Genius Song Lyrics:
Topic Modeling and Text Classification

Text Mining and Search Project

Mattia Simone 901716
Montalbano Daniel 897383

UNIVERSITA' DEGLI STUDI DI MILANO
BICOCCA

Contents

1 Introduction

Music has always held a profound place in human culture, serving as a powerful medium of expression, emotion, and connection. This project delves into the world of music, exploiting the potential of text mining techniques to extract meaningful information from the song lyrics. By integrating topic modelling and text classification methods, the primary goal is to build a predictive model capable of determining genre, identifying relevant topics and suggesting similar songs based solely on the textual content of a given song.

2 Dataset

2.1 Genius Song Lyrics

We acquired on Kaggle[1] a comprehensive song lyrics dataset provided by Genius [2], a platform renowned for its community-driven efforts to upload, annotate, and discuss a wide range of creative work, with a predominant focus on songs.

The **5 million song lyrics** provided by Genius are formatted in a way that requires pre-processing. Key song metadata is often enclosed in square brackets, embedded in the middle of the lyrics. Despite this structuring, the overall composition of the lyrics remains intact. As a result, each entry is likely to contain numerous newline characters, which poses problems during data reading and model integration. Similar attention is required for other columns, such as features, to ensure adequate preparation for subsequent analysis.

Column	**Meaning**
title	Title of the piece
tag	Genre of the piece
artist	Person or group the piece is attributed to
year	Release year
views	Number of page views
features	Other artists that contributed
lyrics	Lyrics
id	Genius identifier
language_cld3	Lyrics language according to CLD3
language_ft	Lyrics language according to FastText's langid
language	Combines language_cld3 and language_ft

Table 1: Description of Columns in the Dataset

2.2 Data Loading

Due to the substantial size of the Genius dataset, we encountered challenges related to its high dimensionality. As a solution, we implemented a loading strategy by splitting the dataset into chunks of size 1,000,000 entries. We loaded and filtered each chunk based on decade and language, and then merged the filtered chunks into a single dataset. To work with a manageable number of documents, we limited the analysis to Italian songs.

2.3 Text processing

Text processing is a crucial step in the text mining pipeline, involving various tasks aimed at enhancing the quality and relevance of textual data for further analysis. In the context of the project, a comprehensive function has been employed to preprocess the input text. The function encompasses several essential tasks, each serving a specific purpose in refining and preparing the text data for subsequent text mining procedures.

Text preprocessing steps:

- *Lowercasing*: this standardization ensures consistency in the representation of words, preventing the model from treating words with different cases as distinct entities.
- *Removing text in square brackets*: this step is particularly useful for discarding meta-information that may not contribute significantly to the overall content, such as *[intro]*.
- *Removing [unctuation and numbers*: this aids in simplifying the language representation and avoiding discrepancies that may arise due to the presence of unnecessary symbols and numbers.
- *Removing extra whitespace*: excessive whitespace is reduced to a single space, contributing to a cleaner and more standardized text representation.
- *Limiting character repetition*: consecutive character repetitions exceeding two are limited to two repetitions, this step aids in handling elongated words or sequences of characters that may not convey additional information.
- *Removing stopwords*: commonly used stopwords, which typically do not contribute significant meaning to the text, are eliminated.
- *Tokenization*: this step breaks down the text into meaningful units, facilitating further analysis at the word level.
- *Lemmatization*: words are lemmatized to reduce inflected forms to their base or dictionary forms, this step helps in standardizing words and grouping variations of a word together.
- *Filtering short tokens*: tokens with a length of less than four characters are removed, this minimizes the inclusion of very short or irrelevant terms, enhancing the quality of the processed text data.

Having obtained our consolidated dataset, we proceeded to split it into training and testing subsets.

2.4 Exploratory analysis

After preprocessing we ended up with 118387 rows, splitted between train (94709) and test (23678) dataset. The dataset is composed by 11 columns:

- title

- tag
- artist
- year
- views
- features
- lyrics
- language
- decade
- lyrics_processed

Now we create some visualization to better understand the distributions of the most important features of the dataset.

2.4.1 Words per song distribution

As we can see in the figure 1, the number of words per text can vary widely. On average, the length is 120 words.

