EMOTION DETECTION IN SONG LYRICS STANZAS

TEXT ANALYTICS - A.Y. 2024/2025

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

Abstract

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 also serve various purposes, such as automatized playlist creation or songs' organization, offering an alternative to the more traditional genre-based classification.

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. The models 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.

The models were trained through transfer learning, by creating the ground truth via an existing model. This proved to be a challenge, because of computational complexity; additionally, the generated labels are imprecise and overlapping, generating **DA FINIRE**

Introduction

This study focuses on capturing emotional dynamics in song lyrics by assigning emotion labels to individual stanzas rather than entire songs. This approach allows for a more granular analysis of emotional shifts within the text. 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

The goal of this report is to provide a comprehensive overview of the project, detailing its methodology, results, and key insights. 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].

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 preprocessing phase began by sampling the dataset while maintaining genre distribution. Text cleaning focused on removing irrelevant noise, such as square-bracketed items, and splitting lyrics into individual stanzas using stanza-related keywords (e.g., ”chorus”, ”verse”). Uninformative stanzas, such as empty or very short ones, were discarded, resulting in a dataset of numbered stanzas. A boolean feature, `is_chorus`, was added to mark repeated or chorus-related stanzas, and duplicate stanzas were removed to eliminate redundancy. Further cleaning involved removing stanza headers and newline characters, producing cleaner stanzas for labeling. Lemmatization was performed using the `spaCy` library, generating tokenized stanzas by filtering out punctuation. ALBERT Base v2 was then fine-tuned to label approximately 100,000 stanzas with emotional categories. A preliminary class distribution analysis showed a slight imbalance, with *joy* being the most frequent (18%) and *disgust* the least (10%). Figure 2 illustrates the distribution across all classes.

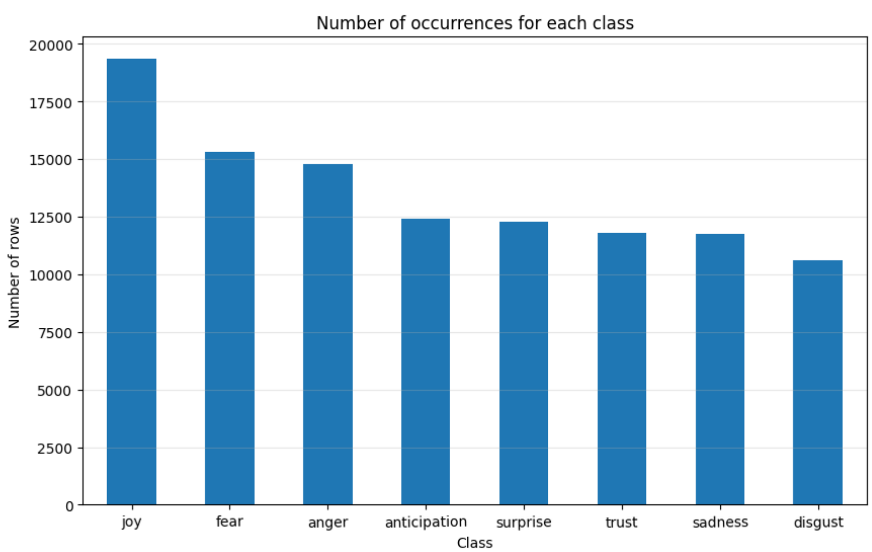

Figure 2: Number of stanzas for each label

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 aimed at providing a performance benchmark for more complex neural networks. Both architectures consisted of a preprocessing layer, followed by a classifier. The preprocessing handled title and `lemmatized_stanzas` using TF-IDF for feature extraction.

To improve classification, feature importance analysis was conducted on the labeled dataset:

- **Custom Stopword List Creation**: A custom list of frequent and generic words, punctuation, and common typographical errors was compiled.
- **Stopword Removal**: Irrelevant words were removed to reduce data noise.
- **TF-IDF Analysis per Label**: TF-IDF scores were computed for each emotion label, with parameters set to minimize the influence of overly common or rare words.

Despite these efforts, the analysis did not yield expected results, with most labels sharing common features and low TF-IDF scores. This was likely due to the repetitive and generic nature of song lyrics.

