Music Emotion Recognition on Lyrics Using Natural Language Processing

Yinan Zhou

Music Technology Area
Department of Music Research
Schulich School of Music
McGill University, Montreal

August 2022

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Arts

ABSTRACT

Music Emotion Recognition (MER) is a promising new direction for automatic music organization and retrieval. The aim of MER is to automatically annotate music based on mood. Researchers have tried to retrieve music mood using audio features, contextual text information, and multimodal methods. Among these music features used in MER, lyrics are an important part of vocal music that MER researchers have been exploiting. Using traditional lexicons and traditional Natural Language Processing (NLP) techniques, researchers have been conducting MER using lyrics alone in several languages. However, there has not been enough data for MER to use NLP on lyrics. Transfer learning methods can bridge the gap between the need of massive data for machine learning methods and the lack of annotated data for MER. This thesis implements transfer learning to classify music only using lyrics in English into four categories: angry, happy, sad, and relaxed. Two pre-trained models, BERT and XLNet, are tested and compared with a traditional machine learning method, Support Vector Machine. Also investigated are the effects of different text preprocessing techniques and their combinations on MER using lyrics.

RÉSUMÉ

La reconnaissance d'émotions dans la musique ("Music Emotion Recognition," MER) est une nouvelle voie prometteuse dans l'automatisation de l'organisation et de la recherche de la musique. Le but de la MER est d'automatiquement annoter la musique selon l'émotion. Des recherches ont déjà été faites pour détecter l'émotion dans la musique en utilisant des caractéristiques sonores, des informations et métadonnées textuelles et des approches multimodales. À l'aide de lexiques et de techniques de traitement automatique du langage naturel (TALN) traditionnels, des études MER n'utilisant que les paroles ont été menées dans plusieurs langues. Toutefois, cette méthode de MER à l'aide des paroles utilisant le TALN souffre de manque de données. L'apprentissage par transfert peut combler à la fois le besoin massif de données pour les méthodes d'apprentissage automatique et le manque de données annotées pour la MER. La présente thèse implémente l'apprentissage par transfert pour classifier la musique en anglais, à l'aide des paroles seulement, dans quatre catégories : fâchée, joyeuse, triste et paisible. Des modèles pré-entraînés, BERT et XLNet, sont testés et comparés avec une méthode traditionnelle d'apprentissage automatique, les machines à vecteur de support. La recherche porte aussi sur les effets de différentes méthodes de prétraitement de texte et de leurs combinaisons sur la MER utilisant les paroles.

ACKNOWLEGEMENTS

I would like to thank my supervisor, Ichiro Fujinaga, for the time and effort spent giving detailed, invaluable guidance and feedback; the other students in the Distributed Digital Music Archive and Library (DDMAL) research lab, for their interesting ideas and suggestions; my parents, for their unconditional support and trust throughout this journey; my best friends, Yuhang Sun and Yuhang Liu, for their constant moral support and encouragement. Finally, special thanks are owed to my colleague at DDMAL, Geneviève, for proofreading the French translation of the abstract.

I am grateful to Compute Canada for providing all the computing resources I needed to run my experiments through their Research Platforms and Portals Grant. This thesis draws on research supported by the Social Sciences and Humanities Research Council.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

Chapter 1 Introduction

With the rapid proliferation in the amount of music produced, music information search strategies need to evolve to fulfill users' expectations of searching and browsing capabilities (Kim et al. 2010). Researchers have explored similarities among music pieces to organize music in groups and recommend suitable music to users. However, most music classification problems concentrate on genre or style classification. Lately, the emotional aspect has become another essential criterion for music classification.

Music psychology studies have revealed that the affective aspects of music are essential. Scruton (1983, 49) believed that the value of music is found in its capacity to express emotion. Schubert (2007, 500) also pointed out that one of music's major attractions may be its capacity to arouse and express emotion. Furthermore, Huron (2000) stated that the primary functions of music are "social and psychological." Huron (2000) also asserted that stylistic and mood indexes are the most useful retrieval indexes for music since they benefit social and psychological search. As a result, music mood emerges as a new type of metadata for music.

In addition, Sloboda and O'Neill (2001, 415) believed that music offers a form of semiotic and affective power that helps individuals construct socially emotional feelings. Eerola and Vuoskoski (2013) asserted that "the emotional power of music is the reason for its application in areas as diverse as the gaming industry, film industry, marketing, and music therapy."

Recently, the demand for automatic methods for music classification and recommendation has arisen with the advent of music streaming and downloading services and the exposure to information on a massive scale. Nevertheless, most music archives do not support emotion-based

music retrieval. Moreover, it is impossible and time-consuming to manually annotate music emotion for all pieces of music.

Automatic Music Emotion Recognition (MER) appeared as a promising new direction for automatic music organization and retrieval. In general, MER is regarded as a multiclass-multilabel classification or regression problem (Kim et al. 2010). Music emotion measurement usually consists of two components: perceptions and inductions (Gabrielsson and Lindström 2001). MER tasks are primarily concerned with recognizing the emotion expressed by music rather than the mood induced by music (Kim et al. 2010).

The goal of MER is to annotate music based on mood automatically. The music can be an entire song or a section of a song. Surveys, social tagging, and annotation games are common methods for gathering ground truth. With the ground truth, contextual text and audio-based music features can be analyzed to recognize emotions.

1.1 Motivation

Music psychology studies have indicated that lyrics are essential to music mood. For example, Stratton and Zalanowski (1994) stated that lyrics seem to have more power to stimulate mood change than audio alone and can endow a particular melody with emotional qualities. According to Cunningham et al. (2006), lyrics were the most frequently mentioned feature by participants when asked why they do not like a song.

Lyrics have been shown to be crucial in MER. For example, Yang and Lee (2009) stated that lyrics alone could be used to generate human-comprehensible classification models. However, most works in MER try to combine lyrics with other modalities, such as audio and social tags (e.g.,

Hu, Downie, and Ehmann 2009; Hu 2010; Xue, Xue, and Su 2015). Only a few MER models used lyrics solely.

This thesis takes the assumption that lyric information can be used to classify music mood and continue exploring its limitations with the state-of-art Natural Language Processing (NLP) techniques. In addition, the effect of different text preprocessing methods on lyric emotion classification is investigated, which has yet to be studied in detail before. It is hoped that the NLP methods deployed on lyrics can properly classify music based on mood.

1.2 Project Overview

This thesis used the two existing English lyrics datasets annotated using mood tags available music archive: MoodyLyrics4Q[1] and AllMusic[2] dataset. Both datasets classify music into four categories based on Russell's model (Russell 1980): *happy*, *relaxed*, *sad*, and *angry*. The lyrics themselves were collected from the Internet using the Genius API[3] since the datasets do not provide lyrics because of license issues.

As a baseline, an automatic music emotion classification based only on lyrics was built using a Support Vector Machine (SVM) classifier with Term Frequency-Inverse Document Frequency (TF-IDF) score as the input lyric feature. For the main experiment, novel word embedding strategies based on transfer learning and Transformers, BERT and XLNet, were finetuned to classify music using lyrics. In addition, for each classifier, the effect of different

[1] http://softeng.polito.it/erion/MoodyLyrics4Q.zip
[2] http://mir.dei.uc.pt/resources/Dataset-Allmusic-771Lyrics.zip
[3] https://genius.com/

combinations of text preprocessing techniques such as lowercase conversion, noise removal, stop-words removal, and stemming or lemmatization, were evaluated.

1.3 Thesis Organization

This thesis contains five chapters. Chapter 1 (the current chapter) introduced MER and the motivation and overview of this thesis research. Chapter 2 first gives the background of MER from the music psychology perspective. Then, it reviews the related works in MER, beginning with an overview of mood as a new type of metadata. Next, it examines the history of two evaluation forums, reviews MER methods based on audio, text, and multimodal features, and ends with an overview of NLP methods. Chapter 3 presents the details of the methodology, including datasets, lyrics collection, evaluation metrics, and a description of machine learning methods. Chapter 4 covers the setup, execution, and results of preliminary experiments and main experiments, and a further discussion on the results. Chapter 5 contains the conclusion of this thesis and provides possible future directions suggested by the results of the experiments.

Chapter 2 Background

This chapter provides the background and popular methods in Music Emotion Recognition (MER). Section 2.1 describes music mood from the perspective of music psychology. Then, Section 2.2 presents the related works in MER. Finally, Section 2.3 gives an overview of Natural Language Processing and its application in Music Information Retrieval, especially in MER.

2.1 Mood in Music Psychology

This section introduces music mood in the view of music psychology and its usage in Music Information Retrieval (MIR). First, Section 2.1.1 provides the nuanced differences and similarities between three terms used in MER: *mood*, *emotion*, and *sentiment*. Next, Section 2.1.2 discusses the sources of mood in music. Section 2.1.3 provides several music psychology principles that are helpful in MER research. Finally, Section 2.1.4 presents two kinds of music emotion representation widely used in MER: categorical representation and parametric representation.

2.1.1 The Terms: Affect, Mood, Emotion, and Sentiment

Affect is a term that refers to a wide range of feelings that people can have. It encompasses both mood and emotion. Both terms, *mood* and *emotion* have been used in music psychology and Music Information Retrieval (MIR) studies to describe the affective impacts of music. Researchers have been focusing on refining the concepts of mood and emotion since the beginning of music psychology research, although they are not entirely unrelated.

In comparison between mood and emotion, moods are distinguishable from emotions by duration. Weld (1912, 283) concluded that “While it is true that the conscious content of the mood is similar to that of the emotion, yet the temporal course and life-history of the former is different

from that of the latter. The emotion is temporary and evanescent; the mood is relatively permanent and stable."

Another main difference between mood and emotion lies in induction. It is usually easy for people to recognize the specific event that triggers a certain emotion whereas it is more difficult to determine the cause of mood (Ekman and Davidson 1994).

Hu (2010, 17) found that the term *emotion* appears to be more prevalent in music psychology while mood seems to be more preferred in MIR research. Hu proposed two reasons for this difference:

> First, MIR researchers try to find the common affective themes of music recognized by the majority, while music psychologists are more interested in human responses to different emotional stimuli. Therefore, MIR studies focus more on the general sentiment that music conveys and choose the term mood over emotion. Second, MIR researchers are searching for a new type of metadata to organize and retrieve music pieces. The research purpose of MIR is on music rather than human responses. It is worth noting that music does not have emotional responses. However, music does convey a certain mood to its audience (Hu 2010, 18).

Although there is a slight difference between them, the two concepts are not completely separated. Hu (2010, 18) also stated that "to some extent, their difference mainly lies in granularity."

Another term, *sentiment*, has also been used in text-involved MIR tasks. According to *The Oxford English Dictionary* (2021), sentiment refers to "A thought or reflection coloured by or proceeding from emotion." In contrast to the term emotion, sentiment is more a resulting opinion

or view towards an object caused by emotion developed over time (Munezero et al. 2014). However, they are frequently used interchangeably as they both refer to experiences resulting from the interaction of biological, cognitive, and social aspects (Stets 2006). Therefore, in this research, the terms mood, emotion, and sentiment are used interchangeably.

2.1.2 Music Mood Sources

The sources of music mood are also a topic of interest to MIR researchers, which is related to music psychology. Is it a result of the intrinsic attributes of music pieces or the extrinsic contexts of music listening behaviours? The answers will significantly impact how mood labels are assigned to music works, whether manually or by computer programs.

The root of the question is whether music meaning lies solely within the context of the musical work itself. There are two opposing perspectives of music meaning in music psychology: absolute and referential views (Meyer 1956, 1). The absolute view sees music meanings as existing only in the context of the music piece. The referential view, however, sees music meanings, which concern the concepts, behaviours, emotional states, and character, as existing outside of music. Meyer (1956, 1) declared that both forms of music meanings exist.

Sloboda and Juslin (2001, 1) reinforced Meyer's idea by proposing two different types of music emotion: "internal" and "external" emotion. Internal emotion is activated by the music's structural attributes, whereas external emotion is evoked by the semantic context associated but outside the music piece. Thus, music mood, they concluded, is a combination of the content and social context of music. In fact, recent MIR studies have started to combine music content, such as audio and lyrics, and social context, such as social tags and reviews (Aucouturier et al. 2007; Hu 2010, 19; Saari et al. 2013b).

2.1.3 Fundamental Principles in Music Psychology

In addition to sources of emotion in music, music psychology studies contain some fundamental principles of music mood perception that can be helpful for MIR research. First, music does have a mood effect on listeners. Early studies have proved that music can change a listener's mood (Capurso et al., 1952). Furthermore, it is widely accepted that assigning mood labels to music pieces appears natural to listeners (Sloboda and Juslin 2001, 94).

Second, different people may perceive certain emotions in music differently. According to Sloboda and Juslin (2001, 94), listeners are often consistent in their perceptions of the emotional effect of music. Schoen and Gatewood (1927, 150) also concluded that music's emotional response is remarkably consistent. Moreover, a study conducted by Lee et al. (2021) analyzed 11,500 emotional ratings of 166 participants from Brazil, South Korea, and the United States over 360 pop songs originating from these three countries. The results showed that basic moods such as sad, joyful, and energetic were rated similarly within and across cultures. It demonstrates that music can be universally classified by certain emotions despite the cultural backgrounds of listeners.

However, music does not evoke all moods equally. To be specific, an experiment by Schoen and Gatewood (1927, 147–148) showed that sadness, joy, and love are more often used than disgust and irritation when describing feelings aroused by music pieces. Furthermore, the level of agreement among listeners is not the same for all the moods. Schoen and Gatewood (1927, 149–150) stated that moods such as joy, amusement, and sadness are relatively consistent among listeners, whereas moods such as disgust and irritation are the moods of low consistency. Edmonds and Sedoc (2021) found that annotators have significantly lower agreement on the mood surprise. Recent research by Lee et al. (2021) demonstrated that there are significant cross-cultural variances

for complex music emotional attributes, such as dreaming and love. For MIR studies, these results imply that some moods will be more challenging to recognize than others.

2.1.4 Music Emotion Representations

Most (70%) of the music and emotion research uses two types of emotion representation: categorical representation and parametric representation (Eerola and Vuoskoski 2013). In a categorical representation, the mood space contains a group of discrete mood categories. The essential idea is that the number of basic emotions is limited (Ekman 1992). Basic emotions are innate and universal emotions, through which all other complex emotional states can be derived (Ekman 1992, 170–171; Damasio 1994, 133). Furthermore, these basic emotions are considered to be adaptable. For example, when dealing with emergencies in life, basic emotions are aroused quickly rather than precisely (Sloboda and Juslin 2001, 76–77).

Ekman (1992, 175–176) defined six universal emotions for facial expressions: anger, disgust, fear, happiness, sadness, and surprise. In the field of music psychology, Hevner (1935) first designed a circle by clustering 66 emotion terms into eight groups (Figure 2.1-1). In the circle, the terms within the same group are similar in meaning, and before hitting a contrast, adjacent clusters differ slightly by accumulating steps. It has been shown that Hevner's cluster circle is consistent despite listeners' music training (Schubert 2003). Another well-known categorization approach is the five clusters (Table 2.1-1) used in Music Information Retrieval Evaluation eXchange (MIREX) Audio Mood Classification (AMC) task (Hu et al. 2008). It is derived by clustering mood labels using popular music from AllMusic.com.[4]

[4] http://www.allmusic.com

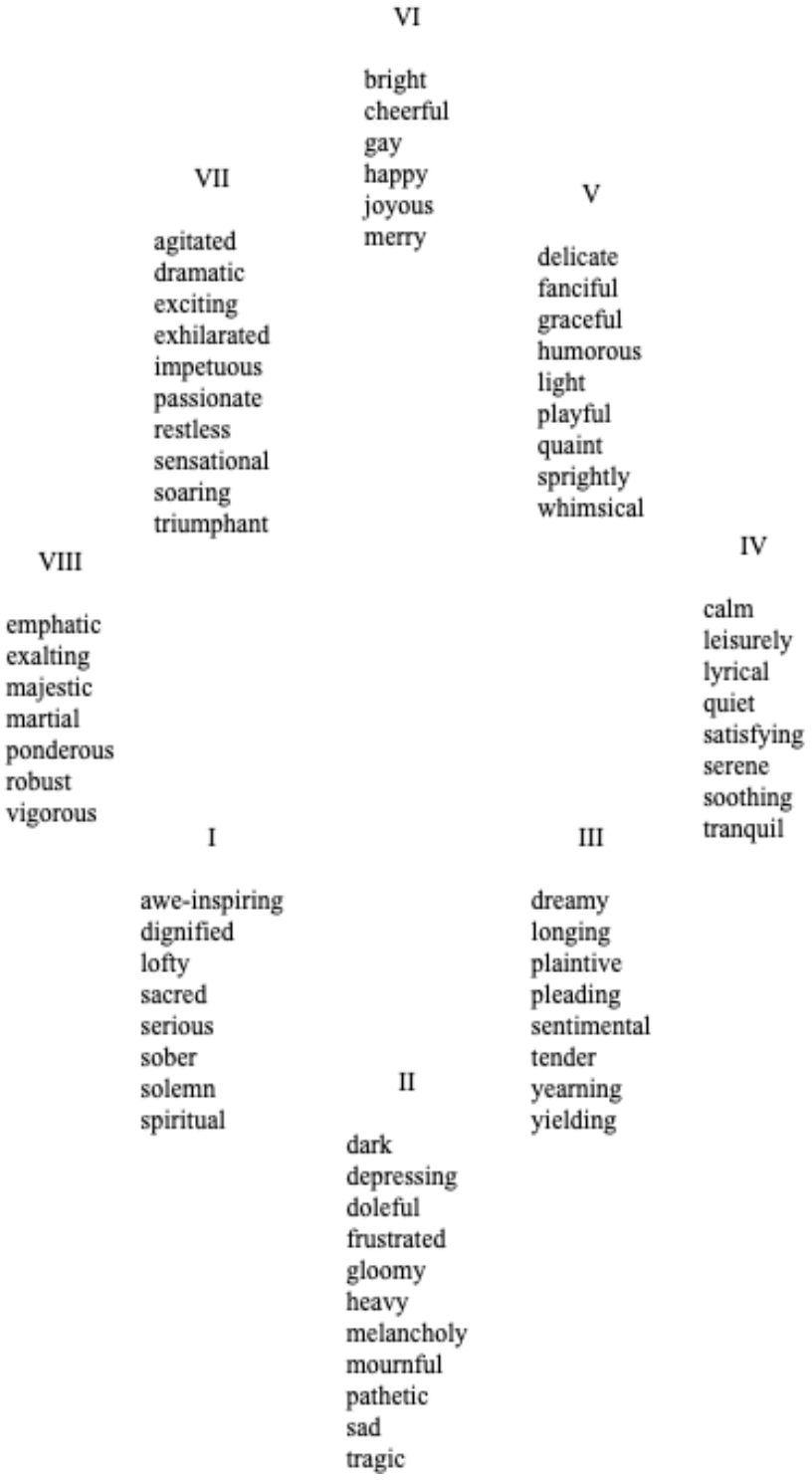

Figure 2.1-1 Hevner's adjective circle. (Farnsworth 1954, 98)

Table 2.1-1 Five mood clusters used in the AMC task (Hu et al. 2008)

Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**
Rowdy	Amiable/ Good natured	Literate	Witty	Volatile
Rousing		Wistful	Humorous	Fiery
Confident	Sweet	Bittersweet	Whimsical	Visceral
Boisterous	Fun	Autumnal	Wry	Aggressive
Passionate	Rollicking	Brooding	Campy	Tense/anxious
	Cheerful	Poignant	Quirky	Intense
			Silly	

A parametric mood representation, on the other hand, continuously measures emotion. The most popular one is the Valence-Arousal (V-A) space (Russell 1980). The V-A space is a two-dimensional space (Figure 2.1-2). Valence describes pleasantness, ranging from negative to positive. Arousal stands for the stimulation level, ranging from low to high. In order to study the influence of internal and external locus on the preference in music, Schubert (2007) first adopted the V-A space to music.

