achieved with the previously described mid-stage fusion model, achieving R^2 scores of 0.22 for valence prediction and 0.23 for arousal prediction.

5.4 Direct Emotion Prediction from Lyrics

Recent efforts have also attempted to develop lyric-only emotion recognition models. Çano et al. create a dataset of mood labeled music by using a bag-of-words approach on song lyrics to synthetically generate mood classifications based on the four quadrants of Russell's Circumplex model[8]. The ANEW lexicon was used to determine the valence and arousal values of individual words, from which a song-level label was taken by finding the sum of valence and arousal ratings of all affective terms identified in the lyrics. The resulting valence/arousal label was then categorized into a mood based on its respective circumplex quadrant. To validate their lyric-to-label approach, mood labels were aggregated from the All Music Guide in a process similar to [5]. The lyrics model was able to achieve 74.25% classification accuracy relative to the AMG mood tags.

Building further on the lyrics-only approach to emotion prediction, Agrawal et al. use a transformer approach to circumplex-quadrant mood classification [1]. `xl-net` is chosen for this lyrical analysis task, as it supports longer-form text inputs (see 3.2). A fine-tuned `xl-net` model is able to achieve 94.78% classification accuracy on the MoodyLyrics dataset, and 88.89% accuracy on the traditionally manually anno-

tated MER dataset (n=180). The classification performance of the transformer-based lyrics-only model demonstrates the potential for this new architecture to be used successfully in music emotion recognition tasks. It also demonstrates that lyrics-only models can outperform even recent audio-based approaches in quadrant classification, whether those models be deep-learning or feature engineering based. Finally, it demonstrates the ability to extract semantic information from music lyrics without the need for feature engineering or the use of manually-rated affective lexicons.

6 Data Collection

In order to test the use of social media conversations as a feature space for music emotion prediction, we first must create a dataset of online discourse to use in our model evaluation. We detail the process for data mining Reddit, Twitter, and YouTube for commentary surrounding the songs featured in four music emotion datasets.

We choose four music affective datasets to test for social media affective prediction: AMG1608 [9], DEAM [2], PmEmo [61], and Deezer's 2018 dataset [13]. Each of these datasets provides an artist name, song title, and accompanying valence and arousal labels for each song. Details on the annotation methodology is provided in 5.1.

From these, we extract the artist names and song titles to be used in our queries. In total, our dataset consisted of 22,827 songs, of which 4,179 are manually annotated and 18,648 are synthetic. Duplicates were not removed, as each dataset will be evaluated independently of one another due to differences in valence-arousal scaling.

Dataset	**Songs**	**Label Type**	**Scaling**
AMG1608	1608	Crowdsourced	$[-1, 1]$
DEAM	1803	Crowdsourced	$[0, 10]$
PmEmo	768	Lab Survey	$[0, 1]$
Deezer	18,648	Synthetic [12]	$[-3, 3]$

Table 6: The label ranges of our four music emotion datasets

From these songs, we build a dataset of social media conversations. We chose to pull comments from Reddit, Twitter, and YouTube. All three of these platforms are large, popular social media websites with music subcultures, where individuals converse about artists, songs, and concerts. In the case of YouTube, many use it as a platform to share and listen to music as well. Because many users will be posting comments immediately after listening to a sample, conversations on this platform are of particular interest from an emotion recognition perspective.

From these platforms, we query for a given artist name and track title. We choose to build queries to strictly include the full artist name and full track title in the title of the submission to ensure any comments are directly relevant to the song itself. Queries are built as artist name followed by track title. Each individual word is wrapped in quotation marks to require exact matching.

For YouTube and Reddit, the search procedure is similar across the two platforms. Once a query is made, a list of submissions matching the request are returned. From

[12] Deezer labels were standardized on $[-1, 1]$. The absolute range is approximately $[-3, 3]$.

this, we pick the ten highest rated submissions, sorting by "likes" and "upvotes" on Reddit and YouTube, respectively. All comments in response to one of these submissions are aggregated. Submission titles and body texts are also recorded, including the original post for Reddit, and the video description for YouTube.

Twitter, being a platform focused more on short-form text posts, does not follow the submission and comments format that the other platforms do. We take a slightly different approach to gathering Tweets. In this case, we pull the top 100 top-level Tweets from a given query. Replies are not recorded in order to avoid a duplicate comments, as a reply to one tweet may also occur in the query as a relevant tweet. Retweets are also not included for a similar reason.

Originally, we stored this information as a series of Comma Separated Values (CSVs), organizing comments by a query index, submission index, and comment indices. However, this was problematic when a song which did not have any submissions returned from the query for a given platform. For example, a song may have data returned from Reddit and YouTube, but have no results from Twitter. In this case, the song would simply be dropped from the Twitter subset of our dataset, leading to inconsistencies between platform datasets. As a result, we decided to pivot to storing discourse in a JSON format, allowing for the structure of songs, submissions, and comments to be maintained without resulting in songs being dropped from the dataset due to a lack of relevant discourse over a specific platform.

Our four music emotion datasets consist of valence and arousal labels generated either synthetically or manually by surveying annotators. Differences in the label ranges used across these datasets makes any direct comparison difficult, so we scale valence and arousal labels to $[-1, 1]$.

		Valence		Arousal	
		Avg.	σ	Avg.	σ
Actual	AMG1608	0.102	0.278	0.140	0.351
	PmEmo	0.597	0.162	0.622	0.185
	DEAM	4.904	1.174	4.814	1.282
	Deezer	-0.067	1.058	0.196	0.961
Scaled	AMG1608	0.025	0.340	0.097	0.437
	PmEmo	0.198	0.411	0.205	0.416
	DEAM	-0.028	0.345	-0.011	0.394
	Deezer	0.126	0.573	0.126	0.573

Table 7: Summary statistics of music emotion dataset valence and arousal labels

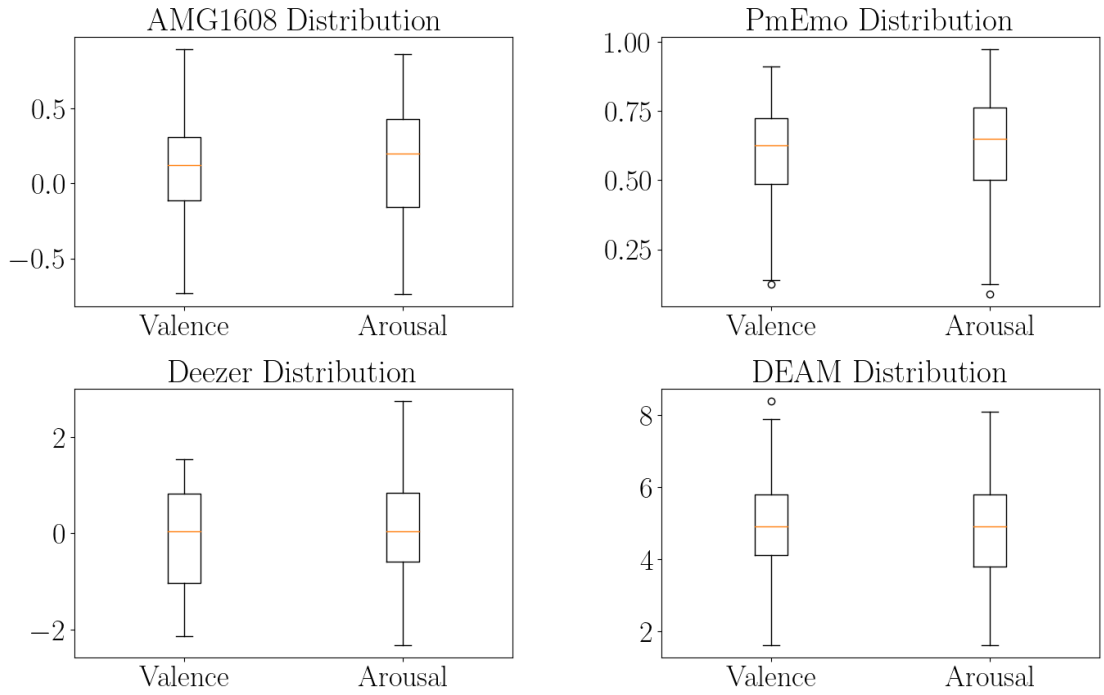


Figure 11: Box-and-whisker plots of music emotion dataset labels

We observe that AMG1608, PmEmo, and Deezer all have a slightly positive average valence. PmEmo, in particular, is centered roughly around (0.2, 0.2) in the first quadrant of the circumplex model. We therefore expect these songs to be more positive and high-energy on average. Given PmEmo was created from a list of pop-genre songs, it may be that these types of songs tend to exhibit moods which fall into the first quadrant of the circumplex model more frequently, expressing positive valence and arousal. However, with a small sample size of 768 songs, it is difficult to make any generalized conclusions.

The valence and arousal labels of these four datasets have fairly high variance, indicating the songlists contain tracks of a wide variety of affects. This is with the exception of valence labels for Deezer. Despite the high variance, and slightly-positive mean valence matching the other datasets, the distribution in Figure 11 demonstrates a lack of strongly positive songs.

6.1 Social Media Dataset

As a part of this work, we contribute a framework for building music discourse datasets from social media sources. The following data was pulled between October and November 2021, scraping social media posts for the aforementioned songs

listed in the four music emotion datasets. If a query for a song yields no submissions, that song is removed from the dataset. Table 8 shows the number of songs included in our dataset per platform

Because Deezer contains the greatest number of songs in its dataset relative to our other songlists, its songs yield the greatest number of total submissions. However, Deezer also has relatively low yield percentages, with only 30% of songs returning data from v.s. 89% and 86% yields from AMG1608 and PmEmo, respectively. Reddit yields on songs from DEAM are also low, with only 11% of songs included in the Reddit subset. However, YouTube returns 84% of DEAM songs. Based on these results alone, it would seem that YouTube has a more diverse set of music-related discourse. However, the less clearly defined gap between the number of Reddit and YouTube results across other datasets makes it difficult to validate this claim.

		Songs	Submissions	Comments	Yield Rates
AMG1608	Reddit	1431	9779	129722	89%
	YouTube	1592	11413	217093	99%
	Twitter	822	1266	5726	51%
PmEmo	Reddit	627	4157	103398	86%
	YouTube	730	6062	121546	95%
	Twitter	331	540	2699	43%
DEAM	Reddit	211	846	15563	12%
	YouTube	1518	5439	53342	84%
	Twitter	86	170	2442	5%
Deezer	Reddit	11915	53823	705406	64%
	Twitter	6116	9601	62399	33%

Table 8: Sizes of our discourse datasets

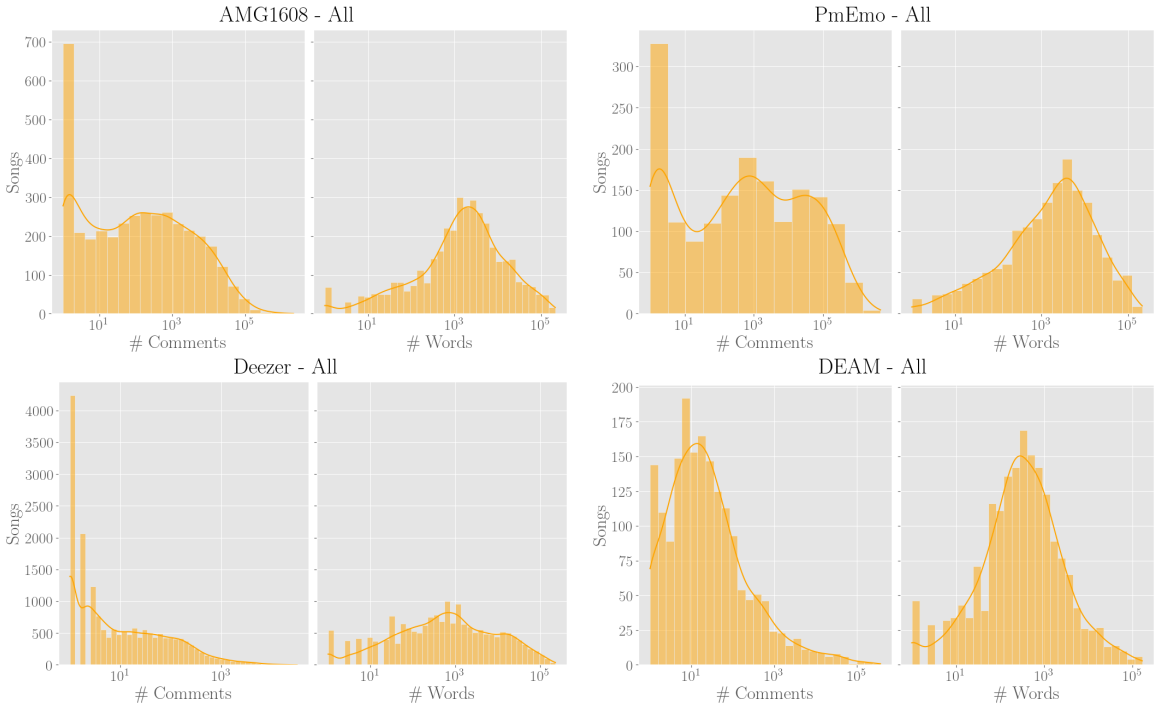


Figure 12: Comment and word distributions of our discourse datasets

		Comments		Words	
		Avg.	σ	Avg.	σ
AMG1608	Reddit	80.67	154.30	2400.83	69.06
	YouTube	135.01	57.71	2128.72	33.66
	Twitter	3.56	7.85	51.07	14.53
PmEmo	Reddit	136.59	218.74	3810.48	56.81
	YouTube	160.56	63.92	2172.13	44.13
	Twitter	3.57	7.26	46.00	15.24
DEAM	Reddit	8.68	62.88	264.43	61.93
	YouTube	29.75	40.12	447.84	32.92
	Twitter	1.36	3.44	10.99	14.79
Deezer	Reddit	37.84	111.21	1220.55	69.66
	Twitter	3.35	11.13	50.02	16.83