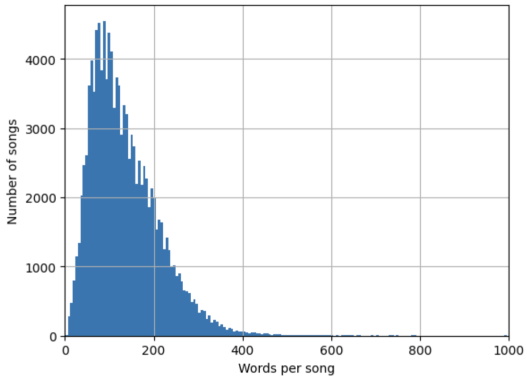

Figure 1: Words per song distribution

2.4.2 Songs per Year distribution

Otherwise the songs are in a range between 1954 and 2020, the figure 2 show that the songs are major distributed in the last decades, so this factor could influence the topic distribution or the terms used in the songs.

2.4.3 Tag's songs distribution

The figure 3 show the tag used in the songs: the mostly used tags are rap and hip-hop. So we can say that for our classification task, a new song will be probably classified as pop or rap because of this class imbalance respect to rock, country and rb. Note that misc is for songs that not represent very well a tag.

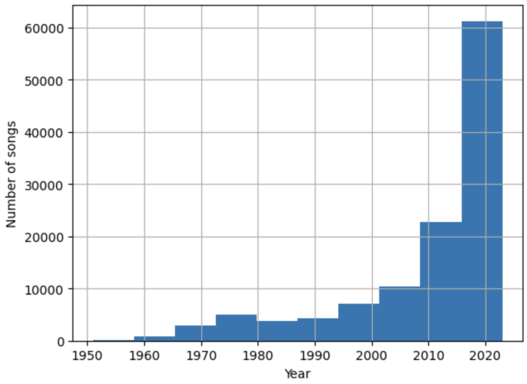

Figure 2: Songs per Year distribution

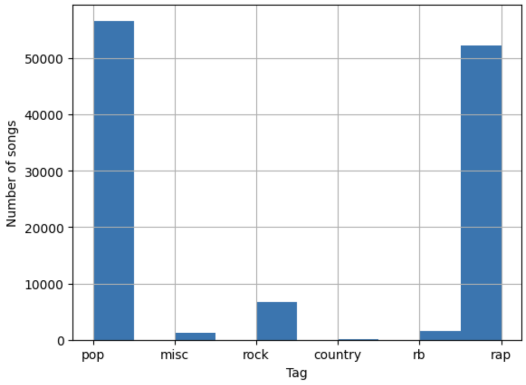

Figure 3: Tag's songs Distribution

3 Topic Modeling

Topic modeling is a powerful technique in natural language processing that allows us to extract meaningful themes or topics from a collection of textual data. In this project, we delve into the implementation and utilization of topic modeling using Latent Dirichlet Allocation (LDA) on a dataset of processed lyrics.

In our code we generate of a Gensim dictionary and corpus based on the preprocessed lyrics from the training dataset. Then, a Term Frequency-Inverse Document Frequency (TF-IDF) model is applied to the corpus to define the importance of words. An LDA model is then trained on the TF-IDF weighted corpus, and the resulting model, along with associated files, is saved for future use.

Finally we iterated through the topics generated by the trained LDA model and prints the top 5 words associated with each topic. The result is a summary of the most representative words for each identified topic.

Topic	Top 5 Words
0	ogni, essere, parole, verità, realtà
1	dentro, sangue, fuoco, nero, morte
2	balla, corri, lontano, volo, ballo
3	solo, quando, sempre, volta, ancora
4	meglio, troppo, molto, ragazza, amico
5	senza, mondo, tempo, vita, mare
6	così, nessun, zucchero, pizza, l'america
7	vieni, andiamo, macchina, long, plastica
8	solo, fare, quando, testa, fuori
9	notte, sole, luna, fuori, quando
10	maria, canto, onda, nata, gesù
11	voglio, mani, morire, natale, muoio
12	ciao, serum, stasera, bere, bevo
13	gente, mondo, anni, nessuno, stato
14	flow, rime, beat, rapper, disco
15	torna, fammi, milano, indietro, fallo
16	comme, cchiù, tengo, pecché, lalala
17	mamma, felice, volare, quaggiù, coro
18	perchè, vivo, dammi, morirò, finchè
19	terra, uomo, guerra, nome, vedrai
20	noia, povero, aiuto, calma, gioia
21	fumo, gang, giro, strada, dentro
22	niente, bella, canzone, portami, frega
23	bene, male, stare, piace, insieme
24	scusa, cento, bomba, eheh, cambierò
25	baby, love, fuck, like, black
26	vuoi, cosa, puoi, dimmi, fare
27	musica, gira, nuovo, festa, suona
28	amore, cuore, lamore, donna, sempre
29	soldi, vuole, bitch, baby, chiama

