Project

Master's degree in Data Science

CRIATIONAL: Generation of Song Lyrics with Emotional Context Using Deep Learning Models

Mariana Oliveira Agostinho

Leiria, September 2025

Project

Master's degree in data science

CRIATIONAL: Generation of Song Lyrics with Emotional Context Using Deep Learning Models

Mariana Oliveira Agostinho

Project developed under the supervision of Doctor Ricardo Manuel da Silva Malheiro, professor at the School of Technology and Management of the Polytechnic Institute of Leiria.

Leiria, September 2025

This page is intentionally left blank

Acknowledgments

I wish to extend my heartfelt gratitude to Professor Doctor Ricardo Malheiro for his guidance, patience, and constant support throughout this project, essential to its completion.

I would also like to sincerely thank my parents, my sister, my friends, and particularly, Nuno Duarte, for their support and understanding during my master's degree, without which this achievement would not have been possible.

This page is intentionally left blank

Abstract

Language and music are fundamental human tools for expression, communication, and emotional connection. Music plays a central role in shaping identity, conveying feelings, and promoting social bonds, making it an intriguing domain for technological exploration. Replicating human creativity in music, especially in songwriting, presents a complex challenge, as natural language processing (NLP) and deep learning (DL) must capture both linguistic structure and emotional nuance.

This study investigates the generation of emotionally contextualised song lyrics using DL models, including LSTM, GPT-2, and T5, guided by Russell's Circumplex Model of Emotions. The models were evaluated on readability, coherence, perplexity, structural consistency, thematic alignment, and emotional accuracy. Results show that GPT-2, particularly when fine-tuned, achieves the best balance of coherence and emotional alignment, although it still lacks some musical features such as rhyme and rhythm. LSTM exhibits patterned sequences but high variability, while T5 struggles with structural consistency and repetitive output, highlighting the challenges of small, non-specialised datasets.

Overall, the work confirms the feasibility of using DL models as creative support in lyric composition, capable of offering emotionally expressive material to inspire musicians, while also pointing to the need for larger datasets and models tailored to musical structure to achieve fully convincing results.

Keywords: deep learning, natural language processing, song lyrics generation, emotion-aware lyrics

This page is intentionally left blank

Resumo

A linguagem e a música são ferramentas humanas fundamentais para expressão, comunicação e ligação emocional. A música desempenha um papel fundamental na formação da identidade, na transmissão de sentimentos e na promoção de laços sociais, tornando-se um domínio particularmente interessante para exploração tecnológica. Replicar a criatividade humana na música, especialmente na composição de letras, constitui um desafio complexo, uma vez que o processamento de linguagem natural e o deep learning devem capturar tanto a estrutura linguística como a nuance emocional.

Este estudo investiga a geração de letras de música contextualizadas emocionalmente através de modelos DL, incluindo LSTM, GPT-2 e T5, orientados pelo Modelo Circumplexo de Emoções de Russell. Os modelos foram avaliados quanto à legibilidade, coerência, perplexidade, consistência estrutural, alinhamento temático e precisão emocional. Os resultados mostram que o GPT-2, particularmente quando ajustado (*fine-tuned*), atinge o melhor equilíbrio entre coerência e alinhamento emocional, embora ainda careça de algumas características musicais, como rima e ritmo. O LSTM apresenta sequências padronizadas, mas com elevada variabilidade, enquanto o T5 tem dificuldades com a consistência estrutural e tende a produzir conteúdos repetitivos, destacando os desafios de trabalhar com conjuntos de dados pequenos e não especializados.

De forma geral, o trabalho confirma a viabilidade de utilizar modelos DL como apoio criativo na composição de letras, capazes de fornecer material emocionalmente expressivo para inspirar músicos, enquanto aponta para a necessidade de conjuntos de dados maiores e modelos ajustados à estrutura musical para alcançar resultados totalmente convincentes.

Palavras-chave: *deep learning*, processamento de linguagem natural, geração de letras de música, letras com contexto emocional

This page is intentionally left blank

Contents

This page is intentionally left blank

List of Figures

This page is intentionally left blank

List of Tables

This page is intentionally left blank

List of symbols and nomenclature

AI – Artificial Intelligence

ANN – Artificial Neural Networks

BERT – Bidirectional Encoder Representations from Text

BPE – Byte-Pair Encoding

DL – Deep Learning

FKGL – Flesch-Kincaid Grade Level

FRE – Flesch Reading Ease

GAI – Generative Artificial Intelligence

GPT – Generative Pre-Training Transformer

GPUs - Graphic Processing Units

LLMs – Large Language Models

LMER – Lyrics-based Music Emotion Recognition

LSTM – Long Short-Term Memory

MLM – Masked Language Models

MER – Music Emotion Recognition

MIR – Music Information Retrieval

MIREX – Music Information Retrieval Evaluation eXchange

ML – Machine Learning

NLG – Natural Language Generator

NLP – Natural Language Processing

NLU – Natural Language Understanding

NN – Neural Networks

POS Tagging – Part-Of-Speech Tagging

RNN – Recurrent Neural Networks

RoBERTa – Robustly optimized BERT approach

SFT – Supervised Fine-Tuning

SLR – Systematic Literature Review

T5 – Text-To-Text Transfer Transformer

This page is intentionally left blank

1. Introduction

Emotions are a primary reason why people connect to music, since it has the power to express our innermost feelings, give us goose pimples, make us cry or trigger specific memories (Gomez-Canon et al., 2021). Creating, listening to, dancing to, and talking about music are fundamental human experiences and an essential way of connecting (Clair, 2024).

Over the last few years, the way we experience music has altered significantly, driven by the rise of Artificial Intelligence (AI). From composition to consumption, AI is reshaping the music industry, offering new tools that support creativity and streamline processes. When used responsibly, AI has the potential to enhance human creativity, enabling artists and composers to develop and grow.

One particularly complex aspect of music creation is songwriting. Composing lyrics requires careful consideration of several components, such as vocabulary, rhyme flow, and emotional coherence. Traditionally, algorithmic methods for generating lyrics were limited in both expressiveness and coherence. However, AI-driven music generation has advanced considerably with advancements in Machine Learning (ML), a type of AI that allows machines to learn from data without requiring explicit programming, and Deep Learning (DL), a subfield of ML that employ neural networks to model and resolve complex problems. As a result, traditional algorithmic approaches have given way to sophisticated models that can capture complex musical structures and styles. Furthermore, ML and DL are being extensively studied as useful tools in the domain of Natural Language Processing (NLP). NLP enables a variety of tasks such as automatic translation, opinion mining, dialogue systems, question-and-answer systems, and text generation. Among these, Natural Language Generation (NLG) has garnered significant attention, as it enables the creation of dialogues and the production of more natural, fluid texts, with performance comparable to that of humans when combined with other tasks (Rodrigues et al., 2022).

Based on these studies, major platforms such as YouTube, TikTok, and Meta have launched AI-powered tools, and AI startups[1] have recently introduced programs that allow users to generate music from text prompts. These tools tend to operate similarly, analysing vast datasets and applying the patterns they contain to make probabilistic predictions.

[1] Suno and Udio

However, any AI music generator is only as effective as the data it has been trained on, which often results in stereotypical sounds within a genre or style instead of creating something unique, original, or exciting (Clair, 2024).

And this is just the beginning, as the field is attracting increasing investment, driven by the growing number of people consuming and producing music, which highlights the need for technological evolution. This has led to the recognition of a relatively recent research field called Music Information Retrieval (MIR), which emerged from the need to manage massive collections of digital music. From this field, Music Emotion Recognition (MER) has appeared as a subfield, representing a new direction for the organization and automatic extraction of music (Zhou, 2022). Together, these domains provide the foundation for exploring how AI can generate lyrics that are not only coherent but also emotionally meaningful.

1.1. Motivation

The integration of emotion into music generation remains a significant challenge in AI. While AI models are increasingly capable of producing lyrics that follow linguistic and structural conventions, they often lack emotional depth and personal nuance. Many existing systems generate lyrics that are generic or stereotypical, reflecting patterns in training data rather than producing content that feels authentic or emotionally engaging (Clair, 2024).

Given the cultural and personal importance of lyrics, this limitation is significant: in music, we cannot merely focus on what sounds good. Lyrics play a fundamental part, not only in the success of a song, but also in its social impact (Ballard et al., 1999). For a piece of music to be truly successful, it is essential to consider the life experiences of the artist, which are deeply rooted in their being (Clair, 2024). This is where the study of human emotions in music becomes essential, as music is, above all, an expression of emotions.

The motivation for this thesis arises from the need to close this gap: to develop AI models that not only generate syntactically correct lyrics but also capture emotional context meaningfully. By combining Deep Learning techniques with insights from NLP and MER, this work aims to explore how emotional awareness can be embedded in the generation of song lyrics, contributing to both academic knowledge and the evolving landscape of AI-assisted music creation.

1.2. Objectives and Contributions

This report describes the work carried out within the scope of the Project Curricular Unit of the master's degree in data science at the School of Technology and Management of Leiria. The main objective of this work is to develop a system capable of automatically generating song lyrics based on specific emotions, using DL based language generation models. The system is designed around Russell's Circumplex Model of Emotions, which allows emotions to be represented in terms of valence and arousal.

To achieve this objective, the work also involves implementing evaluation metrics and methodologies to assess the quality of the generated lyrics, with particular focus on emotional relevance, interpretability, and coherence. The work explores different DL architectures and strategies for conditioning text generation on emotional input, aiming to create a functional prototype that can generate meaningful, emotion-driven lyrics.

This research makes two significant contributions. First, it introduces a new framework for generating song lyrics driven by emotion, combining Russell's Circumplex Model of Emotion with DL techniques. This framework enables the controlled generation of song lyrics that accurately reflect specific emotional states, providing a systematic and reproducible approach to integrating psychological models of emotion with AI methods. Second, it presents a thorough evaluation of the system's outputs, assessing both the linguistic quality of the generated lyrics and how well they align with the target emotional context. This evaluation not only demonstrates the potential of using DL for emotion-based creative text generation but also provides insights into the approach's limitations and challenges, highlighting opportunities for future research and practical applications in AI-assisted music composition.

1.3. Methodology

The project is based on the application of DL techniques for the automatic generation of song lyrics with emotional context, guided by Russell's Circumplex Model of Emotions. The work focuses on the use of pre-trained Large Language Models (LLMs) and Recurrent Neural Network (RNN) based architectures.

The LLMs were employed via the Hugging Face library in Python. These models were applied both through direct inference and fine-tuning, using conditional text generation techniques based on prompt engineering. As these are pre-trained on large-scale

corpora, no additional dataset preparation was required. For the fine-tuning process, a ready-to-use dataset was provided by the supervisor, already cleaned and structured, thereby eliminating the need for data collection or preprocessing. The RNN-based model, using LSTM units, was trained on the same dataset provided by the supervisor. The model was implemented in Python, using the PyTorch framework.

The evaluation of the generated texts comprised both quantitative and qualitative components, focusing on semantic coherence and linguistic quality through the use of fluency and readability tests, as well as standard NLP metrics, compliance with predefined constraints, and verification of the presence of keywords and structural elements.

This methodology enabled a comparative analysis between modern LLM-based approaches and traditional sequential models, evaluating their respective performance within the creative task of emotion-driven lyric generation.

1.4. Structure of the Thesis

The preceding chapter (2) provides the background required to understand the study, including an overview of MER, NLP and DL. Chapter 3 reviews existing literature related to text and lyrics generation. Chapter 4 describes the methodology used to develop the system capable of generating song lyrics based on specific emotions. Chapter 5 focuses on the evaluation of the developed system and the lyrics it generates. Finally, Chapter 6 concludes the work and proposes potential directions for future research.

2. Background concepts

The way we interpret and respond to our surroundings influences who we are and impacts our overall well-being (UWA, 2019). Humans are the most remarkable species, with our greatest accomplishment being the ability to communicate and share information through language. Language is the vehicle for most of our thinking, expression, and cultural development, serving as the primary medium through which we store and transmit knowledge.

Similarly, music stands as one of the most widespread means of expression and communication, present in the daily lives of individuals of all ages and backgrounds around the world (Welch et al., 2020). It plays a vital role in emotional expression, identity, and social connection, making it a compelling domain for technological exploration.

The idea of replicating this ability comes when we discuss human, or natural, language – one of the most intricate and sophisticated aspects of our existence. Creating algorithms that can make sense of natural language is a significant challenge, since the ability to truly comprehend natural language has long eluded machines (Chollet, 2021). That is what NLP is about: using ML and large datasets to give computers the ability not to understand the language – a far more ambitious goal - but to process a piece of language as input and generate something meaningful in return.

In the context of this work, the aim is to explore how such computational approaches can be applied to the generation of emotionally contextualised song lyrics using a DL model. This chapter provides an overview of the key concepts for this task. It explores the connection between music and emotion, examining how emotions are expressed, perceived, and represented in lyrics through LMER. The chapter then introduces NLP and its techniques, which allow computers to process and generate meaningful language. Finally, the chapter presents DL approaches, such as RNNs, LSTMs, and Transformers, along with model optimisation and evaluation strategies, establishing a theoretical and technical foundation for emotion-aware lyric generation.

2.1. Music Emotion Recognition

MER is a computational task aimed at automatically identifying the emotional content conveyed by music, the emotions it induces and those perceived by the listener. To

achieve this, emotionally relevant features are extracted from the music, analysed, and then linked to specific emotions. MER is considered one of the most complex music description tasks within Music Information Retrieval (MIR), an interdisciplinary field focused on developing computational systems that assist humans in better understanding and managing music repertoire (Gomez-Canon et al., 2021).

With the significant growth in paid subscribers to music streaming platforms throughout the years, MIR systems have become increasingly important, particularly in e-commerce and on-demand streaming services where effective information retrieval and recommendation are essential. Although MER has advanced primarily by using acoustic features and social tags to identify and classify musical emotions, the role of lyrics — despite their crucial importance in eliciting emotions, enhancing musical enjoyment, and reflecting user traits and preferences — has been largely overlooked. Even though some studies indicate that emotion classifiers using features extracted from lyrics surpass those based on audio, the potential of lyrics in MER remains underappreciated (Agrawal et al., 2021).

To fully grasp the scope and complexity of MER, it is essential to examine not only how emotion is conceptualized in music but also how it is recognized, represented, and influenced by various musical and lyrical elements. So, to understand the role of emotions in music – and how music, in turn, affects our emotions - it is essential to study emotions and their influence on our musical experience. This represents the foundational concepts for this work and provides the starting point for understanding the project. These aspects are explored in the following sections.

2.1.1. Defining Emotion

Theories and hypotheses about emotions have a long history, dating back centuries. Emotion is far more complex to measure and define properly than many other human responses. As Fehr and Russel stated (Fehr & Russel, 1984) “Everybody knows what an emotion is, until you ask them a definition”.

Emotion, a term derived from the Latin *emovere* (meaning to move out or agitate), generally relates to affective disturbances in our experience that are directed towards events or objects in the world and frequently lead us to respond in particular ways. Rather than being homogeneous experiences, emotions are widely recognized as discrete phenomena, each defined by a unique pattern of feelings, physiological changes,

expressions, and motivations for action. This discrete nature is supported across multiple disciplines. Furthermore, emotions serve an important adaptive function: they provide meaning and guide our interactions with the world, enabling us to evaluate situations and prepare for appropriate actions (Scheve & Slaby, 2019).

In the context of music, these definitions of emotion are particularly relevant, as music is broadly recognised as a powerful vehicle for emotional expression and communication. Musical experiences frequently elicit emotional responses that align with the characteristics described above – bodily sensations, expressive reactions, and subjective feeling. However, unlike other emotionally charged stimuli, music does not necessarily refer to concrete events or objects in the world. Instead, it evokes emotion through abstract structures, such as melodic contour, harmonic progression, rhythm, and tempo. This unique quality of music challenges conventional emotion theories, leading researchers in music psychology to explore whether emotional responses to music truly correspond to basic emotions or represent a distinct category of aesthetic or affective experience (Juslin & Västfjäll, 2008).

2.1.2. Types of Emotion: Expressed, Induced and Perceived

In music, whether in audio or in lyrics, emotions can be divided into three categories (Malheiro, 2016):

- Expressed emotion – is the emotion that the performer aims to convey to the listeners.
- Perceived emotion - relates to the emotion that a listener perceives as being expressed in a song, which might differ from the emotion the performer intended to communicate.
- Induced emotion - describes the emotion that a listener personally feels in response to a song.

Even though the three types of emotions are interrelated, it is important to distinguish them. For example, while a performer may intend to express happiness in a song, a listener might perceive calmness, which could, in turn, induce feelings of sadness in them (Malheiro, 2016).

2.1.3. Representations of emotions

MIR systems use two distinct models to represent emotions: the categorical and the dimensional model. Each model highlights a particular aspect of human emotion, offering insight into how emotions are perceived and interpreted by the mind. These models evaluate an individual's true emotional state (P.S. & Mahalakshmi, 2017).