The V-A space has been validated in many studies (Barrett and Russell 1999; Laurier et al. 2009; Russell 1983). In music-related studies, Cespedes-Guevara and Eerola (2018) argued that music can offer affective information including variations of core affect, valence, and arousal, although they do not cover all the affective information that music can provide. Van Zaanen and Kanters (2010) implemented MER using lyrics with class divisions based on the V-A space and claimed that “there is no large difference in mood prediction based on the mood division.”

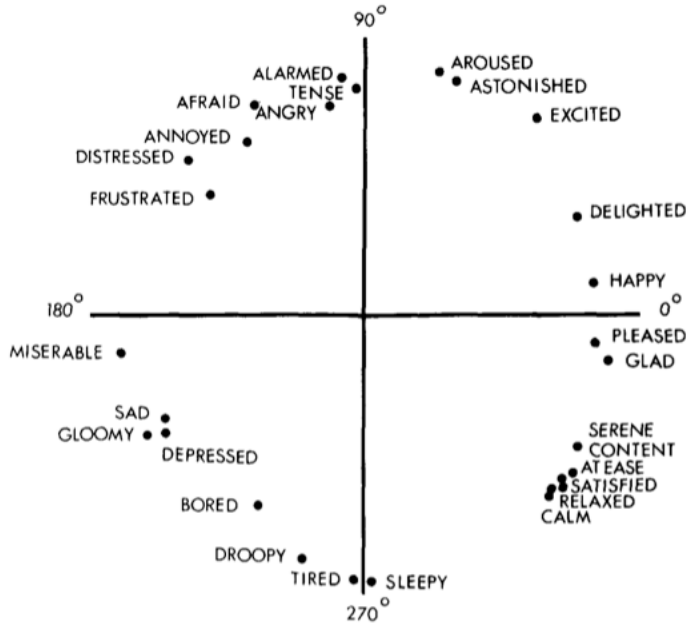

Figure 2.1-2 Unidimensional scaling of 28 affect words on the degree of valence (horizontal axis) and arousal (vertical axis) (Russell 1980)

In addition to two-dimensional models, researchers also favour three-dimensional models of mood. For example, Sjöberg, Svensson, and Persson (1979) proposed that tension is also a basic mood factor besides pleasantness and activation. Schimmack and Grob (2000) supported the need for a three-dimensional model of emotion while it is insufficient to fully capture the structure of affect in all facets.

In fact, categorical and parametric representation cannot be fully separated (Hu 2010, 23; Zentner and Eerola 1993, 200). Gabrielsson and Lindström (2001) claimed that Henver's model revealed an inherent dimensionality similar to valence and arousal in Russell's V-A space. To be specific, valence (sadness – happy) is in the same manner as cluster II – cluster VI, and arousal (exciting/vigorous – serene/dreamy) is similar to cluster VII/VIII – cluster IV/III. Also, Eerola and Vuoskoski (2013) mentioned that the distinctions between happy and sad could also be determined in the valence and arousal, which is the parametric representation.

2.2 Music Emotion Recognition

This section presents the related works in the field of MER. Section 2.2.1 discusses the need and validity of using mood as a type of catalogue metadata in MIR. Section 2.2.2 presents two popular evaluation forums for MER research. Then, MER methods using audio features, text features, and multimodal features are presented respectively in Section 2.2.3, 2.2.4, and 2.2.5.

2.2.1 Mood as Metadata

As stated by Casey et al. (2008), "metadata is the driver of MIR systems." Music information can be accessed by traditional music catalogue metadata, such as title, artist, and genre. With the advent of music streaming and downloading services and the proliferation of music pieces, however, music information search strategies need to be evolved to fulfill the expectations of search and browse capabilities (Casey et al. 2008; Hu 2010, 16; Kim et al. 2010; Yang and Chen 2012). Many studies have pointed out that Music Information Retrieval (MIR) systems need more than traditional bibliographic information for easy and effective access to music information (Casey et al. 2008; Futrelle and Downie 2002; Lee and Downie 2004). Lee and Downie (2004) conducted a user survey about music information needs, uses, and search patterns, and emphasized the importance of contextual metadata.

In recent years, novel metadata has been proposed to organize music efficiently, including perceptual metadata. For example, Hu and Downie (2007) investigated 179 mood labels on AllMusic.com[1] provided by users. Statistical analysis conducted by the authors demonstrated that mood is independent of genre and artist as a new sort of music metadata. Therefore, as a new way to obtain music information, mood may be a helpful complement to traditional metadata.

2.2.2 Evaluation Forums in MER

The most popular evaluation forums in MER are the AMC task in MIREX AMC and the Emotion in Music task held by the Multimedia Evaluation Benchmark (MediaEval). MIREX is a community that has formally evaluated MIR systems and algorithms since 2005. The first AMC evaluation task in MIREX was released in 2007. The MIREX 2007 Mood Classification dataset was selected from the APM[5] collection by the International Music Information Retrieval Systems Evaluation Laboratory (IMIRSEL) at the University of Illinois at Urbana-Champaign (UIUC) (Hu et al. 2008). The set contains a set of 600 audio clips of 30-second long that covers seven major genres. The mood clusters of the audio clips are agreed upon among two out of three human assessors. The audio clips are in 22.05 kHz, mono, and 16-bit WAV format.

In 2013, IMIRSEL and Korea Electronics Technology Institute (KETI) proposed a mood classification dataset with K-POP music. The dataset contains 1,438 songs, each labelled with one of the five mood clusters. However, this dataset is not evenly distributed across the five clusters. In this dataset, each song is annotated by three annotators from the United States and three Korean annotators. In total, there are three independent tasks: one uses annotations by North Americans, one uses annotations by Koreans, and one uses the MIREX 2007 Mood Classification dataset.

MediaEval is a benchmarking initiative that develops and tests novel algorithms and technologies for multimedia analysis, retrieval, access, and exploration. The first Emotion in Music task was released in 2013, composed of two subtasks (Soleymani et al. 2013). The first subtask is to detect arousal and valence on a nine-point scale for each song with multimodal features. The second subtask is to continuously determine the value on arousal and valence

[5] http://www.apmmusic.com

dimensions for the given song per frame. Compared to MIREX, the music used in this task is creative common licensed music from Free Music Archive (FMA)[6] covering different genres of western music. Experimental and low-quality music is excluded to avoid controversial annotations. The dataset consists of 744 songs of 45 seconds. The ground truth is created by human assessors and collected on Amazon Mechanical Turk.

In 2014, an evaluation set of 1000 new clips was released. In addition, the time resolution for the dynamic task was improved to 2 Hz. In addition, the static emotion characterization task was removed. A new subtask, feature design, was added instead. In the following year, the task was composed of three new subtasks: subtask 1 is to find the best regression approach independently of the feature set used, such as linear regression, Feedforward Neural Network (FNN), and Recurrent Neural Network (RNN); subtask 2 is to find the best feature set for the time-continuous estimation independently of the model used; subtask 3 is to find the best global approach. Until 2015, two evaluation metrics were used to compare the performance: Pearson's correlation coefficient and Root Mean Square Error (RMSE).

Then, after a three-year break, the next MediaEval task about music emotion, Emotion and Themes Recognition in Music Using Jamendo, was published in 2019. This time, categorical representation is used instead of dimensional representation. The dataset used is the "autotagging-moodtheme", which is a subset of the MTG-Jamendo dataset7 available under Creative Commons Licenses. The subset contains 18,485 audio tracks in MP3 format annotated with 57 tags. Each

[6] http://freemusicarchive.org/
[7] https://github.com/MTG/mtg-jamendo-dataset

track can possibly have more than one tag. The submission was evaluated using typical performance metrics. The task was also performed in 2020 and 2021.

2.2.3 Audio-based Music Emotion Recognition

The majority of current work on automatic music emotion recognition is based on audio features. According to Panda, Malheiro, and Paiva (2020a), standard audio features available in popular frameworks usually relate to the eight musical concepts: rhythm, dynamics, expressive techniques, melody, tone colour, musical texture, and musical form. Most features are related to tone colour, which concerns timbre. Timbral features are musical surface features that encapsulate the statistic information hidden in the waveform or the spectrum of the signal (McEnnis et al. 2005; Panda, Malheiro, and Paiva 2020a). Here are some common ones:

1) Spectral centroid is the mean of amplitudes in the spectrum of the signal, which is related to the "brightness" of the timbre.

2) Spectral rolloff is the frequency below, which is 85% of the spectral energy. It indicates the skewness.

3) Spectral flux is the spectral correlation between adjoining time frames. It measures how quickly the spectrum changes.

4) Mel Frequency Cepstral Coefficients (MFCCs) are the coefficients of the Mel Frequency Cepstrum (MFC). It is the linear cosine transform of a log power spectrum on the mel scale. The mel scale is based on the human auditory system. MFCC can represent the spectral envelope of a musical signal, which is closely related to the timbre.

In contrast to these low-level features, Panda, Malheiro, and Paiva (2020a) also discussed future research directions for higher-level audio features related to playing techniques and musical layers. Some of those include tone colour, musical texture, rhythms, and expressive techniques.

Tone colour, also known as timbre, is related to the properties of the sound itself. They are crucial in distinguishing the instruments used. The authors proposed that higher-level tone-colour features that capture the types of amplitude envelope (e.g., round, sharp) would be beneficial.

Musical texture pertains to how rhythmic, melodic, and harmonic information are combined in the music work, thus related to the relations between the musical lines or layers. The authors introduced texture features describing the number and density of musical layers.

Rhythm is the musical component of time, reflecting the patterns of long or short notes and rests in music. Therefore, the authors presented some rhythmic features relevant to the tempo and note values, rhythm types, and rests in music. They also suggested that higher-level features about the types of rhythm may be helpful.

Expressive techniques refer to the ornaments and features composers and performers employ to enrich music pieces or express emotions. Expressivity features that the authors proposed illustrate the articulation, ornamentation, vibrato, and tremolo in the music work. Future work about expressivity features can be on the detection of ornamentations other than glissando and portamento, such as trill and slide.

Continuing on their work, Panda, Malheiro, and Paiva (2020b) built an SVM to classify music pieces based on the four quadrants of Russel's model (Russell 1980) using both low-level baseline features and the higher-level features they proposed. The results showed that adding novel features improved the F1-score by 9% compared to baseline-only features. In addition, the authors

ranked both base and novel audio features. The results showed that tone colour features make up the majority of the top five features for each quadrant except quadrant 1. Other than tone colour, texture, rhythm, and tremolo also ranked high.

Classification frameworks used in audio-based MER, that performed well on the MIREX AMC task before the boom of neural networks, were SVM and Gaussian Mixture Model (GMM). Over three-fold cross-validation, a linear SVM by the Marsyas submission (Tzanetakis 2007) ranked the first with an average accuracy of 61.50% in 2007. In 2008, a GMM system using audio features achieved 63.67% and was ranked first (Peeters 2008). In the following year, Cao and Li (2009) proposed to combine SVM and GMM. The input contained audio features, such as MFCC and rhythm patterns. The combined system achieved 65.67% on mood classification. In 2013, Wu and Jang (2013) combined the audio-based Gaussian Super Vector (GSV), visual, and acoustic features. The GSV feature converts the audio input into a set of GMM parameters to present global timbre characteristics. The visual features capture the texture of a spectrogram both locally and globally. The concatenated feature vector was input into an SVM classifier. The proposed model reached an average accuracy of 68.33% and won the contest.

Recently, neural networks have become the most popular method in MER. There are mainly three types of neural networks: Feedforward Neural Network (FNN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN).

FNNs are neural networks where data flows in the forward direction from the input layer to the output layer (see Section 2.3.4.2). CNN specializes in image analysis. It uses a sliding kernel function to extract features from images. RNN is designed for sequence data. In RNN, the data flows in a cycle. Therefore, it has access to previous information when making a prediction. In

other words, RNN has the memory of the information that it has seen before. Long Short-Term Memory (LSTM) is a special kind of RNN that can learn the long-term dependencies across time.

The model by Wu and Jang (2013) maintained its first place until 2017, when the neural network first appeared in the AMC task. Park et al. (2017) applied transfer learning to categorize music mood. They first trained a Deep Convolutional Neural Network (DCNN) on mel-spectrogram to identify different artists. The extracted features were then used for training an SVM classifier on the MER task. The proposed approach achieved 69.93% in 2017 and holds the record on MIREX to this day (2021).

For the MediaEval Emotion in Music task, the best models in terms of RMSE in the years 2013 and 2015 for both arousal and valence dimensions are based on Long Short-Term Memory Recurrent Neural Networks (LSTM-RNNs) (Weninger, Eyben, and Schuller 2013; Xu et al. 2015).

However, in 2014, the best model for the arousal dimension is a State-Space Model (SSM) based on the Kalman filter and Gaussian Process proposed by Markov and Matsui (2014). The LSTM-RNN-based model (Coutinho et al. 2014) ranked first in terms of the valence dimension and second for arousal.

In addition, the Convolutional Recurrent Neural Network (CRNN) structure is also quite popular. For example, Malik et al. (2017) applied CRNN on the MediaEval 2015 Emotion in Music Dataset (Aljanaki, Yang, and Soleymani 2014). The network is one convolutional layer followed by two Recurrent Neural Network (RNN) branches, respectively trained for arousal and valence. The performance was the best reported on the dataset till then. Moreover, Choi et al. (2017) compared three Convolutional Neural Network (CNN) structures and a CRNN structure on the Million Song Dataset (Bertin-Mahieux et al. 2011). The result showed that CRNN outperformed all the CNNs at the expense of computation speed.

2.2.4 Text-based Music Emotion Recognition

Researchers have also tried to retrieve music mood using contextual text information, such as comments, social tagging, and lyrics (An, Sun, and Wang 2017; Bischoff et al. 2009b; Kaminskas and Ricci 2012, 109).

Golder and Huberman (2006, 198) defined social tagging as "the process by which many users add metadata in the form of keywords to shared content." Lamere (2008) analyzed the distribution of tags on last.fm and argued that tags might be the critical solution to many challenging MIR problems, such as MER. Hu et al. (2007) created a set of music mood categories and also a ground truth set corresponding to the categories using tags on artist, album, and tracks on last.fm. Furthermore, Laurier et al. (2009) created a semantic mood space from last.fm tags using Latent Semantic Analysis (LSA). The results demonstrated the relevance of basic emotions: happy, sad, angry, and tender, with the social network.

Other than social tagging, lyrics are also an important part of vocal music that MER researchers have been exploiting. For example, Yang and Lee (2009) used a content analysis package to convert lyrics into psychological feature vectors. They concluded that lyrics alone can be used to generate human-comprehensible classification models.

To perform MER using lyrics, researchers have developed several datasets. Hu, Bay, and Downie (2007) derived a simplified ground-truth set of three mood clusters from last.fm[8] and the USPOP collection[9] using the K-means clustering method. Moreover, to fill the absence of emotions such as surprise or fear in lyrics MER datasets, Edmonds and Sedoc (2021) created a

[8] https://www.last.fm/
[9] https://www.ee.columbia.edu/~dpwe/research/musicsim/uspop2002.html

novel lyrics dataset of 524 songs on Spotify[10], labelled based on Plutchik's eight core emotions: anger, anticipation, disgust, fear, joy, sadness, surprise, and trust (Plutchik 2001). In addition, the dataset by Mihalcea and Strapparava (2012), originally based on Ekman's six core emotions (Ekman 1993), was reannotated using Plutchik's eight emotions at the song level.

Early works on MER using lyrics are mainly based on traditional lexicons and traditional machine learning methods. For instance, Van Zaanen and Kanters (2010) extracted both global and word-based features from lyrics. Global features include character count, word count, and line count. Word-based features contain the Term Frequency-Inverse Document Frequency (TF-IDF) (See Section 2.3.3.1) metric and its components. They also experimented TF-IDF with Laplace smoothing and normalized TF-IDF. The dataset of 5,631 lyrics was provided by the Moody application of Crayonroom.[11] The lyrics were annotated based on 16 partitions of the V-A space. Different class division methods were also implemented. The first-class division, the fine-grained division, uses all 16 classes. The second-class division only focuses on one aspect of the V-A space. The third-class division, the Thayer division, is based on the four quadrants of the V-A space. A k-Nearest Neighbors (kNN) classifier, TiMBL, developed by Daelemans et al. (2002), was used to test the proposed features. The results showed that over 10-fold cross-validation, all the TF-IDF-based features outperformed all global features and their combinations using every class division. The combination of TF and TF-IDF achieved the best accuracy. These results demonstrated that word-oriented metrics only based on lyrics provide a valuable source of information.

[10] https://www.spotify.com/us/
[11] http://www.crayonroom.com

In recent years, pre-trained models, such as GloVe (Pennington, Socher, and Manning 2014) and transformers (Devlin et al. 2019; Yang et al. 2019), have shown promising results in downstream NLP tasks. For example, Akella and Moh (2019) compared traditional and novel machine learning approaches. For traditional methods, they implemented Random Forest with TF-IDF vectors. For deep learning approaches, GloVe vectors of lyrics were used as input for CNN, CRNN, and Bidirectional Long-Short Term Memory (Bi-LSTM). The results showed that CNN with GloVe achieved the best accuracy. Abdillah et al. (2020) also implemented GloVe pre-trained word embeddings on English lyrics. The dataset used was the MoodyLyrics dataset (Çano and Morisio 2017a), which is annotated based on the four quadrants of Russell's model (Russell 1980). The lyrics were first preprocessed with lemmatization, tokenization, stop-word removal, and lowercase conversion. Then, the preprocessed lyrics were used for a GloVe pre-trained word embedding layer, followed by a Bi-LSTM layer. Finally, a dense layer was used as an output layer to produce the mood class. The proposed method was compared with Naïve Bayes, kNN, SVM, LSTM, and CNN. The results showed that the proposed approaches achieved the highest accuracy of 91%.