Table 9: Distribution of comments and words in discourse datasets

Songs from the PmEmo dataset tend to have more discussions on these three

platforms than songs from our other datasets. This may be attributed to the relative popularity of the songs in PmEmo, as the top 100 pop genre songs are selected from a collection of music popularity rankings [61]. It would follow that conversations surrounding widely popular songs are more likely to appear on these social media platforms. This phenomenon appears to be exacerbated with Reddit in particular, where the average number of comments per PmEmo song is 69% greater than that of AMG1608, the next most prolific dataset.

Scraping our platforms for conversations about DEAM songs yielded relatively few results. Despite 84% of songs returning a submission on YouTube, the average number of comments per song is significantly lower than for our other two YouTube datasets. We observe this across all queried platforms, with queries for DEAM song discussions being relatively sparse, even in the cases in which a submission for a song is identified in the first place. Because DEAM selects songs from royalty-free music libraries, it may be that the songs in this dataset are not as popular as those in other datasets, resulting in them not being discussed as frequently.

Average number of words in the comments pertaining to a song seems to be primarily a function of the number of comments in a data source. Interestingly, total word counts were generally greater from our Reddit selections than YouTube, with the notable exception of the DEAM dataset. This is especially evident in the case of PmEmo, where the average number of YouTube comments is greater than that of Reddit comments, despite the average number of words in the Reddit subset being nearly double that of YouTube. This would indicate that music conversations on Reddit tend to be longer than YouTube. We observe that Twitter has drastically lower word count averages than other data sources, however this is a result of the 280-character limit imposed by Twitter on all comments and submissions.

The average number of comments per submission provides further evidence of the popularity of songs in the PmEmo dataset. Despite having as many or fewer posts per song than AMG1608 songs, submissions regarding songs from PmEmo have longer discussions with more comments and longer comments in each post. Likewise, songs in the DEAM dataset have significantly fewer submissions per song, and less discussion within each submission, likely due to the songs in this dataset being more obscure to these social media communities.

		Submissions / Song	Comments / Submission	Words / Submission
AMG1608	Reddit	6.31	13.25	394.80
	YouTube	7.15	19.02	299.94
	Twitter	0.80	3.91	64.30
PmEmo	Reddit	6.02	24.85	694.03
	YouTube	8.16	20.05	271.29
	Twitter	0.74	4.21	63.81
DEAM	Reddit	0.49	16.53	559.21
	YouTube	3.35	9.76	147.61
	Twitter	0.09	4.30	106.47
Deezer	Reddit	3.37	12.98	422.67
	Twitter	0.53	5.19	95.84

Table 10: Relative averages of submissions, comments, and words

In our analysis, we exclude any data from YouTube for the Deezer dataset due to incomplete data collection. Future work should incorporate YouTube comments for Deezer songs.[13]

[13]YouTube's public API provides access to user comments, however it imposes a daily quota of 10,000 units. Assuming an average of 20.05 comments per YouTube post (the average for PmEmo), querying a single song costs approximately 310 units, meaning we can only query 32.25 songs per day. Pulling all YouTube comments for the Deezer dataset would take roughly 1 year and 6 months. Our request for a quota increase is pending review from Google.

7 Approach 1 - Bag of Words Model

We aim to predict valence and arousal labels directly from social media discussions about a song. To accomplish this, we model the valence and arousal of a specific comment, and average these comments to determine the average emotion elicited by that song. We evaluate two approaches for this comment-level modelling. First, we test a naive aggregation method by comparing the words contained in a song's comments to known valence-arousal labels for affective terms and taking an average. We then extend this approach to a collection of valence, arousal, dominance, affect, and sentiment dictionaries to extract emotive features from a series of comments. These features are then used to train a model to predict music valence and arousal values.

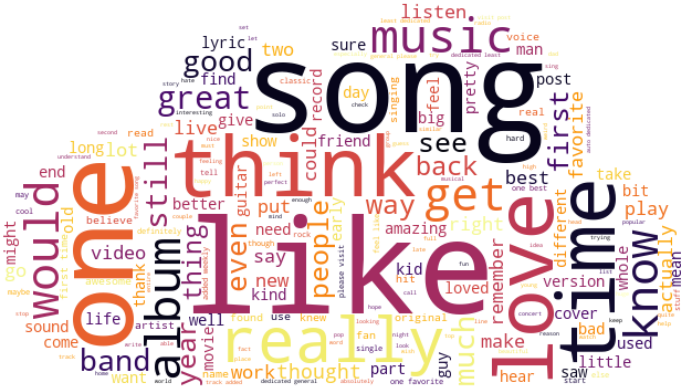


Figure 13: The 100 most frequent affective terms in our social media datasets

To generate affective features from social media comments, we rely on the five affective dictionaries defined in 4.1.2. Each provides a list of unigrams, manually annotated from crowdsourced surveys for emotive qualities. This provides us with a series of wordsets labeled with valence, arousal, dominance, affect presence, affect intensity, and sentiment.

7.1 Aggregate Word Model

Using a similar approach as that described in [8], we generate music valence and arousal labels naively from the aggregate valence and arousal of affective terms identified in comments. However, unlike the MoodyLyrics experiment, we directly generate valence and arousal labels for each song, as opposed to mood categorization by valence-arousal quadrant classification. We compare the EmoVAD and Extended ANEW wordlists for this task, as each dictionary takes a different approach to unigram emotion annotation.

To generate a song-scale valence and arousal label, we create a bag-of-words model for each song by combining all comments. Stopwords are removed using the `nltk` stopwords list[14]. We strip each comment for URLs, phone numbers, deleted comments, or comments which contain no words. The resulting tokens are then lemmatized to match the terms in our affective dictionaries. Duplicate words are not removed, as we consider the number of times a specific term occurs to be semantically relevant. The resulting list of comment words are then matched against a dictionary of terms with valence and arousal labels, and the average of all valence and arousal values for any matched words is taken as our song-level valence-arousal label.

		Valence		Arousal	
		Correlation	R^2	Correlation	R^2
eANEW	AMG1608	0.06	-0.37	0.07	-1.33
	PmEmo	0.11	-0.41	-0.03	-0.92
	DEAM	0.01	-2.55	-0.06	-0.66
	Deezer	0.06	-0.18	0.0	-1.07
EmoVAD	AMG1608	0.04	-0.58	0.13	-0.3
	PmEmo	0.09	-0.40	-0.10	-1.24
	DEAM	0.06	-2.31	0.11	0.41
	Deezer	0.05	-0.09	0.0	-0.76

Table 11: Pearson's Correlations and R^2 values for our bag-of-affective-terms model

The resulting valence-arousal averages do not correlate well with our target values. Across every dataset, R^2 is always negative, indicating a very poor fit between our baseline estimations and either real or synthetic (in the case of Deezer) music emotion labels. However, there are some cases in which these averaged values are slightly more representative of our labels than others. Specifically, when using Extended ANEW

[14] https://gist.github.com/sebleier/554280

features to generate valence labels in PmEmo songs, and EmoVAD for arousal estimation in AMG1608, PmEmo and DEAM. Each of these cases demonstrate a greater R^2 relative to other experiments, as well as having a correlation greater than 0.1. Though these results may still be weak, they deserve further investigation.

We apply this bag-of-words modelling technique to individual social media sources, focusing on AMG1608 and PmEmo as our model consistently achieves better performance on these datasets.

		eANEW		EmoVAD	
		Valence	Arousal	Valence	Arousal
AMG1608	Reddit	0.08	0.03	0.10	0.05
	YouTube	0.08	0.17	0.06	0.06
	Twitter	0.08	-0.03	0.08	-0.10
PmEmo	Reddit	0.04	0.10	0.04	0.15
	YouTube	0.04	0.13	0.05	0.32
	Twitter	0.05	0.07	0.07	0.13

Table 12: Correlations of bag-of-affective-terms model for each social media source

Comments from YouTube are consistently more accurately able to estimate music arousal labels, specifically in the AMG1608 dataset, achieving correlations as high as 0.32 when using the EmoVAD lexicon. These edge cases likely contributed to the less pronounced performance uplift on arousal features on AMG1608 and PmEmo in Table 11. This indicates that the comments contained in our YouTube dataset may contain more affective terms, or a greater variance in terms which are useful for emotion recognition. However, the larger number of YouTube comments in our dataset (see Table 9) is a potential cofactor here. The difference in affective intensity between each social media source should be considered going forward.

Neither affective lexicon outright outperforms the other for music emotion recognition through naive averaging. Though values generated from the EmoVAD dictionary achieve the best performance for arousal prediction, this is only in one specific case. Furthermore, Extended ANEW features appear to outperform those from EmoVAD for valence prediction of AMG1608 and PmEmo songs. Going forward, we include both datasets in our analysis.

7.2 Feature Engineering Approach

We use this bag-of-words approach to design a feature extraction system for the recognition of musical affect from social media comments. For each of our three social media datasets, we append every comment related to a given song, and then tokenize this wordset into individual words and a count of their occurrences. These tokenized wordsets are then compared against each of the five aforementioned affective dictionaries independently, taking words which appear both in the song's discourse and the affective wordlist. For EmoLex, EmoVAD, EmoAff, and Extended ANEW, we take the least affective word, the most affective word, an average of the emotive values of all terms, and the standard deviation in affect for each of the metrics measured for each word in the wordlist. For example, the EmoVAD subset of features would calculate the minimum, maximum, mean, and standard deviation in valence, arousal, and dominance for all words which are measured in the EmoVAD dataset and appear in the comments for a given song.

In the case of MPQA, no such continuous features exist. Instead, words are labeled as positive, negative, and neutral. We assign positive words a value of 1, negative words a value of -1, and neutral words to 0. We then apply the same feature engineering method as described previously, taking the mean and standard deviation of our sentiment wordcounts. Minimum and maximum sentiment are included as well, however these features are likely only relevant in the case where no positive words, or no negative words, are identified in the corpus. In every other case, they would default to a minimum of -1 and a maximum of 1.

In total, each wordlist generates a number of features representing the affective qualities of our music discourse corresponding to the number of features in the wordlist's own annotations, detailed in Table 2, multiplied by the number of summary statistics used to analyze each feature. We combine these individual wordlist analyses to compute a total of 324 features per song, per social media source.

We also investigate the union and intersection of these social media sources. Not every song appears in the discussions on every social media source, as demonstrated in Table 8. Furthermore, not every song has a considerable amount of discussion on platforms even if it does occur, shown in Figure 12. We note in Figure 12 the distribution of comments for a given song is bimodal in the case of AMG1608 and PmEmo, and heavily right skewed in the case of Deezer and DEAM. This would indicate that there are a significant number of songs in our datasets with very little conversation.

To account for this, we investigate two methods for taking the intersection of our three social media source discourse subsets. First, we take the union of all comments from all social media platforms before performing feature extraction, leaving us with 324 features per song, but with a greater number of total words and comments (de-

noted $\cup$). However, the baseline experiments in Table 11 show that there may be some semantic differences between each social media source, and that the source of the discourse itself may be a relevant feature. We also investigate another method of taking our dataset intersection by inner joining the Reddit, YouTube, and Twitter feature sets after feature extraction, resulting in the same number of songs, but 972 features (denoted $\cap$).

