EMOTION DETECTION IN SONG LYRICS STANZAS

TEXT ANALYTICS - A.Y. 2024/2025

Group 2 - Barbieri, Bosco, Ferrara, Rotellini, Zizza

# Abstract

Analyzing the emotional tone of songs' texts can give insights into societal trends and this information can be useful especially for recommendation algorithms.

This study focuses on developing four Machine Learning models - two Static Models and two Neural Networks - to classify emotions conveyed in English song lyrics at the stanza level. These were chosen for their proven effectiveness across various domains and their diverse approaches, providing a thorough investigation of different techniques and depths for emotion classification in text.

# Introduction

Lyrics serve as one of the main foundations of songs, playing a crucial role in expressing feelings in many different ways. The emotional tone of songs can serve various purposes, such as automatized playlist creation or songs' organization, offering an alternative to the more traditional genre-based classification.

To obtain a deeper understanding of emotional fluctuations within the texts, the models assign emotion labels to individual stanzas instead of full songs. The emotion labels correspond to Robert Plutchik's eight primary emotions (shown in figure 1), providing a comprehensive range for representing various emotional states.


Figure 1: Plutchik's eight primary emotions

This report aims to clearly cover and illustrate various aspects of the work. The *Methods* chapter contains a detailed explanation of the data and procedures used in the project, providing descriptions of each part implemented in the project. The *Results* chapter presents an overview of the obtained outcomes. These results are further explored in the final sections, *Discussion* and *Conclusions*, which interpret the general findings, recap the primary objectives of the work, and discuss the importance or potential applications of the results.

# Methods

The dataset used in this project is a sampled subset of English-language songs derived from the *Genius Song Lyrics Dataset*[1]. The original dataset contained numerous attributes; the ones considered relevant for model training are:

- **title:** the song's title;
- **lemmatized_stanzas:** lyrics of the single stanza;
- **stanza_number:** identifies the position of the stanza in the song;
- **is_chorus:** boolean variable that attests whether the stanza is a chorus or not;
- **is_country, is_pop, is_rap, is_rb, is_rock:** boolean variables, result of a one-hot encoding process, that represent songs genres;
- **label:** represents the emotional classification of the stanza, assigned by Albert Base v2[2] model.

All of these attributes, except for `title`, were the result of the preprocessing phase, as described in section . Due to limited computational power, the labeling process was time-intensive, ultimately resulting in a limited dataset, with a few more than 100.000 entries.

## Preprocessing

The initial preprocessing step involved sampling from the original dataset while maintaining the proportional distribution of genres. This approach ensured that the genre representation in the sampled subset accurately reflected that of the full dataset.
The preliminary text cleaning process focused on the `lyrics` attribute, which contained the complete lyrics of each song in string format. Initially, a regular expression (RegEx) was built to remove noise from the lyrics, specifically targeting words enclosed in square brackets that were irrelevant to the stanza splitting process. Many keywords marking different stanzas were written within square brackets, and removing non-keyword items inside brackets was crucial to avoid potential issues.
The next critical step was stanza splitting. After cleaning texts from noisy square-bracketed items, lyrics were split based on various keywords used to denote stanzas (such as ”chorus”, ”verse”, ”intro”, ”outro”, ”refrain”, ”hook”, etc.). The RegEx developed accounted for the different formats in which these keywords appeared, including square brackets, parentheses, or no brackets at all, as well as stanzas separated only by double newline characters. The output of this step was, for each song record, a list of strings representing individual stanzas (each stanza has also a header with the corresponding keyword; this aspect will be discussed in the next paragraph). Next, uninformative strings—such as empty strings or those with fewer than 20 characters—were removed, as they were too short to provide meaningful content. As a result, the output of this preliminary preprocessing phase was a dataset where the records were no longer whole songs but individual stanzas, each numbered according to its position within the song.
A further and more detailed cleaning process on the stanzas involved the creation of the boolean feature `is_chorus`, which was assigned a `true` value for repeated stanzas within the same song or for stanzas with headers such as ”hook”, ”chorus”, ”refrain”, or ”bridge”. Next, stanza headers and newline characters between verses were removed to obtain cleaner stanzas. Since choruses,

hooks, bridges, and refrains often repeat throughout songs, duplicate stanzas were discarded to avoid redundant data. This resulted in a dataset of cleaned, non-duplicate stanzas, which served as the checkpoint for the labeling step and the starting point for the text lemmatization process.