Although most of the lyrics-based analyses used English lyrics, there is other research using languages, such as Chinese, Korean, and Hindi. For Chinese lyrics, He et al. (2008) compared traditional bag-of-words features in unigrams, bigrams, trigrams, and their combinations with three feature representation models: Boolean, absolute term frequency, and TF-IDF weighting. The dataset was provided by YY Music Group.[12] It consists of 1,903 lyrics of Chinese pop music, of which 803 songs are labelled with love, and 1,100 songs are labelled as lovelorn.

[12] https://www.yy.com/

Three machine learning methods, Naïve Bayes, Maximum Entropy (ME) Classification, and SVM, were employed. The results showed that ME with the combination of unigram, bigram, and trigram with TF-IDF weighting achieved the best accuracy, indicating that higher-order bag-of-words features (e.g., bigrams and trigrams) have the ability to capture semantics in terms of mood classification. Moreover, Hu, Chen, and Yang (2009) developed an affective lexicon, called Affective Norms for Chinese Words (ANCW). It is composed of words translated from the Affective Norms for English Words (ANEW) (Bradley and Lang 1999). They used a fuzzy clustering method to classify 500 Chinese songs into four mood categories based on Russell's V-A space.

Kim and Kwon (2011) proposed a lyric-based method using feature extraction based on syntactic analysis of Korean songs. To extract emotion features from lyrics, four kinds of syntactic analysis rules were applied: negative word combination, time of emotions, emotion condition change, and interrogative sentence. Tested on two corpora, the precisions were improved after applying syntactic rules. With the emotion features extracted using the four rules, emotion classification was implemented on 425 songs using SVM, Naïve Bayes, and Hidden Markov Model (HMM). The proposed system achieved the maximum accuracy of 58.8% with SVM using eight emotion categories over 10-fold cross-validation.

For Hindi music, Patra, Das, and Bandyopadhyay (2015) proposed a five-class mood taxonomy based on Russell's model (Russell 1980). Then, they created a corpus of lyrics annotated using both the taxonomy and the polarities (positive and negative). Next, they extracted textual features based on sentiment lexicons, word counts, and n-grams. Finally, an SVM with Radial Basic Function (RBF) kernel was implemented. The model achieved the best F-measure of 68.30% for the polarities and 38.49% for the mood taxonomy.

2.2.5 Multimodal Music Emotion Recognition

In general, information about the same phenomenon can be acquired from various detectors in different contexts. Each of these acquisition frameworks can be said to obtain a different *modality* (Lahat, Adali, and Jutten 2015). Natural phenomena usually have diverse properties, which makes it uncommon to provide comprehensive knowledge of a phenomenon from a single modality. Therefore, researchers have been integrating multiple modalities to understand the phenomena, such as climate change impact (Beuthe et al. 2014).

Multimodal Learning has drawn great attention in recent years in domains, such as Human-Computer Interaction (Jaimes and Sebe 2005), image analysis (Guo, Wang, and Wang 2019), as well as MER. Popular modalities in MER include audio, lyrics, and user tags. One of the early works on Multimodal MER is Laurier, Grivolla, and Herrera (2008), where the researchers investigated both audio and lyrics information for MER studies. They first developed a dataset of 1,000 pop songs divided into four categories based on Russell's four quadrants: happy, sad, angry, and relaxed. For the audio approach, timbral, rhythmic, tonal, and temporal features were extracted and passed to an SVM classifier with a polynomial kernel. For lyrics, the authors implemented two unsupervised learning methods, the standard distance-based method and LSA. In addition, they proposed a method based on the most discriminative terms between language models. The results showed that the two unsupervised learning methods are not as good as the audio-based approach, though they are better than a random guess. Moreover, the third method achieved similar performance to the audio-based method. Furthermore, combining the audio and lyrics features in the same space in a multimodal system showed a significant improvement in the overall performance.

Hu, Downie, and Ehmann (2009) created a ground truth of 5,585 songs divided into 18 mood categories based on social tags. In addition, they found the best performing lyric feature set, among bag-of-words, Part of Speech (POS) and Function Words, was bag-of-words with stemming and TF-IDF weighting (BSTI). They combined this lyrical feature with 63 spectral audio features, which performed best in the MIREX 2007 AMC task. Spectral features, BSTI, and the combination of the two were compared with an SVM classifier over 10-fold cross-validation. The results demonstrated that audio features did not always get higher accuracy than lyric features on MER, and that combining lyric and audio features can improve MER accuracies, but not always. Furthermore, the results showed that, with simply training on lyric features, the combination of lyric and audio features cannot always improve MER performance.

With the same dataset, Hu and Downie (2010) combined basic lyrics features, linguistic features, and audio features using two fusion methods: feature concatenation and late fusion. SVM was chosen as the classifier. The results demonstrated that combining lyrics and audio features can improve system performance, and that the hybrid system can achieve the same or better accuracies with fewer training data than the systems using only lyrics or audio features.

Lu et al. (2010) proposed a two-layer classification method called Fusion by Subtask Merging (FSM), using AdaBoost to classify arousal and valence separately and merging the result. They compared MIDI, audio, lyrics features, and their combination with a dataset of 500 Chinese pop songs divided into four classes based on Russell's model. The results showed that combining three feature domains achieved the highest accuracy.

Wang et al. (2011) analyzed information from Chinese lyrics and rhyme to classify music emotion in the V-A space. A TF-IDF-based new feature space, proposed by Van Zaanen and Kanters (2010), was combined with Part-of-Speech features. Sequential Minimal Optimization

(SMO) SVM was implemented on the proposed lyric feature. In addition, a Naïve Bayes Classifier was applied to the rhyme feature. Both feature-level fusion and classifier-level fusion were evaluated and compared. The results show that feature-level fusion cannot improve performance. As for the classifier-level fusion, the meta-learning method outperforms the baseline learning method.

More recently, Xue, Xue, and Su (2015) incorporated audio and lyric information using a Hough Forest-based fusion and classification schema. The lyrics features were extracted with a Recursive AutoEncoder model after filtering. The method was implemented on a dataset of 781 songs with a total duration of 3,000 minutes.

Other than Western and Chinese songs, Patra, Das, and Bandyopadhyay (2016) have also compared features of audio, lyrics, and their combination on Hindi music. They collected a dataset of 210 60-second clips divided into five categories: angry, calm, excited, happy, and sad. An SVM was used as the classifier. Over 10-fold cross-validation, the results showed that the multimodal system achieved the maximum F-measure.

Apart from integrating audio and lyrics, Bischoff et al. (2009a) have tried to combine audio and social tag information. They implemented MER using audio features and social tags from last.fm[5] based on both MIREX mood clusters and Russell's four quadrants. The classifier for audio features is SVM with Radial Basis Function (RBF) kernel, and for social tags is Naïve Bayes Multinomial. The multimodal method is the linear combination of the two separate classifiers. The results on 1,612 songs showed that the tag-based classifier performed better than the audio-based classifier. In contrast, the multimodal classifier achieved the best performance in terms of recall, precision, F1-Measure, and accuracy.

In addition, Saari et al. (2013b) evaluated the audio-based Semantic Layer Projection (SLP) method proposed in Saari et al. (2013a) for predicting mood ratings with a tag-only system, Affective Circumplex Transformation (ACT), and the combination of semantic tags and audio features in automatic mood annotation. SLP converts audio features into a low-dimensional semantic layer that corresponds to the circumplex model proposed by (Russell 1980). The authors trained SLP on audio features extracted from I Like Music[13] production music corpus with tags provided by last.fm.[8] Based on LSA, ACT technique projects mood terms onto the V-A space. The results suggest that although the audio-based system alone had a good performance, the combination of audio features and tags led to a remarkable increase.

2.3 Natural Language Processing

As discussed in the section above, researchers have been using Natural Language Processing (NLP) methods to analyze contextual texts for MER. This section gives an overview of NLP methods and their applications. First, Section 2.3.1 provides the general definition and workflow of NLP, as well as its applications in poetry and lyrics. Next, Section 2.3.2 presents the text preprocessing stage in the NLP workflow. Then, feature construction methods are discussed in Section 2.3.3. Finally, Section 2.3.4 describes two types of popular classifiers in NLP: SVM and neural networks.

2.3.1 Natural Language Processing

Natural Language Processing (NLP) refers to a set of techniques that strives to make computers understand human language (Eisenstein 2019, 1). NLP encompasses a wide range of

[13] https://web.ilikemusic.com/

tasks and methods. Typical NLP applications include text classification, machine translation, sentiment analysis, and question-answering (Chowdhary 2020, 12).

In the field of NLP, there have been numerous works on poetry computational analysis and classification. For instance, Tizhoosh, Sahba, and Dara (2008) extracted four groups of features: rhyme, meter, shape, and meaning for poetry recognition. Agirrezabal, Alegria, and Hulden (2016) analyzed the rhythmic pattern in poetry using machine learning algorithms, including Naïve Bayes and SVM. Kesarwani (2018) established a poem classification system based on rhyme, diction, and metaphor. Ahmad et al. (2020) classified poems based on emotional states using an attention-based Convolutional Bi-LSTM model.

In addition to poetry analysis and classification, poetry generation is also an NLP task that researchers are interested in. For example, Ghazvininejad et al. (2016) used a Finite-start acceptor together with RNN to generate poetry based on user-provided topics. Also, Van de Cruys (2020) implemented RNN with poetic constraints to generate poems automatically.

As a special form of literature, poetry is similar to lyrics in the sense of similarity features, such as rhyme, diction, and metaphor. Researchers have been using NLP methods to tackle MIR tasks involving lyrics, such as lyric structure and content analysis, as well as lyric perception (Fell, 2020).

MER on lyrics using categorical representation can be regarded as a text classification problem in NLP. A traditional NLP framework for text classification is usually composed of three stages: text preprocessing, feature construction, and classification. The text preprocessing stage often contains tasks including tokenization, stop-word removal, noise removal, etc. The feature construction stage maps text into a vector of numerical values. Finally, the classification stage utilizes pattern classification algorithms, such as SVM and neural networks.

In conclusion, NLP aims to help computers interpret text and speech in a manner similar to that of human beings. Poetry is a popular research object in NLP. Lyrics have similarities with poetry and have also received much attention in the cross field of NLP and MIR. MER on lyrics is a task combined with MIR and text classification. The following sections will introduce each step in the text classification workflow in detail.

2.3.2 Text Preprocessing

Text preprocessing is one of the critical components in NLP classification tasks. It cleans the text data and prepares the data for the model. Text preprocessing methods for English generally include lowercase conversion, noise removal, stop-word removal, stemming or lemmatization, and tokenization.

Lowercase conversion is usually one of the first steps in English text preprocessing. In most NLP tasks, it is accepted that there is no difference between uppercase or lowercase forms of words (Uysal and Gunal 2014). Therefore, prior to any actions, the text is often converted into the same case, preferably lowercase, to avoid information loss due to case differences.

Another widely used first preprocessing step is noise removal. The purpose of noise removal is usually to clean the text data. In this step, all the punctuations, numbers, links, whitespaces, and tags will be removed from the text.

Stop-words are the words that frequently appear in texts but are not related to a specific topic. Typical stop-words include conjunctions, prepositions, articles, etc. These words usually have little or no meaning. As a result, stop-words are considered irrelevant in text classification tasks and are removed during the preprocessing stage.

After text cleaning, the remaining text is ready for the next stage of text preprocessing. Different forms of a word, but with similar meanings, are often used in documents for grammatical reasons. In many cases, it would be helpful for computers to consider these words as a set. Both stemming and lemmatization are procedures that reduce inflectional forms of a word to a common base form (Manning, Raghavan, and Schütze 2008, 32).

Stemming reduces inflection toward their root forms by slicing the words from prefixes or suffixes. For example, the stem of the word "studies" is "studi", and the stem of "studying" is "study". There are many different stemmers available in different languages. One of the most common stemmers for English is the Porter Stemmer, which strips suffixes (Porter 1980). The Porter Stemmer is a rule-based approach. It is composed of five stages of word reductions, within which there are several protocols to select rules.

On the other hand, lemmatization is a dictionary-based approach. It considers the morphology of the words and recovers the words to their canonical or dictionary forms, also known as *lemma*, so that they can be analyzed as a single item. It is worth noting that a lemma is the root form of all its inflectional forms. For example, lemmatization will transform both the word "studies" and "studying" into lemma "study". The WordNet Lemmatizer is one of the most popular lemmatizers. WordNet is a large and publicly available lexical database for the English language with lemmatization features (Miller 1995).

Before feature construction, tokenization is often used as the last step of text preprocessing. Tokenization is the procedure of chopping up sentences into small pieces, called *tokens*. That is to say, tokenization is a form of text segmentation. For example, words, numbers, and punctuations can all be regarded as tokens, which is why numbers and punctuations are usually removed from

the text in the very first step in text preprocessing, for the tasks where they are unrelated to the content and considered noises.

To sum up, text preprocessing is an important step in NLP that converts text into a simpler format and helps improve model performance. Text preprocessing for the English language starts with transforming all letters into the same case, then removing potential noise, converting words to their base forms, and ends with segmenting sentences. It should be noted that not every step of text preprocessing must be performed. Different NLP tasks require different text preprocessing processes (Uysal and Gunal 2014).

2.3.3 Feature Construction

Feature construction encodes the text data into feature vectors for the model. The feature construction stage in NLP usually uses the vector space model proposed by Salton, Wong, and Yang (1975), which represents documents as vectors in the same vector space, to handle the information retrieval operations (Manning, Raghavan, and Schütze 2008, 120). These vectors are known as word embeddings.

Popular word embeddings include Term Frequency-Inverse Document Frequency (TF-IDF), BERT embeddings, and XLNet embeddings. In this section, these three types of word embeddings are described respectively in Section 2.3.3.1, 2.3.3.2, and 2.3.3.3. Finally, Section 2.3.3.4 presents a brief summary of the word embeddings introduced in this section.

2.3.3.1 TF-IDF

TF-IDF is a statistical method that measures the importance of a word to a document. Proposed by Salton and Buckley (1988), TF-IDF is based on the bag-of-words method by Harris (1954), which only describes the occurrence of words, whereas the exact order of words is ignored.

The general idea of TF-IDF is that a term is important to a document if it appears many times in the document, and it is a relatively rare word overall. To be specific, Term Frequency $TF(t,d)$ is usually the count of the term t divided by the length of the document d. The Inverse Document Frequency $IDF(t,D)$ is the number of documents N in the corpus D divided by the number of documents that contain the term t in the logarithmic scale:

$$IDF(t,D) = \log \frac{N}{|\{d \in D : t \in d\}| + 1}, \quad (1)$$

where the 1 in the denominator serves as a smoothing factor, to avoid the division-by-zero situation when the term *t* is not in the corpus. From the expression, it is worth noting that for the same corpus, the IDF score of the same word is the same.

Then, the TF-IDF score is the product of TF and IDF:

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D). \quad (2)$$

TF-IDF has been widely used for various NLP tasks. For example, Bafna, Pramod, and Vaidya (2016) used TF-IDF along with hierarchical agglomerative clustering and fuzzy K-Means to cluster different types of documents. In the field of behavioural finance using NLP techniques, Chen and Chen (2019) developed a public mood dynamic prediction model for stock prices by analyzing financial blogs and news article content with TF-IDF.

Moreover, Van Zaanen and Kanters (2010) combined TF-IDF with Part-of-Speech as a lyrical feature and implemented SMO SVM on the proposed feature to classify music based on emotion. Maheshwari, Bhaveshbhai, and Halder (2021) used TF-IDF to analyze the underlying lyric trend by finding unique evergreen words.

In addition, TF-IDF has also been implemented for MIR tasks other than MER, such as music recommendation using lyrics (Nakamura, Fujisawa, and Kyoudou 2017), music genre classification (Aryafar and Shokoufandeh 2011), and artist identification (Aryafar and Shokoufandeh 2014).

2.3.3.2 Bidirectional Encoder Representations from Transformers (BERT)

The lack of enough training data has been one of the biggest challenges in deep learning. Although a vast amount of text data is available, only a few thousand or a few hundred thousand annotated samples are left when we split it into datasets specific for different tasks, such as lyrics analysis, spam detection, and film comment analysis. However, deep learning-based NLP methods need significantly more data to function well. Transfer learning (Yosinski et al. 2014) thus emerged to bridge the need for massive data of machine learning methods and the lack of annotated data.

The idea of transfer learning is to pre-train a general-purpose language representation model using large-scale unlabeled text and then finetune it for specific downstream tasks using smaller datasets. BERT is a popular pre-trained model proposed by Devlin et al. (2019).

The structure of BERT is based on a Transformer architecture (Figure 2.3-1). Proposed by Vaswani et al. (2017), Transformer is a machine-learning model designed for NLP that uses an attention mechanism to learn the contextual relationships between all words in its input. It is composed of an encoder and a decoder. The encoder compresses the input text into a compact representation, and the decoder reconstructs the original text. The attention mechanism helps the model to utilize the most relevant segment of the input sequence.

To construct word embeddings that best represent the input language, BERT is composed of several encoders, each of which contains three parts: the input part, the multi-head attention mechanism part, and the Feedforward Neural Network (FNN) part.

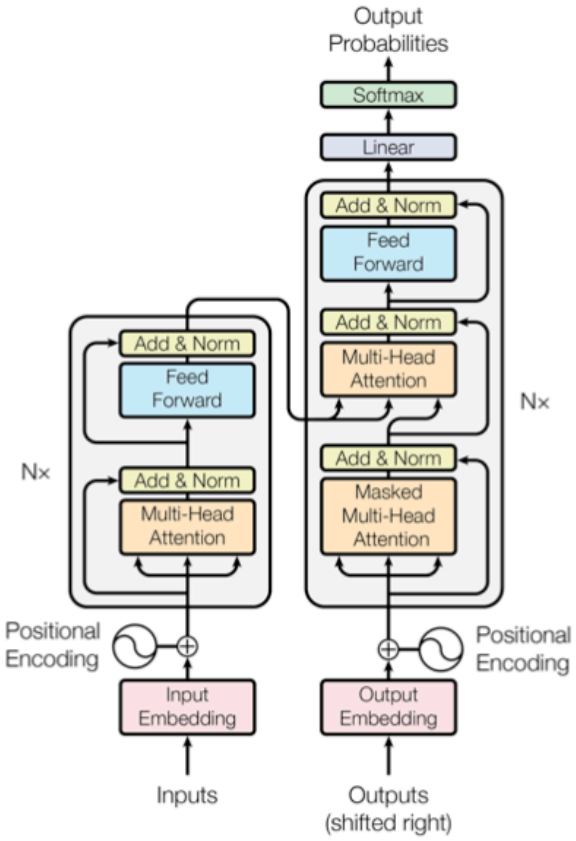

Figure 2.3-1 The model architecture of Transformer (Vaswani et al. 2017, 3)

BERT is pre-trained using two autoencoding tasks, Masked Language Modelling (MLM) and Next Sentence Prediction (NSP) (Devlin et al. 2019). Autoencoding tasks are unsupervised tasks that seek to reconstruct the original data from compressed input. In the input of pre-trained BERT, 15% of the tokens in each sequence are masked out (Devlin et al. 2019). However, the masked tokens will create a discrepancy between pre-training and finetuning. Therefore, to alleviate this, the masked tokens are not always replaced with the [MASK] token if chosen. To be specific, if a token is chosen as a masked token, it will (1) be replaced with the [MASK] token

80% of the time (2) be replaced with a random token 10% of the time (3) remain unchanged 10% of the time.