Categorical Models

In these models, emotions are represented as distinct categories or emotional descriptors. The most well-known model is that of psychologist Paul Ekman, who identified six basic emotions based on facial expressions: happiness, surprise, anger, disgust, fear and sadness (Ekman, 1982). However, the limited number of basic emotions, especially in the context of music, as it often conveys more complex and nuanced emotional states that cannot be fully captured by these categories. Moreover, since these emotions were identified through facial expressions, there is a need to consider emotions that are conveyed through lyrics, which may differ significantly in their depth and complexity.

Another model that can be utilized is Hevner's, which organizes emotional adjectives into eight categories or clusters, represented in a circle (Figure 1). In this circle, the adjectives within the same group share similar meanings. In contrast, the meaning of neighboring clusters gradually changes until they reach a contrast with the cluster in the directly opposite position (Hevner, 1936).

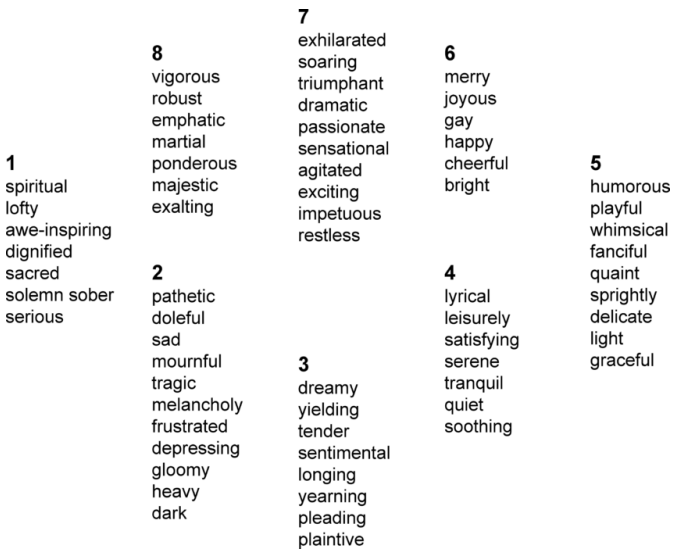

Figure 1 – Hevner's model (Malheiro, 2016).

We can also refer to another model, employed by the MIR research community, the MIREX (Music Information Retrieval Evaluation eXchange) framework, to formally evaluate systems and algorithms. In MIREX, songs are classified into five mood classes (Table 1). These categories were developed by applying clustering methods to a co-occurrence matrix of mood labels for popular music from AllMusic[2] (Malheiro, 2016).

Table 1 – Mood adjectives by cluster used in MIREX. Adapted from (Malheiro, 2016).

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

The categorical model has the advantage of representing human emotions in a simple way with easy-to-understand labels. However, it has some limitations, such as the difficulty of assigning an emotion to a specific category due to cultural, linguistic, or personality differences. Furthermore, it may not always be possible for subjects to select the category that best describes their emotional state, leading to suboptimal emotion detection. Although the categorical model is widely used due to its simplicity, it may not effectively distinguish exact emotional states, especially when emotions do not fit within the predefined categories (P.S. & Mahalakshmi, 2017).

Dimensional Models

In these types of models, a standard set of dimensions connects the various emotional states. These dimensions are typically defined in a two-dimensional (valence and arousal) or three-dimensional (valence, arousal and power) space, with each emotion represented as a point within this space (P.S. & Mahalakshmi, 2017). Among all the models, Russell's proposed Circumplex Model is the most widely recognized and commonly used (Seo & Huh, 2019). In this model, emotions are mapped on a two-dimensional plane, with each axis representing a different dimension: valence, indicating the degree of positivity or negativity of the emotion (x-axis), and arousal, which represents

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

the intensity of the emotion (y-axis) (Figure 2). The Circumplex enables the mapping of emotional states at any point along valence and arousal dimensions (Seo & Huh, 2019).

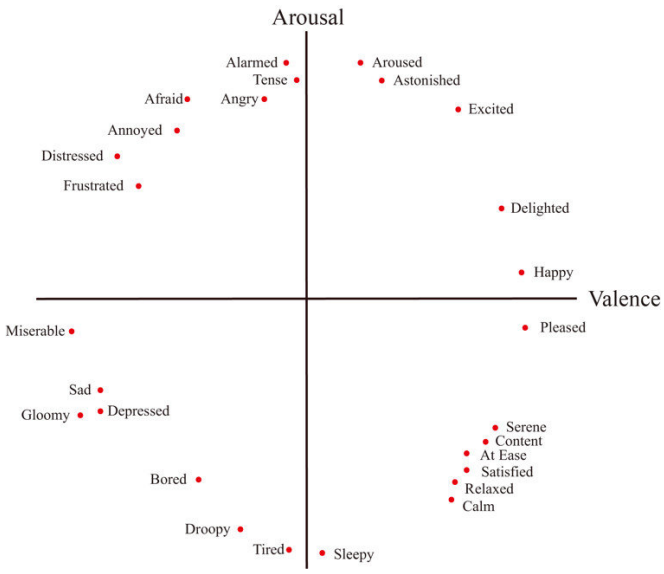

Figure 2 – Russell's circumplex model (Seo & Huh, 2019).

Another dimensional model is the one proposed by Tellegan, Watson and Clark, which uses positive/negative change, pleasantness/unpleasantness, and engagement/disengagement to categorize a broader range of emotions than previous models (Figure 3).

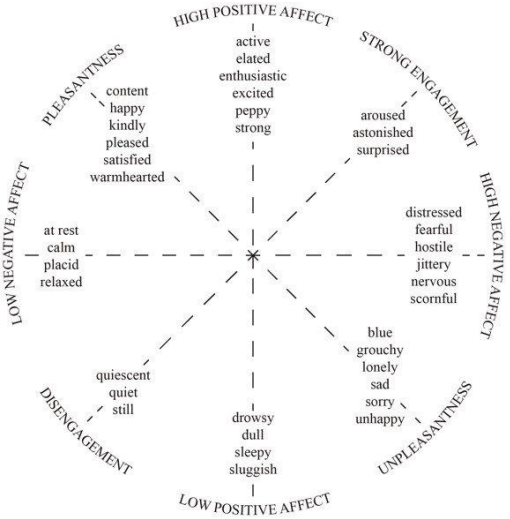

Figure 3 – Tellegen-Watson-Clark model(Seo & Huh, 2019).

The primary benefit of representing emotions within a dimensional structure is that it enables any emotion to be positioned continuously, allowing for the representation of all emotions within the limits of these dimensions.

2.1.4. Lyrics Music Emotion Recognition (LMER)

Most early MER systems concentrated on analyzing audio content, while the detection of emotion from lyrics received comparatively less attention. Later, researchers began combining audio and lyrics, resulting in bimodal MER systems with enhanced accuracy. LMER is a specialized subfield of MER, centered on extracting emotional content from song lyrics (Malheiro, 2016).

In music analysis, lyrics complement audio signals and, combined with other metadata such as album reviews, provide valuable cues for genre identification, song class, and publication time (Ara & Gopalakrishna, 2021). Lyrics provide semantic meaning to music and help reveal melodic, structural, and rhythmic characteristics within the audio signal. For these reasons, lyrics are widely used in music classification, particularly in multimodal music classification (Ara & Gopalakrishna, 2021).

Emotion detection from lyrics is especially relevant for applications such as music retrieval, recommendation, discovery, and playlist generation. Studies indicate that emotional content derived from lyrics can enhance the performance of automated music selection systems, improving user satisfaction and personalization (Ara & Gopalakrishna, 2021).

Nevertheless, LMER faces several challenges. Lyrics frequently convey emotions through figurative language, symbolism, and metaphors, which significantly increases the complexity of emotion recognition tasks. Unlike other text types, such as product reviews or tweets, lyrics typically convey multifaceted or evolving emotions that may change throughout a song. Recent studies have shown that DL-based models, particularly those utilizing NLP, can improve the accuracy of emotion classification in lyrics. Nevertheless, the effectiveness of these models heavily relies on the quality of annotated datasets and their adaptation to the lyrical domain, which is frequently poetic, informal, and highly creative.

A major challenge in lyric emotion recognition lies in the inherent subjectivity of emotion annotation. Listeners often interpret the same lyrics in different emotional ways, influenced by cultural context, personal experiences, or even the musical context within which the lyrics are presented. This variability complicates the creation of reliable datasets and the evaluation of predictive models.

2.2. Natural Language Processing (NLP)

Humans are the most exceptional species on Earth, and our most significant achievement lies in the ability to communicate and share information. According to estimates, only 21 per cent of all data is structured, while the vast majority is unstructured, generated, for instance, through tweets, WhatsApp messages, and posts in various Facebook groups. Much of this content exists as unstructured text, which poses challenges for analysis without advanced computational methods. To extract meaningful insights from such data, it is fundamental to understand how NLP works (Iqbal & Qureshi, 2022).

NLP is a branch of computer science and AI focused on the interaction between computers and humans using natural language (Iqbal & Qureshi, 2022). Its rapid development has been driven by the growing volume of textual data and the need for more natural communication between humans and computers (Patwardhan et al., 2023). NLP employs computational techniques to automatically represent, analyse, and interpret language (Iqbal & Qureshi, 2022). Its primary goal is to develop algorithms that can accurately and fluently understand, generate, and manipulate human language, allowing more intuitive and effective interaction with machines (Patwardhan et al., 2023).

In the context of automatic lyric generation with emotional content, several core tasks in NLP play a fundamental role. These tasks enable the system to interpret, structure, and generate human-like language that is coherent, expressive, and emotionally appropriate for musical composition.

2.2.1. Text Pre-Processing

Pre-processing is the first step in preparing raw text for computational analysis, ensuring that language data is structured, consistent, and ready for modelling. It involves a variety of methods, ranging from simple pattern matching to more advanced linguistic annotation, each contributing to the extraction of meaningful features. The following subsections introduce key techniques employed in NLP pipelines.

Regular Expressions

One of the most useful tasks in text processing is the use of regular expressions (often shortened to regex), which provide a language for specifying text search strings. Regex is particularly effective when searching for specific patterns within a text, whether

in a single document or across a larger corpus. A regular expression search function searches the corpus and returns all instances that match the defined pattern. Depending on the configuration, the search can be set to retrieve only the first match per line or every match found, making it a highly adaptable method for targeted text retrieval and pattern detection (Jurafsky & Martin, 2025).

In the context of lyrics, regular expressions can be used to identify recurring rhymes, repeated phrases, stanza boundaries, or specific linguistic structures such as metaphorical markers.

Tokenization

Before any NLP operation can be performed on a text, the text must be normalized, a task known as normalization. A fundamental component of this process is tokenization (segmentation) of text, which involves segmenting text into its minimal linguistic units, known as tokens. Tokens can represent words, subwords, or punctuation marks (Jurafsky & Martin, 2025).

Tokenization is a crucial pre-processing step in most NLP pipelines, as it prepares the input for further syntactic and semantic analysis. Generally, tokenization algorithms can be divided into two categories (Jurafsky & Martin, 2025):

- Top-down tokenization, in which a standard is defined, and explicit rules are applied to segment the text accordingly. For instance, the sentence "Natural language processing is complex." can be segmented into tokens ["Natural", "language", "processing", "is", "complex", "."] by applying rules based on whitespace and punctuation.
- Bottom-up tokenization, commonly used in modern NLP, which splits text into subword units based on frequency rather than meaning. These subwords can be entire words, word fragments, or even single characters. This approach, however, can be sensitive to rare or unusual words and small textual variations. Two widely adopted algorithms are Byte-Pair Encoding (BPE) and SentencePiece, with the latter implementing both BPE and Unigram Language Modelling methods (Tay et al., 2022). For example, the word "unbelievable" could be split into ["un", "believ", "able"].

In LMER, subword tokenisation plays a crucial role when using transformer-based models.

Part-of-Speech (POS) Tagging

POS tagging is the process of assigning each word in a text its appropriate grammatical category. In the English language, the most common grammatical classes, based on Penn Treebank, include, for example, nouns (NN), adjectives (JJ), determiners (DT), adverbs (RB) and verbs (VB) (Malheiro, 2016). Accurate POS tagging provides syntactic structure, which supports parsing, semantic role labelling, and downstream tasks such as sentiment or emotion analysis. For example, in the sentence "The orange cat carefully followed the playful kitten around the garden", the determiner "The" would be tagged as DT, the adjectives "orange" and "playful" as JJ, the nouns "cat", "kitten" and "garden" as NN, the adverb "carefully" as RB, the verb "followed" as VB, and the preposition "around" as IN (Jurafsky & Martin, 2025).

Named Entity Recognition (NER)

NER involves identifying and classifying spans of text that correspond to named entities, such as person (PER), location (LOC), organization (ORG), or geopolitical entities (GPE). However, the scope of NER may also include other referential expressions, such as dates, times, numerical quantities, and monetary values (Jurafsky & Martin, 2025). For instance, given the sentence "Apple Inc. was founded by Steve Jobs in Cupertino in 1976", a NER system would classify the entities as follows: "Apple Inc." as an organization; "Steve Jobs" as a person; "Cupertino" as a location; and "1976" as a date (Jurafsky & Martin, 2025).

2.2.2. Syntactic and Semantic Analysis

Syntactic and Semantic analysis are two fundamental layers in comprehending and generating natural language.

Syntactic analysis is a fundamental aspect of computational linguistics, concerned with understanding and modelling human language through computational methods. Examining the structure of sentences, how words are combined and relate to one another, is crucial for numerous applications, such as machine translation, information retrieval, and text mining (Muminovich & Istam kizi, 2025). For lyric generation, syntactic analysis helps ensure grammatical correctness and sentence flow, which are critical for musicality and readability.

Semantic analysis, in contrast, is focused on evaluating and representing the meaning of language (Salloum et al., 2020). Modern NLP relies on techniques such as word embeddings, where words are encoded as dense vectors in continuous space, like Word2Vec and GloVe, and contextual embeddings, such as ELMo and BERT (Camacho-Collados & Pilehvar, 2020). For lyric generation, semantic coherence ensures that words selected to convey a particular emotion are also contextually appropriate, even when figurative language or ambiguity is involved.

Accurate syntactic and semantic interpretation is essential in generating emotionally expressive and coherent lyrics that maintain both grammatical integrity and contextual relevance.

2.2.3. Sentiment Analysis and Emotion Detection

Sentiment Analysis is an NLP technique used to evaluate and identify the emotional tone or mood expressed in textual data. It typically involves identifying words and phrases to categorize their valence as positive, negative, or neutral. However, more advanced approaches can detect specific emotions, intentions, or subtle aspects of sentiment (e.g., joy, sarcasm, anger), as well as context-specific sentiments – all of which are particularly relevant when analyzing musical lyrics (Jim et al., 2024).

Many of the features employed in LMER have been used since the beginning in the sentiment analysis field. However, LMER also incorporates lyric-specific features that are absent from most state-of-the-art text-based emotion detection systems. Examples include aspects of the composer's writing style, such as the use of slang or distinctive phrasing, as well as structural features unique to lyrics, like the frequency of chorus repetitions or the recurrence of the song title within the text (Malheiro, 2016). The importance of sentiment analysis stems from its capacity to extract meaningful insights from large volumes of textual data.

2.2.4. Natural Language Generation (NLG)

NLG is a branch of AI and computer linguistics, with its primary aim being the creation of computer programs that can generate text comprehensible to humans (Oliveira et al., 2012). At its core, NLG is the process of representing semantic information from the

input data, which may take the forms of tables, images, or formal languages, into coherent natural language, facilitating effective communication and information understanding.

Traditionally, NLG research has focused on grammatical correctness and semantic coherence. However, creative language generation, especially in the context of music lyrics, requires a more complex interplay of metaphor, emotion, cultural references, and musical rhythm and rhyme (Olatunji, 2024).

One of the central challenges in NLP-generated lyrics is maintaining emotional depth and narrative coherence. While sentiment analysis and affective computing enable models to recognize and mimic emotions, the nuanced human expressions of pain, joy, love, or sorrow in music often elude algorithmic reproduction. To address this, researchers have developed models that incorporate sentiment-aware training data or use reinforcement learning to optimize emotional alignment. Emotional metrics are embedded in the reward function during training, guiding the model toward outputs with stronger emotional tone. Nevertheless, generating lyrics that are simultaneously authentic in emotional expression and musically meaningful continues to be a major open problem (Olatunji, 2024).

Lyrics also differ structurally from other literary forms. They must conform to rhythm and rhyme schemes, and often include repeated elements, like choruses. Transformer models can be fine-tuned on datasets of song lyrics to learn such structural patterns. However, maintaining coherence over multiple verses remains a technical challenge (Olatunji, 2024).

Traditional rule-based systems and early statistical methods struggled to generalize well compared to unseen data and complex linguistic phenomena. As a result, NLP has gradually shifted towards data-driven approaches, particularly those involving ML and DL.

Several strategies have been introduced to enhance coherence and structure in lyric generation. Controlled text generation and prompt engineering can guide models towards predefined lyrical arcs (e.g., Verse 1 – Chorus – Verse 2 – Bridge). Similarly, attention masks and positional embeddings in transformer models enable better handling of repetition, stanza boundaries, and structural transitions. These techniques improve fluency and structure, but balancing creativity, coherence, and emotional depth remains an area of active research (Olatunji, 2024).