For hyperparameter tuning, Random Search and cross-validation were used to estimate model performance more accurately.

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 specifically 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 specifically 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.

In addition to standard supervised learning, the script supports semi-supervised learning. It can be utilized either strictly for transfer learning or in a data augmentation approach. When using the data augmentation approach, the model's weights can be reset prior to training on the newly pseudo-labeled data, allowing for more robust retraining on an expanded dataset. The results obtained

Results

The performance metrics considered for evaluation in these analysis are the following: accuracy, precision, recall, and F1-score. The latter one is the harmonic mean of precision and recall and was therefore chosen for further discussion in this section. Considering all of these metrics is crucial to accurately evaluate how well each model performs. Regarding 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.

For the Random Forest model, the class with the highest F1-score is `anger`, which is the third class as for support, sitting at 4436. `Joy` was the most supported class, with 5854 instances, but it came second as for the F1-score, which was of 0.40. The second class based off support, which was `fear` with 4652 instances, had a lower F1-score of 0.31. The remaining classes showed comparable support and F1-scores, averaging around 3500 instances and 0.25 respectively, only `disgust` had a considerably lower F1-score, at 18%.
In image 3, the ratio of true positives versus false positives is displayed, with the Random Forest classifier scoring lower than a random guesser in two out of eight classes.

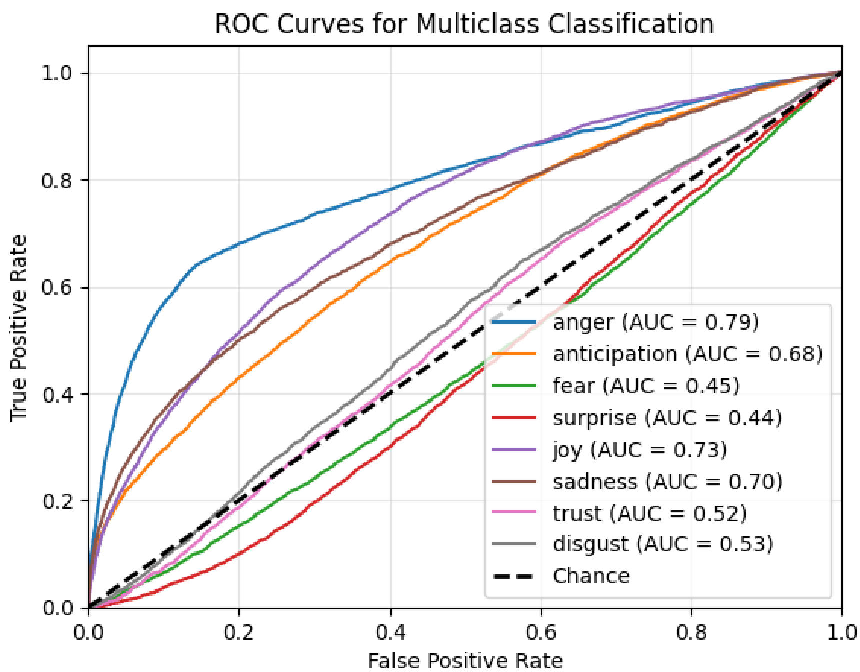

Figure 3: ROC Curve for the Random Forest Classifier

Image **DA AGGIUNGERE IMMAGINE** shows class-wise accuracy.

DA AGGIUNGERE SVM

Neural Networks' performances were generally not on par with the ones obtained by the static models. As mentioned in the previous chapter, the development of neural networks iterated testing phases and adjustments over different configurations for data splitting, preprocessing, architectures and training parameters were tested. Because of the generally poor results, semi-supervised learning did not get important results; downsampling into evenly represented labels gave some minor improvements, for both architectures.

The graphs below show various performance metrics of the best network for the convolutional architecture.