The MLM task attempts to fill in the original value of these randomly masked tokens based on their surroundings to learn the relationships between words, whereas NSP attempts to understand the relations between sentences by predicting the following sentence. Unlike directional models that read text sequentially, BERT can be considered non-directional since it learns the context of a token based on both sides at the same time. In other words, this character enables BERT to learn the contextual information of a token regarding all other tokens in the sentence.

The pre-trained BERT generates contextualized embeddings that can be used for different NLP tasks, such as machine translation (Imamura and Sumita 2019) and text ranking (Yates, Nogueira, and Lin 2021). Pre-trained BERT has also been used on MIR tasks, including symbolic music understanding (Zeng et al. 2021) and English lyrics analysis (Fell et al. 2019; Fell 2020; Sung and Wei 2021).

However, BERT has certain limitations. The masked tokens used by BERT in pre-training are absent from real data in the finetuning process, which leads to a discrepancy between pre-training and finetuning (Yang et al. 2019). Moreover, BERT neglects the high-order and long-range dependencies that exist in natural language as it hypothesizes that the predicted tokens are independent of each other given the non-masked tokens (Dai et al. 2019).

2.3.3.3 XLNet

As mentioned in the previous section, BERT is an autoencoding method with access to contextual information of the whole sentence but suffers from losing relative position information. Autoregressive language modelling is a popular pre-training objective that can preserve relative

position information (Yang et al. 2019). It aims to estimate the conditional distribution of each sentence. In autoregressive language modelling, the likelihood of a text sequence is factorized into a product over the words. However, the probability of each word depends on the words either before or after it. Therefore, the autoregressive language model is not effective at learning contextual information on both sides of a token.

Proposed by Yang et al. (2019), XLNet is a generalized autoregressive pre-training model that combines the benefits of both BERT and autoregressive language modeling while avoiding their pitfalls. XLNet integrates the ideas of autoregressive language modelling and Transformer-XL structure proposed by Dai et al. (2019).

Transformer-XL is an attentive language model that can learn longer dependency than Transformers. XLNet implements an architecture called Two-Stream Self-Attention (TSSA). The two streams in TSSA are content stream attention, which is the standard hidden states in Transformer, and query stream attention, which uses the position information. TSSA borrows both ideas of the segment recurrence mechanism and the relative positional encoding scheme from Transformer-XL. The segment recurrence mechanism stores the information of the current layer into a memory block and feeds it to the next layer. It helps the model to process longer sequences. The relative positional encoding scheme provides a temporal hint to the model on where to pay attention to.

The input of the XLNet pre-training process is the same as BERT, though the pre-training process is different. The pre-train objective of XLNet is a generalized autoregressive method called Permutation Language Modeling (PLM). PLM trains an autoregressive model on all possible permutations of words in a sentence. The pre-training process of XLNet tries to maximize the expected log-likelihood over all possible permutations. In this way, PLM enables XLNet to capture

bidirectional context information of a sequence from all positions, instead of non-directional modelling as BERT and losing relative position information.

Dai et al. (2019) state that XLNet outperforms BERT on twenty NLP tasks using similar experiment settings, including language understanding tasks, reading comprehension tasks, text classification tasks, and document ranking tasks. XLNet has also been used in some MIR tasks, such as music score infilling (Chang, Lee, and Yang 2021) and MER on lyrics (Agrawal, Shanker, and Alluri 2021)

2.3.3.4 Summary

This section introduced three common word embeddings used in NLP. Section 2.3.3.1 presented TF-IDF, which is based on the importance of the word. A term is considered important to a document if it appears frequently throughout the document and is a relatively uncommon word in general. Section 2.3.3.2 and Section 2.3.3.3 described the pre-trained embeddings that learn the representation of language. Specifically, Section 2.3.3.2 presented BERT, a pre-trained model based on Transformer. Section 2.3.3.3 introduced XLNet, a pre-trained model based on Transformer-XL.

2.3.4 Classifiers

As the last stage in the NLP workflow, the classification stage maps the features into different categories, and yields the final classification result. This section presents two popular classifiers in NLP. To be specific, Section 2.3.4.1 discusses SVM, and Section 2.3.4.2 describes FNN. Finally, Section 2.3.4.3 provides a summary of this section.

2.3.4.1 Support Vector Machine

As mentioned in Section 2.2, Support Vector Machines (SVMs), proposed by Boser, Guyon, and Vapnik (1992), were a popular machine learning method used in MER tasks before the use of neural networks.

The SVM is a discriminative machine learning approach, as it particularly defines a hyperplane to separate data points into different classes. In other words, the hyperplane is the decision boundary that helps classify data samples. Obviously, there are many possible hyperplanes that can separate data points. The idea of SVM is to find the optimal separating hyperplane so that future data can be classified with more confidence, in other words, a hyperplane that separates data points into classes and maximizes the distance from the closest data sample to either class. As illustrated in Figure 2.3-2, the distance between the hyperplane and the closest data sample determines the *margin* of the classifier, and the data point is called a *support vector* (Boser, Guyon, and Vapnik 1992).

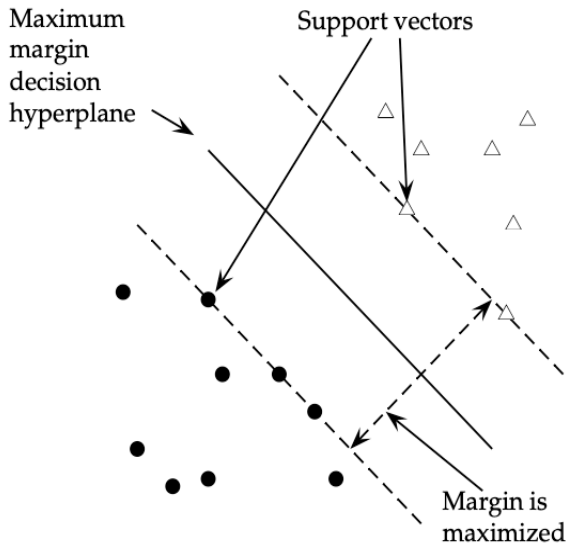

Figure 2.3-2 Optimal separating margin and support vectors (Manning, Raghavan, and Schütze 2008, 320)

In machine learning, the decision boundary is described by points such that $w^T x + b = 0$, so that it is orthogonal to the weight vector w, where x is the feature vector and b represents bias. Therefore, the *weight vector*, which points to the same direction as w, can be described as $w^{normal} = \frac{w}{||w||_2}$, where normal means the vector is normalized and only provides direction information.

In binary SVM, the threshold values are $+1$ and -1. Thus, the binary linear classifier outputs the prediction class $\hat{y}$:

$$\hat{y} = sign(w^T x + b), \quad (3)$$

where a value of $+1$ indicates one class, and -1 indicates the other.

As shown in Figure 2.3-3, if the point on the hyperplane closest to x is labelled as x' then:

$$x' = x - yr\frac{w}{||w||_2}, \quad (4)$$

which also satisfies $w^T x + b = 0$. Therefore:

$$w^T(x - yr\frac{w}{||w||_2}) + b = 0, \quad (5)$$

$$r = y\frac{w^T x + b}{||w||_2}. \quad (6)$$

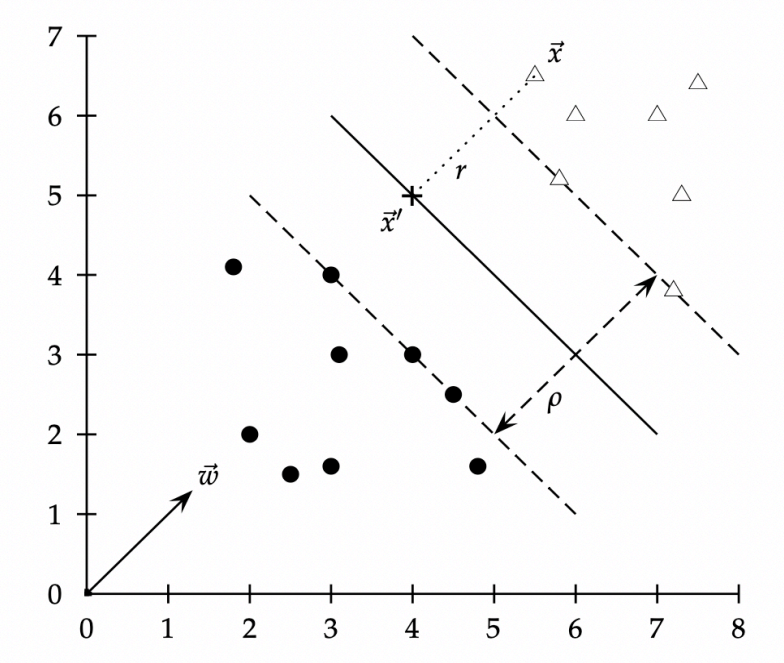

Figure 2.3-3 The maximum margin (Manning, Raghavan, and Schütze 2008, 323)

The problem of finding the maximum margin then becomes a Quadratic optimization problem, which can be solved using standard quadratic programming libraries (Boser, Guyon, and Vapnik 1992). The classifier can then be determined:

$$\hat{y} = sign(\sum_{i=1}^{n} \alpha_i y_i {x_i}^T x + b), \quad (7)$$

where α_i is the Lagrange multiplier, and $w^* = \alpha_i y_i {x_i}^T$ is the optimal weight. To further classify data into multiple classes, multi-class SVM defines the margin between the one class and the nearest other class.

However, the linear margin defined cannot separate data points when they are not linearly separable. To deal with these situations, a kernel SVM with a non-linear decision boundary is needed. This is done by finding a linear decision boundary in an expanded space. The input vector x is replaced by $\Phi(x)$, where Φ is called a feature mapping. In this way, the optimal weight w^* is mapped into an expanded feature space (i.e., a higher-dimensional space).

Then, the inner product of input vectors in the classifier can be substituted by a kernel function K: $R^{m'} \times R^{m'} \rightarrow R$, which can be written as a dot product of feature mappings:

$$K(x_1, x_2) = \Phi(x_1)^T \Phi(x_2)^T, \text{ for some } \Phi. \quad (8)$$

Proposed by Poggio and Girosi (1990), Radial Basis Function (RBF) kernel, also known as the Gaussian kernel, is a widely used kernel function for SVM (Manning, Raghavan, and Schütze 2008, 333). The RBF kernel on two samples is defined as:

$$K(x_1, x_2) = \exp(-\frac{||x_1 - x_2||^2}{2\sigma^2}), \quad (9)$$

where $||x_1 - x_2||^2$ is the squared Euclidean distance between the two feature vectors, and σ is a hyperparameter.

2.3.4.2 Feedforward Neural Network (FNN)

To finetune a pre-trained model, such as BERT and XLNet, on specific downstream tasks, a Feedforward Neural Network (FNN) is usually concatenated to the pre-trained model to generate corresponding outputs (Devlin et al. 2019).

First appeared in the 1950s, FNN is also known as Multi-Layer Perceptrons (MLPs) (Rosenblatt 1957). An FNN classifier aims to approximate some function f that maps an input x into a category y. FNN learns the value of parameters θ in the mapping $y = f(x; \theta)$ that obtain the best approximation. An FNN is composed of an input layer, one or more hidden layers, and an output layer (Figure 2.3-4). A fully-connected layer means each neuron in this layer is connected to every neuron in the previous layer. As shown in Figure 2.3-4, the hidden layer is a fully-connected layer.

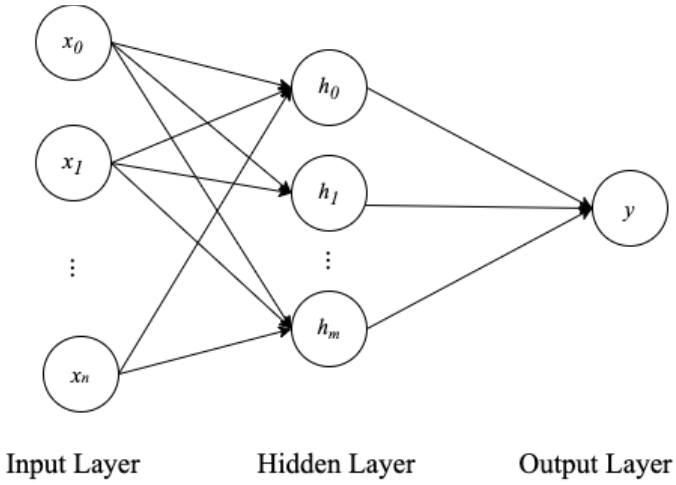

Figure 2.3-4 FNN structure

While training, each neuron receives input from the previous layer, where each input carries a certain weight. Then, each neuron takes the weighted sum of its received input and calculates its output using an activation function. The activation function maps the resulting sum of each neuron to different categories. In neural networks, non-linear activation functions are the most used as they help the model to adapt to a variety of data.

To evaluate the performance of the model, the loss of each iteration will be calculated using a loss function. The most popular cost function for classification problems is the Cross-Entropy loss function (Good 1952). Cross-Entropy loss measures the difference between two distributions. For classification problems, Cross-Entropy loss is calculated by:

$$CE = -\sum_{c=1}^{M} y_c \log (P_c), \quad (10)$$

where M is the number of classes, y_c represents if class c is the correct label, P_c is the predicted probability of class c. FNN will then update the weight on each neuron so that it can minimize the loss using Back Propagation (Rumelhart, Hintont, and Williams 1986).

The finetuning process first flattens the pre-trained embeddings as they are usually of more than one dimension. Then, a fully-connected layer will be added as an output layer to yield the prediction class. The finetuning usually freezes some layers in the pre-trained model and only trains the final layers to prevent overfitting (Kovaleva et al. 2019).

2.3.4.3 Summary

This section presented two common machine learning classifiers in NLP. Section 2.3.4.1 introduces SVM, a discriminative machine learning approach. It finds the decision boundaries that can separate data points into different classes. Section 2.3.4.2 describes FNN, a neural network algorithm, in which the data flows only in one direction.

Chapter 3 Methodology

This chapter presents the details about the dataset, the evaluation metric, and the machine learning methods used for experiments in this thesis. To be specific, Section 3.1 provides the information of the MoodyLyrics4Q Dataset and the AllMusic Dataset, and describes the procedure to collect from the Internet, the words of the lyrics, which were annotated in the datasets but not contained within them. Section 3.2 explains the evaluation metric used in the experiments. Section 3.3 presents the setting of Experiment 1, which implements the SVM method to classify music emotions. The structure of the BERT method and the setting of Experiment 2 are explained in Section 3.4. Section 3.5 describes the details of the XLNet method and Experiment 3.

3.1 Dataset

This section provides detailed information about the dataset used in this thesis. Section 3.1.1 describes the MoodyLyrics4Q Dataset, its statistics, and the mood annotation process used to create the dataset. Section 3.1.2 covers the details of the AllMusic Dataset. However, neither dataset provides lyrics due to copyright concerns. Instead, only titles, artists, and mood categories are provided. Therefore, Section 3.1.3 describes the method and results of collecting lyrics annotated in the MoodyLyrics4Q dataset and the AllMusic Dataset.

3.1.1 MoodyLyrics4Q Dataset

In this thesis, the MoodyLyrics4Q Dataset[14] proposed by Çano and Morisio (2017b) will be used for training and validation. The song sources of this dataset include songs from the Million

[14] http://softeng.polito.it/erion/MoodyLyrics4Q.zip

Song Dataset (Bertin-Mahieux et al. 2011) and the Playlist dataset.[15] The dataset contains 2000 songs annotated using last.fm tags.

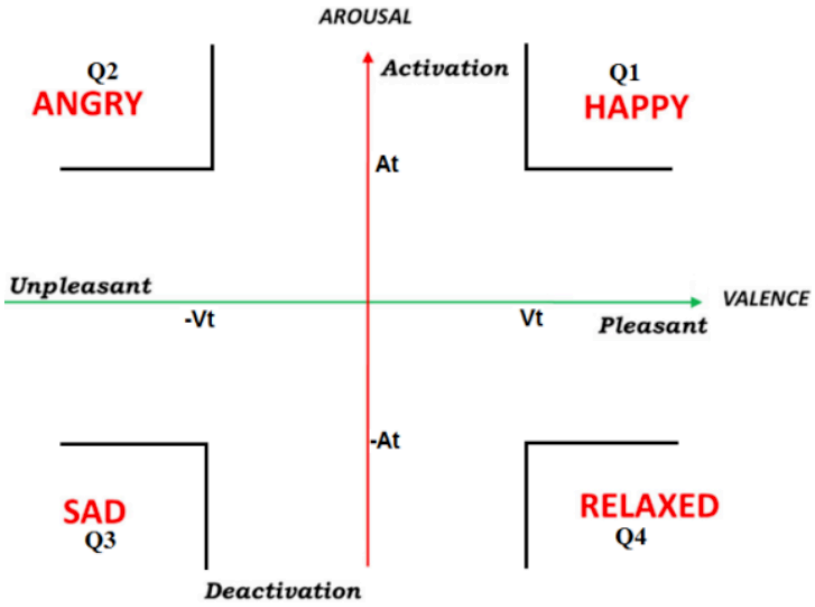

Figure 3.1-1 Dataset annotation (Çano and Morisio 2017b)

The dataset is annotated using Russell's model (Russell 1980). *Happy*, *angry*, *sad*, and *relaxed* are used to represent each cluster corresponding to each quadrant (Figure 3.1-1). Next, the annotation process started with a map between users' tags and Russell's model. The authors manually selected tags provided by AllMusic that clearly fall into one of the four quadrants of Russell's model, using Affective Norms for English Words (ANEW) (Bradley and Lang 1999).

Table 3.1-1 shows the resulting map, where each cluster includes the ten most relevant emotion terms. Each cluster was then extended with related forms of the basic ten terms using lemmatization since they express the same opinion with different forms.

[15] http://www.cs.cornell.edu/~shuochen/lme/data_page.htm

Each song was scored using the resulting clusters with four counters using users' tags provided in last.fm, corresponding to the number of tags in each cluster. To keep high polarity, songs with three or fewer tags were removed. The authors annotated a song to a cluster only when one of the following four rules is met. First, the song has four or more tags of the quadrant and no tags of the others. Second, when the song has tags of more than one quadrant, it has six to eight tags of the selected quadrant and at most one tag of the others. Third, when the song has two tags of the others, it must have nine to 13 tags of the selected quadrant. Finally, the song must have 14 or more tags of one quadrant, when it has at most three tags belonging to the others.