2.3. Deep Learning (DL)

Over the last decade, the NLP field has been transformed by advances in DL techniques. These approaches rely on an architecture composed of multiple layers of Artificial Neural Networks (ANN or NN), which are composed of interconnected processing units, known as neurons, linked together through numerous connections. Each unit receives a set of inputs, transforms them, and transmits the results to other neurons.

Each neuron is linked to others via weighted connections, which determine ow much influence an input has on the neuron's output. These weights, often described as the trainable parameters of a layer, are complemented by an additional parameter known as the bias (b), which shifts the activation threshold of the neuron (Figure 4). As the model is exposed to the training data, these parameters are adjusted, allowing the network to learn an effective mapping from the input features to the desired outputs. Learning can be described as the optimization of these parameters across all layers so that the network can produce outputs that closely match the expected targets (Chollet, 2021).

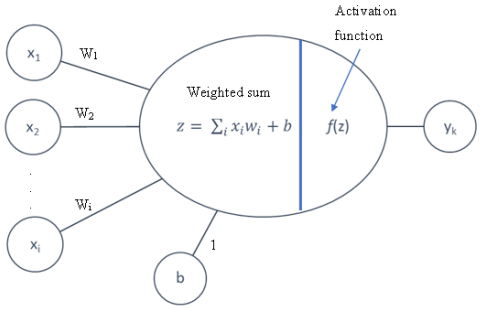

Figure 4 – Artificial Neuron structure. Adapted from (Nielsen, 2019).

Figure 5 illustrates the general structure of a deep NN, showing that such models are typically organized into three principal sections: an input layer, a series of hidden layers, and an output layer. The input layer receives the initial data, whereas each hidden layer progressively transforms and integrates the outputs of the previous layer, enabling increasingly abstract representations. The output layer then consolidates these computations to generate the final prediction of the network. This hierarchical structure allows deep networks to perform complex decision-making tasks efficiently (Chollet, 2021).

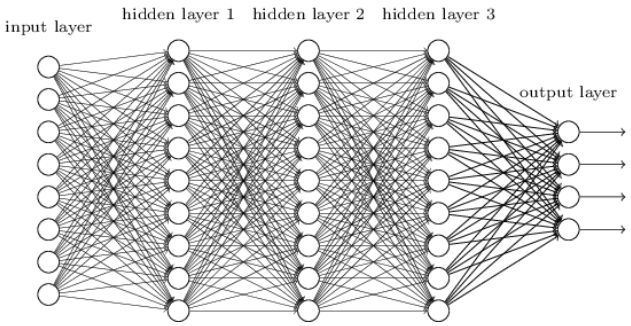

Figure 5 – Deep Neural Network (Nielsen, 2019).

In these networks, the output of each becomes the input for the next, with information flowing in a single direction; therefore, they are called feedforward NNs (Nielsen, 2019).

NNs are often described as fully connected or dense NNs because every neuron in one layer is connected to every neuron in the following layer. Although a dense layer on its own can only represent linear relationships, the introduction of activation functions adds non-linearity, enabling the network to capture more complex patterns in the data (Figure 6).

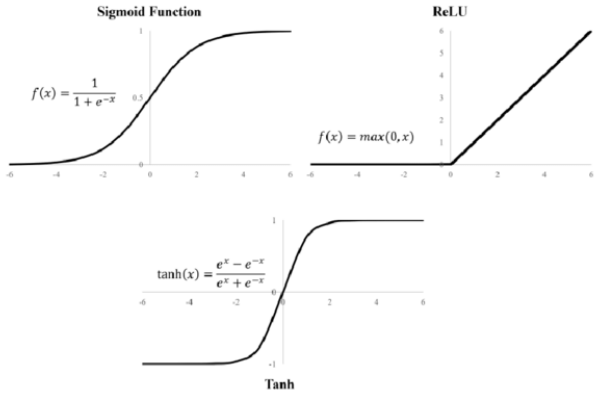

Figure 6 – Most common activation functions (Purnawansyah et al., 2021).

To control the output of an NN, it is necessary to measure the difference between the predicted output and the expected result. This task is handled by the loss function, which calculates a numerical score representing the performance of the network on a given example. This score is then used as feedback to adjust the network's, guiding them in a direction that minimises the loss for the current example. The optimizer, which applies the backpropagation algorithm, propagates the error signal from the output layer back through the network and estimates the contribution of each weight to the overall loss. Gradient descent is then applied to determine the direction and magnitude of the weight updates, gradually reducing the loss (Chollet, 2021).

Initially, the network's weights are randomly assigned; therefore, the output differs significantly from the targets, and the loss is high. As the network processes more examples, the weights are adjusted to gradually minimize the loss. Repeating this process over many iterations leads to a network whose weights minimize the loss function and whose outputs closely match the expected targets, reflecting a trained network (Chollet, 2021).

2.3.1. Recurrent Neural Networks (RNNs)

Traditional NNs process each input independently, without retaining information from prior inputs. In contrast, humans process language by integrating each word into a mental representation of the preceding context and continuously updating their internal understanding as new words are encountered.

Recurrent Neural Networks (RNNs) follow this behavior in a simplified manner. They process sequences by handling each element in turn and maintaining a state that retains information about earlier elements. An RNN is a type of NN that has an internal loop, which means that it depends on what happened before and on what is entering now (Figure 7) (Chollet, 2021).

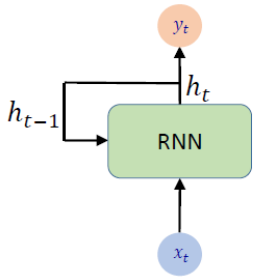

Figure 7 – NN with a loop. Adapted from (Goodfellow et al., 2016).

In RNNs, the internal state is reset between independent sequences, so each sequence is treated as a separate input. Rather than processing the entire sequence at once, the network iterates over its elements, and during training, the outputs and gradients are computed for all elements before being aggregated to update the weights via backpropagation (Chollet, 2021).

Computing gradients in RNNs requires multiplying the weights by themselves several times, which can create challenges in capturing the long-term dependencies. This often leads to problems such as exploding gradients, where algorithms assign high values

to weights, preventing learning, or vanishing gradients, where gradient values become too small, causing the model to cease learning (Chollet, 2021).

2.3.2. Long Short-Term Memory (LSTM)

In simple RNNs, information from earlier inputs can be reduced as more inputs are processed, a phenomenon known as the vanishing gradient problem. Consequently, inputs that occurred far back in the sequence may have little impact on the network output, even if they are important. LSTM networks, a special type of RNN, are designed to address this limitation. They can retain long-term dependencies and relationships in sequential data by preserving relevant information over time, effectively preventing older inputs from being lost during processing (Chollet, 2021).

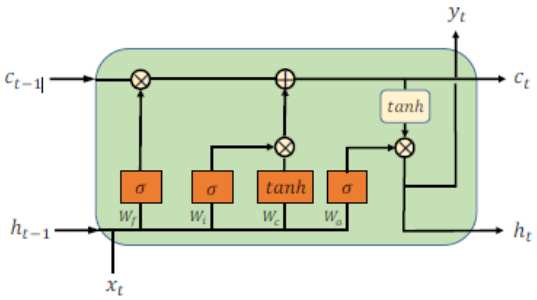

Figure 8 – Structure of LSTM unit. Adapted from (Goodfellow et al., 2016).

LSTM cells contain computational blocks that control the information flow (Figure 8). These units employ structures called gates, specifically, the input, forget, and output gates, to manage the addition or removal of information from the cell state. Each gate plays a distinct role in determining which information must be remembered. The forget gate manages what information in the cell state should be forgotten considering the new information that enters the network. The input gate controls which part of the latest information is stored in the cell state and retains that information. It facilitates the update of the cell state by identifying and preserving attributes deemed relevant for future use. The output gate transmits the hidden state; here, the updated cell state is handed through a tanh function and multiplied by the sigmoid output to calculate the hidden state. Generally, the sigmoid function serves as the activation function for the gates. Its output, ranging between 0 and 1, can be interpreted as a measure of how much information should be allowed through, ranging from none (values near 0) to all (values near 1). One sequence element is presented to the network at each time step (Chollet, 2021).

2.3.3. Transformers

With advancements in NLP, models based on RNNs and LSTM networks have begun to show limitations in handling long-range dependencies and achieving efficient parallel computation. The transformer architecture was introduced to overcome these challenges and has since become the basis of state-of-the-art language models for most NLP tasks (Vaswani et al., 2023). The core innovation of Transformers is the attention mechanism, which enables the model to weigh the relevance of different elements within the input sequence when generating an output. Transformers, unlike RNNs, can process entire sequences in parallel rather than step by step, which greatly improves both the training speed and the capacity to model long-range contextual dependencies. The transformer architecture depends on multi-head attention layers and fully connected feed-forward layers, which are typically organized into encoder and decoder blocks (Figure 9).

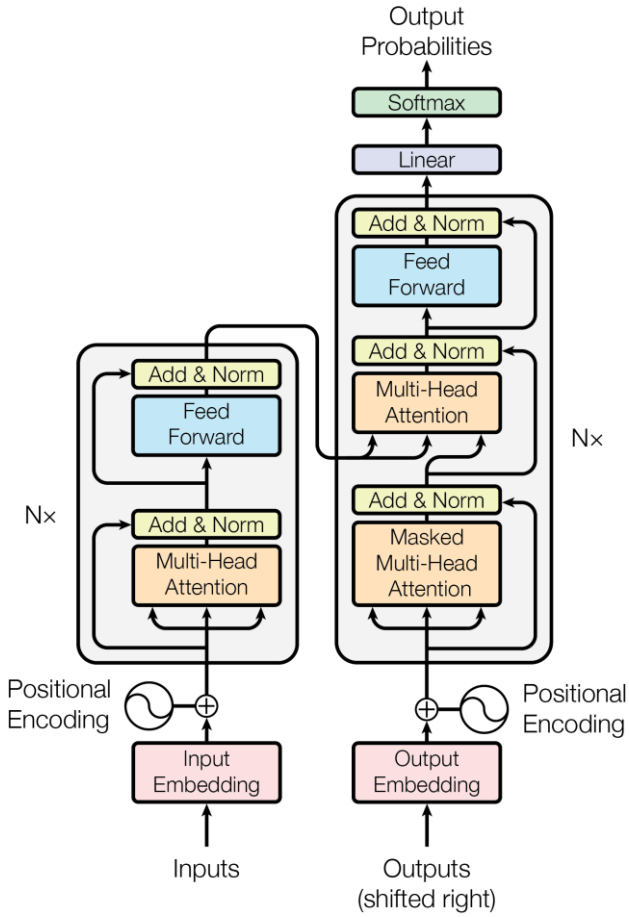

Figure 9 – The architecture of the transformer model. Adapted from (Vaswani et al., 2023).

In the original design by *Vaswani et al. (2017)*, the model consists of a stack of six encoder and six decoder layers (Figure 9). Each layer contains sublayers, with the output of

layer l serving as the input of layer l+1 until the final prediction is reached (Vaswani et al., 2023). The encoder layers contain a multi-head self-attention mechanism and position-wise fully connected feed-forward network sublayers, each with residual connections and layer normalization. The decoder layer introduces a third sublayer that performs multi-head attention over the encoder output and applies masking to ensure autoregressive generation. Inputs and outputs are represented by learned embeddings, with positional encodings based on sinusoidal functions added to incorporate sequence order information (Vaswani et al., 2023).

The attention function maps a query and a set of key–value pairs to an output, with all elements represented as vectors. The output is then calculated as a weighted sum of the values, where the weight given to each value is determined by a compatibility function that evaluates the query against the corresponding key. Scaled Dot-Product Attention computes these weights by taking the dot product of queries and keys, scaling the result to improve numerical stability, and then applying the weights to the values. Multi-head attention further enhances this process by projecting the queries, keys, and values into lower-dimensional subspaces, enabling the model to capture information from multiple representation subspaces simultaneously (Vaswani et al., 2023).

Large Language Models (LLMs)

NLP has evolved considerably, largely due to the advent of Large Language Models (LLMs) (Chen et al., 2025). The models are deep NNs based on transformer architectures trained on massive datasets of unlabelled text, often comprising hundreds of billions of words. These models are designed to learn linguistic patterns, world knowledge, context sensitivity at scale and typically contain hundreds of millions to billions of trainable parameters (Raffel et al., 2023).

At the core of their designs lies a self-supervised learning objective that involves predicting succeeding words in incomplete sentences. Through this process, LLMs develop the capacity to generate coherent and contextually appropriate responses by learning statistical associations between words and phrases (Chen, Zhang, Langrené, & Zhu, 2025).

LLMs function by encoding input text into a high-dimensional vector space that preserves semantic relationships, which is then decoded to generate meaningful output. The quality of these outputs is influenced by numerous factors, such as

prompt formulation, configuration of the model's hyperparameters, and range and variety of the training data. (Chen, Zhang, Langrené, & Zhu, 2025)

Generative Pre-Trained Transformer (GPT)

GPT gained widespread recognition following the launch of ChatGPT by OpenAI, demonstrating the practical utility of large-scale generative AI in natural language processing. As a DL model, GPT is pre-trained on extensive text corpora and can be fine-tuned for a diversity of given tasks, such as sentiment analysis, text classification, machine translation, language generation and language modelling (Yenduri et al., 2024).

The architecture of GPT is based on the transformer framework, representing a major advancement over prior approaches, such as RNNs and LSTMs. It employs a self-attention mechanism, allowing the model to consider the context of the whole sentence when generating the next word, thereby enhancing its capacity to understand and produce language. The decoder component of the transformer architecture is particularly central in GPT, as it generates the output text based on the encoded representations of the input (Yenduri et al., 2024).

GPT has demonstrated remarkable versatility across a variety of tasks. One of its mains strengths lies in Natural Language Understanding (NLU), where it can analyze and interpret the meaning of text, including recognizing entities and relationships in sentences. It also excels in NLG, enabling it to produce text output, such as making creative content or providing thorough and informative answers to questions (Yenduri et al., 2024).

Since its introduction, GPT has evolved through multiple versions, each with its own features and capabilities (Table 2). Early models, such as GPT-1, demonstrated proof of concept, while GPT-2 raised awareness of the potential and risks of generative AI. GPT-3 was scaled to 175 billion parameters, enabling highly generalized language modelling abilities. Subsequent iterations, including GPT-3.5 and GPT-4, have introduced refinements in reasoning, factual accuracy, and alignment with user intent. GPT is a generative model, classified as Generative AI (GAI), which is

designed to generate new content, such as images, music, text or other types of data (Yenduri et al., 2024).

GPT-Neo, developed by EleutherAI, implements the GPT with the architecture originally proposed by OpenAI. Subsequently, EleutherAI developed GPT-J (6B), a more advanced model with six billion parameters. Both models are derived from GPT-3 and have demonstrated strong performance in several language tasks, including text generation, translation and question answering. GPT-Neo, being smaller, is computationally lighter and suitable for experimentation, while GPT-J offers improved contextual understanding and generation quality because of its larger scale (Lin et al., 2022).

Table 2 – Comparison of different versions of the GPT model.

Model	Year	Availability	Tokens	Parameters	Features	Limitations
GPT-1	2018	Open	~512	117M	First generative transformer for text completion	Limited capacity, data, applications and cannot perform complex tasks
GPT-2	2019	Open	~1024	1.5B	Text generation capabilities are enhanced, but there is also an increased risk of misuse	Limited control and data diversity, expensive computational requirements and risk of improper information, like incoherent verses/choruses and frequent repetition
GPT-3	2020	API-only	~2049-4096	175B	Good NLP capabilities, language translation, and summarisation and the generation of text	Limited control and data diversity, lack of explanations and ethical concerns
GPT-3.5	2022	API-only	~4096	Similar or larger than GPT-3	Enhances user experience by providing more accurate and contextually appropriate information.	Limited resources to train and contextual understanding, data bias, lack of explainability and high inference latency
GPT-4	2023	API-only	8192-32768	100T	Creative and technical writing tasks	Computationally expensive
GPT-J	2021	Open	~2048	6B	Performance similar to GPT-3	May struggle with complex tasks due to fewer parameters
GPT-Neo	2021	Open	~2048	125M, 1.3B e 2.7B	Performance similar to GPT-3	May struggle with complex tasks due to fewer parameters

Bidirectional Encoder Representations from Text (BERT)

Compared with newer models, BERT pre-trains deep bidirectional representations from unlabelled text by jointly using left and right context throughout all layers. As a result, the pre-trained model can be adapted to multiple tasks, such as natural language inference and question answering, by adding only a single output layer, eliminating the need for extensive task-specific modifications (Devlin et al., 2019).

The BERT model only has encoder layers and no decoder stacks. The BERT model employs Masked Language Modelling (MLM), in which some input tokens are randomly hidden ("masked"), and the attention layers must learn to comprehend the context. The model predicts the hidden tokens (Rothman, 2024).

Alongside the masked language model, a "next sentence prediction" task can be employed to jointly pre-train text-pair representations (Devlin et al., 2019).

Robustly optimised BERT approach (RoBERTa)

RoBERTa presents a replication and extension study of BERT pretraining, that measures the effects of various key hyperparameters and training data size. The study revealed that BERT was considerably undertrained, and that with optimized settings, it could equal or exceed the performance of subsequent models (Liu et al., 2019).

RoBERTa primarily follows the original BERT optimization hyperparameters but adjusts the learning rate peak and number of warm-up steps, tuning them separately for each configuration (Liu et al., 2019).