Table 2: Most relant words for each topic

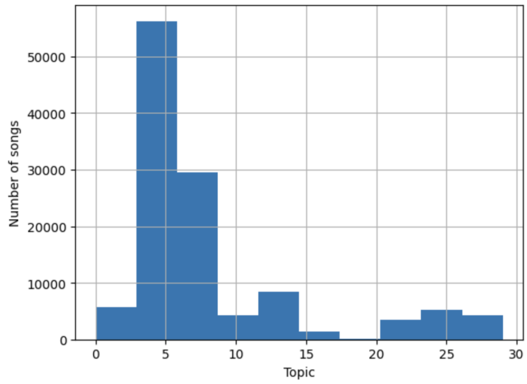

Figure 4: Topic Distribution

Additionally, if the *SHOW_GUI* flag is set to True, an interactive graphical user interface (GUI) using pyLDAvis is displayed. This visualization tool enhances the understanding of topic distribution and relationships, offering a dynamic representation of the identified topics.

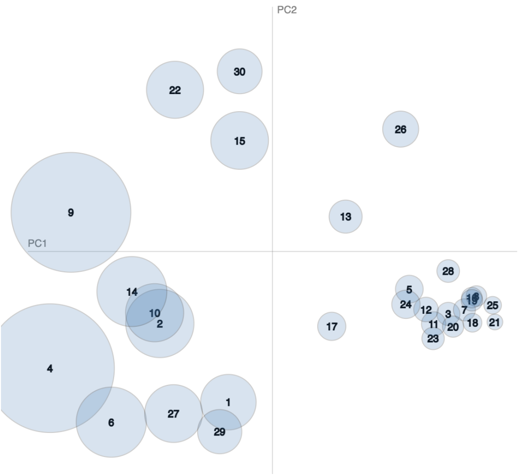

Figure 5: Intertopic Distance Map

At the end, we performed PCA to visualize the most frequent topic for each year:

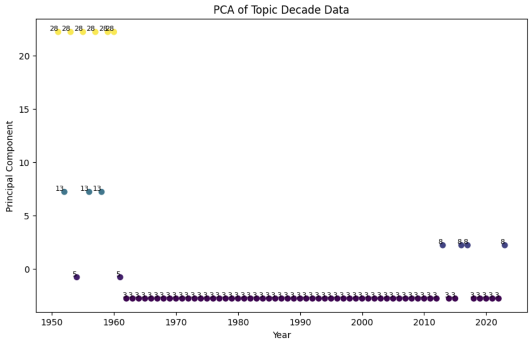

Figure 6: Topic for year

3.1 Latent Dirichlet allocation (LDA)

Latent Dirichlet Allocation (LDA) is a key model in language processing and machine learning. It works by seeing documents as mixes of topics, and topics as mixes of words. Picture it like a recipe: each document is like a blend of different topics, and each topic is made up of various words. LDA cleverly figures out these mixtures by refining them again and again, helping uncover the main themes in a bunch of text. Its flexibility and ability for spotting hidden patterns make LDA super useful for tasks like grouping documents, figuring out main topics, and analyzing feelings in texts.

In our implementation of Latent Dirichlet Allocation (LDA), we initiate the process by creating a Gensim dictionary and corpus. If either the $FORCE_TOPIC_MODELING$ flag is set to True or the LDA model file doesn't exist, we proceed with constructing the dictionary and corpus. The lyrics from the training dataset, once processed, serve as the basis for this dictionary and corpus generation. Subsequently, a Term Frequency-Inverse Document Frequency (TF-IDF) model is applied to the corpus, enhancing the importance of words based on their frequency across documents.