Table 3.1-1 Cluster of terms (Çano and Morisio 2017b)

Q1-Happy	**Q2-Angry**	**Q3-Sad**	**Q4-Relaxed**
happy	angry	sad	relaxed
happiness	aggressive	bittersweet	tender
joyous	outrageous	bitter	soothing
bright	fierce	tragic	peaceful
cheerful	anxious	depressing	gentle
humorous	rebellious	sadness	soft
fun	tense	gloomy	quiet
merry	fiery	miserable	calm
exciting	hostile	funeral	mellow
silly	anger	sorrow	delicate

3.1.2 AllMusic Dataset

The test dataset used in this thesis is the AllMusic Dataset[16] proposed by Malheiro et al. (2018). Similar to Çano and Morisio (2017b), the authors started with a map of AllMusic mood tags to four quadrants in Russell's model using ANEW. Then, the authors extracted 400 songs with more emotion tags for each quadrant. A song was annotated to a selected quadrant only if all the mood tag terms belonged to the corresponding cluster. Finally, the annotated initial dataset was validated by three people. Only the songs with at least one agreement between AllMusic's annotation and human annotators were validated. The resulted dataset contains 771 songs.

3.1.3 Lyric Collection

As aforementioned, the dataset only provides titles, artists, and mood categories in an Excel table, and does not contain lyrics due to copyright issues. In this thesis, the lyrics were extracted from the Genius website, a community that provides music metadata including song titles, artists, lyrics, etc.

A crawler in Python was first written to scrape lyrics according to the song title and artist name on the Genius website for each Excel file using the Genius API. Then, songs with non-English lyrics and acoustic songs without lyrics were discarded. Finally, the common pattern tags, such as [Verse 1], in the lyrics were removed.

For the MoodyLyrics4Q Dataset, lyrics of 1,900 songs from 1,227 artists were extracted. As shown in Figure 3.1-2 (a), the resulting dataset is well-balanced. Table 3.1-2 lists the detailed numbers of lyrics in each quadrant. The dataset contains 524,397 words in total, and 285,764 words

[16] http://mir.dei.uc.pt/resources/Dataset-Allmusic-771Lyrics.zip

without stop-words. The vocabulary size without stop-words is 25,263. Figure 3.1-2(b) shows the distribution of word count per song. It can be observed that most lyrics contain less than 500 words.

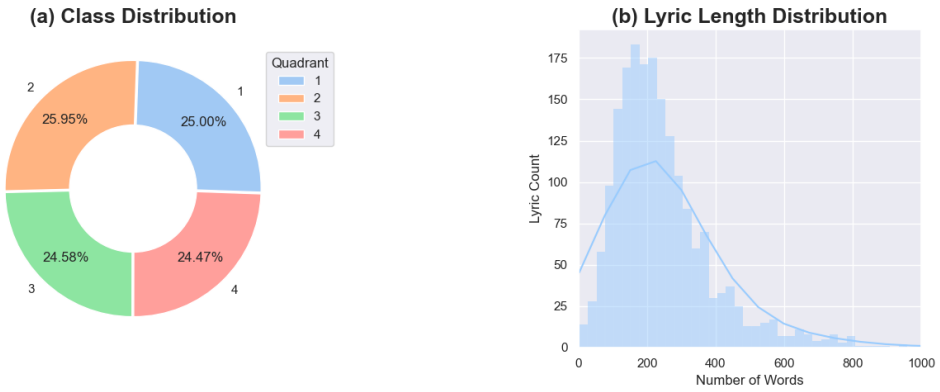

Figure 3.1-2 MoodyLyrics4Q Dataset Distributions

757 lyrics by 321 artists were extracted from the AllMusic Dataset. Figure 3.1-3 (a) shows the distribution over four classes. The specific numbers of lyrics from each quadrant are listed in Table 3.1-2. The dataset consists of 205,146 words, and 111,430 words without stop-words with a vocabulary size of 12,773. Figure 3.1-3 (b) shows the lyric length distribution. Similar to the MoodyLyrics4Q Dataset, the majority of lyrics have less than 500 words.

Table 3.1-2 Dataset Distribution

Quadrant	**MoodyLyrics4Q**	**AllMusic**
1	475	206
2	493	202
3	467	203
4	465	146
Total	1900	757

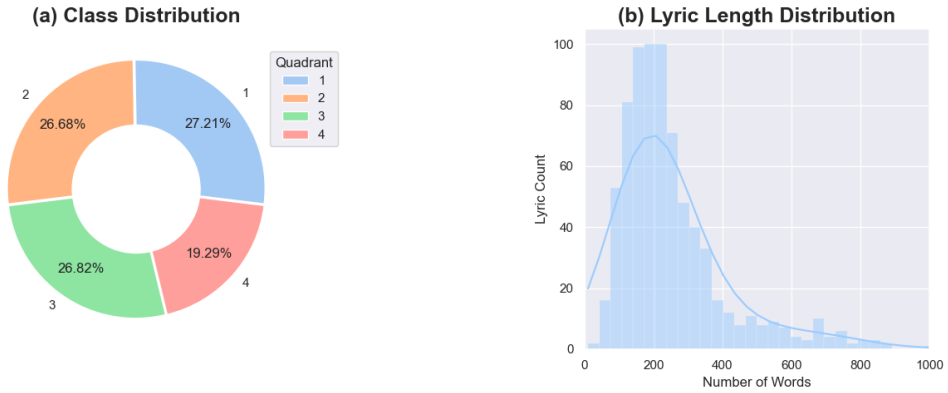

Figure 3.1-3 AllMusic Dataset Distributions

3.2 Evaluation Method

For each classifier and preprocessing setting, the model will be trained and validated on half of the MoodyLyrics4Q Dataset, and the other half will be used for testing.

To avoid overfitting and to better compare different models, a 5x2 repeated cross-validation (Bouckaert and Frank 2004) will be implemented. This step will randomly divide the MoodyLyrics4Q Dataset into two parts. For each fold, the model will be trained on one of them and validated using the rest part. This process will repeat five times.

The average F1-score of each fold in each repeat will serve as the performance metric for the model. The F1-score is calculated using $F1 = 2 \times \frac{(precision \times recall)}{(precision + recall)}$, where $precision = \frac{tp}{tp+fp}$ and $recall = \frac{tp}{tp+fn}$, tp represents the true positives, fp is the false positives, and fn means the false negatives.

Text preprocessing methods will be compared and evaluated using the validation results. To be specific, the models without any text preprocessing will be compared with the models with

all combinations of the text preprocessing techniques. The text preprocessing combinations that significantly improved the system performance will be discussed.

In addition, text preprocessing with similar functions will also be compared and evaluated, both alone and when combined with the same set of other methods. Specifically, noise removal and stop-words removal both remove noisy tokens in the lyrics. However, noise removal eliminates punctuations and numbers, whereas stop-words removal removes words that do not have specific meanings. Moreover, both stemming and lemmatization convert the token into its basic form with different methods. Stemming simply slices the word from prefix or suffix. However, lemmatization takes morphology into account and transforms the word into its canonical form. Although the two methods in these two sets have similar functions in processing text, the differences in their methodology may lead them to have completely different effects on MER.

In the end, the model that achieves the highest average F1-score will be trained on 60% of the MoodyLyrics4Q Dataset, validated using 40%, and tested on the AllMusic Dataset to yield the final performance for each classifier.

3.3 Experiment 1: SVM

Experiment 1 implements RBF SVM with different text preprocessing techniques on MER. This experiment will be conducted on a MacBook Air (M1, 2020) with 16 GB of memory.

The lyrics will first be preprocessed and converted into vectors using the TF-IDF score for the classifier to process. Then, all possible permutations of no preprocess, lowercase conversion, noise removal, stop-words removal, and stemming or lemmatization will be explored.

As mentioned in Section 3.2.3, most lyrics contain less than 500 words. Therefore, the vector size starts from 512. In addition, preliminary experiments have narrowed the vector size to 512, 1024, and 4096 since a larger size resulted in a significant drop.

Then, the Scikit-learn Python Library[17] will be used to implement the RBF SVM algorithm. Scikit-learn is an open-source library that provides tools for machine learning.

3.4 Experiment 2: BERT

Experiment 2 uses BERT to perform MER and compares different text preprocessing settings. The lyrics will first be tokenized using the pre-trained model 'bert-base-cased' provided in the library Transformers[18] built by Hugging Face,[19] an Artificial Intelligence community. The 'bert-base-cased' pre-trained model contains 12 layers, 768 hidden nodes, 12 attention heads, and 110 million parameters in total.

The pre-trained model was trained using BookCorpus,[20] a dataset that contains 11,038 books, and English Wikipedia,[21] an online encyclopedia. The model was trained on four cloud TPUs for one million steps with a batch size of 256. For 90% of the steps, the sequence length was set to 128 tokens, and 512 tokens for the rest. The optimizer used is Adam with a fixed weight decay of 0.01, as proposed in Loshchilov and Hutter (2019).

After tokenization, the lyrics will be padded or cut down to 512 words since most lyrics have less than 500 words. The resulting vector will be fed into the pre-trained BERT to finetune

[17] https://scikit-learn.org/stable/
[18] https://huggingface.co/transformers/v2.11.0/index.html
[19] https://huggingface.co/
[20] https://yknzhu.wixsite.com/mbweb
[21] https://en.wikipedia.org/wiki/English_Wikipedia

the parameters. The model is the 'bert-base-cased' pre-trained model with a linear layer of four nodes on top of its output. The optimizer used is the same as the pre-trained model.

Experiment 2 will be conducted on the Cedar[22] cluster provided by Compute Canada.[23] Due to the memory limitation of the hardware (CUDA[24]) available at Cedar, the batch size will be set to 8.

In addition, early stopping will be implemented to avoid overfitting. Early stopping is a regulation method that terminates the training process when a monitored metric stops improving. In this experiment, the early stopping method observes the validation loss, which assesses the error on the validation set. Specifically, if the validation loss does not drop for a certain number of epochs, the training process will be forced to stop. The number of epochs is called patience in early stopping. If the training process is not stopped, the model will be trained for 500 epochs at most.

As aforementioned, the pre-trained BERT model contains a large number of parameters, thus needing a long time to run. Therefore, several preliminary experiments will be carried out first to narrow down the possible hyperparameter settings. The details are covered in Chapter 4.

3.5 Experiment 3: XLNet

For the XLNet method, the lyrics will first be tokenized using the pre-trained model 'xlnet-base-cased' provided in the Transformers[18] Python library. The 'xlnet-base-cased' model consists of 12 layers, 768 hidden nodes, and 12 attention heads.

[22] https://docs.computecanada.ca/wiki/Cedar
[23] https://www.computecanada.ca/
[24] https://developer.nvidia.com/cuda-zone

The training dataset for their model included BookCorpus,[20] English Wikipedia,[21] Giga5,[25] ClueWeb 2012-B,[26] and Common Crawl.[27] The model was trained for 500,000 steps with an Adam weight decay optimizer and a batch size of 8,192 (Yang et al. 2019).

To finetune the model to fit the MER problem, a linear layer of four nodes will be added on top of the pre-trained model, which matches the four target classes. Experiment 3 will use the same optimizer as the pre-trained model. The batch size will be set to 8 because of the hardware (CUDA) memory limitation. Early-stopping will also be implemented using the same criteria for the BERT model.

As in Experiment 2, preliminary experiments will first be conducted to find the reasonable range of hyperparameters (i.e., learning rate and the early-stopping patience). The details of preliminary experiments are included in Chapter 4.

[25] https://catalog.ldc.upenn.edu/LDC2011T07
[26] https://lemurproject.org/clueweb12.php/
[27] https://commoncrawl.org/

Chapter 4 Experiments

This chapter presents the detailed settings, results, and discussions of each experiment described in Chapter 3. Section 4.1 includes the details of the experiments conducted on the MoodyLyrics4Q Dataset. Section 4.2 presents the experiments and results on the AllMusic Dataset. Finally, discussions of both experiments' results are covered in Section 4.3.

4.1 Experiments and Results on MoodyLyrics4Q Dataset

This section provides the experiments and results on the MoodyLyrics4Q Dataset. Section 4.1.1 presents Experiment 1, which uses a Support Vector Machine (SVM) classifier with different maximum token sizes and text preprocessing techniques. Section 4.1.2 covers Experiment 2, which implements BERT (Section 2.3.3.2) with different hyperparameter settings. Experiment 3 is described in Section 4.1.3, which uses XLNet (Section 2.3.3.3) with different hyperparameter settings. Finally, Section 4.1.4 briefly summarizes the main findings.

4.1.1 Support Vector Machine (SVM)

The hyperparameters of the SVM experiments include the maximum length of the input tokens and the text preprocessing steps. As mentioned in Section 3.1.3, most lyrics have fewer than 500 words. Therefore, the lower limit of the maximum length was set to 512. The upper limit of 4096 was chosen because preliminary experiments have shown that there was a dramatic drop in the F1-score beyond 4096. Thus, the possible values for the maximum length were: 512, 1024, 2048, and 4096. Lyrics that have less words than the maximum length would be zero-padded. For lyrics with more words, the exceeding words would be cut out.

Then, the SVM classifiers with combinations of the four different maximum lengths and all possible permutations of text preprocessing techniques were trained and validated using the

training subset of the MoodyLyrics4Q Dataset. As mentioned in Section 3.2, text preprocessing methods with similar functions were compared and evaluated. Group 1 compares stemming and lemmatization as they both convert the token into its basic. Group 2 compares noise removal and stop-words removal because they both remove noisy tokens.

Table 4.1-1 and Table 4.1-2 present all the pairs, where LC means lowercase conversion, NR represents noise removal, SR means stop-words removal, Stem denotes stemming, and Lemma represents lemmatization. The pairs on the same row will be compared to control variables.

The repeated 5x2 cross-validation results are shown in Figure 4.1-1. In this figure, the corresponding p values of the paired t-test between the pairs in Group 1 and Group 2 less than 0.05, which indicates a significant difference, are labelled. Also labelled are the pairs that show significant differences between with and without text preprocessing.

Table 4.1-1 Text preprocessing method Group 1

Stem	Lemma
Stem + LC	Lemma + LC
Stem + NR	Lemma + NR
Stem + SR	Lemma + SR
Stem + LC + NR	Lemma + LC + NR
Stem + LC + SR	Lemma + LC + SR
Stem + NR + SR	Lemma + NR + SR
Stem + LC + NR + SR	Lemma + LC + NR + SR

Table 4.1-2 Text preprocessing method Group 2

NR	SR
NR + LC	SR + LC
NR + Stem	SR + Stem
NR + Lemma	SR + Lemma
NR + LC+ Stem	SR + LC + Stem
NR + LC + Lemma	SR + LC + Lemma

As indicated in Figure 4.1-1, the SVM classifier, among the models with text preprocessing, only stemming (Stem), noise removal and stemming (NR + Stem), and noise removal and lemmatization (NR + Lemma) performed significantly better than the model without any text preprocessing. In addition, the model with stemming (Stem) achieved a better F1 score than the model with lemmatization (Lemma). This is the case even when they are combined with lowercase conversion, noise removal, stop-words removal (LC + NR + SR + Stem; Lemma), although the difference is insignificant when they are combined with other text preprocessing methods. In addition, no significant difference shows in Group 2.

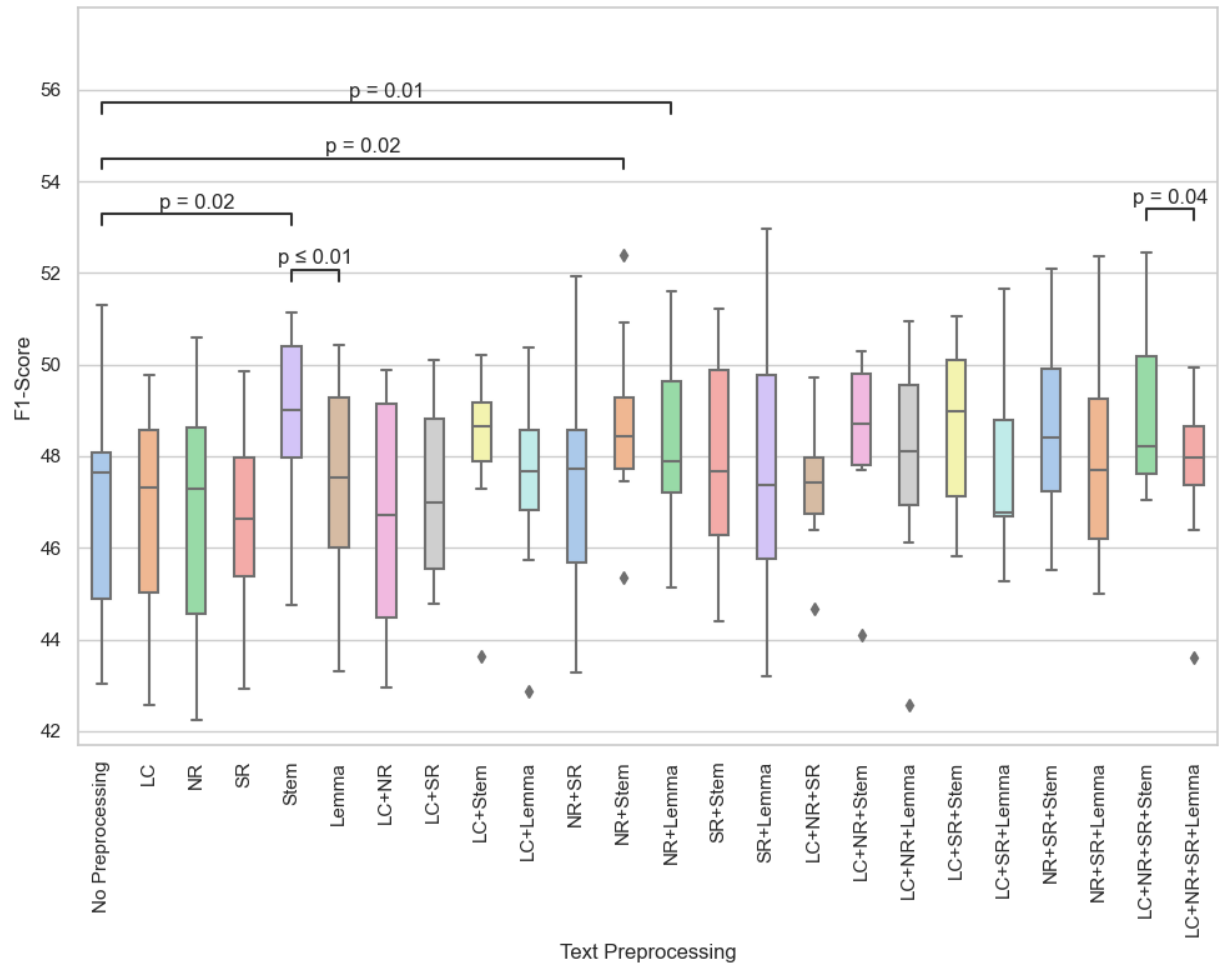

Figure 4.1-1 F1-Scores for SVM validation results of different text preprocessing configurations. The significant differences are shown with brackets with the corresponding p-values.