Text-to-Text Transfer Transformer (T5)

The T5 model was developed to facilitate multitask learning within the NLP domain. Its architecture comprises two main levels: pre-training, establishing a shared knowledge base applicable to a wide range of sequence-to-sequence tasks, and fine-tuning, which adapts the model for specific downstream tasks (Mastropaolo et al., 2021).

The T5 model is built upon the transformer architecture. When an input sequence is provided, it is converted into a sequence of embeddings and passed to the encoder. Each encoder block is composed of two sublayers: a multi-head self-attention mechanism followed by a feed-forward network. Each subcomponent first receives layer normalization, and a residual skip connection then combines its input with its resulting output. The model incorporates dropout at multiple stages: inside the feed-forward layers, along the skip connections, on the attention weights, and at the stack's input and output (Mastropaolo et al., 2021).

Decoders function much like encoders, with each self-attention layer succeeded by another attention mechanism that focuses on the output produced by the encoder. The final decoder block's output is passed through a dense layer with a softmax activation, generating probability distributions over the vocabulary. Unlike the original transformer model, T5 uses a simplified positional embedding scheme in which each position is represented by a single scalar added to the corresponding logit used in computing attention weights. For efficiency, the authors also shared the positional embedding parameters across all layers (Mastropaolo et al., 2021).

Table 3 provides a comparative overview of the NLP and recurrent models discussed, with key aspects such as architecture, training, tasks, strengths, and limitations summarized to offer a clear and concise view of the differences and capabilities of each model.

Table 3 – Overview of the NLP and sequential models presented and compared in this study.

Model	Architecture	Training	Tasks	Features / Strengths	Limitations
LSTM	LSTM cells	Trained in sequential data; can learn long-term dependencies	Time series prediction, sequence modelling, speech recognition, text generation	Capable of learning long-term dependencies, mitigates the vanishing gradient problem, and retains relevant sequential information over time	Slower to train than transformers on large datasets, limited parallelization, performance may decline on very long sequences
GPT (GPT-1 to GPT-4, GPT-J, GPT-Neo)	Transformer (decoder)	Pre-trained on large corpora; can be fine-tuned	Text generation, translation, sentiment analysis, classification, and language modelling	Self-attention, contextual understanding, advanced NLU and NLG, scalability to large models, versatile	High computational cost, data bias, limited explainability, struggles with complex tasks in smaller models
BERT	Transformer (encoder)	Pre-trained with Masked Language Modelling (MLM) and Next Sentence Prediction (NSP)	Question answering, natural language inference, text classification	Deep bidirectional contextual understanding, easy to fine-tune for multiple tasks	Not generative, unsuitable for text creation, computationally intensive to train
RoBERTa	Transformer (encoder)	Optimized pre-training of BERT (more data, tuned hyperparameters)	Similar to BERT, with improved performance	Improved BERT with longer training, better hyperparameters, and superior benchmark performance	Still not generative, high computational cost
T5	Transformer (encoder+decoder)	Pre-trained seq2seq multitask; fine-tuned for specific downstream tasks	Translation, summarization, classification, text generation	Text-to-text model, multitask support, flexible for any NLP task formulated as input → output	More complex than encoder-only models, high computational cost

2.3.4. Model Optimisation

Although LLMs are powerful, they are often too general for specialized tasks. To enhance their performance, strategies such as fine-tuning and prompt engineering can be employed. This section presents the theoretical foundations of both approaches.

Fine-Tuning

Fine-tuning consists of adapting a pre-trained model to a specific task or domain by continuing its training on a domain-specific dataset that differs from the data used during the model's initial training. This approach takes advantage of the model's existing knowledge, allowing it to transfer learned patterns and features to new contexts. As a result, fine-tuning can improve task performance while reducing the need for extensive data and computational resources compared to training a model from the ground up (Parthasarathy et al., 2024).

There are some commonly adopted approaches to fine-tuning, each with its own advantages and trade-offs depending on the application domain (Parthasarathy et al., 2024):

- Unsupervised Fine-Tuning - This approach does not require labelled data. Instead, the model is exposed to a large volume of unlabelled text from the target domain, allowing it to refine its understanding of domain-specific vocabulary, idioms and stylistic patterns. While particularly useful in specialized fields such as law or medicine, it is less precise for tasks requiring explicit structure or categorisation.
- Supervised Fine-Tuning (SFT) - In SFT, the model is trained on labelled datasets that are specifically designed for the target task. For instance, fine-tuning an LLM for emotion-based lyric generation might require datasets annotated with emotions. This method typically yields high performance but comes at the cost of collecting and curating large, high-quality labelled datasets.

In the context of creative tasks, such as lyric generation, fine-tuning provides a powerful mechanism for adapting general-purpose LLMs to stylistic, emotional, and cultural requirements. This approach ensures that the generated content aligns more closely with artistic conventions, emotional tones, and audience expectations, while maintaining coherence and originality.

Prompt-Engineering

The evolution of LLMs highlights major advances in AI research, marked by greater model complexity, improved training methods, and the broadening of their potential applications (Chen et al., 2025). Users interact with LLMs using prompts, which are strategically designed task-specific or natural language instructions (Gao, 2023). However, their effectiveness is highly dependent on the quality of the prompts that guide them (Marvin et al., 2024). This has given rise to prompt engineering, which is vital for maximizing the utility and accuracy of LLMs and ensuring they can meet varied and evolving user requirements (Chen et al., 2025).

Prompt engineering is the systematic process of designing and optimizing input prompts to guide LLMs responses, ensuring that the generated output is relevant, accurate and coherent (Figure 10). This process is crucial to unlock the model's potential, enhancing its usability and extending its applicability through an extensive range of domains (Chen et al., 2025).

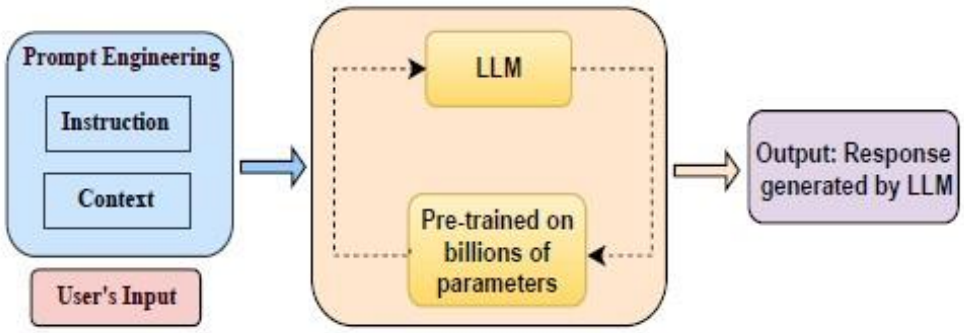

Figure 10 – Prompt engineering components. Adapted from (Sahoo et al., 2025).

By incorporating certain elements, it is possible to design effective prompts that enable LLMs to generate high-quality responses (Chen et al., 2025):

- Give Instructions – When given only a simple instruction, the model has a wide range of possible outputs, often resulting in overly broad responses. A more detailed and comprehensive description helps stimulate precise and contextually relevant outputs.
- Ensure clarity and precision – Prompts that are clear and precise reduce uncertainty and guide the model towards producing outputs that align closely with the intended requirements. Detailed and precise prompts are particularly effective in ensuring accuracy and relevance, thereby reducing the model's uncertainty.

- Role Prompting - Assigning the model a specific role to adopt, such as an expert in a relevant field, can improve the quality of responses (Gao, 2023). This method helps the model tailor its output to better align with the desired output.
- Try several times - Owing to the non-deterministic nature of LLMs, running the same prompt several times and selecting the best output can be advantageous. This strategy helps mitigate variability and increases the chances of obtaining high-quality responses.
- One-shot or Few-shot Prompting - In one-shot prompting, the model is provided with a single example to learn from, whereas few-shot prompting involves providing several examples. It is crucial to ensure that the examples are diverse and balanced (Gao, 2023). The choice depends on task complexity and model capability: simple tasks or highly capable models may require only one example, while more complex tasks often benefit from a few-shot approach.
- Chain of Thought - This method guides LLMs to break down complicated tasks into multiple intermediate steps (Gao, 2023). This approach involves supplying the model with prompts that guide its reasoning process, such as “Let’s think step by step” or by giving examples that include detailed reasoning leading to the answer. It also provides a clearer structure for the model’s reasoning process, thereby improving the interpretability of its outputs for users.

Thus, prompt engineering is increasingly recognized not simply as a technical trick but as a systematic methodology for steering generative models towards desired outcomes, particularly in creative and high-stakes applications such as lyric generation (Marvin et al., 2024).

2.3.5. Evaluation

To evaluate the model's capability to generate authentic and semantically meaningful lyrics, the output can be assessed through a combination of objective and subjective criteria (Ram et al., 2021). This dual approach provides a thorough evaluation of both the technical performance of the models and the artistic value of the generated lyrics.

Readability

The definition of a well-written or readable text depends on the intended audience. Readability measures how easily a text can be read and comprehended in terms of its linguistic characteristics (Crossley et al., 2023). In practice, most research on readability focuses on text comprehension rather than reading speed. Comprehension is influenced by lexical sophistication, syntactic complexity, and discourse structure, as well as the reader's prior knowledge and proficiency. For this study, we focus exclusively on text-intrinsic features, ignoring reader-specific variables (Crossley et al., 2023).

Among the most widely used measures of text readability are Flesch Reading Ease (FRE) and Flesch-Kincaid Grade Level (FKLG) formulas. These traditional formulas use proxy indicators for lexical and syntactic features: the number of words per sentence estimates syntactic complexity, while the number of characters per word estimates lexical sophistication. Although these indices do not account for semantic cohesion, narrative structure, or discourse-level features, they remain practical and interpretable methods for evaluating text clarity (Crossley et al., 2023).

FRE uses two text metrics – the average number of words per sentence and the average number of syllables per word – to produce a score between 0 and 100, with higher scores indicating easier comprehension. FKGL uses the same two metrics as FRE and provides an intuitive measure of the reading proficiency required to understand the content (Crossley et al., 2023).

Perplexity

Perplexity is a key metric in NLP for evaluating LLMs. It measures how effectively a model predicts unseen text, based on the probability it assigns to a given sequence (Jurafsky & Martin, 2025). Formally, the perplexity of a word sequence is calculated as the geometric mean of the inverse probabilities of the words (Roh et al., 2020).

Lower perplexity indicates that the model assigns higher probability to the observed sequence, reflecting improved fluency and coherence. Conversely, a model with high perplexity struggles to predict the next word, which often results in text that is less coherent or natural (Jurafsky & Martin, 2025).

A limitation of this measure is its sensitivity to tokenisation. Models employing different tokenisation strategies may yield perplexity scores that are not directly comparable. As a result, perplexity is most reliable when applied to models that share a consistent tokenisation scheme (Jurafsky & Martin, 2025).

Lexicons

An alternative method of evaluation involves sentiment and affective lexicons. Instead of treating every word as an independent feature, this approach targets words that strongly signal affect or sentiment. These lexicons are based on the idea that words possess affective meanings or connotations – defined here as elements of a word's meaning linked to a writer's or reader's emotions, opinions, or evaluations (Jurafsky & Martin, 2025).

In their simplest form, sentiment lexicons categorize words as positive or negative. More nuanced lexicons assign continuous values across affective dimensions (Jurafsky & Martin, 2025). The NRC Valence, Arousal, and Dominance (VAD), for example, assign valence, arousal, and dominance scores to more than 55,000 English words. Decoders function much like encoders, with each self-attention layer succeeded by another attention mechanism that focuses on the output produced by the encoder (Mohammad, 2025).

To construct the NRC VAD Lexicon (Mohammad, 2018a), words and emoticons were selected from prior lexicons and obtained annotations via crowdsourcing using the best–worst scaling method. In this method, annotators are presented with N items (generally 4) and instructed to identify the item that is the best (highest) and the one that is the worst (lowest) with respect to a particular property. Decoders function much like encoders, with each self-attention layer succeeded by another attention mechanism that focuses on the output produced by the encoder (Jurafsky & Martin, 2025).

3. State of the Art Review

The review and analysis of existing literature and research studies related to the project topic are essential to the development of this work. For this purpose, a Systematic Literature Review (SLR) is conducted, a research methodology designed to identify, evaluate, and synthesise relevant literature. Its primary aim is to provide a critical evaluation of current knowledge and understanding in the field, identify gaps, and highlight areas for further research.

3.1. Methodology

This SLR was conducted in accordance with the guidelines for performing SLRs as proposed by Kitchenham and Charters (Amara et al., 2016). Before conducting the review, a structured protocol was developed to enhance the rigour and reproducibility of the study. This protocol serves as a comprehensive framework for conducting the review (Figure 11 - adapted from (Amara et al., 2016)), outlining key aspects such as research questions, search strategies for identifying relevant studies, criteria for inclusion and exclusion, and the approach for data extraction and synthesis.

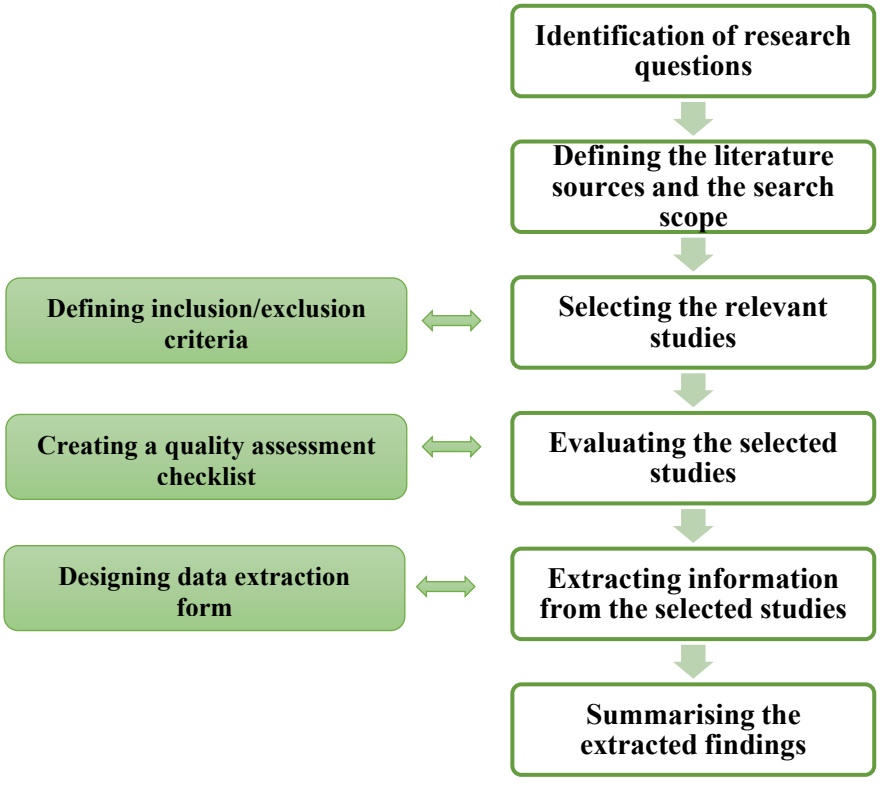

Figure 11 – SLR process.

3.2. Research Questions

To understand the role of DL models in generating emotionally expressive song lyrics, it is important to address key research questions, as they help explore their effectiveness, challenges and potential. The following research questions in Table 4 guide this study:

Table 4 – Research questions.

No.	Research Question (RQ)
RQ1	Which DL algorithms are the most frequently used for generating song lyrics in the literature?
RQ2	How do different DL models compare in their ability to generate emotionally aware song lyrics?
RQ3	How can DL models be trained and tuned to generate song lyrics that effectively express specific emotions?
RQ4	How can we measure and evaluate the emotional accuracy of song lyrics generated by DL models?
RQ5	What are the best metrics and methods for assessing the coherence and stylistic consistency of automatically generated song lyrics?
RQ6	In what ways can the output of DL models be refined to enhance the originality and uniqueness of the generated song lyrics, while maintaining emotional expressiveness?

3.3. Search Strategy and Sources

For a systematic review, a well-planned search strategy is essential to identify all relevant studies while preventing them from being overshadowed by less pertinent ones. To achieve this, online search engines, such as Google Scholar, were used alongside platforms that provide access to scientific content, namely ResearchGate and B-on. Additionally, major online databases, including IEEE Xplore, ScienceDirect, arXiv, Taylor & Francis and SpringerLink were consulted.

Once the literature sources had been identified, a search string was constructed, consisting of four main key terms: Emotions, Music, Generation and AI. These terms were

complemented with alternatives and, in the case of AI, also with specific alternatives that have been widely used in paper titles. The search strings were then constructed by joining alternative spellings and synonyms, and the main search terms (Table 5).

Table 5 – Search strings.

No.	Search string (SS)
SS1	Emotional creativity in music
SS2	Emotion-aware text generation
SS3	Song lyrics generation
SS4	Artificial Intelligence for song lyrics generation
SS5	Computational creativity in AI-generated lyrics
SS6	Lyrics generation with Deep Learning
SS7	Emotion-aware transformer-based lyrics generation
SS8	Large Language models for music generation
SS9	Neural networks for emotion-driven lyrics generation
SS10	Prompt engineering for song lyrics generation

3.4. Inclusion and Exclusion Criteria

To ensure that only the most relevant, scientifically robust, and applicable studies are included in the report, it is essential to establish clear inclusion and exclusion criteria (Table 6).