The subsequent step involved lemmatizing the stanzas using the `spaCy` library. A list of lemmatized tokens was created by filtering out punctuation and empty words. Lemmatization was chosen over stemming because it produces more accurate and meaningful results, particularly for tasks requiring semantic understanding, such as the one at hand.

Since the dataset was not pre-labeled at the stanza level, ALBERT Base v2 was employed for this task. This transformer model is specifically designed to be fine-tuned on tasks that require an understanding of the entire sentence, such as sequence classification.

## Models

The selected model architectures are:

- **Random Forest**: A robust ensemble learning method known for its ability to handle complex, high-dimensional datasets effectively.
- **Support Vector Machine (SVM)**: A powerful classifier that excels in separating classes by finding the optimal hyperplane, particularly effective in text classification tasks.
- **One-Dimensional Convolutional Neural Network (1D-CNN)**: Designed to capture local patterns in sequential data, leveraging convolutional layers to learn hierarchical features.
- **Recurrent Neural Network (RNN)**: Utilized for its strength in processing sequential data, with the ability to capture contextual relationships between words across different stanzas.

Their different approaches and depths are an important point of the study, as they offer interesting insights into the possible different techniques and levels of complexity required for detecting emotional tones in complex pieces of text.

## Static Models

The development of static models was then simple and straight forward. Static models are here meant to provide a performance comparison for the more complex Neural Networks. The two architectures are the same, consisting of a preprocessing layer to handle the inputs, followed by the classifier itself.

The preprocessing layers managed two key textual inputs, `title` and `lemmatized stanzas`, using Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction. An additional analysis aimed at identifying the most significant features for each emotion label in the dataset was conducted to enhance classification.

The feature importance analysis was performed on the already labeled and lemmatized dataset, following these steps:

Custom Stopword List Creation: A custom stopword list was compiled, consisting of the most frequent and generic words in the dataset, along with additional punctuation marks and common typographical errors not covered by the default NLTK stopword lists.

Stopword Removal: Irrelevant words identified by the custom stopword list were removed from the lemmatized stanzas to reduce noise in the data.

TF-IDF Analysis per Label: A function was developed to compute TF-IDF scores for a given text, with parameters `min_df` set to 2 and `max_df` set to 0.80. This configuration ensured that words appearing in fewer than two documents or in more than 80% of the documents were ignored, minimizing the influence of extremely rare or overly common words. The function was applied separately to the cleaned stanzas for each label, with the aim of identifying the most relevant features per emotion category.

The results, however, did not meet expectations, though the outcome was not entirely surprising. Most labels shared at least two common features, and certain labels (such as surprise and trust) shared all features. Additionally, all identified features exhibited very low TF-IDF scores, below 0.05. This result appears to be inherent to the nature of the dataset: song lyrics frequently contain repetitive and generic language, making it difficult to distinguish specific emotions based solely on textual features. Consequently, the analysis was concluded at this point.

The preprocessing layers handle both `title` and `lemmatized_stanzas` through TF-IDF. To help with classification, additional features displaying the five most frequent words for each class was added. This was also obtained through TF-IDF. A word was considered informative for a class if it appeared at least twice and in up to 80% of the stanzas associated with that class. These features quantify the number of times that certain word appears for each class. The boolean attributes are converted to integers, while `stanza_number` is scaled.

Random Search was chosen for hyperparameter tuning, for both models. Cross validation is also used in order to provide a more accurate estimate of model performance.