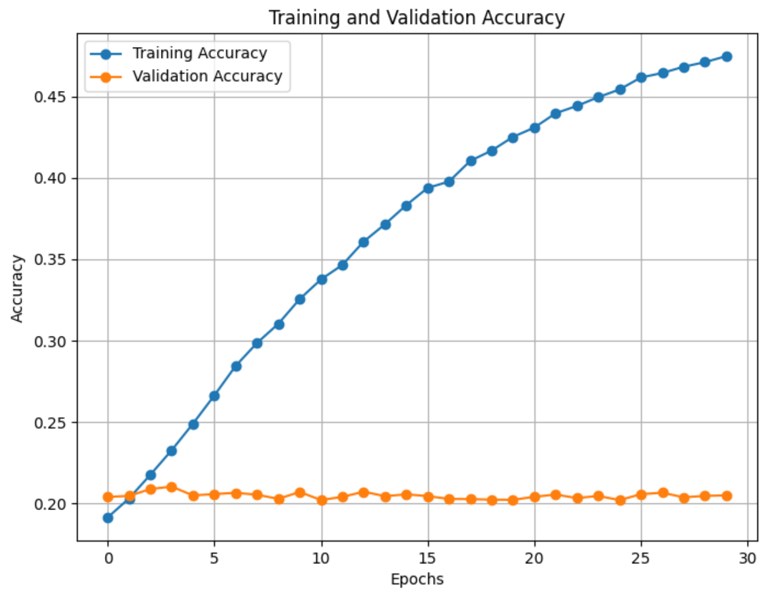

Figure 4: Convolutional Neural Network's training and validation accuracy

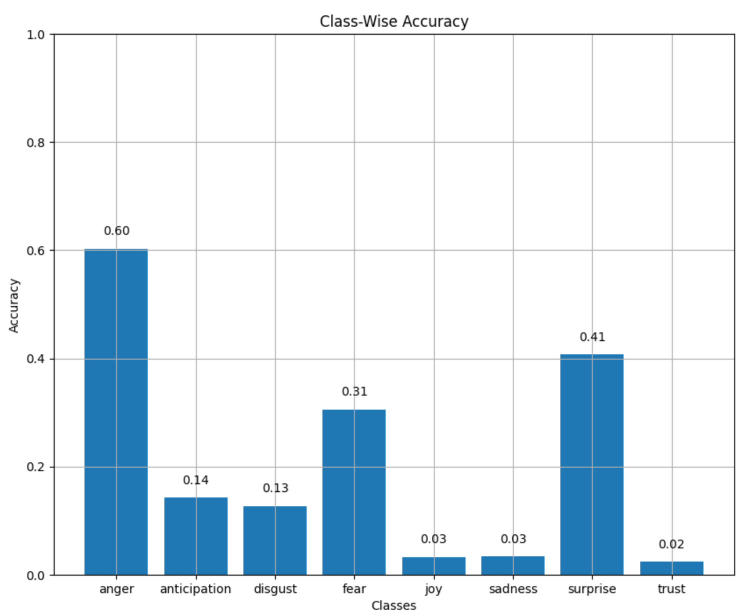

Figure 5: Convolutional Neural Network's class-wide accuracy

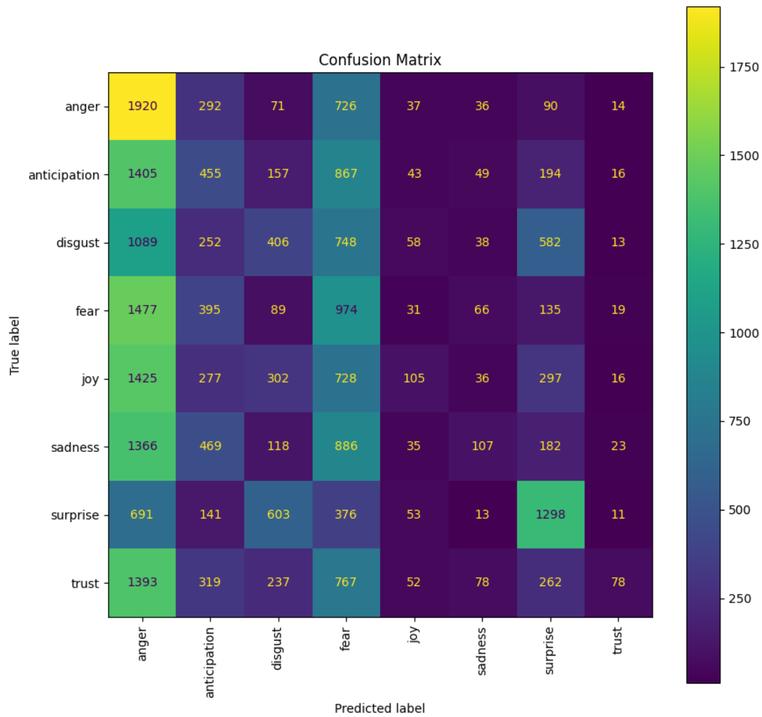

Figure 6: Convolutional Neural Network's confusion matrix

Discussion

The results presented in the previous chapter show general uncertainty in the models. This has a few possible explanations: one of the most likely is general overlapping of features. One of the most likely explanations is the significant overlap of features.