Table 6 – The including and excluding criteria for selecting relevant studies.

Inclusion Criteria	Exclusion Criteria
Original research articles, review papers, theses, academic books, and peer-reviewed journals	Non-academic sources (e.g., blogs, Wikipedia, Reddit), opinion pieces without empirical or theoretical basis, and unpublished or non-peer-reviewed work
Research articles written in English	Non-English articles; duplicate records

Studies published primarily between 2015 and 2023	Studies published outside the defined period
Studies that align with the defined search criteria and research objectives	Studies do not match the search criteria
Only one copy of each study is included when stored in multiple sources. If a study has multiple versions published, only the most recent is kept.	Redundant copies or outdated versions of the same study

3.5. Quality Assessment

At this stage, a quality assessment was conducted on the selected studies to evaluate and determine their quality. Table 7 presents the set of questions used to assess each of the selected studies in terms of both the quality of the used method and the quality of the reporting.

Table 7 – Quality Assessment Checklist

No.	Question
QA1	Is the study relevant to the research?
QA2	Is there a clear statement of the research aim?
QA3	Are the findings clearly articulated and presented?
QA4	Is the experimental or research procedure explained?
QA5	Are the data sources or datasets clearly described?
QA6	Has the paper been cited by other researchers?

3.6. Data Extraction

The data that has been summarised previously is used to address the research questions. There is certain information that can be easily extracted, such as the title,

author's name, year of publication, type (article, journal, book), research problem, algorithm, tool, and dataset. Other information requires greater precision, including the limitations of the methods or architectures used, potential directions for future research, and the outcomes of the studies.

3.7. Data Synthesis

The data synthesis phase aims to summarise and present the key results obtained from the analysis of the selected studies. To accomplish this, the following strategy was employed:

- Address each research question individually by referencing the data extracted in the previous stage.
- Identify additional findings beyond those directly related to the research questions.
- Highlight gaps in the research and provide recommendations for future work.

3.8. Overview of the reviewed studies

The SLR allowed the identification of predominant approaches as well as some notable gaps in the literature regarding the generation of lyrics using DL models. Table 8 shows the studies that provide a theoretical foundation for research in the field of music lyric generation.

In general, current approaches appear to be quite fragmented, and there are very few models capable of producing lyrics that are both coherent and emotionally meaningful. This gap is the main reason behind this study, which seeks to develop a methodology for generating song lyrics with emotional context using deep learning techniques. Additionally, this research attempts to address challenges that have not been fully explored yet, including the simultaneous maintenance of semantic coherence and the expression of nuanced emotional content in generated lyrics, thereby contributing to a more comprehensive understanding of automated lyric generation.

Table 8 – Most relevant studies.

Authors	Title	Year	Summary	Methods	Data	Results	Limitations	Relevance & Contributions
Naveen Ram et al.	Say What? Collaborative Pop Lyric Generation Using Multitask Transfer Learning	2021	The authors developed a system that uses the T5 model to generate pop lyrics.	T5	Private dataset (Genius)	The model produces semantically meaningful lyrics, suitable for pop music. Evaluated with BLEU and human judgement (MTurk).	Private dataset; focused only on pop; automatic metrics may not fully capture creativity	Directly targets lyric generation using modern transformers; demonstrates that multitask learning enhances semantic coherence and suitability for pop music
Jacob Devlin et al.	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	2019	The BERT model was introduced for pre-training deep bidirectional transformers.	BERT	BooksCorpus + Wikipedia	Achieved new state-of-the-art across 11 NLP tasks: GLUE, MultiNLI, SQuAD v1.1 F1 and SQuAD v2.0 F1.	Not lyric-specific; does not generate creative text directly	Foundational model for later lyric generation research; enables fine-tuning on lyric-specific tasks
Harrison Gill et al.	Deep Learning in Musical Lyric Generation: An LSTM-Based Approach	2020	An LSTM is used to generate lyrics for a specific genre	LSTM	Private dataset (built from Genius lyrics API)	Generated rap and pop lyrics, preserved line length and variation across genres, close to training text.	Can produce repetitive text; struggles with long-term coherence	Demonstrates LSTM's effectiveness for genre-specific lyric generation; highlights importance of structured and clean datasets
Hugo Gonçalo Oliveira	Tra-la-Lyrics 2.0: Automatic Generation of Song Lyrics on a Semantic Domain	2019	The author presents a system that aims to generate text Based on the	N/A	Private Dataset	Produced semantically coherent lyrics aligned with the input rhythm.	Private dataset; limited model details; generalisation is restricted	Shows importance of aligning lyrics with musical rhythm; emphasises semantic coherence

			rhythm of a song melody, given as input.					
Hugo Gonçalo Oliveira	PoeTryMe: a versatile platform for poetry generation	2012	In this paper, the author presents a platform to generate poetry automatically.	N/A	N/A	Generates poetry across multiple genres using semantic and structural constraints.	Focused on poetry, not lyrics; does not use deep learning	Early creative text-generation system; demonstrates that semantic and structural constraints improve coherence and style, inspiring lyric generation
Sung-Hwan Son et al.	Korean Song-lyrics Generation by Deep Learning	2019	In this paper, the authors propose a system to create Korean song lyrics	LSTM	N/A	Successfully generated Korean lyrics resembling training data.	Dataset not disclosed; limited generalisation to other languages	Demonstrates applicability of LSTM to non-English lyrics; highlights importance of linguistic diversity
Liu et al.	RoBERTa: A Robustly Optimised BERT Pretraining Approach	2019	Introduces an optimised BERT variant with improved training strategies.	RoBERTa	BooksCorpus + CC-News, OpenWebText, Stories	Outperformed BERT on multiple NLP benchmarks: GLUE, RACE, and SQuAD.	Not lyric-specific; does not generate creative text directly	Influences subsequent transformer-based lyric generation; shows that optimised pretraining improves downstream performance

4. Methods

This chapter presents the methodology used for developing the project. It outlines the approach followed to design, implement, and evaluate a system capable of generating emotionally expressive song lyrics through DL techniques.

Before implementation, creating a project plan is a crucial step, as it directly influences the ability to achieve results within the stipulated timeframe. To ensure that the defined objectives are met, a Gantt chart was developed to illustrate the division of the work and its respective durations (Figure 12). The project planning has been divided into five main tasks, each further divided into subtasks to facilitate execution and management.

The first task involves a comprehensive review of the state of the art. The knowledge gathered at this stage supports the design choices and guides subsequent development.

The second task focuses on defining the DL models studied in the first task and selecting the metrics to be used for evaluating these models.

In the following task, the chosen models were implemented, and the process of generating song lyrics began.

Based on this, the next task focused on system evaluation, which was conducted through a combination of objective and subjective criteria.

The system was developed through an iterative cycle of training, evaluation, and refinement. After each evaluation phase, the models are retrained with adjusted parameters to improve their accuracy, creativity, and emotional expressiveness. This iterative process continues until the system demonstrates the ability to produce coherent and emotionally rich song lyrics.

All stages of the project were documented to ensure reproducibility and transparency. The final system represents the culmination of these tasks, combining a robust technical foundation with the creative potential required for lyric generation.

Figure 12 – Gantt chart of the project. Grey indicates the actual time taken, and red indicates the planned time.

4.1. Model Architecture and Training

The model architecture and training process are central to the development of a system capable of generating emotionally expressive song lyrics. In this project, the design prioritised flexibility and scalability, allowing experimentation with multiple deep learning models while maintaining control over the emotional content of the generated lyrics.

4.1.1. Model Selection

The system is primarily based on transformer architectures, which have shown outstanding performance in NLG tasks due to their capacity to capture long-range dependencies and contextual nuances.

In this project, an LSTM model was trained to predict the next token in a sequence based on preceding tokens, with emotional conditioning implemented via the concatenation of embeddings representing the quadrants of the Russell Circumplex Model, enabling the network to generate lyrics that reflect the intended affective state.

The GPT-2 model was also employed. The model consists of multiple layers of self-attention, layer normalization, and feed-forward networks, enabling it to capture long-range contextual dependencies in textual sequences. It generates text sequentially, predicting each token based on preceding tokens, and can be conditioned via designed textual prompts to control style, structure, and emotional content.

Additionally, the T5 model was utilised for lyric generation. Its encoder-decoder structure, with multiple layers of self-attention, cross-attention, and feed-forward networks, enables it to process input sequences and generate contextually coherent outputs. For this study, T5 was conditioned through textual prompts specifying the desired emotional quadrant, thematic content, and lyrical structure, allowing the generation of lyrics that are both stylistically consistent and emotionally aligned.

4.1.2. Emotional Conditioning

To integrate emotional control, the system leverages the Russell Circumplex Model of Emotions, which maps emotions across two axes: valence (positive vs. negative) and arousal (high vs. low). The model defines four quadrants corresponding to different emotional states, guiding the style and tone of the generated lyrics (Table 9). For the

system to “understand” the model, it was explicitly presented within the prompt construction process, each quadrant including representative emotions and stylistic instructions that inform the language models (Table 9). This ensures that the generated output aligns with the intended affective state.

Table 9 – Mapping of the Russell Circumplex Model quadrants to representative emotions and stylistic instructions used in lyric generation.

Quadrant	Valence	Arousal	Emotions	Stylistic Instructions
1	Positive	High	Happy, Excited, Joyful, Pleased	Use energetic, uplifting, and lively language, vivid imagery and fast rhythm.
2	Negative	High	Angry, Anxious, Tense, Frustrated	Use intense, fiery, and raw language, convey urgency and emotional conflict.
3	Negative	Low	Sad, Lonely, Tired, Depressed	Use slow, soft, and somber language, reflective tone with emotional depth.
4	Positive	Low	Calm, Relaxed, Peaceful, Serene	Use gentle, soothing, and tranquil language, flowing rhythm and harmonious imagery.

This structured representation enables the system to condition language models on specific emotional states, allowing for the generation of lyrics that are not only contextually coherent but also emotionally expressive. By linking each quadrant to well-defined stylistic instructions, the system can guide the selection of vocabulary, rhythm, and tone, effectively translating abstract emotional states into concrete linguistic patterns suitable for song lyrics.

4.1.3. Prompts for Lyrics Generation

Prompt engineering plays a critical role in the system. Before generating lyrics, the system constructs a detailed textual prompt that specifies the desired emotional quadrant. Optionally, the prompt can include thematic content, keywords, and the structure of the lyrics (such as the number of stanzas and lines per stanza). This approach allows the

models to produce outputs that are not only grammatically and stylistically coherent but also emotionally aligned with the intended affective state.

Table 10 – Examples of prompts given to the models.

Quadrant	Example Prompt
1	Write song lyrics from the 1st quadrant.
1	Write song lyrics from the 1st quadrant. The theme is about “summer festival”. Include the words “dance” and “laughter”. The structure of the lyrics should be 3 stanzas, each stanza with 4 verses.
2	Write song lyrics from the 2nd quadrant.
2	Write song lyrics from the 2nd quadrant. The theme is about “raging wildfire”. Include the words “flames” and “screams”. The structure of the lyrics should be 4 stanzas, each stanza with 3 verses.
3	Write song lyrics from the 3rd quadrant.
3	Write song lyrics from the 3rd quadrant. The theme is about “loneliness”. Include the words "silence" and "tears". The structure of the lyrics should be 2 stanzas, each stanza with 5 verses.
4	Write song lyrics from the 3rd quadrant.
4	Write song lyrics from the 4th quadrant. The theme is about “peaceful morning”. Include the words "sunrise" and "breeze". The structure of the lyrics should be 2 stanzas, each stanza with 5 verses.

4.1.4. Parameters

During text generation, the system relies on hyperparameters that directly influence the style, diversity, and coherence of the produced lyrics. The most relevant of these, and those employed in this work, are:

- Temperature: plays a crucial role by controlling the randomness of the generated output. Lower values (e.g., 0.3–0.7) result in more deterministic and conservative outputs, whereas higher values (e.g., 0.8–1.0) promote creativity and diversity (Chen et al., 2025).
- Top-p: manage the nucleus sampling, a technique for introducing randomness to the model’s output (Chen et al., 2025).

- Top-k: restricts sampling to the top k most probable tokens at each step of output generation (Chen et al., 2025).
- Maximum length: the number of tokens generated.

4.2. Dataset

For the training of the models, a pre-compiled dataset was provided by the supervisor, which was constructed according to Russell's Circumplex model of emotion. Lyrics were collected from three main sources[3] and subjected to preprocessing steps, including orthographic correction, removal of non-English or short texts (<100 characters), elimination of irrelevant elements (e.g., artist names, structural markers such as [Chorus x2]), and completion of repeated sections based on the corresponding audio (Malheiro, 2016).

Annotation was carried out by 39 participants with diverse backgrounds. The procedure consisted of reading each lyric, identifying the predominant emotion, assigning values between -4 and 4 for valence and arousal, and refining these values through relative ranking across samples. On average, each lyric received eight annotations, and the final values were computed using a 10% trimmed mean to reduce the influence of outliers. Lyrics with annotation standard deviation above 1.2 were discarded, resulting in a final dataset of 180 lyrics. Annotation consistency was evaluated using Krippendorff's alpha, yielding 0.87 for valence and 0.82 for arousal, indicating strong agreement among annotators. Although familiarity with the material was relatively low (≈12%), the reliability was ensured by the number of annotations per lyric and the high inter-annotator agreement (Malheiro, 2016).

The final dataset is distributed across quadrants as follows: 44 lyrics in Quadrant 1, 41 in Quadrant 2, 51 in Quadrant 3, and 44 in Quadrant 4. In terms of arousal, 85 lyrics were classified as positive and 95 as negative, while for valence, 88 were positive and 92 were negative (Malheiro, 2016).

Since the dataset was already prepared, no further text cleaning or preprocessing was required. This ensured consistency in data quality and avoided biases introduced during data collection.

[3] Lyrics.com, ChartLyrics and MaxiLyrics

4.3. Fine Tuning

The adaptation of pre-trained language models to the specific task of emotionally expressive lyric generation was achieved through a process of SFT. While models such as GPT-2, T5, and LSTM architectures possess strong general language capabilities, they are not inherently optimised for musical or poetic text. SFT allows the models to learn stylistic, structural, and emotional patterns specific to song lyrics, while retaining the broader linguistic knowledge acquired during pre-training.

Fine-tuning was performed with attention to several key hyperparameters:

- Learning rate - Selected to allow gradual adaptation without catastrophic forgetting of pre-trained knowledge.
- Batch size - Balanced between stability of training and computational efficiency.
- Number of epochs - Determined empirically to ensure sufficient exposure to all emotional quadrants while preventing overfitting.
- Optimiser - AdamW was used to improve convergence and maintain stability during gradient updates.
- Maximum sequence length: Set to cover the typical length of song verses, ensuring full contextual understanding

These parameters were iteratively adjusted based on preliminary experiments to optimise the emotional alignment, coherence, and stylistic quality of the generated lyrics.

4.4. Evaluation criteria

The metrics used to evaluate the evaluated generated lyrics were chosen based on prior literature on the assessment of machine-generated text and song lyrics, ensuring that both structural and emotional properties were considered.

4.4.1. Readability

For the evaluation of the readability of texts generated by language models, an architecture based on the automatic analysis of the text at three main levels – sentences, words, and syllables – was adopted. Initially, the text undergoes preprocessing that normalises characters and separates the content into sentences and words, ensuring correct

tokenisation. Next, the number of words and sentences is counted, alongside an estimation of the number of syllables per word, using both simple heuristics and, where available, pronunciation dictionaries for greater accuracy. With this data, the formulas for the FRE) and FKGL indices are applied, which combine averages of words per sentence and syllables per word to quantify reading ease and the educational level required to understand the text. Finally, the results are adjusted and formatted for analysis, enabling an objective assessment of the linguistic complexity of the lyrics produced by the models.

4.4.2. Perplexity

The GPT-Neo model was employed as a reference for evaluating the linguistic quality of the song lyrics. It was chosen due to its computational efficiency, allowing perplexity to be calculated over a relatively large set of texts without requiring extensive computational resources. Its primary role was specifically to compute the perplexity of the generated texts, thereby providing a quantitative measure of fluency and naturalness. The choice of this model aligns with the strategy of relying on an external and consistent evaluator, rather than assessing texts with the same models responsible for their generation. In this way, the results obtained reflect an independent measure of linguistic performance, ensuring greater robustness and impartiality in the comparative analysis carried out throughout the study.

4.4.3. VAD

The system follows a modular workflow for the automatic processing of song lyrics and the generation of emotional classifications. It begins by reading the lyrics from text files alongside a mapping file containing the ground-truth quadrants. Each lyric undergoes text pre-processing, including normalization and tokenization, before the tokens are compared against a reference lexicon to extract emotional attributes and compute mean values. Songs are then assigned to an emotional quadrant, and the predicted quadrants are evaluated against the ground-truth labels using performance metrics and confusion matrices. This modular design supports scalability and reusability across different lyric datasets. Confusion matrices are employed to evaluate alignment between predicted and intended quadrants, providing both quantitative and visual insights into model performance.