For each text preprocessing setting, the maximum length with the highest F1-score was tested on the MoodyLyrics4Q test subset. The results and the corresponding maximum token lengths are shown in Table 4.1-3. For most text preprocessing settings, the SVM model worked the best with a maximum token length of 2048. The performance of the SVM models ranges from 46.44% to 50.82%. The SVM model without any text preprocessing achieved an F1-score of 47.09%.

Table 4.1-3 SVM test results on MoodyLyrics4Q

Preprocessing	**Maximum Token Length**	**F1-Score**
None	2048	47.09%
LC	2048	46.87%
NR	2048	47.58%
SR	2048	47.78%
Stem	2048	47.54%
Lemma	2048	48.06%
LC + NR	2048	47.54%
LC + SR	2048	49.04%
LC + Stem	2048	49.99%
LC + Lemma	2048	48.46%
NR + SR	2048	46.81%
NR + Stem	2048	48.25%
NR + Lemma	2048	47.60%
SR + Stem	2048	47.84%
SR + Lemma	4096	47.68%
LC + NR + SR	2048	47.48%
LC + NR + Stem	2048	48.85%
LC + NR + Lemma	2048	47.53%
LC + SR + Stem	1024	49.02%
LC + SR + Lemma	4096	48.33%
NR + SR + Stem	4096	49.28%
NR + SR + Lemma	2048	46.44%
LC + NR + SR + Stem	4096	**50.82%**
LC + NR + SR + Lemma	2048	49.35%

4.1.2 BERT

As mentioned in Section 3.4, preliminary experiments were carried out first to determine the number of frozen layers of BERT in the finetuning process. The 'bert-base-cased' model contains 11 BERT layers in total. The preliminary experiments started with freezing all but the last BERT layer and stopped when the model started to overfit the training set. Figure 4.1-2 shows the validation loss change of the BERT model with different numbers of unfrozen layers.

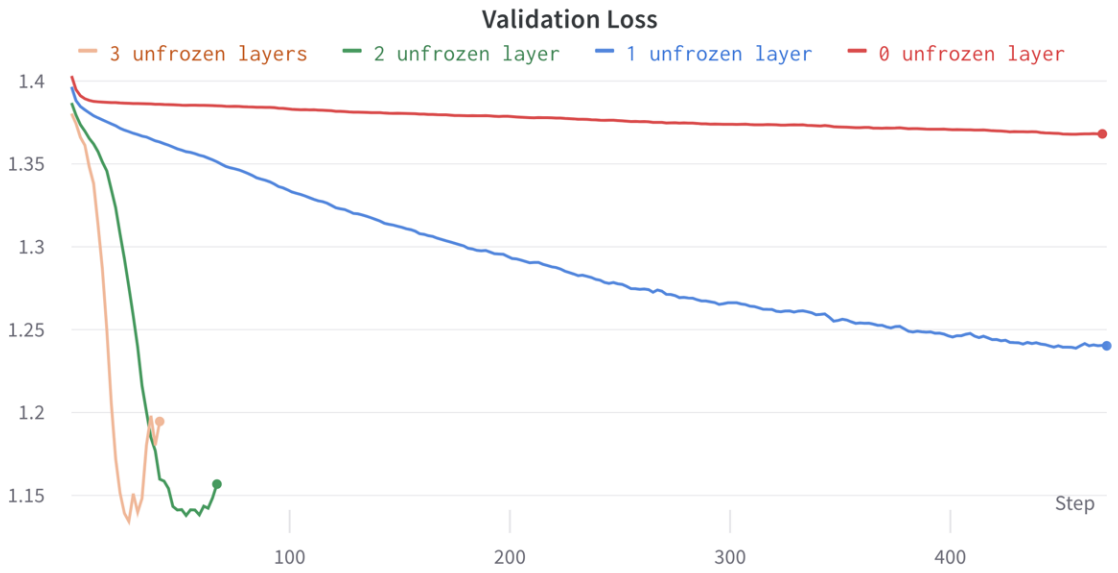

Figure 4.1-2 Preliminary experiment results with different numbers of unfrozen layers of BERT

When all the layers of the pre-trained model were frozen, the model learned slowly because only the fully-connected layer was updated during the training process. Unfreezing more layers reduces the number of steps it takes to converge to a local minima. When more than one layer in the pre-trained model were unfrozen, the loss dropped faster, but it started to overfit the training dataset after a few epochs. This overfitting can be compensated by early stopping with an appropriate patience value.

However, when three layers were unfrozen, the model learned too fast and overfit the training data too fast, making it impossible to early stop exactly at the lowest point. Furthermore, adding more unfrozen layers results in more parameters to be trained and more processing time needed for each step (more than 40 hours needed for an entire 5x2 cross-validation). Therefore, only two unfrozen layers were used in the pre-trained model. This configuration provided a reasonable learning speed and training time while reducing overfitting.

Then, the preliminary experiments also explored the reasonable learning rate for finetuning. The results show that when the learning rate was set to $1e^{-5}$ or $1e^{-7}$, the validation F1-score of the model stopped increasing much earlier than that of the model with a learning rate of $1e^{-6}$. It is possible that the step that each batch takes in gradient descent was too large when the learning rate was $1e^{-5}$, so that the model bounced back and forth between the convex function of gradient descent. By contrast, when the learning rate was $1e^{-7}$, the steps were so small that the model barely learned. Therefore, the learning rate was set to $1e^{-6}$.

Next, preliminary experiments tested different early-stopping patience values. The experiment started with 5 and kept increasing with a step of 5. The results showed that when the early-stopping patience value was set to 15, the difference between the training and validation F1-score was greater than 15%, which indicates the model overfits the training set. Therefore, the early-stopping patience values were limited to 5 and 10.

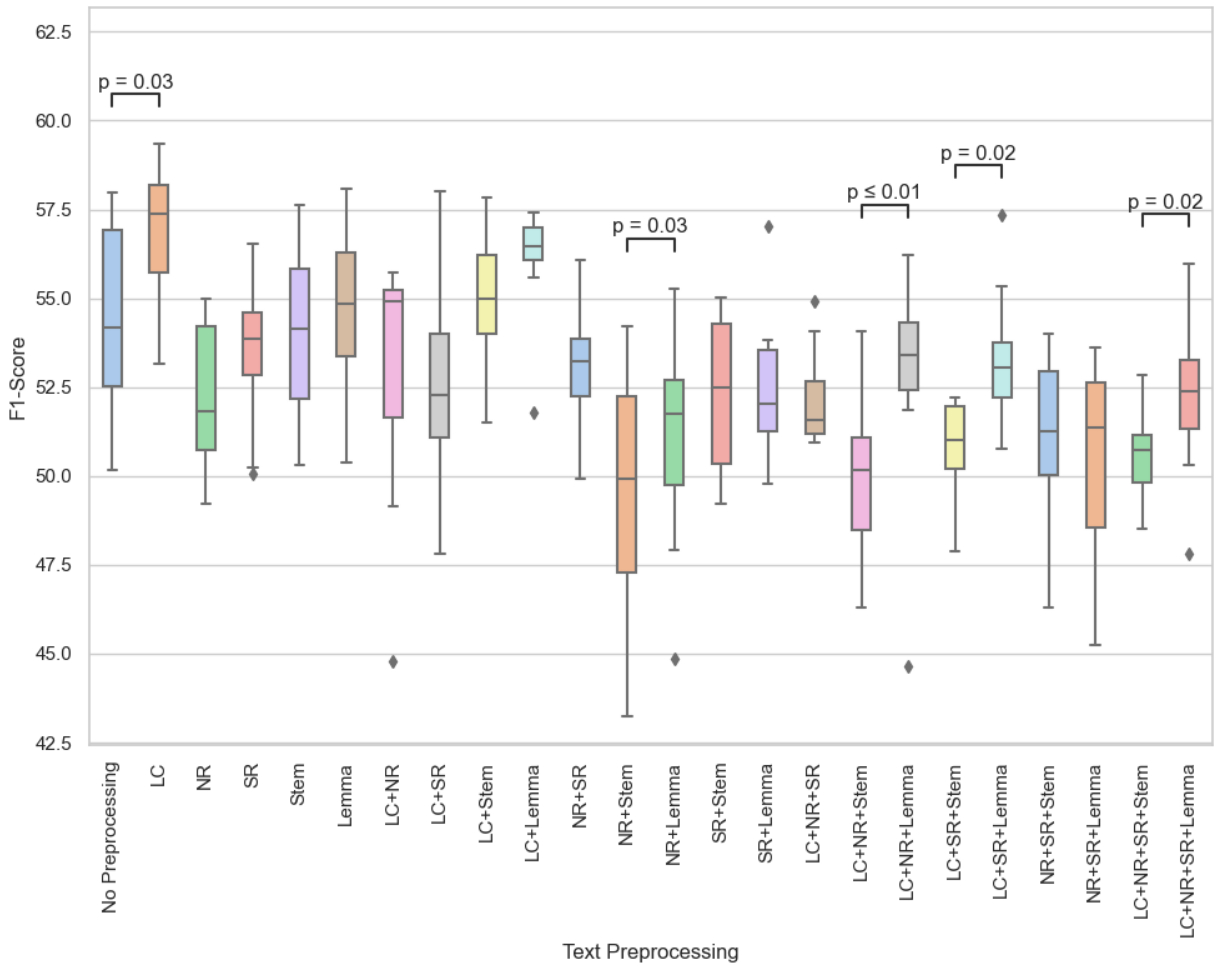

Figure 4.1-3 F1-Scores for BERT validation results of different text preprocessing. The brackets show the significant differences with the corresponding p-values.

Next, the BERT model with combinations of the two different early-stopping patience values (5 or 10) and all possible permutations of text preprocessing techniques were trained and validated using the training subset of the MoodyLyrics4Q Dataset.

Figure 4.1-3 shows the results of the model with the highest F1-score in each text preprocessing setting. It is observed that only the model with lowercase conversion (LC) performed significantly better than the model without any text preprocessing techniques. In addition, although stemming and lemmatization do not show a large difference when applied alone,

lemmatization had significantly better results than stemming when combined with noise removal (NR + Stem; Lemma), lowercase conversion (LC + NR + Stem; Lemma), stop-words removal (LC + SR + Stem; Lemma), and their combinations (LC + NR + SR + Stem; Lemma).

The early-stopping patience value for each text preprocessing setting was thus selected based on the cross-validation result. The selected models were then tested on the MoodyLyrics4Q test subset. The results are shown in Table 4.1-4. The performance of the BERT models ranges from 47.48% to 53.31%. The BERT model without any text preprocessing obtained an F1-score of 51.13%.

4.1.3XLNet

Similar to BERT, several preliminary experiments were conducted for XLNet to find the optimal number of unfrozen layers of the pre-trained model and the early-stopping patience values during finetuning. To determine the number of unfrozen layers, the preliminary experiments started by freezing all the layers in the pre-trained model and kept increasing the number of unfrozen layers until the model started to overfit the training set.

As shown in Figure 4.1-4, adding unfrozen layers increased the learning speed. When two layers were unfrozen, the model learned faster, but the validation loss started to increase after a few epochs, indicating the model overfits the training set. However, the validation loss of the model with three unfrozen layers directly jumped back when reaching the bottom and started to increase rapidly. In this case, it is much harder to compensate for the overfitting than in the model with only two unfrozen layers. Hence, only two layers out of 11 XLNet layers in the pre-trained model were unfrozen.

Table 4.1-4 BERT test results on MoodyLyrics4Q

Preprocessing	**Early-stopping Patience**	**F1-Score**
None	10	51.13%
LC	10	52.50%
NR	10	49.24%
SR	10	52.46%
Stem	10	51.72%
Lemma	10	**53.31%**
LC + NR	10	51.49%
LC + SR	10	50.92%
LC + Stem	10	49.38%
LC + Lemma	10	52.34%
NR + SR	10	52.10%
NR + Stem	10	47.87%
NR + Lemma	10	48.13%
SR + Stem	10	51.95%
SR + Lemma	10	52.39%
LC + NR + SR	10	50.15%
LC + NR + Stem	10	48.19%
LC + NR + Lemma	10	52.20%
LC + SR + Stem	10	50.34%
LC + SR + Lemma	10	52.96%
NR + SR + Stem	10	49.49%
NR + SR + Lemma	10	52.02%
LC + NR + SR + Stem	10	47.48%
LC + NR + SR + Lemma	10	49.43%

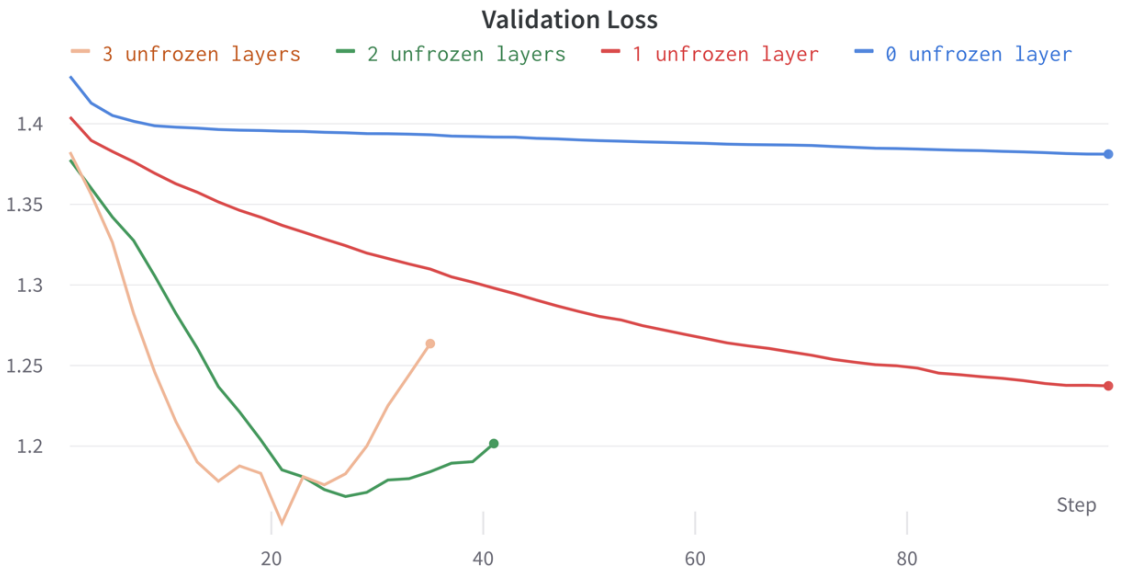

Figure 4.1-4 Preliminary Experiment Results of XLNet

Then, the preliminary experiments investigated the proper learning rate and early-stopping patience values for the XLNet model. The results showed that the XLNet model behaved similarly to the BERT model with the same learning rate and the same early-stopping patience values (See Section 4.1.2). Therefore, the learning rate was set to $1e^{-6}$, and the early-stopping patience values were limited to 5 and 10.

Next, the XLNet model with combinations of the two different early-stopping patience values (5 or 10) and all possible permutations of text preprocessing settings were finetuned and validated using the training subset of the MoodyLyrics4Q Dataset.

The results of the 5x2 cross-validation are shown in Figure 4.1-5. The results indicate that the application of stemming (Stem) significantly decreased the performance of the model compared to the model with lemmatization (Lemma). Even when combined with lowercase conversion, noise removal, or stop-words removal, the model that implemented lemmatization (LC

+ Lemma; NR + Lemma; SR + Lemma) obtained better performance than the model with stemming (LC + Stem; NR + Stem; SR + Stem), respectively.

In addition, the application of noise removal (NR; LC + NR; NR + Lemma) decreased the performance significantly compared to the models with stop-words removal (SR; LC + SR; SR + Lemma).

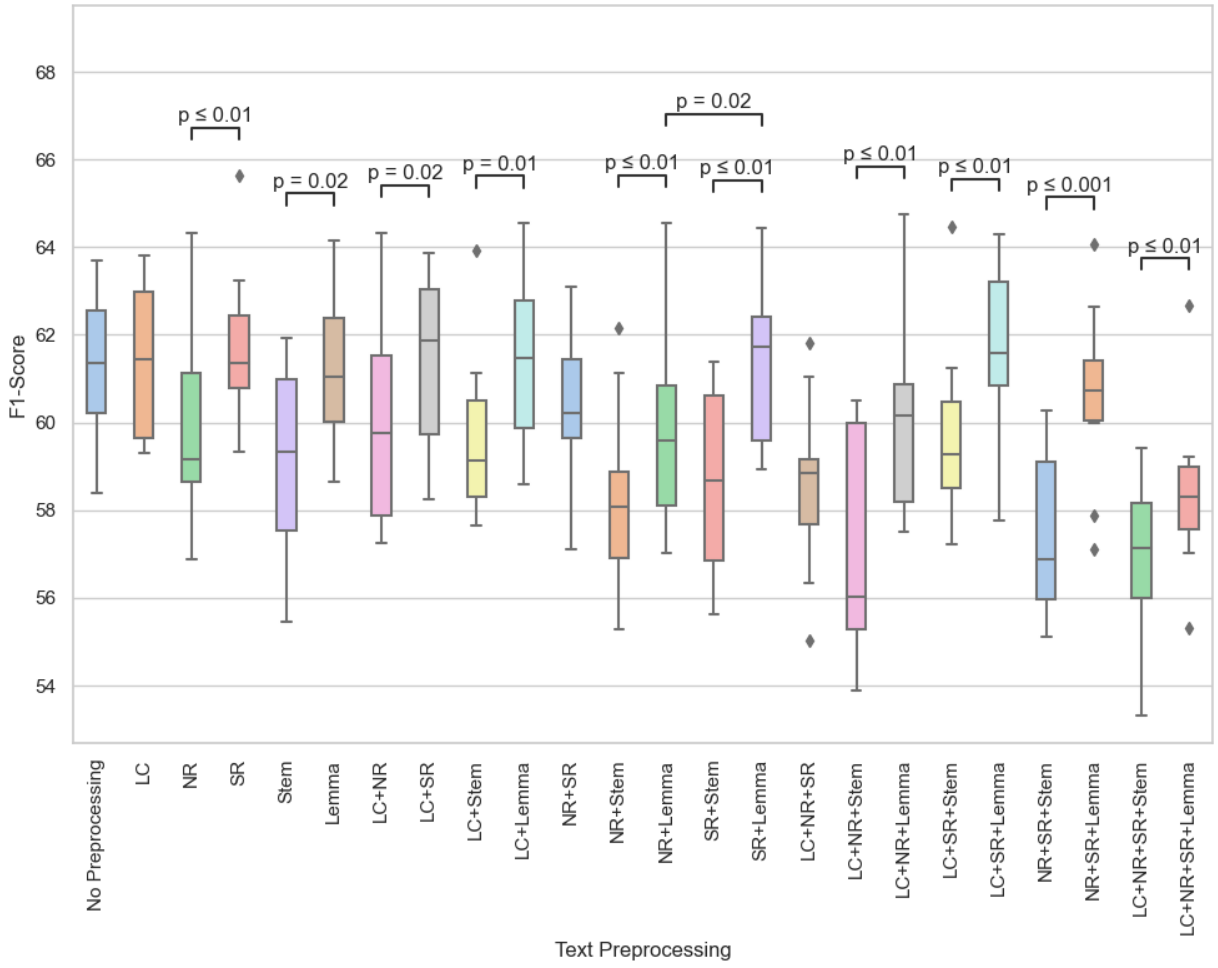

Figure 4.1-5 F1-Scores for XLNet validation results of different text preprocessing settings. The significant differences are shown with brackets with the corresponding p-values.