There are multiple tradeoffs to consider when taking the intersection of our datasets. First, we exclude a significant portion of our datasets as many songs have conversations occurring on Reddit and YouTube for example, but not on Twitter. We may risk introducing bias into our models as well, by only selecting songs which are of a genre or property which is correlated with popularity on Reddit, YouTube, and Twitter. Furthermore, in the case where all comments are appended prior to feature extraction, introducing too many comments into the affective averages risks our feature sets converging on the mean valence and arousal for the given wordlist. However, removing songs with too little online presence may reduce outliers or cases where one particularly negative or positive comment ends up being the only representation a song receives in our model. Furthermore, increasing the number of available features may improve a model's ability to properly delineate sample.

	AMG1608	PmEmo	Deezer	DEAM
# Songs	822	331	6116	86

Table 13: Number of songs in each intersection feature set

7.2.1 Baseline: Linear Regression

We test a variety of models for predicting music valence and arousal targets from the affective term feature sets. We start by applying a simple linear regression model to the task. Using implementations from the scikit-learn library of machine learning functions [7], we set any missing features to 0, to indicate that no words were present which matched that wordlist in the given social media source. Valence and arousal labels are clipped to a range of $[-1, 1]$ by applying min-max scaling to ensure consistency between the four music emotion datasets. We use a randomized test-train split of 0.80 and 0.20, respectively.

		Valence	Arousal		Valence	Arousal
Reddit	AMG1608	0.13	0.22	PmEmo	0.05	0.21
	DEAM	0.14	0.15	Deezer	0.16	0.10
YouTube	AMG1608	0.41	0.52	PmEmo	0.11	0.22
	DEAM	-0.03	0.03			
Twitter	AMG1608	0.12	0.16	PmEmo	0.05	-0.09
	DEAM	-0.30	0.06	Deezer	0.01	0.01
$\cup$	AMG1608	0.12	0.16	PmEmo	0.19	0.09
	DEAM	-0.19	0.19	Deezer	0.02	0.00
$\cap$	AMG1608	0.11	0.21	PmEmo	0.43	0.18
	DEAM	0.09	0.12	Deezer	0.07	0.35

Table 14: Pearson's correlations of a linear regression model's predictions

This set of experiments demonstrates that there is significant variance in each social media platform's music conversations and how well they correlate to the emotion elicited by the song. It appears that features generated from YouTube comments provide the strongest source of information for music emotion extraction from the three platforms tested here. This is consistent with the findings from Table 12. However, the best-performing models for valence prediction on Deezer and DEAM both were using Reddit features. In the case of Deezer, a lack of data from YouTube means it is unknown if a YouTube based feature set would outperform our Reddit model. As for DEAM, this dataset seems to be an outlier to the otherwise consistent performance of YouTube features for emotion label prediction. However, as shown in Table 8, there are significantly fewer overall comments for songs from DEAM than for other datasets, lending less credibility to any performance comparison between DEAM predictive models and other datasets.

We observe that models based on Twitter features tend to perform significantly worse than their corresponding Reddit and YouTube models. This is likely a result of the significantly fewer source comments available in the Twitter subset.

Taking the union of these three social media sources yielded poor performance, with Pearson's correlations between -0.19 and 0.19. This approach never outperforms all three source-specific models in both valence and arousal prediction. Because we observe no improvement over models with less source data, social media source union feature sets will be excluded in our future experiments.

Interestingly, the intersection of our sources performed significantly above that of our individual source models, trading with our YouTube model for highest correlations with respect to valence or arousal. For example, in the case of valence prediction for songs from PmEmo, an intersection feature set achieves a correlation of 0.43 versus the YouTube model's 0.05. Between the poor performance of union feature sets and the varied performance of source-specific feature sets we can conclude that, in the context of music emotion recognition, the source of an individual comment or set of comments impacts the affective qualities of that comment. Preserving the separation between comments across social media platforms at the feature level may improve the performance of music emotion prediction models.

7.2.2 Model Comparison

We evaluate three types of popular machine learning approaches for this task, testing a support vector machine, a k-nearest-neighbors approach, and three separate ensemble methods. The same feature sets as described in 7.2 and 7.2.1 are used, excluding union feature sets.

Our feature space is of high dimensionality, with over 900 features in the case of our intersection feature sets. This potentially impacts both model performance and training speed. Many dimensionality reduction techniques rely on selecting a subset of the feature space which best explains the variance in the dataset. Methods like Principal Component Analysis (PCA), however, reduce dimensionality while maintaining the greatest possible variance between components using Singular Value Decomposition. By specifying a desired percentage of explained variance to maintain, `scikit-learn`'s implementation of PCA will return the minimum number of components necessary to represent the original feature space without reducing explained variance below the previously set threshold [7].

To test if our feature space would benefit from PCA, we design a small A/B experiment to compare the performance of a random forest model on the AMG1608 dataset on our original features versus the reduced feature space. We choose to test the intersection feature set as well as YouTube features, as those feature sets resulted in the best performance from our linear regression model. We test PCA with an explained variance of 95% using a random forest model for prediction.

		YouTube		∩	
		Valence	Arousal	Valence	Arousal
Baseline	AMG1608	**0.53**	**0.61**	**0.41**	**0.60**
	PmEmo	**0.60**	**0.39**	**0.58**	**0.38**
PCA	AMG1608	0.44	0.57	0.38	0.48
	PmEmo	0.59	0.33	0.53	0.36

Table 15: Effect of PCA on random forest model performance

In every test case described in Table 16, applying PCA yields worse correlations than using the whole feature space. In the future, with a sufficiently robust model, PCA may be considered to improve runtime performance. However, for our experiments, we will not use dimensionality reduction.

Because these features are derived from affective wordlists of varying coverage of our comment datasets, there exist cases in which a song has a set of features for one wordlist, but not another. For example, a song may have sufficient comments to match one or more words in the EmoVAD lexicon, but matches none in the smaller MPQA lexicon, leading to missing data across our feature space. Furthermore, setting missing values to zero may be misrepresenting our data. as zero could have significance in our feature space. For example, if all words matched to the MPQA lexicon for a given song are neutral, then we may expect the average sentiment feature to be zero. This is a very different scenario than one where no words in the comments matched the MPQA dictionary, in which case this feature would be N/A and later substituted for 0.

To potentially resolve this, we test two different methods for handling missing data. First, we drop any rows which have fewer than 30% of features as containing valid data. Secondly, we combine this with the truncation of any features which have coverage for fewer than 80% of songs. These methods are tested against a baseline where missing data was set to zero instead of being removed from the dataset. It is important to note that in these comparisons, each test set will be of varying size. Direct comparisons can not be drawn from the performance of each model across null-data handling methods as a result. The following correlation values provide a general guideline for how a model will respond to the removal of missing data.

		YouTube		∩	
		Valence	Arousal	Valence	Arousal
Baseline	AMG1608	**0.53**	**0.61**	**0.41**	**0.60**
	PmEmo	**0.60**	0.39	**0.58**	0.38
Drop Songs	AMG1608	0.45	0.56	0.41	0.60
	PmEmo	0.53	0.39	0.57	0.37
Drop Features	AMG1608	0.41	0.55	0.39	0.59
	PmEmo	0.58	**0.43**	0.58	**0.41**

Table 16: Effect of dropping null samples and features on random forest model performance

Except in the case of predicting arousal for songs from PmEmo when removing null features, dropping null data always has a negative impact on the performance of our random forest model. The slight increase in correlation to true arousal labels in the PmEmo dataset may be a result of the test subset having certain samples omitted, artificially inflating performance in comparison to our baseline. Improvement in arousal correlations with null data omission is within 0.03 to 0.04 to our baseline, in the cases where there is an improvement. However, reduction in valence prediction performance on AMG1608 songs falls by as much as 0.12, and fails to match baseline performance in four separate cases. We believe handling null values by dropping the respective songs or features to negatively impact model performance, and will include all data in our feature sets going forward.

With our preprocessing pipeline now well defined, we choose seven models to evaluate for the task of music emotion recognition from bag-of-words affective terms features. We use a support vector machine and a k-nearest-neighbors regression model in comparison to our ensemble approaches: a random forest model, AdaBoost, and a histogram-based gradient boosting regression tree based on LightGBM [24]. We focus on ensemble models because of their ability to handle complex decision boundaries by training a series of weaker models on the task. AdaBoost is of particular interest here, as it updates the weights of each training sample during each iteration of the training loop to focus on examples which are more difficult to delineate. This allows the model to fit to difficult problems, at the risk of introducing too much bias and prioritizing extreme outliers.

We tune the hyperparameters of each model using a grid search to test every combination of a range of parameters. The ranges used in tuning for our seven models are presented in 11. Not all of scikit-learn's models support multi-target regression, re-

quiring us to predict valence and arousal separately. During hyperparameter tuning, models were scored by mean squared error and measured against valence labels, as valence prediction is consistently the more difficult of the two problems [26] [13]. The intersection features from AMG1608 ($n = 774$) were used, with 5-fold cross validation for each candidate model.

Each tuned model is tested against our four music emotion datasets, again predicting valence and arousal as separate targets. We test each social media source independently, as well as including intersection feature sets.

LightGBM		AdaBoost		Random Forest	
Max Iter.	500	# Estimators:	150	# Estimators:	500
Min Samples Leaf:	10	Max Depth:	5	Boostrap:	True
L2 Regularization:	0.0	Learning Rate:	1.5	Criterion:	MSE
Loss:	MSE	Loss:	Linear	# Features:	Auto
Learning Rate:	0.05	SVM		Min Samples Leaf:	2
Max Depth:	None	Kernel:	rbf	Min Samples Split:	5
Max Leaf Nodes:	15	Gamma:	$1e^{-4}$	Max Depth:	30
KNN		Tolerance:	0.001	CCP α	0
Minkowski's p:	3	Loss:	Linear		
n Neighbors:	10				

Table 17: Optimal parameters for valence prediction across five models

		AdaBoost		KNN		LightGBM		RF		SVM	
		V	A	V	A	V	A	V	A	V	A
AMG1608	Reddit	0.25	0.35	0.06	0.15	0.22	0.36	**0.30**	**0.36**	0.22	0.31
	YouTube	0.47	0.60	0.25	0.42	**0.51**	**0.60**	0.35	0.55	0.44	0.58
	Twitter	0.01	0.02	-0.01	-0.04	0.01	0.12	**-0.14**	**0.24**	0.08	-0.06
	∩	0.40	**0.61**	0.17	0.27	0.36	0.58	**0.41**	0.60	0.39	0.55
DEAM	Reddit	0.35	0.47	-0.03	0.14	**0.40**	**0.53**	0.24	0.24	0.05	0.10
	YouTube	0.09	0.11	0.05	0.09	**0.15**	0.21	0.07	**0.24**	0.13	0.18
	Twitter	-0.50	0.13	-0.17	**0.57**	-0.11	0.10	-0.33	0.19	-0.33	0.40
	∩	0.45	0.53	**0.56**	**0.63**	0.07	0.54	0.43	0.59	0.10	0.40
PmEmo	Reddit	0.40	0.40	0.37	**0.44**	0.34	0.33	**0.55**	0.38	0.41	0.30
	YouTube	**0.63**	0.34	0.45	0.29	0.60	0.34	**0.63**	**0.52**	0.54	0.43
	Twitter	-0.22	0.02	**0.34**	**0.27**	0.11	0.07	0.19	-0.21	-0.05	0.22
	∩	**0.65**	0.44	0.48	0.37	0.57	**0.47**	0.58	0.33	0.61	0.46
Deezer	Reddit	0.11	0.11	0.11	0.08	0.21	0.17	**0.24**	**0.21**	0.16	0.11
	Twitter	0.05	0.01	0.03	-0.01	0.04	0.04	**0.05**	**0.12**	0.04	0.00
	∩	0.07	0.16	0.00	0.03	0.15	0.17	**0.18**	0.12	0.16	**0.17**

Table 18: Valence and arousal Pearson's correlations of 5 models for music emotion prediction

We achieve maximum Pearson's correlations to valence and arousal targets of (0.51, 0.60) with LightGBM on AMG1608, (0.56, 0.63) with k-nearest-neighbors in DEAM, and (0.63, 0.52) and (0.24, 0.21) with a random forest model in PmEmo and Deezer, respectively. Except in the case of Deezer, YouTube or intersection feature sets always result in a model which achieves the highest correlations. For Deezer, where no YouTube feature set exists, Reddit features outperform both models trained on Twitter and intersection features.

AdaBoost generates predictions which achieve better correlation to valence than any model in the PmEmo intersection feature set. However, AdaBoost is consistently outperformed by a random forest model. Initially it was believed that AdaBoost would be the best performing of the ensemble approaches due to its iterative weighting mechanism, but this does not seem to be the case.

