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 to classify emotions conveyed in English song lyrics at the stanza level. The classification employs Plutchik's eight primary emotions, offering a nuanced understanding of emotional expression in lyrical content. The selected model architectures are:

- **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 CON CONCLUSIONI

Introduction

Songs have the unique ability to engage people in ways that few other mediums can match. 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 songs' 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 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* section provides a detailed explanation of the data and procedures used in the project, with a particular focus on the pipeline implemented to construct the models. Following this, the *Results* chapter presents an overview of the obtained outcomes, using plots and figures to highlight significant findings. 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 consisting of (**QUANTE? AGGIUNGEREI NUMERO STROFE**).

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.

Static Models

Firstly, it is important to illustrate the feature creation step applied in order to enhance model performance. This step involved generating five features for each emotion class using Term Frequency-Inverse Document Frequency (TF-IDF), a statistical measure that evaluates how informative and important a word is within a document or class. In this project, TF-IDF was applied to identify the five most informative words for characterizing each emotion. The `min_df` and `max_df` parameters were employed during the TF-IDF computation to establish minimum and maximum thresholds. Specifically, a word was considered informative for a class if it appeared at least twice and in up to the 80% of the stanzas associated with that class. These features quantify the number of times that

certain word appears for each class. This step was incorporated into the project's pipeline because the initial model training, that will be illustrated in the following sections, revealed suboptimal classification performance. The addition of these features led to a slight improvement in the results.**E' EFFETTIVAMENTE VERO?**

The development of static models was then simple and straight forward. Static models were mainly developed 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 handle both `title` and `lemmatized_stanzas` through TF-IDF. 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 Neural Network models can be trained with the `neural_networks.py` script. This script offers the possibility of configuring certain parameters for training. The architectures were developed and tuned through empirical, reiterated testing.

Both the Recurrent and One-Dimensional Convolutional Neural Network have the same non recurrent architecture. Except for `lemmatized_stanzas` and `title`, other attributes are preprocessed as in Static Models' preprocessing.

Term Frequency-Inverse Document Frequency is applied to `title`, followed by Non-Negative Matrix Factorization, to perform topic modeling.

`lemmatized_stanzas` is first tokenized, and then padded in order to get an input with consistent shape.

The script offers the option of downsampling classes in order to get uniform label representation.

cnn architecture

rnn architecture

optional semisupervised learning, reset

graphs

Evaluation

The performances of the models were evaluated using the `classification_report` function from the `scikit-learn` library. This function is particularly useful as it offers an overview of key evaluation metrics commonly used in Machine Learning, i.e. accuracy, precision, recall, and F1-score. 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 and 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.

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