Using the validation result, the proper early-stopping patience value for each text preprocessing setting was determined. The selected models were then tested on the MoodyLyrics4Q test subset. The results are shown in Table 4.1-5. The performance of the XLNet

models ranges from 54.27% to 59.08%. The XLNet model without any text preprocessing achieved 58.98% in F1-score. Only the XLNet model with lemmatization outperformed the XLNet without any text preprocessing with an F1-score of 59.08%.

Table 4.1-5 XLNet Test Result on MoodyLyrics4Q

Preprocessing	**Early-stopping Patience**	**F1-Score**
None	10	58.98%
LC	10	56.20%
NR	10	57.32%
SR	10	57.17%
Stem	10	55.31%
Lemma	10	**59.08%**
LC + NR	10	56.48%
LC + SR	10	56.01%
LC + Stem	10	56.23%
LC + Lemma	10	56.17%
NR + SR	10	56.40%
NR + Stem	10	56.49%
NR + Lemma	10	57.21%
SR + Stem	10	56.64%
SR + Lemma	10	57.50%
LC + NR + SR	10	57.12%
LC + NR + Stem	10	55.44%
LC + NR + Lemma	10	56.45%
LC + SR + Stem	10	54.27%
LC + SR + Lemma	10	55.47%
NR + SR + Stem	10	57.09%
NR + SR + Lemma	10	56.84%
LC + NR + SR + Stem	10	55.85%
LC + NR + SR + Lemma	10	56.19%

4.1.4 Summary of Cross-Validation Results

The cross-validation results show that text preprocessing methods do not always improve MER performance. When using SVM, only stemming (Stem), noise removal and stemming (NR + Stem), and noise removal and lemmatization (NR + lemma) significantly performed better than SVM without any text preprocessing (Figure 4.1-1). In addition, only the application of lowercase conversion (LC) increased the performance of the BERT model (Figure 4.1-3). However, no text preprocessing method significantly improved the MER performance when using XLNet (Figure 4.1-5). Yet, in general, XLNet (53.32%–65.64%) outperformed both SVM (42.25%–52.97%) and BERT (43.25%–59.34%), regardless of text preprocessing methods used (Figure 4.1-1, Figure 4.1-3, and Figure 4.1-5).

Furthermore, the effects of text preprocessing methods are different for different algorithms. Namely, in Group 1, the SVM model with stemming (Stem: 48.68%; LC + NR + SR + Stem: 49.01%) performed significantly better than SVM with lemmatization (Lemma: 47.47%; LC + NR + SR + Lemma: 47.73%) (Figure 4.1-1), whereas the BERT model (Figure 4.1-3) and the XLNet model (Figure 4.1-5) observed the opposite results. The results of Group 2 showed that XLNet with stop-words removal (SR: 61.68%; LC+SR: 61.47%; SR + Lemma: 61.29%) performed better than XLNet with noise removal (NR: 59.79%; LC+NR: 59.91%; NR + Lemma: 59.92%), yet such difference does not appear for SVM (Figure 4.1-1) and BERT (Figure 4.1-3).

4.2 Experiments and Results on AllMusic Dataset

The best hyperparameter settings and text preprocessing techniques for SVM, BERT, and XLNet were selected using the results of the experiments on the MoodyLyrics4Q Dataset. To further compare the three methods, the best models were also tested using the AllMusic Dataset.

To make the validation set and the test set similar in size to prevent overfitting, the models were trained on 60% of the MoodyLyrics4Q Dataset, validated using 40% of the MoodyLyrics4Q Dataset, and tested on the entire AllMusic Dataset. The results are shown in Table 4.2-1. SVM worked best when using all types of text preprocessing methods, with an F1-score of 66.87% whereas both BERT and XLNet achieved the best performance with only lemmatization.

Table 4.2-1 Test Result on AllMusic Dataset

Method	**Preprocessing**	**F1-Score**
SVM	LC + NR + SR + Stem	66.87%
BERT	Lemma	63.59%
XLNet	Lemma	**70.09%**

4.3 Discussion

This section analyses the validation and test results and presents possible explanations. First, Section 4.3.1 discusses the validation results on the MoodyLyrics4Q dataset. Then, Section 4.3.2 analyses the test result differences between the MoodyLyrics4Q dataset and the AllMusic dataset.

4.3.1 Discussion on MoodyLyrics4Q Validation Results

As concluded in Section 4.1.4, the validation results show that for different algorithms, the impacts of text preprocessing methods are different. This is probably because different algorithms deal with text information differently. First, in Group 1, the SVM model with stemming outperformed the SVM model with lemmatization significantly, whereas the BERT and XLNet models had the opposite outcomes. The different effects of stemming and lemmatization on SVM may be caused by the fact that the process of stemming cuts the word, whereas lemmatization

transforms the word into its basic form. Therefore, stemming does not remove all the conjugation information embedded in the word. SVM can make use of this information, and better classify lyrics.

However, BERT and XLNet can handle unseen words in the vocabulary. To be specific, they break down the unobserved words into sub-words (Devlin et al. 2019). For example, the word "studying" would be divided into "study+##ing," where ## is used to represent sub-words. When dealing with words like "studies," stemming would cut the word into "studi," which would hinder the model from grouping "studies" and "study." On the contrary, lemmatization does not change the word for conjugation related to person and number. Therefore, lemmatization does not impede the unseen word process of BERT and XLNet.

Second, in Group 2, only XLNet observes the difference between stop-words removal and noise removal. This suggests that the punctuations and numbers in the lyrics may be useful for XLNet to detect music emotion, whereas the stop-words do not provide much help.

Moreover, for the BERT model, only the model with lowercase conversion (LC) significantly outperformed the model without any text preprocessing techniques (see Figure 4.1-3). This indicates that BERT is not sensitive to the division of sentences as the uppercase letter usually labels the beginning of a new sentence. This characteristic of BERT may relate to its Next Sentence Prediction pre-training task (Section 2.3.3.2), which learns the relationship between sentences. The improvement of performance by lowercase conversion suggests that the division of sentences in lyrics may not affect the ability of BERT to detect the emotion of music.

Overall, the validation results of the three methods on the Moodylyrics4Q Dataset showed that text preprocessing techniques have different effects, when combined and when used with

different methods. Therefore, researchers need to carefully analyze all the possible combinations of text preprocessing techniques, instead of simply applying them or disabling them.

4.3.2 Discussion on Test Results

The test results on the MoodyLyrics4Q dataset and the AllMusic Dataset are quite different. The MoodyLyrics4Q test results show that BERT obtained better performance than SVM, and XLNet performed better than BERT. However, for the AllMusic Dataset, the test results show that although XLNet achieved a higher F1-score than SVM, BERT performed worse than SVM.

The difference between the annotation processes of the two datasets may explain this difference. After the automatic annotation based on the word map, the AllMusic Dataset was checked manually, whereas the annotation of the MoodyLyrics4Q Dataset did not have this procedure. Therefore, the MoodyLyrics4Q Dataset may contain more noise, where the lyrics were annotated incorrectly.

As a result, the BERT model may overfit this noise, whereas SVM and XLNet were better able to handle it. This may be due to the discrepancy between pre-training and finetuning caused by the masked tokens in the Masked Language Modelling task (Section 2.3.3.2). The masked tokens used in the pre-training process are absent from the finetuning. These absent tokens may contain useful information that is crucial in detecting this noise.

Another possible explanation for the different results between the two datasets is that although transfer learning makes use of the knowledge obtained from the pre-training process, thus learning much faster than other models with the same amount of data, transfer learning models may easily overfit the training data with a small dataset. This thesis tried to avoid overfitting by

freezing most layers in the pre-trained model. However, the datasets used may still be too small compared to the amount of data used during the pre-training process.

Chapter 5 Conclusion

This chapter presents the concluding remarks on the results of the experiments previously conducted. Also discussed are some ways this thesis could be improved and possible directions for future research.

5.1 Concluding Remarks

The SVM model achieved a 66.87% F1-score on the AllMusic Dataset with lowercase conversion, noise removal, stop-words removal, and stemming. The BERT model obtained 63.59%, and the XLNet model achieved 70.09%. This result showed that although transfer learning methods can improve performance, they do not always work better than traditional machine learning methods when the dataset is relatively small.

The text preprocessing results are different for each classifier, suggesting that all combinations of text preprocessing should be carefully examined for different methods, even when the tasks are the same.

Among the three methods, the application of lowercase conversion showed a significant improvement for BERT. Lowercase conversion removes the division between sentences, which BERT learns in the pre-trained process. This improvement suggests that sentence division in lyrics may not be important in analyzing the emotion of music.

Word normalization techniques, stemming and lemmatization, showed different effects on different methods. The traditional machine learning model, SVM, cannot deal with unseen words, so stemming significantly improved its performance. However, BERT and XLNet have their way of processing unobserved words. Large changes in the word form would hinder them from learning new words instead.

In addition, for the XLNet model, the model with stop-words removal performed significantly better than the model with noise removal. This indicates that the stop-words in the lyrics may not contain emotional information, whereas the punctuations and numbers may contain some useful information.

5.2 Future Research

Possible directions for improving the performance of MER on lyrics include:

- Creating an annotated MER lyrics dataset on a much larger number of songs. Although transfer learning reduces the need for a large amount of finetuning data, a small finetuning dataset can easily lead to overfitting. This thesis tried to avoid overfitting by freezing most layers of the pre-trained model. More data may be more helpful in improving system performance and preventing overfitting.
- Exploring pre-trained Transformer-based word embeddings on lyrics in other languages. This thesis only focused on English lyrics. However, whether pre-trained models can extract useful information related to music mood from lyrics in other languages remains unclear.
- Trying all the possible combinations of frozen layers and active layers of the pre-trained model in the finetuning process. The preliminary experiments of this thesis started with freezing layers from the end, and stopped when the model did not overfit the training data. However, other combinations of layer settings may give different results.
- Experimenting with larger word embedding lengths. In this thesis, the maximum length of word embeddings was set to 512 for deep learning models, as 92.67% of

the lyrics in the datasets contain fewer than 500 words. However, several lyrics consist of more words.

- Implementing different types of neural networks to finetune the pre-trained word embeddings, such as different Feedforward Neural Network structures, Convolutional Neural Networks and Recurrent Neural Networks. This thesis applied the finetuning methods stated in Devlin et al. (2019). Nevertheless, researchers have tried to combine pre-trained word embeddings and other types of neural network structures.

REFERENCES

Aljanaki, Anna, Yi-Hsuan Yang, and Mohammad Soleymani. 2017. "Developing a Benchmark for Emotional Analysis of Music." *PloS One* 12 (3): e0173392.

Aucouturier, Jean-Julien, François Pachet, Pierre Roy, and Anthony Beurivé. 2007. "Signal + Context = Better Classification." In *Proceedings of the 8th International Conference on Music Information Retrieval*, 425–30.

Abdillah, Jiddy, Ibnu Asror, and Yanuar Firdaus Arie Wibowo. 2020. "Emotion Classification of Song Lyrics Using Bidirectional LSTM Method with GloVe Word Representation Weighting." *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)* 4 (4): 723–29.

Aryafar, Kamelia, and Ali Shokoufandeh. 2011. "Music Genre Classification Using Explicit Semantic Analysis." In *Proceedings of the 1st International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies*, 33. ACM Press.

Aryafar, Kamelia, and Ali Shokoufandeh. 2014. "Multimodal Music and Lyrics Fusion Classifier for Artist Identification." In *Proceedings of 13th International Conference on Machine Learning and Applications*, 506–9.

Agirrezabal, Manex, Iñaki Alegria, and Mans Hulden. 2016. "Machine Learning for Metrical Analysis of English Poetry." In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, 772–81.

Akella, Revanth, and Teng-Sheng Moh. 2019. “Mood Classification with Lyrics and ConvNets.” In *Proceedings of the 18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 511–14.

Ahmad, Shakeel, Muhammad Zubair Asghar, Fahad Mazaed Alotaibi, and Sherafzal Khan. 2020. “Classification of Poetry Text Into the Emotional States Using Deep Learning Technique.” *IEEE Access* 8: 73865–78.

An, Yunjing, Shutao Sun, and Shujuan Wang. 2017. “Naive Bayes Classifiers for Music Emotion Classification Based on Lyrics.” In *16th IEEE/ACIS International Conference on Computer and Information Science*, edited by Guobin Zhu, Shaowen Yao, Xiaohui Cui, and Simon Xu, 635–38.

Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. “A Training Algorithm for Optimal Margin Classifiers.” In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, 144–52.

Bischoff, Kerstin, Claudiu S. Firan, Raluca Paiu, Wolfgang Nejdl, Cyril Laurier, and Mohamed Sordo. 2009a. “Music Mood and Theme Classification: A Hybrid Approach.” In *Proceedings of the International Society for Music Information Retrieval Conference*, 657–62.

Bischoff, Kerstin, Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. 2009b. “How Do You Feel about ‘Dancing Queen’?: Deriving Mood & Theme Annotations from User Tags.” In *Proceedings of the 2009 Joint International Conference on Digital Libraries*, edited by Fred Heath, Mary Lynn Rice-Lively, and Richard Furuta, 285–94.

Barrett, Lisa Feldman, and James A. Russell. 1999. "The Structure of Current Affect: Controversies and Emerging Consensus." *Current Directions in Psychological Science* 8 (1): 10–14.

Beuthe, Michel, Bart Jourquin, Nathalie Urbain, Inge Lingemann, and Barry Ubbels. 2014. "Climate Change Impacts on Transport on the Rhine and Danube: A Multimodal Approach." *Transportation Research Part D: Transport and Environment* 27: 6–11.

Bradley, Margaret M., and Peter J. Lang. 1999. "Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings." Vol. 30, No. 1. Technical report C-1, the center for research in psychophysiology, University of Florida.

Bafna, Prafulla, Dhanya Pramod, and Anagha Vaidya. 2016. "Document Clustering: TF-IDF Approach." In *Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 61–66.

Bouckaert, Remco R., and Eibe Frank. 2004. "Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms." In *Advances in Knowledge Discovery and Data Mining*, edited by Honghua Dai, Ramakrishnan Srikant, and Chengqi Zhang, 3056:3–12.

Bertin-Mahieux, Thierry, Daniel Ellis, Brian Whitman, and Paul Lamere. 2011. "The Million Song Dataset." In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, 591–96.

Capurso, Alexander, Vineent R. Fisichelli, Leonard Gilman, Emil A. Gutheil, Jay T. Wright, and Frances Paperte. 1952. *Music and Your Emotions*. New York, NY: Liveright Publishing Corporation.

Chang, Chin-Jui, Chun-Yi Lee, and Yi-Hsuan Yang. 2021. "Variable-Length Music Score Infilling via XLNet and Musically Specialized Positional Encoding." In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 97–104.

Cao, Chuan, and Ming Li. 2009. "Thinkit's Submissions for MIREX2009 Audio Music Classification and Similarity Tasks." *Music Information Retrieval Evaluation EXchange (MIREX)*.

Coutinho, Eduardo, Felix Weninger, Björn W. Schuller, and Klaus R. Scherer. 2014. "The Munich LSTM-RNN Approach to the MediaEval 2014 'Emotion in Music' Task." In *Working Notes Proceedings of the MediaEval 2014 Workshop*. Vol. 1263.

Çano, Erion, and Maurizio Morisio. 2017a. "MoodyLyrics: A Sentiment Annotated Lyrics Dataset." In *Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*, 118–24.

Cano, Erion, and Maurizio Morisio. 2017b. "Music Mood Dataset Creation Based on Last FM Tags." In *Proceeding of the 4th International Conference on Artificial Intelligence and Applications*, 15–26.

Cespedes-Guevara, Julian, and Tuomas Eerola. 2018. "Music Communicates Affects, Not Basic Emotions – A Constructionist Account of Attribution of Emotional Meanings to Music." *Frontiers in Psychology* 9 (February): 215.

Chowdhary, K.R. 2020. *Fundamentals of Artificial Intelligence*. New Delhi: Springer India.

Choi, Keunwoo, Gyorgy Fazekas, Mark Sandler, and Kyunghyun Cho. 2017. "Convolutional Recurrent Neural Networks for Music Classification." In *Proceedings of the 2017 IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2392–96.

Casey, Michael A., Remco Veltkamp, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney. 2008. "Content-Based Music Information Retrieval: Current Directions and Future Challenges." *Proceedings of the IEEE* 96 (4): 668–96.

Damasio, Antonio R. 1994. *Descartes' Error: Emotion, Reason, and the Human Brain*. New York, NY: Grosset/Putnam.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–86.

Daelemans, Walter, Jakub Zavrel, Kurt van der Sloot, and Antal van den Bosch. 2002. "Timbl: Tilburg Memory-Based Learner." Technical Report ILK 02-10. Tilburg, the Netherlands: Tilburg University.

Dai, Zihang, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context." In *Proceedings of the 57th Conference of the Association for Computational Linguistics, Volume 1: Long Papers*, 2978–88.

Edmonds, Darren, and João Sedoc. 2021. "Multi-Emotion Classification for Song Lyrics." In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment*

and Social Media Analysis, edited by Orphée De Clercq, Alexandra Balahur, João Sedoc, Valentin Barrière, Shabnam Tafreshi, Sven Buechel, and Véronique Hoste, 221–35.

Eisenstein, Jacob. 2019. *Introduction to Natural Language Processing*. Adaptive Computation and Machine Learning Series. MIT Press.

Ekman, Paul. 1992. "An Argument for Basic Emotions." *Cognition and Emotion* 6 (3–4): 169–200.

Ekman, Paul. 1993. "Facial Expression and Emotion." *American Psychologist* 48 (4): 384–92.

Ekman, Paul, and Richard J. Davidson. 1994. *The Nature of Emotion: Fundamental Questions.* Series in Affective Science. New York: Oxford University Press.

Eerola, Tuomas, and Jonna K. Vuoskoski. 2013. "A Review of Music and Emotion Studies: Approaches, Emotion Models, and Stimuli." *Music Perception* 30 (3): 307–40.

Futrelle, Joe, and J Stephen Downie. 2002. "Interdisciplinary Communities and Research Issues in Music Information Retrieval." In *Proceedings of the 3rd International Conference on Music Information Retrieval*, 215–21.Farnsworth, Paul R. 1954. "A Study of the Hevner Adjective List." *The Journal of Aesthetics and Art Criticism* 13 (1): 97–103.