There is significant improvement over both baseline experiments (see Table 14 and Table 11) from ensemble models. However, a k-nearest-neighbors regressor seems to outperform these ensemble models in the case of Twitter features for DEAM and PmEmo songs. Both of these datasets have relatively few tweets associated with

them. It is possible our k-nearest neighbors approach outperforms ensemble models in low-yield datasets with relatively sparse comment sets. Overall, our random forest approach achieves the highest performance in either valence or arousal in 9 of our 15 subsets, indicating it is the strongest model for identifying music emotion given our feature space.

7.3 Discussion

Features generated from YouTube comments seem to outperform all other single-source models, and match or exceed performance from our combined features sets depending on the regression model or dataset. This is counterintuitive, as the Reddit subset for AMG1608 and PmEmo tends to have more words per song on average than YouTube or Twitter. Without YouTube data for Deezer, it is difficult to say if the success of YouTube feature sets is specific to AMG1608 and PmEmo, or if it is generalizable. YouTube features perform very poorly for valence and arousal prediction on DEAM songs, though this again may be due to the relatively few YouTube comments associated with the DEAM dataset and the obscurity of the songs chosen for DEAM.

Twitter feature sets consistently perform the worst of any source, regardless of model, with the sole exception of predicting arousal in DEAM. Even in this case, Twitter is outperformed by a combined feature set. The consistently poor performance of Tweets for music valence/arousal prediction leads us to believe one of three factors is influencing Twitter-based models. First, it is possible that limiting our data pull to 100 tweets per song was insufficient to capture affectively relevant conversations from the platform. A more thorough data mining approach should be considered in the future to better understand the limits of Twitter as a source for music information retrieval. Secondly, limiting the number of characters in a social media submission may result in conversations which omit key affective terms which our affective wordlists need in order to extract semantic information. However, the extensive work on Twitter sentiment analysis, including the use of affective word dictionaries for emotion recognition, would seem to refute this [35].

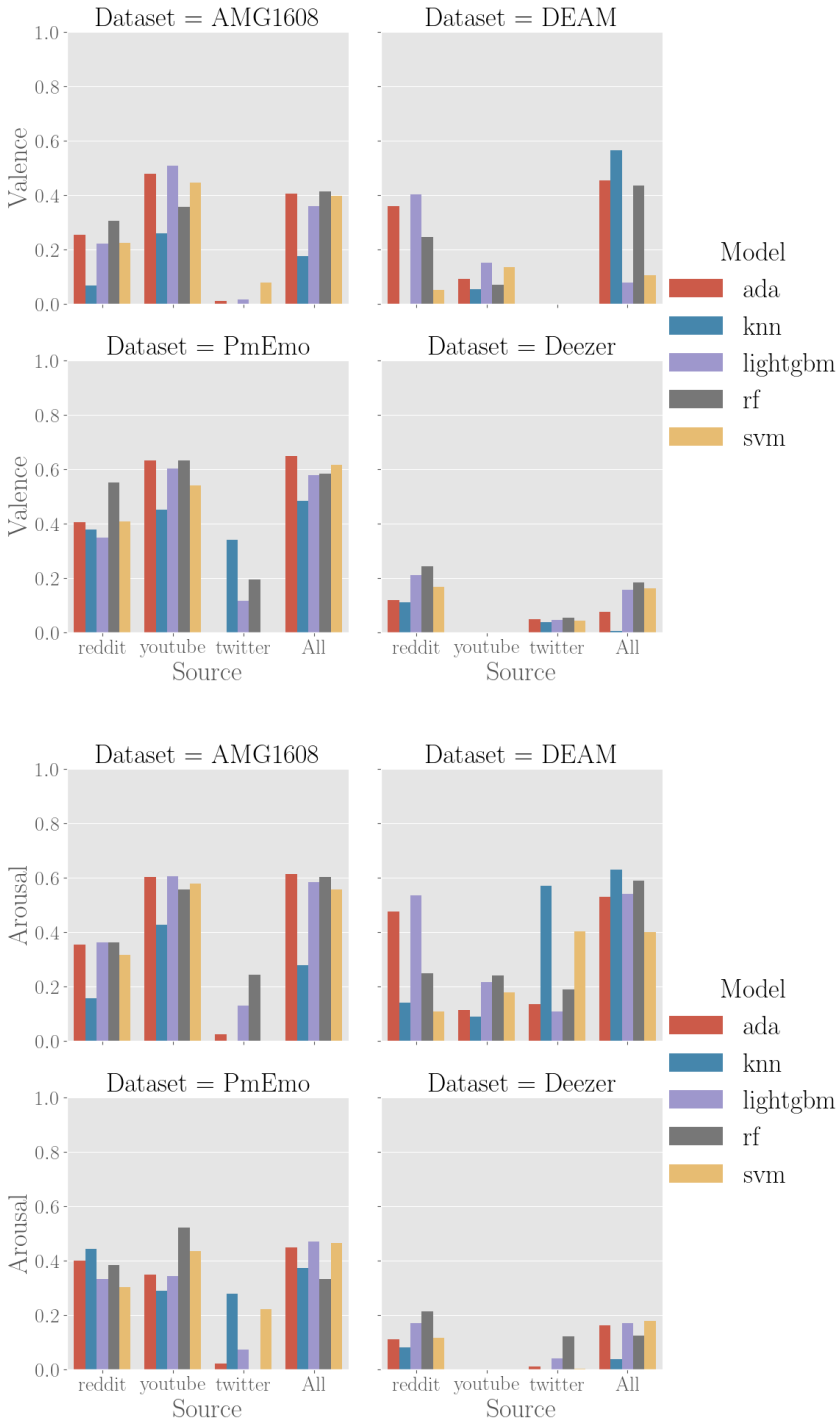


Figure 14: Comparison of model performance for music valence and arousal prediction

Finally, it is possible that when discussing a song, not every tweet will directly refer to the specific song title and artist name. Our current query methods for Twitter varies from that for Reddit and YouTube in that it does not pull all replies within a tweet. Our concern was that by querying for specific tweets, and all replies to that tweet, we may end up with duplicate data or cyclical conversations, as a reply tweet may be matched as a top-level tweet later on in the query. However, conversations in reply to a specific mention of a song and artist are potentially just as or more semantically relevant than the original comment itself, as indicated by the comparatively strong performance of Reddit and YouTube comment based feature engineering approaches.

This valence-arousal prediction experiment is limited by a lack of multi-target regression, and models being tuned specifically for valence prediction. Valence and arousal are connected dimensions [47], not disjoint ones, and as such predicting either independently of the other weakens the results of any emotion recognition system. By tuning our model hyperparameters on valence, we create models which are tuned for at estimating the valence of a song, instead of models which are tuned for estimating the emotion of a song.

Our random forest model averages a Pearson's correlation of 0.25 across all data subsets. Omitting Twitter features, this average increases to 0.36. We find maximum performance for valence prediction in the PmEmo subset, with valence and arousal correlations of 0.63 and 0.52. This considerably exceeds the baseline linear regression performance of 0.11 and 0.22.

It seems that, though our predicted labels follow the same trend line as the labels in the PmEmo test set, as well as the overall trend of the entire PmEmo dataset seen in Figure 9, the predictions are tightly clustered around the mean. It is possible that our models are underfit, or that there exists too little variance in the feature sets to reasonably discern between examples. This issue persists in our other models, and is not limited to ensemble methods. In fact, the distribution of arousal predictions produced by our support vector machine is less variant than that of corresponding ensemble methods.

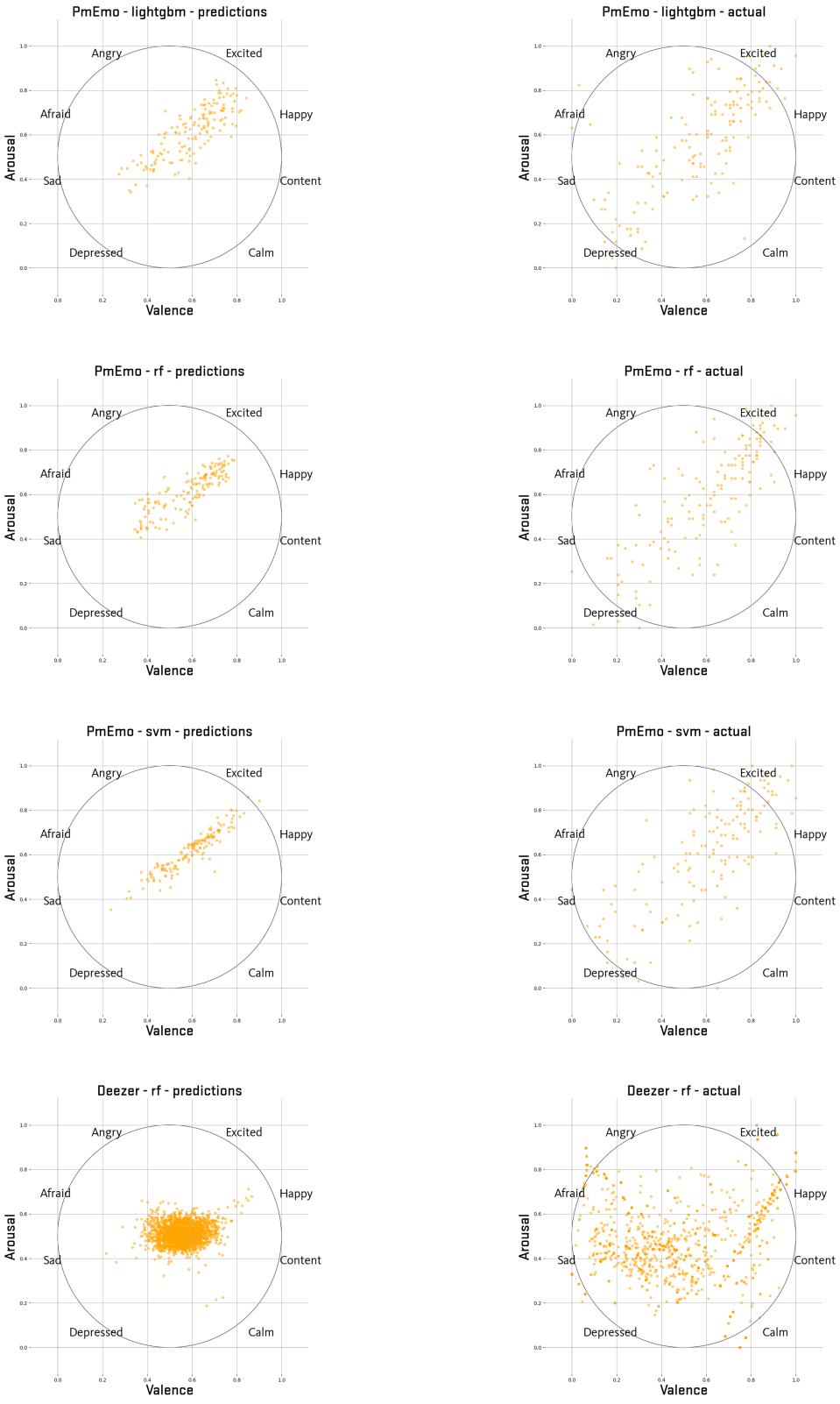


Figure 15: Circumplex distributions of a selection of models and datasets

Performance on the Deezer dataset is considerably worse across all models, dimensions, and sources. The intersection feature set on Deezer is not comparable to that of our other datasets due to the omission of YouTube data from these features. However, even in comparing Reddit feature subsets, all models consistently perform worse on Deezer than any other dataset. This is an unexpected result, as generally models perform better in the presence of an abundance of data. Given that Deezer consists of an order of magnitude more songs than our other datasets, the poor performance across its labels is alarming. It is possible that the synthetic nature of Deezer's labels hinders the performance of social-media based emotion prediction models. With no published human annotated subset of Deezer labels, we can not currently compare synthetic and manual annotations on Deezer songs for correlation to valence and arousal values generated from social media sentiment analysis. It also can not be ruled out that the inclusion of YouTube features may bring model performance on Deezer songs to comparable levels with our other datasets. Our analysis of social media model performance on Deezer songs is inconclusive and will be considered in future work.

A random forest model is the highest performing method for Deezer songs using the Reddit feature subset. The issue of predictions being clustered toward the mean is exacerbated even further in this dataset however, as seen in Figure 15. Low variance in predictions persists between sources, models, and datasets, and indicates that our affective term features may not be capturing the variance between the comments of each song. Aggregating a series of distinct comments into a single bag-of-words model loses the semantic meaning of each unique comment and the conversations which follow. Furthermore, the use of affective dictionaries restricts our feature generation to a whitelist, potentially omitting valuable semantic information during pre-processing. This feature engineering approach also has no method for handling sentiment negation or bigrams. A model which can understand the structure of each comment, as well as how each comment relates to one another, may be more suited to this learning task.