There are clear indications of this issue in the performance of neural networks, particularly the recurrent architecture. This is evident during training, where significant improvements in training accuracy occur only in later epochs, as shown in the training and validation accuracy plot (figure 4): once training accuracy begins to rise meaningfully, the model quickly overfits. Additional insights are provided by the confusion matrix and class-wise accuracy plot (figures 5, 6), which consistently show a dominant class with a high number of predictions and a secondary class with relatively high accuracy, while the remaining classes receive significantly fewer predictions overall.

The convolutional architecture exhibits similar behavior, albeit less prominently (figures [...]). In this case, training accuracy improves earlier in the epochs, but with minimal corresponding gains in validation accuracy. The confusion matrix and class-wise accuracy plot reveal analogous patterns, though to a lesser extent.

There are a few different solutions to solve this problem: by applying a filter to exclude the most common words, it's possible that the remaining words carry less ambiguous emotional meanings. Another unexplored approach is the use of custom metrics during training, designed to prioritize class-wise accuracy rather than overall model accuracy.

Another problem is the general fallacy of the generated ground truth. This can also explain general poor The **left section** of the graph in Figure 7 displays the predicted probabilities for each class. In the **center section** feature importances are ranked from most to least relevant and divided into two groups: on the right features with a positive influence on the predicted label; on the left, those with a negative influence that suggest the model should consider other classes. The **right section** of the graph highlights the values of the most important features, using bright colors to indicate features with a positive influence on the prediction.

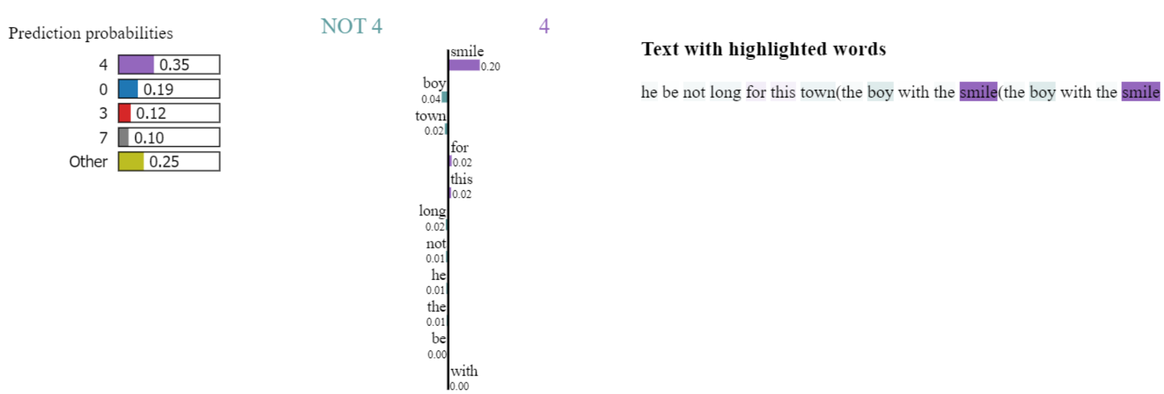

Figure 7: Explainability - visualization

The figure above illustrates a prediction where the model assigned the label *joy* to the stanza under analysis, but the correct label, assigned by the ALBERT model (as mentioned in the *Methods* section), was *sadness*. However, the word *smile*, which is brighty highlighted, intuitively suggests that *joy* might be a more plausible class for this stanza, even one that ALBERT could reasonably assign. This observation raises a critical issue: the transfer learning approach used to create the ground truth appears to have some limitations; in some instances the SVC model assigns a label that seems more contextually appropriate for the stanza, yet it differs from the supposedly correct label provided by ALBERT.

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