Fell, Michael. 2020. "Natural Language Processing for Music Information Retrieval: Deep Analysis of Lyrics Structure and Content." PhD Dissertation. Université Côte d'Azur.

Fell, Michael, Elena Cabrio, Michele Corazza, and Fabien Gandon. 2019. "Comparing Automated Methods to Detect Explicit Content in Song Lyrics." In *Proceedings of Recent Advances in Natural Language Processing*.

Gabrielsson, Alf, and Erik Lindström. 2001. “The Influence of Musical Structure on Emotional Expression.” In *Music and Emotion: Theory and Research*, edited by Patrik N. Juslin, and John A. Sloboda, 223–48. Oxford University Press.

Good, I. J. 1952. “Rational Decisions.” *Journal of the Royal Statistical Society. Series B (Methodological)* 14 (1): 107–14.

Ghazvininejad, Marjan, Xing Shi, Yejin Choi, and Kevin Knight. 2016. “Generating Topical Poetry.” In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1183–91.

Golder, Scott A, and Bernardo A Huberman. 2006. “Usage Patterns of Collaborative Tagging Systems.” *Journal of Information Science* 32 (2): 198–208.

Guo, Wenzhong, Jianwen Wang, and Shiping Wang. 2019. “Deep Multimodal Representation Learning: A Survey.” *IEEE Access* 7: 63373–94.

He, Hui, Jianming Jin, Yuhong Xiong, Bo Chen, Wu Sun, and Ling Zhao. 2008. “Language Feature Mining for Music Emotion Classification via Supervised Learning from Lyrics.” In *Advances in Computation and Intelligence*, edited by Lishan Kang, Zhihua Cai, Xuesong Yan, and Yong Liu, 426–35.

Hevner, Kate. 1935. “Expression in Music: A Discussion of Experimental Studies and Theories.” *Psychological Review* 42 (2): 186–204.

Huron, David. 2000. “Perceptual and Cognitive Applications in Music Information Retrieval.” *Perception* 10 (1): 83–92.

Hu, Xiao. 2010. “Improving Music Mood Classification Using Lyrics, Audio and Social Tags.” PhD Dissertation. University of Illinois at Urbana-Champaign.

Hu, Xiao, Mert Bay, and J. Stephen Downie. 2007. "Creating a Simplified Music Mood Classification Ground-Truth Set." In *Proceedings of the 8th International Conference on Music Information Retrieval*, 309–10.

Hu, Xiao, and J. Stephen Downie. 2010. "Improving Mood Classification in Music Digital Libraries by Combining Lyrics and Audio." In *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, 159–68.

Hu, Xiao, J. Stephen Downie, and Andreas F. Ehmann. 2009. "Lyric Text Mining in Music Mood Classification." In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 411–16.

Hu, Xiao, Stephen J. Downie, Cyril Laurier, Mert Bay, and Andreas Ehmann. 2008. "The 2007 MIREX Audio Mood Classification Task: Lessons Learned." In *Proceedings of the 9th International Conference on Music Information Retrieval*, 462–67.

Hu, Yajie, Xiaoou Chen, and Deshun Yang. 2009. "Lyric-Based Song Emotion Detection with Affective Lexicon and Fuzzy Clustering Method." In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 123–28.

Harris, Zellig S. 1954. "Distributional Structure." *Word* 10 (2–3): 146–62.

Imamura, Kenji, and Eiichiro Sumita. 2019. "Recycling a Pre-Trained BERT Encoder for Neural Machine Translation." In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, 23–31. Association for Computational Linguistics.

Jaimes, Alejandro, and Nicu Sebe. 2005. "Multimodal Human Computer Interaction: A Survey." In *Computer Vision in Human-Computer Interaction*, edited by Nicu Sebe, Michael Lew, and Thomas S. Huang, 1–15.

Juslin, Patrik N., and Petri Laukka. 2004. “Expression, Perception, and Induction of Musical Emotions: A Review and a Questionnaire Study of Everyday Listening.” *Journal of New Music Research* 33 (3): 217–38.

Kim, Minho, and Hyuk-Chul Kwon. 2011. “Lyrics-Based Emotion Classification Using Feature Selection by Partial Syntactic Analysis.” In *Proceedings of the IEEE 23rd International Conference on Tools with Artificial Intelligence,* 960–64.

Kaminskas, Marius, and Francesco Ricci. 2012. “Contextual Music Information Retrieval and Recommendation: State of the Art and Challenges.” *Computer Science Review* 6 (2): 89–119.

Kovaleva, Olga, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. “Revealing the Dark Secrets of BERT.” In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 4364–73.

Kesarwani, Vaibhav. 2018. “Automatic Poetry Classification Using Natural Language Processing.” Master’s Thesis. University of Ottawa.

Kim, Youngmoo E., Erik M. Schmidt, Raymond Migneco, Brandon G. Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A. Speck, and Douglas Turnbull. 2010. “Music Emotion Recognition: A State-of-the-Art Review.” In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 255–66.

Laurier, Cyril, Jens Grivolla, and Perfecto Herrera. 2008. “Multimodal Music Mood Classification Using Audio and Lyrics.” In Proceedings of the *7th International Conference on Machine Learning and Applications*, 688–93.

Laurier, Cyril, Mohamed Sordo, Joan Serra, and Perfecto Herrera. 2009. "Music Mood Representations from Social Tags." In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 381–86.

Lahat, Dana, Tülay Adali, and Christian Jutten. 2015. "Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects." *Proceedings of the IEEE* 103 (9): 1449–77.

Lee, Harin, Frank Hoeger, Marc Schoenwiesner, Minsu Park, and Nori Jacoby. 2021. "Cross-Cultural Mood Perception in Pop Songs and Its Alignment with Mood Detection Algorithms." In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 366–373.

Lee, Jin Ha, and J. Stephen Downie. 2004. "Survey of Music Information Needs, Uses, And Seeking Behaviours: Preliminary Findings." In *Proceedings of the 5th International Conference on Music Information Retrieval*, 441–46.

Loshchilov, Ilya, and Frank Hutter. 2019. "Decoupled Weight Decay Regularization." In *Proceedings of the 7th International Conference on Learning Representations.*

Lamere, Paul. 2008. "Social Tagging and Music Information Retrieval." *Journal of New Music Research* 37 (2): 101–14.

Lu, Qi, Xiaoou Chen, Deshun Yang, and Jun Wang. 2010. "Boosting for Multi-Modal Music Emotion Classification." In *Proceedings of the 11th International Society for Music Information Retrieval Conference,* 105–10.

Manning, Christopher D., Prabhakar. Raghavan, and Hinrich. Schütze. 2008. *Introduction to Information Retrieval*. New York, NY: Cambridge University Press.

McEnnis, Daniel, Cory McKay, Ichiro Fujinaga, and Philippe Depalle. 2005. “JAudio: A Feature Extraction Library.” In *Proceedings of the 6th International Conference on Music Information Retrieval*, 600–3.

Miller, George A. 1995. “WordNet: A Lexical Database for English.” *Communications of the ACM* 38 (11): 39–41.

Markov, Konstantin, and Tomoko Matsui. 2014. “Dynamic Music Emotion Recognition Using State-Space Models.” In *Working Notes Proceedings of the MediaEval 2014 Workshop*. Vol. 1263.

Meyer, Leonard B. 1956. *Emotion and Meaning in Music*. Chicago, IL: University of Chicago Press.

Munezero, Myriam, Calkin Suero Montero, Erkki Sutinen, and John Pajunen. 2014. “Are They Different? Affect, Feeling, Emotion, Sentiment, and Opinion Detection in Text.” *IEEE Transactions on Affective Computing* 5 (2): 101–11.

Malik, Miroslav, Sharath Adavanne, Konstantinos Drossos, Tuomas Virtanen, Dasa Ticha, and Roman Jarina. 2017. “Stacked Convolutional and Recurrent Neural Networks for Music Emotion Recognition.” In *Proceedings of the 14th Sound and Music Computing Conference*, 208–213.

Mihalcea, Rada, and Carlo Strapparava. 2012. “Lyrics, Music, and Emotions.” In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language* Learning, 590–99.

Malheiro, Ricardo, Renato Panda, Paulo Gomes, and Rui Pedro Paiva. 2018. "Emotionally-Relevant Features for Classification and Regression of Music Lyrics." *IEEE Transactions on Affective Computing* 9 (2): 240–54.

Maheshwari, Tanish, Tarpara Nisarg Bhaveshbhai, and Mitali Halder. 2021. "The Power of Visual Analytics and Language Processing to Explore the Underlying Trend of Highly Popular Song Lyrics." *Engineering and Applied Science Letters* 4 (3): 19–29.

Nakamura, Keita, Takako Fujisawa, and Takasaki Kyoudou. 2017. "Music Recommendation System Using Lyric Network." In *Proceedings of IEEE 6th Global Conference on Consumer Electronics (GCCE)*, 1–2.

Oxford English Dictionary Online, s.v. December 2021. "Sentiment." Oxford University Press. Accessed 8 March 2022, https://www-oed-com.proxy3.library.mcgill.ca/view/Entry/176056?redirectedFrom=sentiment

Patra, Braja Gopal, Dipankar Das, and Sivaji Bandyopadhyay. 2015. "Mood Classification of Hindi Songs Based on Lyrics." In *Proceedings of the 12th International Conference on Natural Language Processing*, 261–67.

Patra, Braja Gopal, Dipankar Das, and Sivaji Bandyopadhyay. 2016. "Multimodal Mood Classification Framework for Hindi Songs." *Computación y Sistemas* 20 (3): 515–26.

Peeters, Geoffroy. 2008. "A Generic Training and Classification System for MIREX08 Classification Tasks: Audio Music Mood, Audio Genre, Audio Artist and Audio Tag." In *Proceedings of the 9th International Conference on Music Information Retrieval*.

Peeters, Geoffroy. 2008. “A Generic Training and Classification System for MIREX08 Classification Tasks: Audio Music Mood, Audio Genre, Audio Artist and Audio Tag.” In *Extended Abstract MIREX08*.

Park, Jiyoung, Jongpil Lee, Juhan Nam, Jangyeon Park, and Jung-Woo Ha. 2017. “Representation Learning Using Artist Labels for Audio Classification Tasks.” *Music Information Retrieval Evaluation EXchange (MIREX)*.

Pyrovolakis, Konstantinos, Paraskevi Tzouveli, and Giorgos Stamou. 2022. “Multi-Modal Song Mood Detection with Deep Learning.” *Sensors* 22 (3): 1065.

Porter, Martin F. 1980. “An Algorithm for Suffix Stripping.” *Program* 14 (3): 130–37.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. “Glove: Global Vectors for Word Representation.” In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 1532–43.

Plutchik, Robert. 2001. “The Nature of Emotions: Human Emotions Have Deep Evolutionary Roots, a Fact That May Explain Their Complexity and Provide Tools for Clinical Practice.” *American Scientist* 89 (4): 344–50.

Panda, Renato, Ricardo Manuel Malheiro, and Rui Pedro Paiva. 2020a. “Audio Features for Music Emotion Recognition: A Survey.” *IEEE Transactions on Affective Computing*, 1–1.

Panda, Renato, Ricardo Malheiro, and Rui Pedro Paiva. 2020b. “Novel Audio Features for Music Emotion Recognition.” *IEEE Transactions on Affective Computing* 11 (4): 614–26.

Poggio, Tomaso, and Federico Girosi. 1990. “Networks for Approximation and Learning.” *Proceedings of the IEEE* 78 (9): 1481–97.

Rosenblatt, Frank. 1957. “The Perceptron–a Perceiving and Recognizing Automation.” Cornell Aeronautical Laboratory.

Russell, James A. 1980. “A Circumplex Model of Affect.” *Journal of Personality and Social Psychology* 39 (6): 1161–78.

Russell, James A. 1983. “Pancultural Aspects of the Human Conceptual Organization of Emotions.” *Journal of Personality and Social Psychology* 45 (6): 1281–88.

Rangarajan, Rohit. 2015. “Generating Music from Natural Language Text.” In *Proceedings of 10th International Conference on Digital Information Management (ICDIM)*, 85–88.

Sung, Bo-Hsun, and Shih-Chieh Wei. 2021. “BECMER: A Fusion Model Using BERT and CNN for Music Emotion Recognition.” In *Proceedings of IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI)*, 437–44.

Strapparava, Carlo, and Alessandro Valitutti. 2004. “WordNet-Affect: An Affective Extension of WordNet.” In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, 1083–86.

Schubert, Emery. 2003. “Update of the Hevner Adjective Checklist.” *Perceptual and Motor Skills* 96 (3): 1117–1122.

Schubert, Emery. 2007. “The Influence of Emotion, Locus of Emotion and Familiarity upon Preference in Music.” *Psychology of Music* 35 (3): 499–515.

Salton, Gerard, Anita Wong, and Chung-Shu Yang. 1975. “A Vector Space Model for Automatic Indexing.” *Communications of the ACM* 18 (11): 613–20.

Salton, Gerard, and Christopher Buckley. 1988. “Term-Weighting Approaches in Automatic Text Retrieval.” *Information Processing & Management* 24 (5): 513–23.

Sloboda, John A., and Patrik N. Juslin. 2001. “Psychological Perspectives on Music and Emotion.” In *Music and Emotion: Theory and Research*, edited by Patrik N. Juslin, and John A. Sloboda, 71–104. Oxford University Press.

Stets, Jan E. 2006. “Emotions and Sentiments.” In *Handbook of Social Psychology*, edited by John Delamater, 309–35. Boston, MA: Springer US.

Sjöberg, Lennart, Erland Svensson, and Lars-Olof Persson. 1979. “The Measurement of Mood.” *Scandinavian Journal of Psychology* 20 (1): 1–18.

Schoen, Max, and Esther L. Gatewood. 1927. “The Mood Effects of Music.” In *The Effects of Music*, edited by Max Schoen, 131–51. Routledge.

Soleymani, Mohammad, Michael N. Caro, Erik M. Schmidt, and Yi-Hsuan Yang. 2013. “The MediaEval 2013 Brave New Task: Emotion in Music.” In *Proceedings of the MediaEval Multimedia Benchmark Workshop*. Vol. 1043.

Sloboda, John A., and Susan A. O’Neill. 2001. “Emotions in Everyday Listening to Music.” In *Music and Emotion: Theory and Research*, edited by Patrik N. Juslin, and John A. Sloboda, 415–429. Oxford University Press.

Saari, Pasi, Tuomas Eerola, Gyorgy Fazekas, and Mark Sandler. 2013a. “Using Semantic Layer Projection for Enhancing Music Mood Prediction with Audio Features.” In *Proceedings of Sound and Music Computing Conference*, 722–28.

Saari, Pasi, Tuomas Eerola, György Fazekas, Mathieu Barthet, Olivier Lartillot, and Mark B. Sandler. 2013b. “The Role of Audio and Tags in Music Mood Prediction: A Study Using Semantic Layer Projection.” In *Proceedings of the 14th International Society for Music Information Retrieval Conference,* 201–6.

Scruton, Roger. 1983. *The Aesthetic Understanding: Essays in the Philosophy of Art and Culture.* South Bend, IN: St. Augustine's Press.

Schimmack, Ulrich, and Alexander Grob. 2000. "Dimensional Models of Core Affect: A Quantitative Comparison by Means of Structural Equation Modeling." *European Journal of Personality* 14 (4): 325–45.

Stratton, Valerie N., and Annette H. Zalanowski. 1994. "Affective Impact of Music Vs. Lyrics." *Empirical Studies of the Arts* 12 (2): 173–84.

Tzanetakis, George. 2007. "Marsyas Submissions to MIREX 2007." *Music Information Retrieval Evaluation EXchange (MIREX)*.

Tizhoosh, Hamid R., Farhang Sahba, and Rozita Dara. 2008. "Poetic Features for Poem Recognition: A Comparative Study." *Journal of Pattern Recognition Research* 3 (1): 24–39.

Uysal, Alper Kursat, and Serkan Gunal. 2014. "The Impact of Preprocessing on Text Classification." *Information Processing & Management* 50 (1): 104–12.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." In *Advances in Neural Information Processing Systems*. Vol. 30.

Van de Cruys, Tim. 2020. "Automatic Poetry Generation from Prosaic Text." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2471–80.

Van Zaanen, Menno, and Pieter Kanters. 2010. "Automatic Mood Classification Using TF* IDF Based on Lyrics." In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 75–80.

Weninger, Felix, Florian Eyben, and Björn W. Schuller. 2013. “The TUM Approach to the MediaEval Music Emotion Task Using Generic Affective Audio Features.” In *Proceedings of the MediaEval Multimedia Benchmark Workshop*. Vol. 1043.

Wu, Ming-Ju, and Jyh-Shing Roger Jang. 2013. “MIREX 2013 Submissions for Train/Test Tasks (Draft).” *Music Information Retrieval Evaluation EXchange (MIREX)*, 88–89.

Wang, Xing, Xiaoou Chen, Deshun Yang, and Yuqian Wu. 2011. “Music Emotion Classification of Chinese Songs Based on Lyrics Using TF* IDF and Rhyme.” In *Proceedings of the 12th International society for Music Information Retrieval Conference*, 765–70.

Xue, Hao, Like Xue, and Feng Su. 2015. “Multimodal Music Mood Classification by Fusion of Audio and Lyrics.” In *MultiMedia Modeling - 21st International Conference, Proceedings, Part II*, edited by Xiangjian He, Suhuai Luo, Dacheng Tao, Changsheng Xu, Jie Yang, and Muhammad Abul Hasan, 8936:26–37.

Xu, Hu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. “BERT Post-Training for Review Reading Comprehension and Aspect-Based Sentiment Analysis.” In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, 2324–35.

Yang, Yi-Hsuan, and Homer H. Chen. 2012. “Machine Recognition of Music Emotion: A Review.” *ACM Transactions on Intelligent Systems and Technology* 3 (3): 1–30.

Yang, Dan, and Won-Sook Lee. 2009. “Music Emotion Identification from Lyrics.” In *Proceedings of the 11th IEEE International Symposium on Multimedia*, 624–29.

Yates, Andrew, Rodrigo Nogueira, and Jimmy Lin. 2021. “Pretrained Transformers for Text Ranking: BERT and Beyond.” In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 1154–56. ACM.

Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. “How Transferable Are Features in Deep Neural Networks?” In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, 3320–28.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. “XLNet: Generalized Autoregressive Pretraining for Language Understanding.” In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*, 5754–64.

Zentner, Marcel, and Tuomas Eerola. 1993. “Self-Report Measures and Models.” In *Handbook of Music and Emotion: Theory, Research, Applications*, edited by Patrik N. Juslin, 187–221. Oxford University Press.

Zeng, Mingliang, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu. 2021. “MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training.” In *Findings of the Association for Computational Linguistics*, 791–800.