8 Approach 2 - Transformer Models

The feature engineering approach to music emotion prediction from social media commentary demonstrates a weak, yet present, correlation between these social media conversations and the emotion elicited in a listener by a piece of music. Our baseline experiments demonstrate that by measuring the valence and arousal of the words used to describe a song by users, we can approximate the song's affective label. These predictions were further enhanced by generating a set of summary statistics regarding the affect presence, affect intensity, valence, arousal, dominance, and sentiment of these words, and training an ensemble model on the resulting feature set.

We believe the limitations of our machine learning approach to stem from the bag-of-words feature extraction process. Our feature sets are based on appending all comments into a single string, matching all words in that comment set to our affective dictionaries, and then finding the minimum, maximum, mean, and standard deviation of all those values, across our affective dimensions. This results in a feature set which roughly captures the affective terms used by Reddit, YouTube, and Twitter users to describe a song, however it does not capture the relationship between these words or these comments, nor does it preserve word order. We also risk biasing our dataset by the types of words which are measured by our affective dictionaries.

To address these concerns, we design a transformer based approach for estimating a comment's music valence and arousal. Transformer models have been applied with success in music mood categorization, using lyrics as features [1]. By leveraging the nature of the transformer architecture's context awareness, we preserve the structure of each comment. This allows our model to understand not just how the individual words used in a comment set relate to the affective properties of that song, but also the structure of each comment and how each word is used.

Transformer models naively support multi-target regression through the use of multiple output nodes at the end of the network. This allows our model to predict valence and arousal as co-dependent values instead of independent labels as done in prior approaches. These dimensions are inherently related [47], and annotators often place songs on a valence-arousal plane instead of labeling the values independently of one another [9]. We anticipate that the use of a multi-target regression model will improve both valence and arousal correlations by understanding the relationship between them.

8.1 Methods

Our model architecture consists of a pre-trained transformer model and a densely connected neural network to predict regression targets from the last hidden state of the transformer, often referred to as a regression head. We use the `TFDistilBertBase` and `TFRobertaBase` model implementations provided by the Huggingface deep natural language processing library[15]. Raw comments are first filtered for URLs and HTML tags. Stopwords and neutral sentiment words are not filtered out, as filtering stopwords goes against best practice when using BERT-like language models. Because the model trains on full English text, it expects relatively standard grammatical structure. Removing stopwords would disrupt the sentence-level contextual understanding encoded in the output embeddings.

We randomly split our dataset into training, validation, and test, using a 0.70, 0.15, 0.15 split. We split at the song level, not the comment level, to prevent information leakage from our holdout set and to match the intended application of the model. Valence and arousal labels are normalized and scaled to a range of $[0, 1]$.

Each input to the model consists of one comment, with corresponding music valence and arousal labels. Sentences are tokenized using Huggingface's `TokenizerFast` library. A maximum sequence length of 128 is chosen based on implementation defaults[16]. Comments longer than 128 words will be truncated, and comments shorter than this sequence length will be right-padded with 0-tokens. We take the IDs and attention mask from our tokenized social media comments and input them to our language model. The default language model architectures are used - six layers and twelve self-attention heads in the case of DistilBERT, and twelve layers with twelve self-attention heads for RoBERTa.

Finally, in order to enable multi-target regression, we develop our own regression head to use the outputs from the language model to predict a valence and arousal target. Two densely connected layers are used, one of the same dimension as our maximum token length (128), and one output layer consisting of two nodes, representing our two predicted labels. The language model outputs a series of hidden states, of dimension (128, 768). We take only the last hidden layer, as it embeds the context of all previous layers into a single vector with a length of 768. Mean squared error is used as a loss function, and Adam [25] as an optimizer with a learning rate of 1×10^{-5}. Model outputs consist of a valence and arousal prediction for each comment. The mean of these predictions across all comments for a given song is then used as the final song-level output.

[15] https://huggingface.co/models

[16] Using excessively large sequence lengths may result in slow model training times and GPU out-of-memory errors.

8.1.1 RoBERTa v.s. DistilBERT

We choose to compare two powerful pre-trained transformer models for natural language understanding – DistilBERT [48] and RoBERTa[30]. Both of these models are based on transformer architectures, described in 3.2. The compute resources required to train RoBERTa are significant, meaning that DistilBERT will fine tune on our downstream task much faster than RoBERTa, and require fewer compute resources. In the interest of developing a flexible and reproducible model for music emotion recognition, we compare DistilBERT and RoBERTa on valence and arousal prediction on songs from PmEmo and AMG1608, using the combined social media commentary from YouTube, Reddit, and Twitter. Songs are included as long as they are included in at least one of the three sources.

BERT-like models are known to converge quickly due to their pre-trained nature [14]. Models only need to be fine-tuned on downstream tasks for a few epochs, and deeper training risks overfit. Two to four epochs are recommended for fine-tuning BERT. We test DistilBERT and RoBERTa independently for 10 epochs each on the PmEmo-All and AMG1608-All to identify an ideal number of epochs for training. We measure the performance of the model against a held-out validation set at the end of each training epoch, and choose an ideal stopping point by measuring the point at which the validation loss either increases or stops decreasing.

		DistilBERT	RoBERTa
AMG1608	Valence	0.46	0.56
	Arousal	0.63	0.65
	Runtime (approx.)	54 min.	1 hr 56 min
	Best Epoch	3	2
PmEmo	Valence	0.70	0.70
	Arousal	0.60	0.58
	Runtime (approx.)	40 min	1 hr 25 min
	Best Epoch	1	2

Table 19: RoBERTa and DistilBERT model performance, fine tuned for 10 epochs using 4 Nvidia v100 GPUs

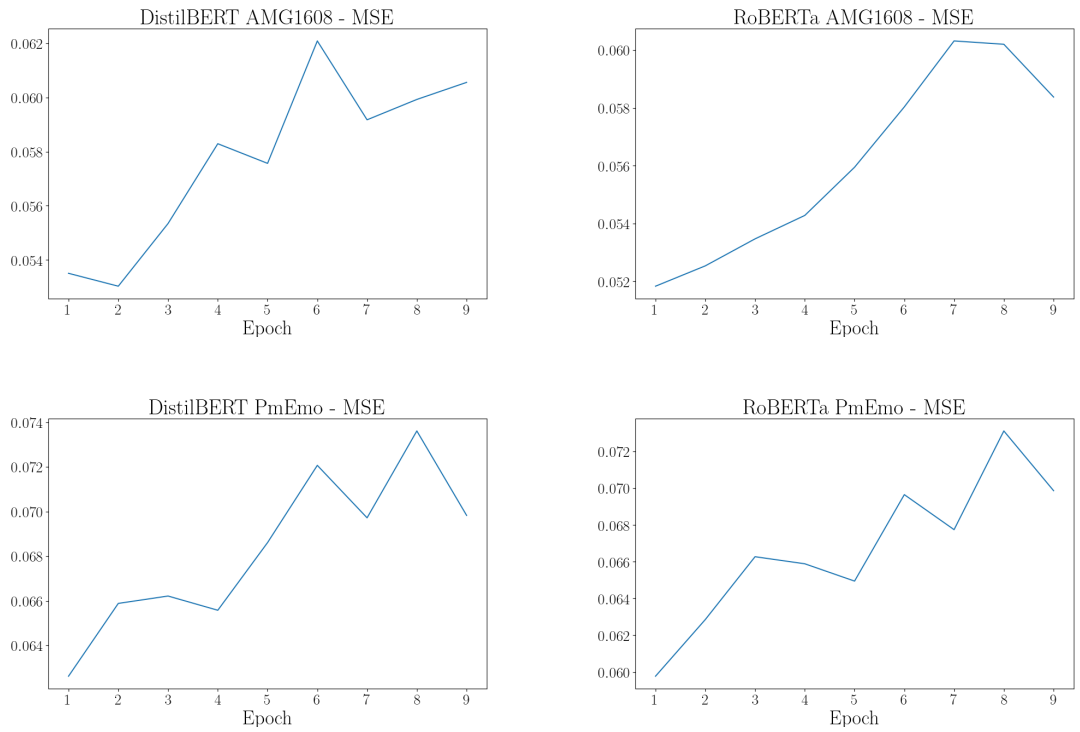


Figure 16: Loss curves for DistilBERT and RoBERTa

Our transformer language models seem to tune to this task in between 1 and 3 epochs, where a best epoch is defined as the epoch with the lowest mean squared error loss for our validation subset. Initial results seem to indicate that RoBERTa marginally outperforms DistilBERT in valence and arousal prediction on these two datasets. In order to compare the performance of these two models while controlling for overfit, we repeat this experiment, training each model on 2 epochs.

		DistilBERT	RoBERTa
AMG1608	Valence	0.49	0.51
	Arousal	0.64	0.63
PmEmo	Valence	0.72	0.71
	Arousal	0.64	0.64

Table 20: Comparison of DistilBERT and RoBERTa performance, fine tuned for 2 epochs

The performance delta between RoBERTa and DistilBERT for these two datasets is very close. In the case of songs from the PmEmo dataset, DistilBERT marginally

outperforms RoBERTa in valence prediction accuracy, while matching it in arousal prediction. For AMG1608, DistilBERT outperforms RoBERTa for arousal prediction by an equally slim margin, while losing to RoBERTa for valence prediction. However, the difference between computational cost for these models can not be understated. In both of our 10-epoch training cases, DistilBERT completed training in less than half the runtime as RoBERTa. Because the models are comparable in predictive performance, these runtimes justify our decision to choose DistilBERT for our experiments.

8.1.2 Source-specific Models

We compare five dataset slices for prediction on AMG1608 songs. In a similar approach presented in Table 14, each social media source is tested as an independent training set. Additionally, we test two methods of source aggregation − one of which takes the intersection of songs contained in all three sources as used in Table 14 and Table 18, which we again denote as ∩, and one simply concatenating all comments from all sources. Because our language model predicts at the comment level, no additional join logic is needed to include all comments for all songs included in any source.

We fine tune DistilBERT for two epochs against these five cases. Valence and arousal predictions are made on the comment level, and the mean of these comment-level predictions are taken per song to generate a song-level label.

	Reddit	Twitter	YouTube	∩	Concat.
Valence	0.32	0.23	0.62	0.47	0.49
Arousal	0.56	0.34	0.72	0.62	0.64

Table 21: Results of DistilBERT trained for 2 epochs on AMG1608 songs.

Our best model in Table 18 was able to achieve a correlation of 0.41 and 0.60 to valence and arousal labels, respectively, for AMG1608 songs in the intersection subset. A DistilBERT-based deep learning approach is able to outperform a random forest model on this subset of AMG1608, achieving valence and arousal correlations of 0.47 and 0.62. This model also dramatically outperforms prior models on AMG1608-YouTube prediction, improving from (0.51, 0.60) to (0.62, 0.72).

The concatenated approach to data subset prediction proves to be the more effective of the two aggregations, achieving correlations of (0.49, 0.64). However, this is still unable to outperform a YouTube-only model for AMG1608. Using only YouTube comments (and only songs which occur in the YouTube subset as a result) improves valence correlations by 0.13 and arousal correlations by 0.08. However, a direct performance comparison of these models is not possible, as the test set for AMG1608-Concat

contains more songs than AMG1608-YouTube. The YouTube subset only consists of 1592 of the 1608 songs in AMG1608, while the concat subset contains 1607 songs. These data subsets are very close to one another in size, but not identical. Despite marginally worse performance, our concatenated model covers a broader range of songs.

8.2 Results

We test our model on valence and arousal prediction of our four music emotion datasets, comparing models trained on a concatenated feature set to those trained exclusively on YouTube comments.