4.5. Tools and Frameworks

The development of the emotionally controlled lyric generation system relied on a combination of programming languages, libraries, and deep learning frameworks, selected for their robustness, flexibility, and support for NLP and text generation tasks:

- Python was chosen as the programming language due to its widespread adoption in NLP research, extensive library ecosystem, and ease of integration with deep learning frameworks.
- Frameworks: PyTorch, for model implementation, training, and fine-tuning, offering dynamic computation graphs and seamless integration with the Hugging Face Transformers library. The Transformers library provided pre-trained models (GPT-2, T5) and tokenizers, as well as pipelines for text generation and evaluation.
- Text processing libraries: NLTK (Natural Language Toolkit) library was used for tokenization, part-of-speech tagging, and basic text preprocessing. The textstat library was employed to compute readability metrics such as the Flesch-Kincaid score. Regular Expressions were applied to parse prompts and extract keywords, themes, and structural instructions.
- Data visualization and analysis: Matplotlib and Seaborn were used for plotting results and analyzing metrics. Pandas was employed for handling tabular data, metrics, and evaluation results. Scikit-learn was used to compute confusion matrices and other classification metrics.
- GPU Acceleration: Model training and evaluation were performed on CUDA-enabled GPUs, improving performance and reducing runtime.
- Development Platforms: The project was implemented and tested using Google Colab and Jupyter Lab, which provided interactive notebooks for experimentation and visualization.

5. Implementation and Results

This chapter describes the implementation of the emotionally controlled lyric generation system and presents the main results obtained. The system was designed to generate song lyrics aligned with specific emotional states, as defined by the Russell Circumplex Model, while optionally incorporating themes, keywords, and structural constraints.

5.1. Implementation Process

The implementation followed a structured pipeline that integrates user input, prompt construction, model selection, text generation, post-processing, and evaluation.

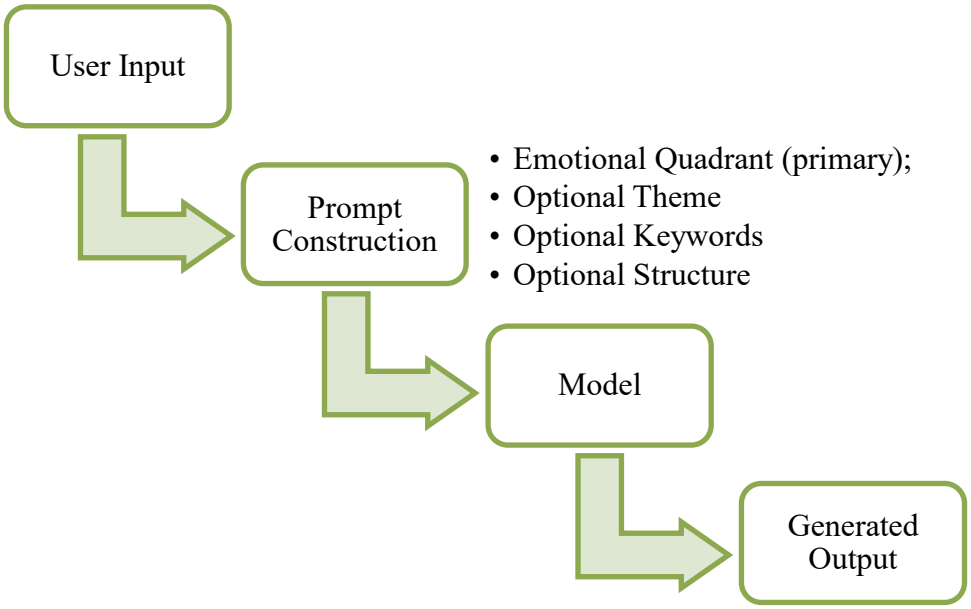

Figure 13 – System pipeline.

The main stages of this pipeline are illustrated in Figure 13, and are described as follows:

1. User Input: The system begins by accepting user specifications, which include the desired emotional quadrant (as defined by the Russell Circumplex Model) and may also include thematic elements, keywords, and structural constraints such as the number of stanzas and verses per stanza. These inputs provide the foundation for prompt construction and guide the stylistic and emotional direction of the generated lyrics.

2. Prompt Construction: Based on the user input, a detailed textual prompt is generated for each model. The prompt explicitly encodes the emotional state, optional theme, keywords, and structural specifications, ensuring that the models are conditioned to produce outputs that respect both stylistic and affective requirements. This step is crucial for controlling the content of the generated lyrics and achieving optimal emotional alignment. Consistent prompts were used across all models to maintain comparability between outputs and maximise emotional alignment.
3. Model Selection and Generation: Depending on the experimental setup, one of the pre-trained language models is selected for lyric generation. The GPT-2 and T5 models were implemented using the Hugging Face Transformers library, which provided pre-trained weights, tokenizers, and ready-to-use pipelines for text generation. Each of them was trained as is and subsequently fine-tuned on the provided dataset (all of 180 lyrics were used) to capture stylistic, structural, and emotional patterns. The LSTM model was implemented directly in PyTorch, allowing precise control over the number of layers, hidden size, sequence length, and dropout. PyTorch also handles the dataset provided (all of 180 lyrics were used), manages backpropagation and optimization during training, and supports sequential text generation for inference, predicting the next token step by step based on the learned patterns. For text generation, parameters such as temperature, top-k, top-p, and maximum sequence length were configured to achieve a desired balance between creativity, coherence, and adherence to the prompt. During fine-tuning, key hyperparameters were selected to balance adaptation to the lyric dataset with the retention of pre-trained knowledge. These parameters include the learning rate, batch size, number of epochs, maximum sequence length, hidden size (for LSTM), and encoder-decoder attention layers (for T5). The specific values used for both fine-tuning and text generation for each model are summarized in Table 11.

Table 11 – Parameters used for fine-tuning and text generation per model

Model	Hyperparameter	Values
LSTM	Learning rate	1,00E-03
	Batch size	16

	Number of epochs	15
	Maximum sequence length	300
	Hidden size / layers	256
GPT-2	Learning rate	5,00E-05
	Batch size	16
	Number of epochs	5
	Temperature	0,88
	Top-k	40
	Top-p	0,95
	Maximum sequence length	300
T5	Learning rate	5,00E-05
	Batch size	16
	Number of epochs	5
	Temperature	0,88
	Top-k	40
	Top-p (nucleus sampling)	0,95
	Maximum sequence length	300
	Encoder-decoder attention layers	12 layers each

4. Lyrics Generation: The selected model generates lyrics sequentially, token by token, conditioned on the constructed prompt. For transformer-based models, such as GPT-2 and T5, self-attention and contextual embeddings enable the generation of coherent and contextually relevant sequences, whereas the LSTM model relies on learned sequential dependencies and emotion-conditioned embeddings. For this study, a total of 80 lyrics were generated for each model, with 20 lyrics corresponding to each quadrant of Russell's model. Within each quadrant, 10 lyrics were generated using prompts that specified only the quadrant, while the remaining 10 employed prompts that additionally included the intended structure, keywords, and theme, allowing for a more guided generation.
5. Post-Processing: Generated outputs undergo post-processing to ensure consistency with the user-specified structure, keywords and themes. This includes formatting lyrics according to the requested stanza and line structure,

correcting spacing and punctuation, and removing potential repetition or malformed sequences.

6. Evaluation: Finally, the generated lyrics are evaluated using a combination of quantitative and qualitative metrics. Quantitative measures include perplexity, Flesch-Kincaid readability scores, and emotion classification accuracy (using VAD-based models and confusion matrices). Qualitative assessment involves manual review of stylistic coherence, emotional alignment, and lyrical fluency, providing a comprehensive view of the system's performance. For each model, all 80 generated lyrics were evaluated, allowing for a thorough comparison across the different quadrants of Russell's model and generation conditions.
    a. The readability evaluation was implemented in Python using the textstat library to calculate the FRE and FKGL indices. Texts were pre-processed to normalise characters and tokenised into sentences and words. Syllable counts, as well as word and sentence counts, were computed automatically by textstat to generate the readability scores. For comparison and validation, a manual implementation was also developed using Python's standard libraries, which estimates syllable counts using vowel-based heuristics.
    b. For the implementation of perplexity, the GPT-Neo-125M model was employed, with the Hugging Face Transformers library used to handle tokenization and model inference, thereby ensuring reproducibility and compatibility with widely adopted NLP tools. Each lyric file was read, pre-processed, and evaluated with the model to obtain the cross-entropy loss, from which perplexity values were computed. As the stability of perplexity scores can vary for shorter sequences, the final value reported for each lyric corresponds to the mean perplexity across the processed text. These results were subsequently compiled into structured tables and graphs to facilitate systematic analysis and comparison.
    c. The NRC VAD Lexicon (Mohammad, 2025) was used as the reference resource for extracting valence and arousal scores. The NRC VAD Lexicon was loaded from a tab-separated text file and normalized from [-1, 1] to [0, 1]. Lyrics were read from text files and

pre-processed by converting to lowercase, removing punctuation, and tokenizing words. For each token found in the lexicon, valence and arousal scores were retrieved and averaged to compute the mean values per lyric. Based on these means, each lyric was assigned to one of the four quadrants. Predicted quadrants were then compared against the ground-truth labels from the CSV, and confusion matrices were generated and visualized using Matplotlib. All results, including per-lyric valence, arousal, and predicted quadrant, were stored for further analysis.

5.2. Generation Results and Analysis

This section presents the results obtained from the emotionally controlled lyric generation system. The analysis focuses on evaluating the quality, coherence, and emotional alignment of the generated lyrics across the different models. Both quantitative metrics, such as readability, perplexity, and emotion classification accuracy, and qualitative assessments of stylistic and structural properties are considered. The aim is to provide a comprehensive understanding of each model's performance and the effectiveness of the prompt-based conditioning on emotional expression.

Readability

The mean FRE and FKGL scores for each language model are presented in Table 12 and in Table 13, respectively, providing an overview of the readability characteristics of the generated lyrics.

Table 12 – Mean FRE for different language models.

Model	Mean
GPT-2	79.30
GPT-2_finetuned	70.50
LSTM	-30.25
T5	68.88
T5_finetuned	68.75

The results indicate notable differences among the models. GPT-2 exhibited the highest mean FRE (79.30), suggesting that its outputs are generally easier to read. This

suggests that GPT-2 is likely to produce lyrics that are simple, direct, and accessible. The fine-tuned version (GPT-2_finetuned) showed a lower mean score (70.50), indicating that fine-tuning slightly increased the complexity of the text, potentially adding more varied structures or richer vocabulary to the lyrics.

In contrast, the LSTM model presented a negative mean FRE (-30.25), highlighting that its outputs are considerably more difficult to read. This indicates that LSTM-generated lyrics are highly inconsistent and may be challenging to interpret, making them less suitable for conventional lyric writing.

Both T5 and T5_finetuned produced similar mean FRE scores (68.88 and 68.75, respectively), suggesting that fine-tuning had minimal impact on readability. While T5 outputs are less readable than those from GPT-2, they remain more consistent and easier to read than LSTM-generated texts.

In summary, the analysis of mean FRE scores suggests that GPT-2 is well-suited for generating highly readable, simple lyrics, LSTM produces complex and unpredictable outputs, and T5 offers a stable, moderately complex alternative.

Table 13 – Mean FKGL for different language models.

Model	Mean
GPT-2	8.92
GPT-2_finetuned	12.73
LSTM	50.35
T5	5.94
T5_finetuned	5.97

The results reveal clear differences among the models. GPT-2 produced lyrics with a mean FKGL of 8.92, indicating moderately complex text that remains accessible to most adult readers. The fine-tuned version (GPT-2_finetuned) showed a higher mean FKGL of 12.73, suggesting that fine-tuning introduced additional linguistic complexity, potentially resulting in more sophisticated or poetic lyrics.

In contrast, the LSTM model showed an extremely high mean FKGL of 50.35, demonstrating that its outputs are highly complex and inconsistent. Such high complexity makes LSTM-generated lyrics largely unsuitable for conventional songwriting.

Both T5 and T5_finetuned produced similar mean FKGL scores (5.94 and 5.97, respectively), indicating that their outputs are highly readable. This simplicity, combined

with consistency, makes T5 particularly well-suited for generating accessible, straightforward lyrics, as typically found in popular music. Fine-tuning had minimal effect on the readability of T5-generated lyrics.

In summary, the FKGL analysis suggests that GPT-2 generates moderately complex lyrics, GPT-2_finetuned produces more sophisticated outputs, T5 generates highly readable and simple lyrics, and LSTM produces overly complex and inconsistent lyrics.

Perplexity

The perplexity of the generated lyrics was evaluated using the GPT-Neo-125M model, and the mean values are presented in Table 14.

Table 14 – Comparison of Text Perplexity for the models.

Model	Mean
GPT-2	16.82
GPT-2_finetuned	16.53
LSTM	23.54
T5	23.54
T5_finetuned	23.86

The results reveal distinct differences in predictability and coherence among the models. GPT-2 and GPT-2_finetuned produced lyrics with the lowest mean perplexity values, indicating that these models generate text that is relatively predictable and internally consistent according to GPT-Neo-125M. The slight reduction in perplexity for GPT-2_finetuned suggests that fine-tuning marginally improved the alignment of the generated lyrics with the linguistic patterns recognized by GPT-Neo-125M.

In contrast, LSTM, T5, and T5_finetuned exhibited higher perplexity scores, with means around ranging from 23.5 to 23.9. These elevated values suggest that the lyrics produced by these models are less predictable, which may reflect greater variability in word choice, syntax, or sentence structure. While higher perplexity does not necessarily indicate lower quality, it highlights differences in the consistency of the generated text compared to the reference model.

NRC VAD

The emotional characteristics of the generated lyrics were evaluated using the NRC Valence-Arousal-Dominance (VAD) framework. Each lyric was classified into one of four quadrants based on high/low valence (V) and high/low arousal (A):

- HV/HA: Positive Valence / High Arousal.
- BV/HA: Negative Valence / High Arousal.
- BV/LA: Negative Valence / Low Arousal.
- HV/LA: Positive Valence / Low Arousal.

Tables 15 to 19 present the distribution of generated lyrics across these quadrants for each model.

Table 15 – Distribution of GPT-2-generated lyrics across true and predicted NRC VAD quadrants.

		Predicted Label			
		HV/HA	BV/HA	BV/LA	HV/LA
Actual Label	HV/HA	10	2	3	5
	BV/HA	6	4	6	4
	BV/LA	4	4	7	5
	HV/LA	6	4	9	1

For GPT-2, the results show a relatively balanced emotional distribution. The model correctly predicted a substantial number of lyrics in the HV/HA and BV/LA quadrants, indicating proficiency in generating energetic positive and calm negative lyrics. However, true HV/LA lyrics were frequently misclassified as BV/LA or HV/HA, suggesting occasional confusion between calm positive and calm negative moods. Similarly, true BV/HA lyrics were spread across all quadrants, reflecting moderate difficulty capturing high-arousal negative emotions. Overall, GPT-2 demonstrates a wide emotional range, with a slight bias toward misclassifying low-arousal positive lyrics.

Table 16 – Distribution of GPT-2 fine-tuned generated lyrics across true and predicted NRC VAD quadrants.

		Predicted Label			
		HV/HA	BV/HA	BV/LA	HV/LA
Actual Label	HV/HA	8	1	2	9
	BV/HA	5	3	7	5
	BV/LA	2	0	10	8
	HV/LA	6	2	8	4

The fine-tuned GPT-2 shows improved alignment with true emotional labels, particularly for BV/LA and HV/HA lyrics. Fine-tuning increased the correct predictions in

these quadrants while slightly reducing misclassifications of high-arousal negative lyrics. True HV/HA lyrics were still occasionally misclassified as HV/LA, indicating persistent challenges in distinguishing high-energy positive from low-arousal positive moods. True BV/LA lyrics were sometimes predicted as HV/LA, reflecting occasional confusion between low-arousal negative and positive lyrics. Despite these minor misclassifications, fine-tuning overall improved emotional control and reduced extreme mispredictions, enhancing the model's ability to produce calm and reflective lyrics alongside energetic positive content.

Table 17 – Distribution of T5 generated lyrics across true and predicted NRC VAD quadrants.

		Predicted Label			
		HV/HA	BV/HA	BV/LA	HV/LA
Actual Label	HV/HA	4	4	3	9
	BV/HA	8	7	2	2
	BV/LA	3	5	10	2
	HV/LA	4	4	5	6

For T5, the emotional predictions are more variable. True BV/HA and BV/LA lyrics were predicted with reasonable accuracy, indicating that T5 captures negative moods moderately well. However, true HV/HA lyrics were frequently misclassified as HV/LA or BV/HA, and HV/LA lyrics were distributed across all quadrants, suggesting inconsistency in representing high-arousal positive and low-arousal positive moods. This pattern indicates that T5 tends to overpredict calm or negative moods, potentially underrepresenting energetic positive emotions.

Table 18 – Distribution of T5 fine-tuned generated lyrics across true and predicted NRC VAD quadrants.

