Abstract

This study focuses on developing four Machine Learning models to classify emotions conveyed in English song lyrics at the stanza level. The classification leverages Plutchik's framework of eight primary emotions, offering a nuanced understanding of emotional expression in lyrical content.
The selected model architectures are as follows:

- **Random Forest**
- **Support Vector Machine (SVM)**
- **One-Dimensional Convolutional Neural Network (1D-CNN)**
- **Recurrent Neural Network (RNN)**

These models were chosen for their proven effectiveness across various domains and their diverse approaches, providing a thorough investigation of different techniques for emotion classification in text.

DA FINIRE e far stare nella stessa pagina con l'introduzione

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.

The goal of this project is the development of 4 Machine Learning models that perform emotion detection on song lyrics stanzas. 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 are assigned based on Robert Plutchik's eight primary emotions (shown in figure 1), offering a comprehensive range for representing diverse emotional states.

Figure 1: Plutchik's eight primary emotions

This report tries to cover and illustrate clearly various aspects of this work. In the *Method* section, we will provide a detailed explanation of the data and procedures used in the project, in particular describing the pipeline taken to implement the models. Then, the *Results* chapter will provide an overview of the obtained results with the aid of plots and figures, highlighting the significant outcomes. This section will be connected to the last two, i.e. the *Discussion* and *Conclusions* ones, which will explain what the general findings mean, recapping the primary objective of the work and discussing the importance or potential applications of the results.

1. Methods

The dataset used in this project represents 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;
- **tag:** represents the genre of the song. For easier handling, this attribute of the original dataset has been one-hot encoded into various boolean variables (is_country, is_pop, is_rap, is_rb, is_rock);
- **label:** represents the emotional classification of the stanza, assigned by Albert Base v2[2] model.

All of these attributes, except for the `title` one, were the result of the preprocessing phase, as will be described later in section 1.1
Due to limited computational power, the labeling process was time-intensive, ultimately resulting in a limited dataset consisting of (**QUANTE? AGGIUNGEREI NUMERO STROFE**).

1.1 Preprocessing

The first step in the preprocessing phase of this dataset involved sampling from the original dataset while preserving the proportions of the different genres. This ensured that the genre distribution in the subset remained representative 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 the strings of noisy square-bracketed items, the 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 by double newline characters. The output of this step was, for each song record, a list of strings representing individual stanzas, with each stanza labeled by its corresponding header keyword. Subsequently, 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 led to 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.

For the labeling step, Albert Base v2[2] was employed. This transformer model is specifically designed to be fine-tuned on tasks that require an understanding of the entire sentence, such as sequence classification.

2. Static Models

2.1 Random Forest

2.2 SVM

3. Neural Networks

3.1 One-Dimensional Convolutional Neural Network

3.2 Recurrent Neural Network

Key findings and conclusions

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