		AMG1608	PmEmo	Deezer	DEAM
Concat.	Valence	0.49	0.72	0.35	0.09
	Arousal	0.64	0.66	0.30	0.08
YouTube	Valence	0.62	0.68	N/A	0.25
	Arousal	0.72	0.52	N/A	0.26

Table 22: Correlations for valence and arousal prediction trained on concatenated and YouTube comment sets

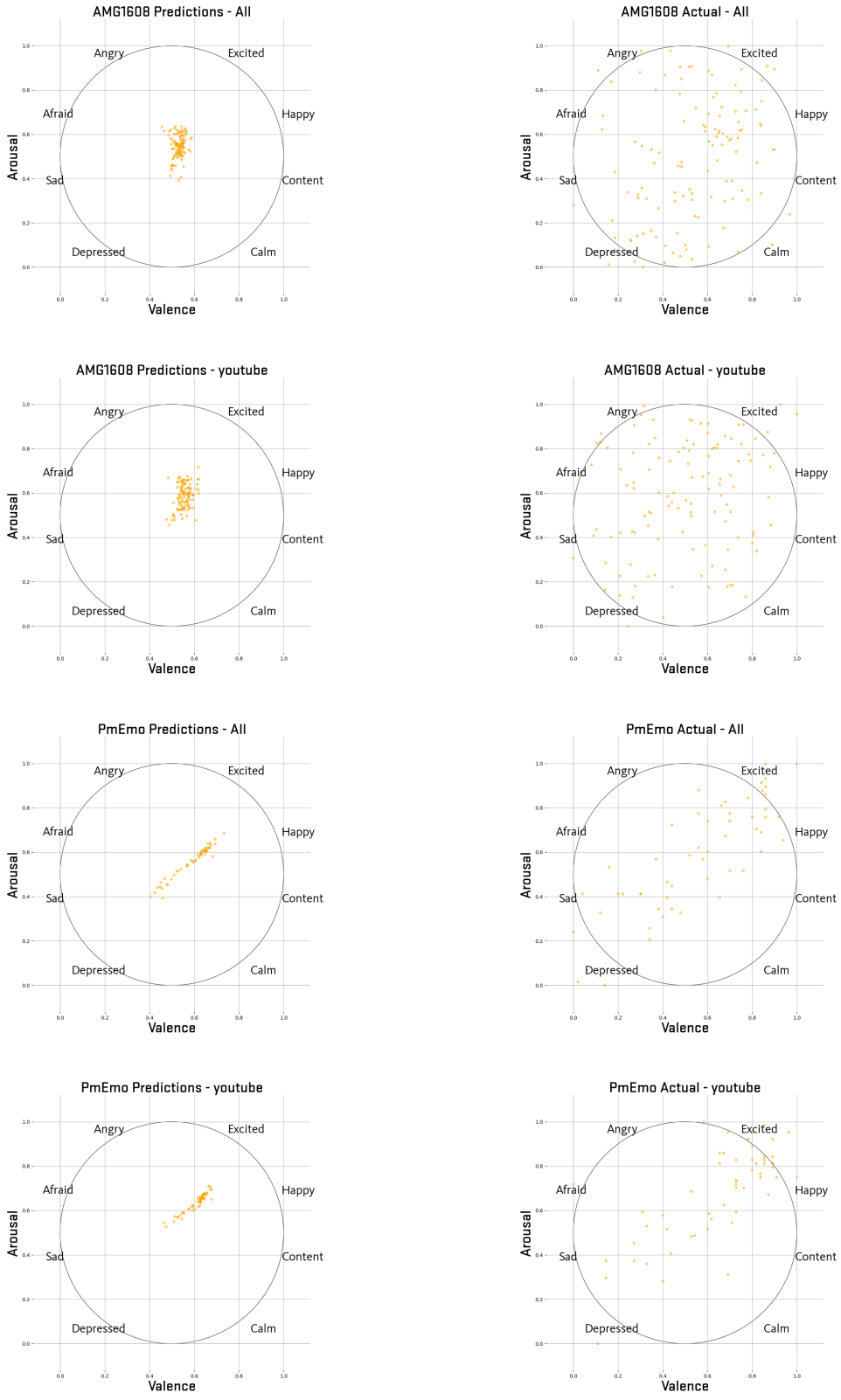


Figure 17: Distribution of DistilBERT predictions on AMG1608 and PmEmo songs

Based on correlation measures, direct comment-level music-valence and music-arousal label prediction and aggregation using a pre-trained transformer model outperforms every feature-engineering based machine learning model, for both YouTube-only models and concatenated comment set models, with the exception of songs from the DEAM dataset. The best condition for our random forest model, predicting PmEmo songs from the YouTube subset, achieves a maximum correlation of (0.63, 0.52). A DistilBERT model is able to achieve (0.68, 0.52) under those same conditions, and provide a maximum correlation of (0.72, 0.66) on the PmEmo concatenated comment set.

We believe there to be an insufficient number of Reddit, YouTube, and Twitter comments associated with the DEAM dataset for our deep learning approach to yield any meaningful performance uplift from a traditional machine learning model. BERT-like models work best in the presence of very large training sets [14], and DistilBERT's poor performance on the DEAM can be attributed to the lack of social media conversation surrounding these royalty-free songs on these three platforms. Model performance on the Deezer dataset continues to be relatively poor across all approaches.

YouTube specific models continue to closely match or outperform those models which incorporate comments from all three sources. Notably, our concatenated dataset does significantly outperform a YouTube model for songs from the PmEmo dataset. However, both DEAM and AMG1608 labels more closely correspond with the predictions from a model trained on only their YouTube subsets than full concatenated models. YouTube based feature sets have consistently outperformed other source-specific datasets in both our approaches, across models and techniques, indicating that conversations from YouTube comments under posts about a song may be more semantically relevant to the task of predicting that song's valence and arousal than those conversations from Reddit or Twitter. The primary difference between these platforms is that YouTube is commonly used as a music listening platform, versus similar communities on other platforms which exist to discuss music. The affective descriptors used by a user who has just listened to a song on YouTube may be different from those used by one discussing that song or artist at a later date on, for example, Reddit, and the difference in the semantic differences between these platforms warrants future investigation.

Though our DistilBERT model was able to achieve an improvement in Pearson's correlation to real affective labels compared to feature engineering approaches, the fundamental issue of predictions trending towards the mean persists between these modeling techniques. Figure 17 demonstrates the valence and arousal distributions of our true labels compared to DistilBERT predictions in AMG1608 and PmEmo songs, across both concatenated and YouTube data subsets. The low variance in predictions

across all four examples closely resembles the behavior of feature engineering based models shown in Figure 15. This indicates an underlying issue with our social media discourse datasets, and how we extract affective information out of them.

9 Discussion

The precedent for including social media information in music emotion recognition models was set by Laurier et al. [5], using user-annotated song metadata from Last.FM to predict mood labels. This approach was later extended to the estimation of valence and arousal labels for the Deezer dataset [13]. However, these methods relied on users tagging songs with descriptive metadata provided from music-specific online communities. We investigate the use of sentiment analysis on musical discourse to predict valence and arousal labels directly from online social media conversations.

9.1 Contributions

We assess the feasibility of predicting music emotion from social media discourse by developing two novel approaches to learning valence and arousal labels from online commentary. These models demonstrate that the conversations from social media platforms like Reddit, YouTube, and Twitter contain semantic information which is relevant to the task of music valence and arousal prediction.

We create a dataset of conversations related to music from Reddit, YouTube, and Twitter. This social media discourse dataset is based on submissions related to the songs listed in four music emotion datasets, enabling any model trained on our social media data to directly compare to both human annotated labels as well as existing methods for estimation of musical affect. We include a sentiment analysis framework for extracting affective features from this discourse.

Ensemble models perform particularly well at making valence and arousal predictions from these affective feature sets. We observe a modest correlation of the valence and arousal predictions of a random forest model to human annotated emotion labels, with a maximum performance of (0.63, 0.52). This indicates the validity of using social media discourse for music emotion prediction.

We repeat this experiment using a different approach to sentiment analysis from social media text, relying on recent advancements in deep natural language processing through the use of pre-trained transformer models such as BERT [14]. We find that, despite RoBERTa having almost twice as many parameters as DistilBERT, both achieve comparable performance on our datasets. We choose to use DistilBERT for our model to leverage it's relatively low computational costs. We also find that pre-trained transformer models can be fine-tuned to our learning task in relatively few epochs, further reducing both training time and computational cost for our model. DistilBERT marginally outperforms a random forest model on all tasks except prediction on DEAM songs, while achieving a maximum correlation of (0.72, 0.66)

YouTube comments are found to be the most important source for emotion prediction, outperforming combined comment sets at times. Unlike in our feature engineering experiments, removing all songs which do not occur in all three social media sources does not seem to improve performance significantly in our transformer approach. Deep learning models perform well in conditions with very large training sets, and filtering by dataset intersection can reduce the number of samples by up to 58%.

9.2 Limitations

Our analysis of music emotion recognition models trained on social media discourse compared to those generated synthetically from Last.FM tags is restricted by a lack of YouTube data for Deezer. The quota restrictions imposed by Google on the YouTube Data API allowed for the aggregation of YouTube comments for smaller datasets. However, scraping YouTube comments for Deezer would involve data mining potentially up to 180,000 videos, and could take upwards of 1 year and 6 months with the current daily API limit. This limits our ability to draw comparisons between models trained on Deezer versus those trained on other datasets due to an incomplete dataset.

All models trained exclusively on Tweets performed considerably worse than other source-specific or aggregate model. This can not be explained by claiming that Twitter comments do not contain semantically relevant information, as the usefulness of Tweets for social media sentiment analysis has been demonstrated by the NLP research community [35]. Instead, we believe this indicates an issue with our querying and data accumulation process for Twitter. The process for gathering Tweets differed from that of gathering YouTube comments and Reddit threads, as the platform has no top-level posts from which comment threads can be aggregated. We designed a data mining system which only referenced the 100 most relevant tweets strictly containing the artist name and track title. This fails to capture reply tweets or conversations surrounding a song in the same manner that our Reddit or YouTube datasets will.

We also find that the distribution of either model's predictions tend to be clustered closely to the center of the valence-arousal space. Both of our models perform some form of comment-level aggregation, taking a mean of comment-level or word-level affective values to generate a value or set of values representing the affect of all comments related to a given song. This aggregation risks discarding valuable semantic information contained within comments with opposing sentiment by reducing them into an average neutral sentiment.

9.3 Future Work

A primary focus of any future work should be to revise the existing data mining strategies. First, YouTube comments for Deezer should be obtained. We are currently in the process of applying for an academic license to YouTube's API to increase our quota, pending Google's response. An increase in permitted daily transactions to YouTube should make the completion of the Deezer social media discourse subset trivial by the use of our existing framework.

Our methods for pulling user conversations from Twitter deserve to be revisited as well. The relatively few tweets captured by our initial data scraping indicate that the restriction to only the 100 most relevant tweets should be relaxed. Furthermore, reply tweets should be pulled as well as top-level tweets which reference the artist name and track title, in order to capture responding comment threads similarly to Reddit or YouTube. It is important to design a system to avoid duplicate tweets, as a tweet which occurs in our top n results may also be present as the reply to another tweet.

We plan to investigate the use of SoundCloud and Last.FM comments as well. Last.FM, in particular, has proven repeatably useful in generating features for music emotion recognition thanks to its community annotated tags [5]. Last.FM has recently enabled "Shouts", allowing users to post free-form comments in response to a song. To our knowledge no existing work has attempted to use sentiment analysis on Last.FM conversations for music emotion prediction.

Both models we present result in outputs clustered towards an average, indicating that our data may be too noisy for meaningful sentiment analysis. A revision of our data pre-processing pipeline is necessary. Comments may need to be dropped from our dataset in some cases. For example, Reddit marks comments which were deleted by a user as "[deleted]", and those removed by a moderator as "[removed]". These comments provide very little semantic information, and likely hinder the performance of our transformer model. Since the bag-of-words model learns based on a whitelist of affective terms, affectively irrelevant comments are less of a concern. A combination of the pre-processing steps of our random forest model and the direct comment-level valence and arousal predictions offered by DistilBERT may be a step forward in the future, dropping any comments which do not contain an affective word before tokenizing. Comments of insufficient length, of low or negative score, or those generated by bots may also have adverse effects on our model performance.

Future work should investigate model architectures and feature extraction methods which adequately represent the variance within comments and understands the relationship between comments. Recent developments in relation-aware transformer architectures may allow a BERT-like model to encode relationships between comment-

levels samples. This could indicate each comment's connection to a specific song, or even embed inter-comment dependencies to better represent the nested comment-tree structure of platforms such as Reddit [54].

Which stage to perform song-level aggregations at should be considered for future iterations of our transformer model. Currently, we aggregate valence and arousal predictions at the end of the network. In the future, the last-hidden-state output of our language model at each comment could be fed into a pooling layer before making a valence and arousal prediction. To do so would require a song to have a fixed number of comments provided as input, either left-padding or truncating all songs which do not have enough comments, and picking a top n comments from those songs with too many. Picking this top n songs across social media sources would be difficult, as not every social media source ranks posts similarly[17].