		Predicted Label			
		HV/HA	BV/HA	BV/LA	HV/LA
Actual Label	HV/HA	8	3	3	6
	BV/HA	5	7	5	2
	BV/LA	3	5	3	9
	HV/LA	1	7	6	5

The fine-tuned T5 shows improvements in capturing HV/HA and HV/LA lyrics. Fine-tuning reduced the overprediction of negative moods and balanced the emotional distribution. Nevertheless, true BV/LA lyrics were occasionally misclassified as HV/LA, and HV/LA lyrics were sometimes predicted as BV/HA or BV/LA, highlighting ongoing challenges in distinguishing low-arousal moods of differing valence. Overall, fine-tuning increased the model's control over emotional output and produced a more balanced

distribution of positive and negative moods, while maintaining diversity in emotional expression.

Table 19 – Distribution of LSTM generated lyrics across true and predicted NRC VAD quadrants.

		Predicted Label			
		HV/HA	BV/HA	BV/LA	HV/LA
Actual Label	HV/HA	3	6	3	8
	BV/HA	5	6	3	5
	BV/LA	5	4	4	7
	HV/LA	3	8	4	4

In contrast, LSTM demonstrates the least accurate predictions and the highest variability. True HV/HA lyrics were frequently misclassified as HV/LA or BV/HA, while HV/LA and BV/LA lyrics were scattered across multiple quadrants. Although the model captured some BV/HA and HV/LA lyrics correctly, the overall distribution is inconsistent, reflecting weak alignment with true emotional labels. This high misclassification rate indicates that LSTM-generated lyrics are less reliable in terms of reproducing intended emotional tones and show considerable unpredictability in their emotional content.

Table 20 – F1-scores of generated lyrics across models.

Model	**F1-Score**
GPT-2	26.0%
GPT-2_finetuned	28.2%
LSTM	21.7%
T5	34.5%
T5_finetuned	29.5%

The results shown in Table 20 indicate that the models struggled to generate lyrics with clearly distinguishable emotions when analysed using the NRC-VAD lexicon. The F1-scores obtained were generally low, with T5 achieving the highest score at 34.5%, followed by T5 fine-tuned at 29.5%, GPT-2 fine-tuned at 28.2%, GPT-2 at 26.0%, and LSTM performing worst at 21.7%. Considering that the random baseline for four emotional quadrants is 25%, these findings suggest that the generated lyrics generally lack clear emotional differentiation. This outcome is expected, as GPT-2 and T5 were not explicitly trained to produce emotionally controlled text, and NRC-VAD captures only lexical indicators of valence and arousal, which may not fully reflect more subtle or complex emotional cues present in the lyrics. Overall, the results highlight the limitations

of current text generation models in producing lyrics with consistent emotional content and emphasise the challenges of evaluating generated text using lexicon-based methods.

Structure

The LSTM model, when provided with the structure of stanzas and lines per stanza, was able to generate song lyrics that respected this format. However, the dataset used was very small, which limited the model's ability to learn to generate lyrics independently.

Similarly, the GPT-2 model generated lyrics according to the specified structure when prompted with stanza and line information. In the absence of such guidance, GPT-2 produced continuous text. This behaviour is likely due to GPT-2 having been pre-trained primarily on continuous text and not exposed to song lyrics. Even when provided with a dataset, GPT-2 does not reliably generate lyrics with the expected structured format.

In contrast, the T5 model did not consistently generate text following the expected structure, often returning incomplete or empty lines. This illustrates the main challenge: the fine-tuned T5 learned to reproduce patterns seen during training, but the training data did not provide consistent structural information. As a result, the model is unable to produce new lyrics based on creative prompts and tends to generate text line by line rather than in stanzas.

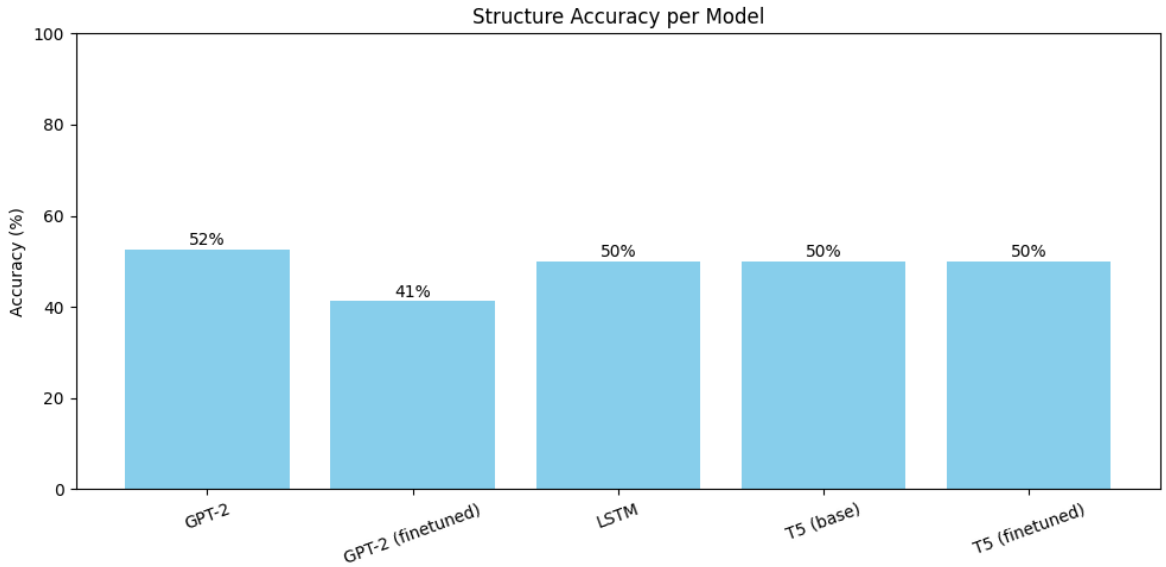

Figure 14 – Structure Accuracy per Model.

In this experiment, each model generated 80 lyrics, but only 40 of these (10 per quadrant) were prompted with a clearly specified structure. Given this setup, the maximum achievable structure accuracy is 50% if all structured prompts were followed perfectly. As shown in Figure 14, the observed structure accuracies range from 41% to 52%, which is

consistent with expectations. GPT-2 slightly outperformed the other models, while fine-tuned GPT-2 showed a lower structure adherence (41.25%). These findings confirm that models can follow structural instructions to some extent, but the probabilistic nature of text generation and the absence of explicit guidance for half of the dataset prevent perfect adherence. The remaining 40 lyrics, without explicit structural instructions, were not expected to conform to any specific format, explaining why the accuracy does not approach 100%.

Keywords and Theme

The LSTM model could include some of the words specified in the prompt, particularly those appearing early in the input. However, due to the limited size of the dataset and the inherent sequential memory constraints of LSTMs, thematic coherence across the entire text was limited, and the model often reproduced patterns observed during training rather than generating fully novel content.

GPT-2 was able to include some of the keywords specified in the prompt, demonstrating that it can partially follow thematic guidance. However, this capability is limited: the model does not consistently integrate all requested keywords or maintain coherent thematic development across longer texts.

During training, T5 learned to reproduce patterns from the dataset and prompt, like keywords and themes. As a result, the model cannot reliably generate entirely new lyrics based on creative prompts. Instead, it tends to reproduce sequences like what it has seen during training, often ignoring or only partially incorporating the requested themes or keywords. This behaviour is expected given the small dataset and the absence of structured or semantically guided training signals.

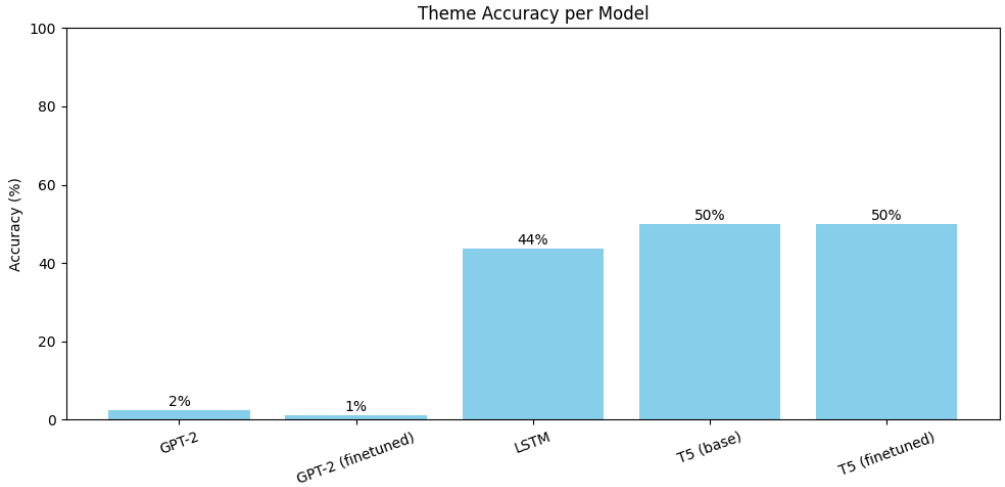

Figure 15 – Theme Accuracy per Model.

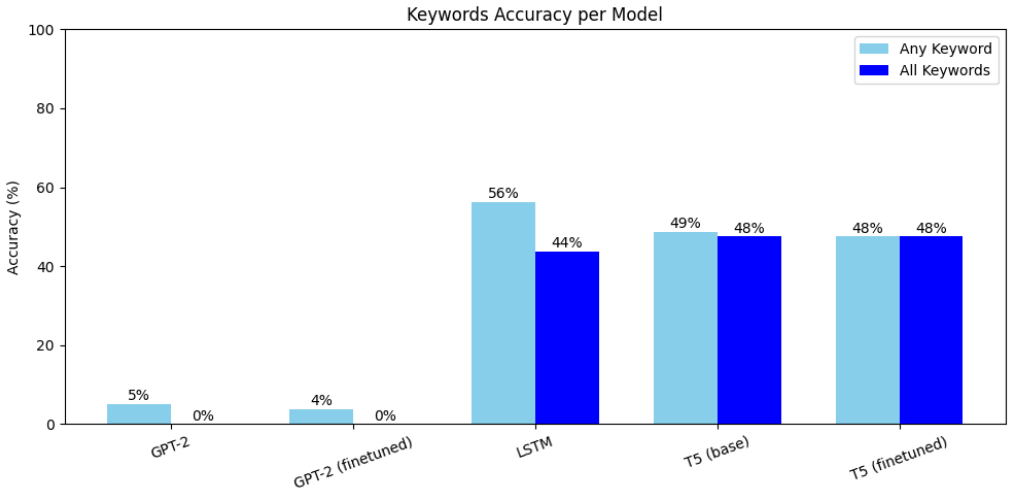

Figure 16 – Keywords Accuracy per Model.

The analysis of theme adherence shows a clear distinction between models (Figure 15 – Theme Accuracy per Model.), considering only the 40 lyrics with explicitly defined themes. Regarding theme adherence, the GPT-2 models performed poorly, with theme accuracy below 3%, indicating minimal ability to maintain thematic consistency. In contrast, LSTM and T5 models achieved theme accuracy between 43.75% and 50%, reflecting a better, but still partial, capacity to generate lyrics consistent with the specified theme. It is important to note that this higher performance is largely due to these models frequently reproducing content directly from the prompts or from the training dataset, rather than generating fully original lyrics.

The analysis of keyword adherence shows a clear distinction between models (Figure 16 – Keywords Accuracy per Model.), considering only the 40 lyrics with explicitly defined themes. The GPT-2 models, both base and fine-tuned, rarely included the keywords in the generated lyrics, with only 5% of GPT-2 lyrics and 3.75% of GPT-2 fine-tuned lyrics containing at least one keyword, while for both models the accuracy of including all keywords was consistently 0%. In contrast, the LSTM model included at least one keyword in 56.25% of lyrics and all keywords in 43.75%, while the T5 models ranged from 47.5% to 48.75% for at least one keyword and 47.5% for all keywords. This higher adherence is partly due to these models frequently copying content directly from the prompts or training dataset, rather than independently incorporating the keywords into novel lyrics.

5.3. Discussion of findings

The evaluation of the lyric generation models reveals clear differences in their performance across readability, coherence, emotional alignment, structure, and thematic consistency (Table 21).

Table 21 – Comparative summary of readability, coherence, structure, keyword and theme adherence, and overall performance for each lyric generation model.

Model	FRE	FKGL	Perplexity	Structure Accuracy (%)	Any Keyword Accuracy (%)	All Keyword Accuracy (%)	Theme Accuracy (%)
GPT-2	79.30	8.92	16.82	52.50	5.00	0.00	2.50
GPT-2_finetuned	70.50	12.73	16.53	41.25	3.75	0.00	1.25
LSTM	-30.25	50.35	23.54	50.00	56.25	43.75	43.75
T5	68.88	5.94	23.54	50.00	48.75	47.50	50.00
T5_finetuned	68.75	5.97	23.86	50.00	47.50	47.50	50.00

Considering the FRE and FKGL scores, GPT-2 generates lyrics that are simple and accessible, with a FRE of 79.3 and FKGL of 8.92. The fine-tuned version increases linguistic complexity (FKGL of 12.73), suggesting that fine-tuning introduced more varied vocabulary and syntactic structures. In contrast, LSTM outputs are extremely complex and difficult to read (FRE of -30.25 and FKGL for 50.35), reflecting inconsistency and making them less suitable for conventional songwriting. T5 and T5 fine-tuned provide a balance of readability and simplicity (FRE around 68.8 and FKGL around 5.95), producing texts that are consistent and accessible, which aligns well with the characteristics of popular music lyrics.

In terms of predictability and coherence, GPT-2 and its fine-tuned version achieved the lowest perplexity values, indicating consistent and internally coherent outputs. LSTM and T5 produced higher perplexity scores, reflecting greater variability in word choice and sentence structure. While this variability may contribute to creativity, it also reduces stability and predictability compared to GPT-2.

Emotional alignment further highlights distinctions among the models. GPT-2 captures high-energy positive and calm negative moods most accurately, and fine-tuning improves control over emotional expression. T5 benefits from fine-tuning primarily in

recognizing positive and calm emotions, though it still tends to overpredict calm or negative moods. LSTM exhibits the weakest emotional alignment, with outputs frequently misclassified across quadrants, reflecting inconsistent conveyance of intended emotions. Across all models, F1-scores indicate that clearly distinguishable emotional content remains challenging to achieve, partly due to the reliance on lexicon-based evaluation.

When considering structure and adherence to prompts, all models achieved structure accuracy between 41% and 52% when only the 40 lyrics with explicit structural prompts were considered. GPT-2 base slightly outperformed the other models, while GPT-2 fine-tuned achieved lower structure adherence. Concerning keywords and theme adherence, LSTM and T5 showed higher rates. However, it is important to note that this high adherence largely results from these models frequently reproducing content directly from the prompts or the training dataset, rather than generating fully original lyrics. GPT-2, even when fine-tuned, rarely integrated keywords or maintained thematic coherence, highlighting its capacity for readability and coherence but its limited ability to follow complex instructions.

Overall, GPT-2, particularly the fine-tuned version, demonstrates the best balance among readability, coherence, emotional control, and structural fidelity, making it the most suitable for prompt-based, emotionally guided lyric generation. T5 provides consistent and readable outputs but relies heavily on prompts and the training dataset for high adherence, while LSTM generates highly variable and less reliable lyrics. LSTM outputs, although capable of following some structural and keyword cues, remain highly variable, complex, and less reliable. These findings underscore the importance of model architecture, dataset quality, and fine-tuning strategy in producing high-quality, emotionally expressive lyrics and suggest that prompt-based conditioning alone is insufficient to ensure fully controlled or original lyrical generation.

6. Conclusion

This chapter summarises the main contributions and findings of this work. It is divided into two sections: Concluding Remarks, which highlights the key outcomes, and Limitations and Future Research, which addresses the main constraints and suggests directions for further investigation.

6.1. Concluding Remarks

The main purpose of this project was to develop a system capable of automatically generating song lyrics based on specific emotions, using DL models. The work was guided by Russell's Circumplex Model of Emotions, which provided a structured psychological foundation for conditioning lyrics on valence and arousal. In addition, the study also aimed to design evaluation methods to assess the quality of the generated lyrics, not only in terms of linguistic aspects such as coherence and readability, but also in terms of their emotional accuracy. In doing so, the research aimed to bridge the gap between NLG and creative tasks in music composition, while also identifying the limitations of existing DL models in this domain.

From the experiments conducted, we can conclude that each model has its own strengths and limitations when it comes to generating song lyrics. LSTM is capable of capturing sequences of words and reflecting some emotional tendencies, producing lyrics that follow certain patterns. However, it struggles with overfitting and maintaining rhythm and consistent meter, which limits its ability to generate fully structured and coherent lyrics independently.

GPT-2 is better at generating coherent and grammatically correct text, and it can follow basic instructions. Nevertheless, it does not naturally produce lyrics with a musical style. This is largely because GPT-2 was pre-trained on large corpora of prose-like text, which is different from the structure and stylistic conventions of song lyrics. As a result, its output often lacks the flow and rhythm typical of songs, including rhyme patterns, line breaks, and repetitions that are common in music.