## Neural Networks

The architectures were developed and tuned through empirical, reiterated testing. These parameters helped with the process, and can be used for further experimentation. Both the Recurrent Neural Network and the One-Dimensional Convolutional Neural Network share the same preprocessing architecture. Most attributes are processed in the same manner as in the Static Models; specific steps are applied to `lemmatized_stanzas` and `title`. Non-Negative Matrix Factorization is applied to `title` in addition to Term Frequency-Inverse Document Frequency to extract latent topics, providing a richer representation of the textual data.

`lemmatized_stanzas` are handled by Convolutional and Recurrent pipelines of the two Networks. Elements are first tokenized, and then padded in order to get an input with consistent shape, which is essential for both types of recurrent layers.

### One-Dimensional Convolutional Neural Network

The Convolutional part of the architecture is designed to extract and learn local patterns in `embedding_lyrics`. Its structure consists of three convolutional layers, each applying filters of varying sizes. This allows to detect patterns at different granularities. These layers are followed by Global Max Pooling, to reduce the previous output's dimension to a fixed-length vector, as well as retaining focus on the most informative patterns. A dropout layer is then applied, to introduce regularization and prevent overfitting.

### Recurrent Neural Network

The Recurrent part of the architecture is designed to extract and learn local patterns in `embedding_lyrics`. Its structure consists of three Gated Recurrent Units (`GRU` layers) to model temporal relationships. These are characterized by progressively smaller numbers of units; this allows pattern capture at different abstraction levels. All three layers in the architecture use the `tanh` activation function to compute the hidden state and the `sigmoid` activation function for the recurrent gate. The first and second layers return the full sequence of hidden states for each time step in the input sequence, enabling richer learning of patterns over time. Dropout is applied on every layer, to prevent overfitting and add regularization.

## Shared Components

The remaining features are handled by a simple pipeline, which concatenates their Input layers. The inputs are passed them through a dense layer to create a compact representation. The output of the lyrics-processing branch is concatenated with the processed additional features. The combined representation is passed through a Dense layer with 32 units (ReLU activation), followed by a Dropout layer with a rate of 0.3. Finally, the output layer uses 8 units with a softmax activation, corresponding to the classification into 8 emotion categories.

The models are trained using categorical cross-entropy as the loss function, as it consistently produced better-performing results for both types of neural networks. As evaluation metric, the better performing one was categorical accuracy. Other metrics were tested for evaluation purposes, particularly top-k categorical accuracy with $k = 2$, which yielded interesting insights by considering a prediction correct if the true label is among the top two predicted classes. While it showed potential in improving generalization and preventing overfitting, it was ultimately discarded as a primary evaluation metric. This is because, with $k = 2$, the model has a 25% baseline chance of being correct when there are eight possible classes, which reduces the precision required for accurate learning and limits performance improvements.

# Results

The performance metrics taken into account for evaluation are the following: accuracy, precision, recall, and F1-score. Taking all of them into account is crucial to accurately evaluate how well each model performs.
For the static models implemented in this project, the classification report revealed an accuracy of 34% for the Random Forest algorithm and 43% for SVM. These results can be considered reasonable, given that the task at hand is a multi-class classification problem with 8 classes.
**AGGIUNGERE ALTRO**

# Discussion

# Conclusions

**AGGIUNGERE INFO SPECIFICHE AL PROGETTO**
In conclusion, this study aimed at demonstrating how emotion detection in song lyrics stanzas can provide valuable insights into the emotional landscape of music and how it can be implemented with Machine Learning models. These findings have practical applications, such as improving music recommendation systems and creating mood-based playlists. At the same time, the study faced challenges, particularly with interpreting ambiguous or context-dependent lyrics, which highlights opportunities for further research in this field.

## Future developments

# Bibliography

[1] *Genius Song Lyrics*. URL: https://www.kaggle.com/datasets/carlosgdcj/genius-song-lyrics-with-language-information?select=song_lyrics.csv.

[2] *Albert Base v2*. URL: https://huggingface.co/albert/albert-base-v2.

# List of figures