Different base language models should be evaluated for this task. We only test `RoBERTa-base` and `DistilBERT-base-uncased` in our experiments. Newer architectures such as `xl-net` have been applied with success to music mood classification from lyric analysis thanks to it's support for larger input sequences [1]. More recently, Longformer contributes a linearly scaling self-attention mechanism which allows for sequences with significantly more tokens. This would again allow for predictions to be output from the model architecture at the song-level, modifying the input by concatenating a fixed number of comments. Similar issues as the approach described above are anticipated, mainly from the need for a method to pick the top n most semantically relevant comments.

9.4 Applications

An automatic system for music emotion recognition enables large music libraries to be rated for estimated emotive response. With the increasing prevalence of online music streaming platforms, affective modeling allows music recommender algorithms to improve user music discovery by enabling mood filtering [13]. Emotion-aware playlist generation efforts benefit from a large source of emotion labels. Systems for generating playlists which smoothly transition between affective states rely on continuous valence and arousal emotion labels [15].

By demonstrating that social media conversational data can be used to estimate a song's affective qualities, we introduce a new feature space for existing music emotion annotation systems. Hybrid models already integrate sentiment analysis of song lyrics and Last.FM tags with acoustic information to build more robust mood classifiers [22]. Existing input spaces used in current music emotion recognition models

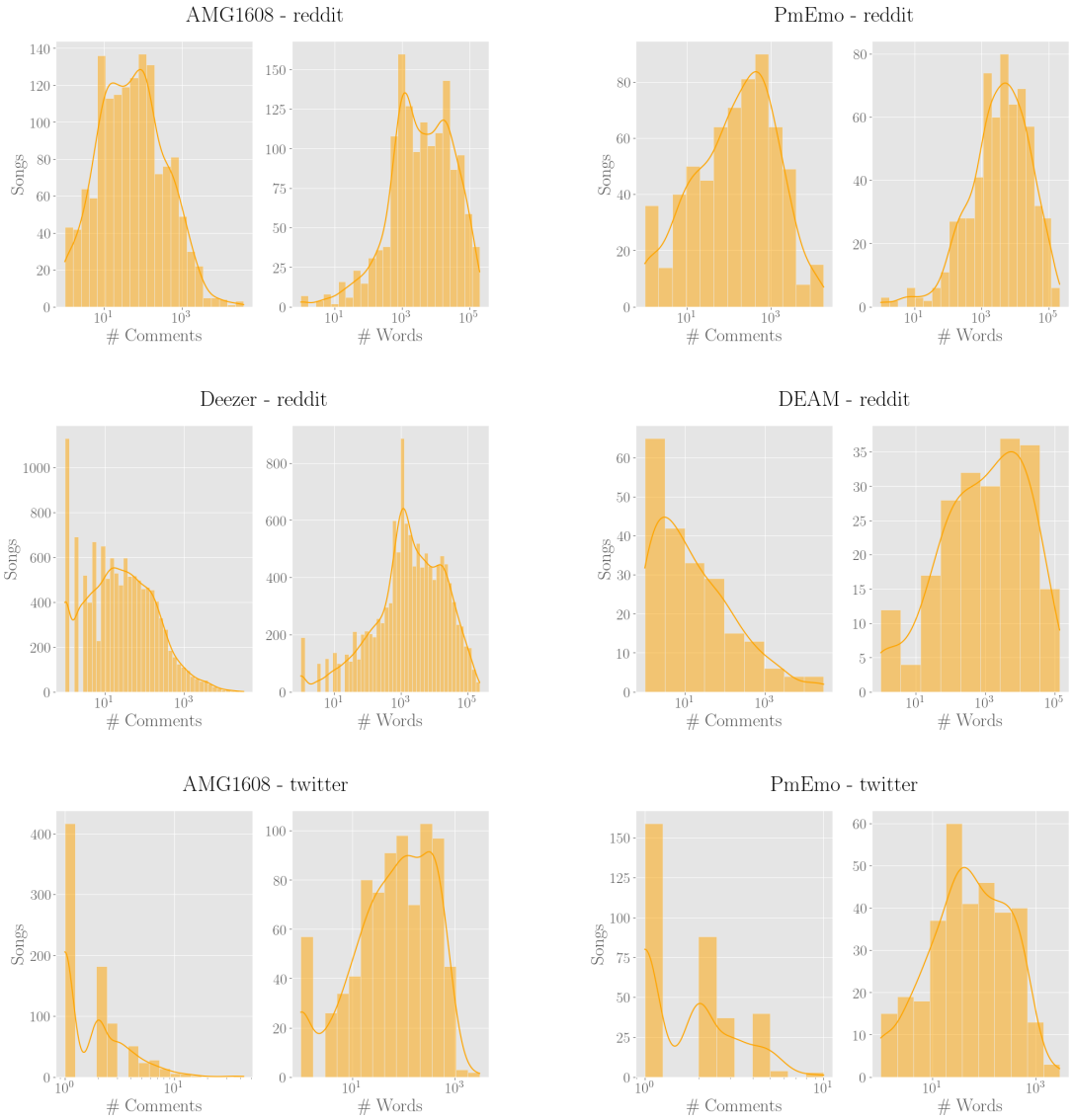
[17] https://blog.youtube/news-and-events/update-to-youtube/

suffer from challenges with regards to explainability. Furthermore, the semantic gap limits the ability of models built exclusively from low-level acoustic features from adequately explaining the human perception of a song [39]. We believe that augmenting these existing approaches with musical discourse may improve model performance.

9.5 Conclusion

We find that directly predicting musical affect from social media discourse using pre-trained transformer models outperforms feature engineering based models. This transformer approach accomplishes Pearson's correlations above 0.7 to valence and arousal labels in specific datasets. This indicates that the semantic information embedded in these comments is correlated to the manually annotated emotion labels of AMG1608 and PmEmo. Therefore, it is possible to predict the affective qualities of a given song directly from the conversations users have online about that song. To the best of our knowledge, this is the first approach to music valence and arousal regression to use conversational information from social media platforms.

10 Appendix A: Social Media Distributions



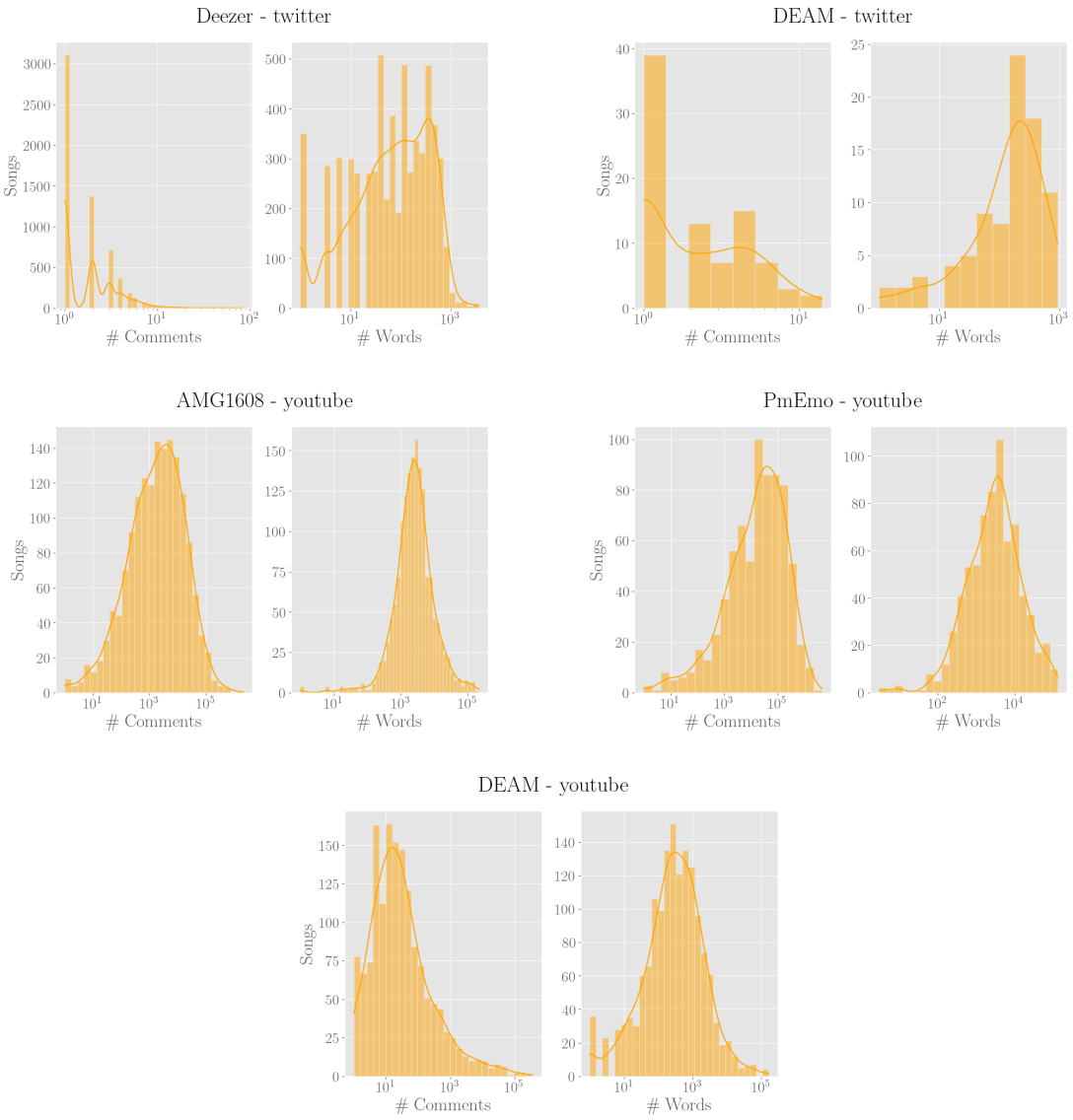


Figure 19: Distributions of music discourse from each unique social media platform

11 Appendix B: Model Parameter Tuning

Random Forest	
# Estimators	100, 150, 200
Bootstrapping	True, False
Criterion	Squared Error, Poisson
Max Features	Auto, $\sqrt{n}$, 30%
Min Samples Leaf	1, 2, 4
Min Samples Split	2, 5, 10
Max Depth	10, 20, 30, 40, 50, 75, 100
CCP α	0.0, 0.01, 0.02, 0.03
LightGBM	
Max Iterations	50, 100, 150, 200, 250, 500
Min Samples Leaf	10, 20
L2 Regularization	0.0, 0.001, 0.01, 0.1, 1.0
Learning Rate	0.05, 0.1, 0.5, 1
Max Depth	None, 10, 25, 50
Max Leaf Nodes	15, 25, 31, 50, None
AdaBoost	
# Estimators	50, 100, 150, 200
Base Estimator Max Depth	3, 5, 7, 10, None
Learning Rate	0.5, 1.0, 1.5, 3.0. 5.0
Loss Function	Linear, Exponential
Support Vector Machine	
Kernel	Linear, Polynomial, RBF
Gamma	Scale, Auto, 1e-4, 1e-3, 0.01, 0.1, 0.2, 0.5, 0.6, 0.9
Tolerance	1e-4, 1e-3, 0.01
Regularization	1e-4, 1e-3, 0.01, 0.1, 1.0, 10, 100, 1000, 10000
K-Nearest Neighbors	
# Neighbors	3, 5, 7, 10
Minkowski P-value	2, 3, 4, 5

Table 23: Hyperparamter ranges used in the tuning of our five models tested against affective features extracted from social media music commentary

References

[1] Yudhik Agrawal, Ramaguru Guru Ravi Shanker, and Vinoo Alluri. "Transformer-based approach towards music emotion recognition from lyrics". en. In: *Advances in Information Retrieval. ECIR 2021* 12657 (2021). arXiv: 2101.02051, pp. 167–175. DOI: 10.1007/978-3-030-72240-1_12.

[2] Anna Aljanaki and Mohammad Soleymani. "A data-driven approach to mid-level perceptual musical feature modeling". In: *Proceedings of the 19th International Society for Music Information Retrieval Conference*. ISMIR. June 2018, pp. 615–621. DOI: 10.48550/arXiv.1806.04903. URL: http://arxiv.org/abs/1806.04903.

[3] N. S. Altman. "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression". In: *The American Statistician* 46.3 (1992), pp. 175–185. DOI: 10.1080/00031305.1992.10475879.

[4] Iz Beltagy, Matthew E. Peters, and Arman Cohan. "Longformer: The Long-Document Transformer". In: arXiv:2004.05150 (Dec. 2020). DOI: 10.48550/arXiv.2004.05150. URL: http://arxiv.org/abs/2004.05150.

[5] Kerstin Bischoff et al. "Music Mood and Theme Classification - A Hybrid Approach". en. In: *Proceedings of the 10th International Society for Music Information Retrieval Conference*. ISMIR. 2009, pp. 657–662.