In the experiments with T5, the model was able to generate coherent text, but it often produced repetitive lines or referenced external content. Even with fine-tuning, T5 struggled to maintain coherence across verses and stanzas, which shows the difficulty of

using seq2seq models for creative and structured lyric generation. This demonstrates the limitations of using a small, non-specialized dataset for training.

Considering the objectives defined, the system developed in this work achieved the central aim of demonstrating that it is possible to generate emotion-driven song lyrics using DL models. The integration of Russell's Circumplex Model of Emotions into the workflow functioned as expected, allowing prompts to be conditioned on emotions based on their valence and arousal. In addition, evaluation methodologies were established, materialised through the use of metrics such as readability (FRE, FKGL), coherence (perplexity), emotional alignment (NRC-VAD), and structural adherence. These metrics enabled a systematic comparison between the different models and provided concrete evidence of their strengths and limitations.

With these objectives accomplished, it is now possible to address the research questions one by one:

- RQ1: Which DL algorithms are the most frequently used for generating song lyrics in the literature? The review conducted confirmed that LSTM, GPT-based architectures, and encoder–decoder models such as T5 are the most prominent approaches. These were therefore selected as the core models for experimentation in this work.
- RQ2: How do different DL models compare in their ability to generate emotionally aware song lyrics? The research showed that GPT-2, particularly the fine-tuned version, offered the best balance of readability, coherence, and emotional alignment, while LSTM and T5 were more effective at integrating keywords and themes, though often by copying from prompts or training data rather than generating novel content.
- RQ3: How can DL models be trained and tuned to generate song lyrics that effectively express specific emotions? Fine-tuning improved GPT-2's emotional precision, while LSTM required explicit structure guidance and T5 benefited from conditioning but struggled with consistency. These findings suggest that training on larger, domain-specific datasets would further enhance performance.
- RQ4: How can we measure and evaluate the emotional accuracy of song lyrics generated by DL models? The use of the NRC-VAD lexicon provided a structured way of mapping generated lyrics to valence and arousal values,

enabling quantifiable comparisons with intended emotions. This confirmed the feasibility of automatic evaluation of emotional alignment in generated text. Future work could build on this by employing more advanced evaluation methods, such as perplexity AI or embedding-based emotion models, to capture subtler aspects of emotional expression.

- RQ5: What are the best metrics and methods for assessing the coherence and stylistic consistency of automatically generated song lyrics? A combination of perplexity for coherence, readability indices for linguistic clarity, and structural adherence checks for stanza/verse format proved effective. Together, these methods provided a multidimensional assessment of lyric quality beyond subjective evaluation.
- RQ6: In what ways can the output of DL models be refined to enhance the originality and uniqueness of the generated song lyrics, while maintaining emotional expressiveness? The results highlight that a larger, more specialised lyric datasets, reinforcement learning from human feedback, and stronger conditioning mechanisms could improve originality and emotional expressiveness. Additionally, combining statistical models with creative post-processing tools may support more musically convincing results.

In summary, all three models, LSTM, GPT-2, and T5, demonstrated useful capabilities. GPT-2, especially when fine-tuned, offered the most consistent balance between readability, coherence, and emotional control. However, none of the models managed to fully capture the complexity of real song lyrics. This shows that, while progress has been made, there is still a lot of room for improvement, especially regarding dataset quality, originality, and more advanced methods for conditioning generation on emotions.

6.2. Limitations and Future Research

This study faced several limitations that shaped the outcomes and pointed towards promising directions for future work.

Some transformer-based models, such as BERT and RoBERTa were explored, but lyrics generation with these models did not produce satisfactory results due to their bidirectional architecture, which is designed for context understanding and classification

tasks, rather than sequential text generation. As these models consider context from both the left and right of each token simultaneously, they are not naturally suited to predict the next word autonomously, which is essential for text generation tasks.

Adapting a large pre-trained model to a specific task is not a straightforward process. Pre-trained models carry deep "habits" from their training data, and these can be difficult to change. In the case of GPT-2, its prose-like style is deeply ingrained. Without large, specialized lyric datasets and substantial computational resources, it is challenging to transform GPT-2 into a fully musical lyric generator. While the generated text is grammatically coherent, it does not always feel like actual song lyrics, missing important aspects such as rhyme, rhythmic structure, and lyrical repetition. Similarly, T5 models frequently struggled with coherence across longer lyrics, repeating words or producing incomplete stanzas. These challenges underscore the importance of dataset design and training strategies tailored to the unique demands of musical language.

Another limitation concerns memory and coherence. GPT-2 lacks long-term memory for extended lyrics, which constrained its ability to generate longer, thematically consistent pieces, while T5 models often defaulted to fragmented or repetitive structures when uncertain. This suggests that architecture designed for longer dependencies, or fine-tuned with explicit structural markers, would be more suitable for lyric generation.

The dataset size was also a restrictive factor. The relatively small corpus limited the models' capacity to generalize and to produce stylistically diverse lyrics. Expanding the dataset, either through larger collections of lyrics or via data augmentation techniques, would allow models to learn more varied stylistic and rhythmic patterns. Moreover, training on corpora that combine lyrics with other creative writing forms could further enhance originality and expressiveness.

This study aimed to evaluate 80 song lyrics generated by deep learning models, distributed across the four quadrants of Russell's model, using Perplexity AI to assess emotional perception and coherence. However, this analysis could not be carried out in full due to limitations of the tool and the constraints of the study. The lyrics often exceeded the maximum text length allowed for upload, and the tool did not support batch processing of large volumes, requiring manual analysis. In addition, the time and computational resources available did not permit a complete evaluation, restricting the study to a representative subset. Future research could address these limitations by using APIs capable of processing texts in batches, dividing lyrics into smaller segments, or employing

alternative tools with greater analytical capacity, thereby allowing a comprehensive assessment of lyrics across all four quadrants of Russell's model.

Future work should focus on larger and more diverse lyric datasets, architectures better suited to creative text, and training strategies that incorporate musical and poetic constraints to improve the quality of automatically generated lyrics. By addressing these limitations, future work has the potential to produce systems that are not only more robust and reliable but also capable of generating lyrics that are expressive, musically aligned, and emotionally consistent.

References

Agrawal, Y., Shanker, R. G. R., & Alluri, V. (2021). Transformer-based approach towards music emotion recognition from lyrics. *Lecture Notes in Computer Science*, *12657*, 167–175. https://doi.org/10.1007/978-3-030-72240-1_12

Amara, S., Macedo, J., Bendella, F., & Santos, A. (2016). Group Formation in Mobile Computer Supported Collaborative Learning Contexts: A Systematic Literature Review. *Educational Technology & Society*, *19*(2), 258–273.

Ara, A., & Gopalakrishna, R. (2021). A Study on Emotion Identification from Music Lyrics. In *Lecture Notes on Data Engineering and Communications Technologies* (Vol. 72, pp. 396–406). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-3-030-70713-2_37

Ballard, M. E., Dodson, A. R., & Bazzini, D. G. (1999). Genre of music and lyrical content: Expectation effects. *Journal of Genetic Psychology*, *160*(4), 476–487. https://doi.org/10.1080/00221329909595560

Camacho-Collados, J., & Pilehvar, M. T. (2020). Embeddings in Natural Language Processing. *Proceedings of the 28th International Conference on Computational Linguistics: Tutorial Abstracts*, 10–15. https://doi.org/10.18653/v1/2020.coling-tutorials.2

Chen, B., Zhang, Z., Langrené, N., & Zhu, S. (2025). Unleashing the potential of prompt engineering for large language models. *Patterns*, *6*(6). https://doi.org/10.1016/j.patter.2025.101260

Chollet, F. (2021). *Deep Learning with Python*.

Clair, A. (2024). *What AI in music can — and can't — do*. https://www.vox.com/the-highlight/358201/how-does-ai-music-work-benefits-creativity-production-spotify

Crossley, S., Heintz, A., Choi, J. S., Batchelor, J., Karimi, M., & Malatinszky, A. (2023). A large-scaled corpus for assessing text readability. *Behavior Research Methods*, *55*(2), 491–507. https://doi.org/10.3758/s13428-022-01802-x

Devlin, J., Chang, M.-W., Lee, K., Google, K. T., & Language, A. I. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Proceedings of NAACL-HLT 2019, 4171–4186. https://doi.org/https://doi.org/10.48550/arXiv.1810.04805

Ekman, P. (1982). *Emotion in the Human Face* (2nd ed.). Cambridge University Press.

Fehr, B., & Russel, J. A. (1984). "Concept of Emotion viewed from a prototype perspective." *Journal of Experimental Psychology: General*, *113*(3), 464–486.

Gao, A. K. (2023). *Prompt Engineering for Large Language Models A brief guide with examples for non-technical readers*. https://ssrn.com/abstract=4504303

Gomez-Canon, J. S., Cano, E., Eerola, T., Herrera, P., Hu, X., Yang, Y. H., & Gomez, E. (2021). Music Emotion Recognition: Toward new, robust standards in personalized and context-sensitive applications. *IEEE Signal Processing Magazine*, *38*(6), 106–114. https://doi.org/10.1109/MSP.2021.3106232

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Hevner, K. (1936). Experimental Studies of the Elements of Expression in Music. *The American Journal of Psychology*, *48*(2), 246–268.

Iqbal, T., & Qureshi, S. (2022). The survey: Text generation models in deep learning. In *Journal of King Saud University - Computer and Information Sciences* (Vol. 34, Issue 6, pp. 2515–2528). King Saud bin Abdulaziz University. https://doi.org/10.1016/j.jksuci.2020.04.001

Jim, J. R., Talukder, M. A. R., Malakar, P., Kabir, M. M., Nur, K., & Mridha, M. F. (2024). Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review. *Natural Language Processing Journal*, *6*, 100059. https://doi.org/10.1016/j.nlp.2024.100059

Jurafsky, D., & Martin, J. H. (2025). *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models Third Edition draft Summary of Contents*.

Juslin, P. N., & Västfjäll, D. (2008). Emotional responses to music: The need to consider underlying mechanisms. *Behavioral and Brain Sciences*, *31*(5). https://doi.org/10.1017/S0140525X08005293

Lin, S., Hilton, J., & Evans, O. (2022). *TruthfulQA: Measuring How Models Mimic Human Falsehoods*. http://arxiv.org/abs/2109.07958

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. http://arxiv.org/abs/1907.11692

Malheiro, R. M. da S. (2016). *Emotion-based Analysis and Classification of Music Lyrics*. University of Coimbra.

Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. (2024). Prompt Engineering in Large Language Models. In *Lecture Notes in Computer Science* (Vol. 13633, pp. 387–402). https://doi.org/10.1007/978-981-99-7962-2_30

Mastropaolo, A., Scalabrino, S., Cooper, N., Palacio, D. N., Poshyvanyk, D., Oliveto, R., & Bavota, G. (2021). *Studying the Usage of Text-To-Text Transfer Transformer to Support Code-Related Tasks*. http://arxiv.org/abs/2102.02017

Mohammad, S. M. (2025). *NRC VAD Lexicon v2: Norms for Valence, Arousal, and Dominance for over 55k English Terms*. http://saifmohammad.com/WebPages/nrc-vad.html

Muminovich, A. R., & Istam kizi, N. N. (2025). *SYNTACTIC ANALYSIS IN COMPUTATIONAL LINGUISTICS: EXPLORING COLLOCATIONS AND VALENCY Nurmuxammedova Nurjaxon Istam kizi*. https://www.academicpublishers.org/journals/index.php/ijai

Nielsen, M. (2019). *Why are deep neural networks hard to train?*

Olatunji, S. (2024). *LANGUAGE, LEARNING, AND LYRICS: NLP APPLICATIONS IN SONGWRITING AND LYRICISM*. https://www.researchgate.net/publication/391060127

Oliveira, H. R. G., Cardoso, F. A., & Pereira, F. C. (2012). *Tra-la-Lyrics: An approach to generate text based on rhythm*. http://pdos.csail.mit.edu/scigen/rooter.pdf

Parthasarathy, V. B., Zafar, A., Khan, A., & Shahid, A. (2024). *The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities*. http://arxiv.org/abs/2408.13296

Patwardhan, N., Marrone, S., & Sansone, C. (2023). Transformers in the Real World: A Survey on NLP Applications. In *Information (Switzerland)* (Vol. 14, Issue 4). MDPI. https://doi.org/10.3390/info14040242

P.S., S., & Mahalakshmi, G. S. (2017). *Emotion Models: A Review*. https://www.researchgate.net/publication/319173333

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2023). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. http://arxiv.org/abs/1910.10683

Ram, N., Gummadi, T., Bhethanabotla, R., Savery, R. J., & Weinberg, G. (2021). Say What? Collaborative Pop Lyric Generation Using Multitask Transfer Learning. *HAI 2021 - Proceedings of the 9th International User Modeling, Adaptation and Personalization Human-Agent Interaction*, 165–173. https://doi.org/10.1145/3472307.3484175

Rodrigues, M. A. G., Oliveira, A. de P., Moreira, A., & Possi, M. de A. (2022). *Lyrics Generation supported by Pre-trained Models*. https://www.vagalume.com.br/

Roh, J., Oh, S.-H., & Lee, S.-Y. (2020). *Unigram-Normalized Perplexity as a Language Model Performance Measure with Different Vocabulary Sizes*. http://arxiv.org/abs/2011.13220

Rothman, D. (2024). *Transformers for Natural Language Processing and Computer Vision Third Edition Explore Generative AI and Large Language Models with Hugging Face, ChatGPT, GPT-4V, and DALL-E 3*.

Salloum, S. A., Khan, R., & Shaalan, K. (2020). A Survey of Semantic Analysis Approaches. *Advances in Intelligent Systems and Computing*, *1153 AISC*, 61–70. https://doi.org/10.1007/978-3-030-44289-7_6

Scheve, C. von, & Slaby, J. (2019). Emotion, emotion concept. In *Affective Societies* (pp. 42–51). Routledge. https://doi.org/10.4324/9781351039260-3

Seo, Y. S., & Huh, J. H. (2019). Automatic emotion-based music classification for supporting intelligent IoT applications. *Electronics (Switzerland)*, *8*(2). https://doi.org/10.3390/electronics8020164

Tay, Y., Tran, V. Q., Ruder, S., Gupta, J., Chung, H. W., Bahri, D., Qin, Z., Baumgartner, S., Yu, C., & Metzler, D. (2022). *Charformer: Fast Character Transformers via Gradient-based Subword Tokenization*. http://arxiv.org/abs/2106.12672

UWA. (2019). *T H E S C I E N C E O F E M O T I O N Exploring the Basics of Emotional Psychology*. https://online.uwa.edu/news/emotional-psychology/

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need*. http://arxiv.org/abs/1706.03762

Welch, G. F., Biasutti, M., MacRitchie, J., McPherson, G. E., & Himonides, E. (2020). Editorial: The Impact of Music on Human Development and Well-Being. In *Frontiers in Psychology* (Vol. 11). Frontiers Media S.A. https://doi.org/10.3389/fpsyg.2020.01246

Yenduri, G., Ramalingam, M., Selvi, G. C., Supriya, Y., Srivastava, G., Maddikunta, P. K. R., Raj, G. D., Jhaveri, R. H., Prabadevi, B., Wang, W., Vasilakos, A. V., & Gadekallu, T. R. (2024). GPT (Generative Pre-Trained Transformer) - A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions. *IEEE Access*, *12*, 54608–54649. https://doi.org/10.1109/ACCESS.2024.3389497

Zhou, Y. (2022). *Music Emotion Recognition on Lyrics Using Natural Language Processing*.

This page is intentionally left blank

Appendices

Table 22 – Prompts presented to each model.

Quadrant	Prompt
1	Write song lyrics from the 1st quadrant.
1	Write song lyrics from the 1st quadrant. The theme is about “sunny days”. Include the words "beach" and “party”. The structure of the lyrics should be 2 stanzas, each stanza with 5 verses.
1	Write song lyrics from the 1st quadrant. The theme is about “summer festival”. Include the words “dance” and “laughter”. The structure of the lyrics should be 3 stanzas, each stanza with 4 verses.
2	Write song lyrics from the 2nd quadrant.
2	Write song lyrics from the 2nd quadrant. The theme is about “a stormy night”. Include the words "thunder" and "shadows". The structure of the lyrics should be 2 stanzas, each stanza with 5 verses.
2	Write song lyrics from the 2nd quadrant. The theme is about “raging wildfire”. Include the words “flames” and “screams”. The structure of the lyrics should be 4 stanzas, each stanza with 3 verses.
3	Write song lyrics from the 3rd quadrant.
3	Write song lyrics from the 3rd quadrant. The theme is about “loneliness”. Include the words "silence" and "tears". The structure of the lyrics should be 2 stanzas, each stanza with 5 verses.
3	Write song lyrics from the 3rd quadrant. The theme is about “lost love”. Include the words “memories” and “rain”. The structure of the lyrics should be 3 stanzas, each stanza with 4 verses.
4	Write song lyrics from the 4th quadrant.
4	Write song lyrics from the 4th quadrant. The theme is about “peaceful morning”. Include the words "sunrise" and "breeze". The structure of the lyrics should be 2 stanzas, each stanza with 5 verses.
4	Write song lyrics from the 4th quadrant. The theme is about “quiet forest”. Include the words “morning mist” and “river”. The structure of the lyrics should be 4 stanzas, each stanza with 3 verses.