[6] Margaret M. Bradley and Peter J. Lang. "Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings". In: 1999.

[7] Lars Buitinck et al. "API design for machine learning software: experiences from the scikit-learn project". In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122.

[8] Erion Çano and Maurizio Morisio. "MoodyLyrics: A Sentiment Annotated Lyrics Dataset". In: *Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*. ISMSI '17. New York, NY, USA: Association for Computing Machinery, Mar. 2017, pp. 118–124. ISBN: 978-1-4503-4798-3. DOI: 10.1145/3059336.3059340. URL: https://doi.org/10.1145/3059336.3059340.

[9] Yu-An Chen et al. "The AMG1608 dataset for music emotion recognition". en. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. South Brisbane, Queensland, Australia: IEEE, Apr. 2015, pp. 693–697. ISBN: 978-1-4673-6997-8. DOI: 10.1109/ICASSP.2015.7178058. URL: http://ieeexplore.ieee.org/document/7178058/.

[10] Andrea Chiorrini et al. "Emotion and sentiment analysis of tweets using BERT". en. In: *Workshop Proceedings of the EDBT/ICDT 2021 Joint Conference*.

[11] Yoonjung Choi and Janyce Wiebe. "+/-EffectWordNet: Sense-level Lexicon Acquisition for Opinion Inference". en. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1181–1191. DOI: 10.3115/v1/D14-1125. URL: http://aclweb.org/anthology/D14-1125.

[12] Shreyan Chowdhury et al. "Towards Explainable Music Emotion Recognition: The Route Via Mid-Level Features". en. In: *Proceedings of the 20th International Society for Music Information Retrieval Conference*. ISMIR. 2019, pp. 237–243.

[13] Remi Delbouys et al. "Music Mood Detection Based On Audio And Lyrics With Deep Neural Net". en. In: *Proceedings of the 19th International Society for Music Information Retrieval Conference*. ISMIR. 2018, pp. 370–375.

[14] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". en. 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)*". May 2019, pp. 4171–4186. URL: http://arxiv.org/abs/1810.04805.

[15] Patrick Donnelly and Shaurya Gaur. "Mood Dynamic Playlist: Interpolating a musical path between emotions using a KNN algorithm". In: *International Conference on Human Interaction & Emerging Technologies: Artificial Intelligence*. 2022.

[16] Harris Drucker et al. "Support Vector Regression Machines". In: *Proceedings of the 9th International Conference on Neural Information Processing Systems*. NIPS'96. Denver, Colorado: MIT Press, 1996, pp. 155–161.

[17] Paul Ekman. "An argument for basic emotions". en. In: *Cognition and Emotion* 6.3–4 (May 1992), pp. 169–200. ISSN: 0269-9931, 1464-0600. DOI: 10.1080/02699939208411068.

[18] Evelyn Fix and Joseph L. Hodges. *Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties*. Randolph Field, Texas., 1951.

[19] Yoav Freund and Robert E. Schapire. "A desicion-theoretic generalization of on-line learning and an application to boosting". In: *Computational Learning Theory*. Ed. by Paul Vitányi. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 23–37. ISBN: 978-3-540-49195-8.

[20] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine." In: *Annals of Statistics* 29 (2001), pp. 1189–1232.

[21] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*. Vol. 1. Aug. 1995, 278–282 vol.1. DOI: 10.1109/ICDAR.1995.598994.

[22] Xiao Hu and J Stephen Downie. "When Lyrics Outperform Audio for Music Mood Classification: A Feature Analysis". en. In: *Proceedings of the 11th International Society for Music Information Retrieval Conference*. ISMIR. 2010, pp. 619–624.

[23] Xiao Hu and J. Stephen Downie. "Improving mood classification in music digital libraries by combining lyrics and audio". In: *Proceedings of the 10th annual joint conference on Digital libraries*. JCDL '10. New York, NY, USA: Association for Computing Machinery, June 2010, pp. 159–168. ISBN: 978-1-4503-0085-8. DOI: 10.1145/1816123.1816146. URL: https://doi.org/10.1145/1816123.1816146.

[24] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems 30 (NIP 2017)*. Dec. 2017. URL: https://www.microsoft.com/en-us/research/publication/lightgbm-a-highly-efficient-gradient-boosting-decision-tree/.

[25] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations* (Jan. 2017). arXiv:1412.6980 [cs]. DOI: 10.48550/arXiv.1412.6980. URL: http://arxiv.org/abs/1412.6980.

[26] M.D. Korhonen, D.A. Clausi, and M.E. Jernigan. "Modeling emotional content of music using system identification". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 36.3 (June 2006), pp. 588–599. ISSN: 1941-0492. DOI: 10.1109/TSMCB.2005.862491.

[27] Cyril Laurier, Jens Grivolla, and Perfecto Herrera. "Multimodal Music Mood Classification Using Audio and Lyrics". In: *2008 Seventh International Conference on Machine Learning and Applications*. Dec. 2008, pp. 688–693. DOI: 10.1109/ICMLA.2008.96.

[28] Tao Li and M. Ogihara. "Toward intelligent music information retrieval". In: *IEEE Transactions on Multimedia* 8.3 (June 2006), pp. 564–574. ISSN: 1941-0077. DOI: 10.1109/TMM.2006.870730.

[29] M. S. Likitha et al. "Speech based human emotion recognition using MFCC". In: *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. Mar. 2017, pp. 2257–2260. DOI: 10.1109/WiSPNET.2017.8300161.

[30] Yinhan Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: arXiv:1907.11692 (July 2019). arXiv:1907.11692 [cs]. DOI: 10.48550/arXiv.1907.11692. URL: http://arxiv.org/abs/1907.11692.

[31] Lie Lu, D. Liu, and Hong-Jiang Zhang. "Automatic mood detection and tracking of music audio signals". In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.1 (Jan. 2006), pp. 5–18. ISSN: 1558-7924. DOI: 10.1109/TSA.2005.860344.

[32] George A. Miller. "WordNet". In: *WordNet, An Electronic Lexical Database*. May 1998.

[33] Saif Mohammad. "Obtaining Reliable Human Ratings of Valence, Arousal, and Dominance for 20,000 English Words". en. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 174–184. DOI: 10.18653/v1/P18-1017. URL: http://aclweb.org/anthology/P18-1017.

[34] Saif Mohammad. "Word Affect Intensities". In: *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference (LREC-2018)* (2018). URL: https://www.saifmohammad.com/WebDocs/word-affect-intensities.pdf.

[35] Saif Mohammad et al. "SemEval-2018 Task 1: Affect in Tweets". en. In: *Proceedings of The 12th International Workshop on Semantic Evaluation*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 1–17. DOI: 10.18653/v1/S18-1001. URL: http://aclweb.org/anthology/S18-1001.

[36] Saif M Mohammad and Peter D Turney. "Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon". en. In: *NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text* (June 2010), pp. 26–34.

[37] Saif M Mohammad and Peter D Turney. "Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon". en. In: *NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text* (June 2010), pp. 26–34.

[38] Charles Osgood, George Soci, and Percy Tannenbaum. "The dimensionality of the semantic space". In: *The Measurement of Meaning*. 1957. ISBN: 9780252745393.

[39] Renato Panda, Ricardo Manuel Malheiro, and Rui Pedro Paiva. "Audio features for music emotion recognition: a survey". In: *IEEE Transactions on Affective Computing* (2020).

[40] Bo Pang and Lillian Lee. "Opinion mining and sentiment analysis". en. In: *Foundations and Trends in Information Retrieval* (2008), p. 94.

[41] Sungjoon Park et al. "Dimensional Emotion Detection from Categorical Emotion". en. In: "*Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*" (Sept. 2021). arXiv: 1911.02499, pp. 4367–4380. URL: http://arxiv.org/abs/1911.02499.

[42] Harshali P. Patil and Mohammad Atique. "Sentiment Analysis for Social Media: A Survey". In: *2015 2nd International Conference on Information Science and Security (ICISS)*. Dec. 2015, pp. 1–4. DOI: 10.1109/ICISSEC.2015.7371033.

[43] Max Pellert, Simon Schweighofer, and David Garcia. "The individual dynamics of affective expression on social media". en. In: *EPJ Data Science* 9.1 (Dec. 2020), p. 1. ISSN: 2193-1127. DOI: 10.1140/epjds/s13688-019-0219-3.

[44] Robert Plutchik. "A General Psychoevolutionary Theory of Emotion". en. In: *Theories of Emotion*. Ed. by Robert Plutchik and Henry Kellerman. Academic Press, Jan. 1980, pp. 3–33. ISBN: 978-0-12-558701-3. DOI: 10.1016/B978-0-12-558701-3.50007-7. URL: https://www.sciencedirect.com/science/article/pii/B9780125587013500077.

[45] Daniel Preoţiuc-Pietro et al. "Modelling Valence and Arousal in Facebook posts". en. In: *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. San Diego, California: Association for Computational Linguistics, 2016, pp. 9–15. DOI: 10.18653/v1/W16-0404. URL: http://aclweb.org/anthology/W16-0404.

[46] Mahnaz Roshanaei, Richard Han, and Shivakant Mishra. "Features for mood prediction in social media". en. In: *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. Istanbul, Turkey: IEEE, Aug. 2012. ISBN: 978-1-4673-2497-7.

[47] James A. Russell. "A circumplex model of affect." en. In: *Journal of Personality and Social Psychology* 39.6 (1980), pp. 1161–1178. ISSN: 0022-3514. DOI: 10.1037/h0077714.

[48] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". en. In: (Oct. 2019). URL: https://arxiv.org/abs/1910.01108v4.

[49] D.P. Solomatine and D.L. Shrestha. "AdaBoost.RT: a boosting algorithm for regression problems". In: *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*. Vol. 2. July 2004, 1163–1168 vol.2. DOI: 10.1109/IJCNN.2004.1380102.

[50] Phillip Stone et al. "The General Inquirer". In: *The General Inquirer*. Dec. 1966.

[51] Yla R. Tausczik and James W. Pennebaker. "The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods". en. In: *Journal of Language and Social Psychology* 29.1 (Mar. 2010), pp. 24–54. ISSN: 0261-927X, 1552-6526. DOI: 10.1177/0261927X09351676.

[52] G. Tzanetakis and P. Cook. "Musical genre classification of audio signals". In: *IEEE Transactions on Speech and Audio Processing* 10.5 (July 2002), pp. 293–302. ISSN: 1558-2353. DOI: 10.1109/TSA.2002.800560.

[53] Ashish Vaswani et al. "Attention is All You Need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 9781510860964.

[54] Bailin Wang et al. "RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers". en. In: arXiv:1911.04942 (Aug. 2021). arXiv:1911.04942 [cs]. URL: http://arxiv.org/abs/1911.04942.

[55] Ju-Chiang Wang et al. "Exploring the relationship between categorical and dimensional emotion semantics of music". en. In: *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies - MIRUM '12*. Nara, Japan: ACM Press, 2012, p. 63. ISBN: 978-1-4503-1591-3. DOI: 10.1145/2390848.2390865. URL: http://dl.acm.org/citation.cfm?doid=2390848.2390865.

[56] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. "Norms of valence, arousal, and dominance for 13,915 English lemmas". en. In: *Behavior Research Methods* 45.4 (Dec. 2013), pp. 1191–1207. ISSN: 1554-3528. DOI: 10.3758/s13428-012-0314-x.

[57] Tien-Lin Wu and Shyh-Kang Jeng. "Probabilistic estimation of a novel music emotion model". In: *Proceedings of the 14th international conference on Advances in multimedia modeling*. MMM'08. Berlin, Heidelberg: Springer-Verlag, Jan. 2008, pp. 487–497. ISBN: 978-3-540-77407-5.

[58] Dan Yang and WonSook Lee. "Disambiguating Music Emotion Using Software Agents". en. In: *Proceedings of the 5th annual meeting of the International Society for Music Information Retrieval*. 2004, p. 6.

[59] Yi-Hsuan Yang and Homer H. Chen. "Machine Recognition of Music Emotion: A Review". en. In: *ACM Transactions on Intelligent Systems and Technology* 3.3 (May 2012), pp. 1–30. ISSN: 2157-6904, 2157-6912. DOI: 10.1145/2168752.2168754.

[60] Zhilin Yang et al. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: arXiv:1906.08237 (Jan. 2020). arXiv:1906.08237 [cs]. DOI: 10.48550/arXiv.1906.08237. URL: http://arxiv.org/abs/1906.08237.

[61] Kejun Zhang et al. "The PMEmo Dataset for Music Emotion Recognition". en. In: *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*. Yokohama Japan: ACM, June 2018, pp. 135–142. ISBN: 978-1-4503-5046-4.