The LDA model is then trained on this TF-IDF weighted corpus, configured with the specified number of topics (NUM_TOPICS) and passes (PASSES). If the LDA model is newly created, it is saved for future use. Additionally, the TF-IDF weighted corpus and dictionary are stored as files for efficient retrieval. On subsequent runs, or if the LDA model file exists, we load the pre-existing dictionary, corpus, and LDA model.

3.2 Evaluation

Topic modeling evalutaion metrics:

h **Coherence:**

- The coherence is assessed using the c_v coherence score, which stands at **0,405**. This score suggests a moderate level of interpretability and meaningfulness in the topics generated by the LDA model. A higher coherence score generally indicates more coherent and distinct topics. In this context, the LDA model demonstrates a reasonable ability to generate topics that are consistent, clear, and relevant.

h **Perplexity:**

- The perplexity score is measured at $-12,066$, reflecting how well the LDA model predicts the test dataset. A lower perplexity score is desirable, and this value suggests a relatively good predictive performance of the LDA model on the test dataset. The negative sign does not affect the interpretation, as the focus is on the magnitude of the score, where a lower value indicates a more confident and accurate predictive capability.

h **Diversity:**

- The diversity score is determined to be **0,032**, indicating a low to moderate level of diversity among topics across different songs in the dataset. This metric measures the dissimilarity of topics, and a higher diversity score would suggest a broader range of topics present in the dataset. In this context, the LDA model exhibits a degree of diversity, but the score suggests room for improvement in achieving a more varied set of topics across the dataset.

4 Text Classification

Text classification a task where the goal is to categorize or label text documents into predefined classes or categories: in this case, the task is to classify song lyrics into music genres.

4.1 TF-IDF Vectors

Earlier attempts involved employing multiple machine learning models with TF-IDF vectors as features: TF-IDF, or term frequency-inverse document frequency, is a numerical statistic that reflects the importance of a term within a document relative to a collection of documents (corpus). It is calculated by multiplying the term frequency (TF) in a document by the inverse document frequency (IDF) of the term across the corpus.

We trained three models with TF-IDF vectors as features and evaluated the accuracy on the test set:

- Logistic Regression
- Random Forest
- K-Nearest Neighbors (KNN)

The evaluation show that Logistic Regression is the best model (accuracy of 82.46%). We also tried some feature selection techniques (Chi-square and truncated SVD) but the achieved accuracy dropped significantly (70-75%).

4.2 Word Embeddings

We then tried a Deep Learning-based approach using word embeddings as features. Word embeddings are a type of representation for words that capture their meaning and relationships with other words, they have been shown to be very effective for text classification. The model's architecture is as follows:

- *Text Vectorization Layer*: this layer converts the preprocessed song lyrics into numerical representations by assigning each unique word a unique integer index, it maintains a vocabulary of up to MAX_FEATURES words, limiting the complexity of the model.
- *Embedding Layer*: this layer transforms the numerical representations into dense vectors of embedding dimension 1000, these embedding vectors represent the semantic meanings of words, capturing subtle relationships between them.

- *Dropout Layer*: a dropout layers is incorporated to address overfitting by randomly dropping out a certain percentage of neurons during training, preventing the model from memorizing the training data and generalizing better to unseen data.
- *Dense Layer*: the final dense layer consists of 6 neurons, one for each genre. It applies a softmax activation function to output a probability distribution across the genres, indicating the predicted genre for the given song lyrics.

To prevent overfitting and improve generalization, early stopping and a learning rate reduction strategy were employed. The achieved classification accuracy of 85.71% indicates the model's effectiveness in classifying song genres.

5 Prediction

Finally we design a function to use all the trained models to determining genre, identifying relevant topics, and suggesting similar songs based solely on the textual content of a given song.

To obtain similar songs, we first calculate a matrix containing the topic vector of each song in the training set. Then, when making a prediction, we find the 5 songs in the training set with the shortest Euclidean distance to the topic vector that was predicted for the specified song.

6 Conclusions

This project has demonstrated the potential of text mining techniques to extract meaningful information from song lyrics. By integrating topic modeling and text classification methods, we have developed a predictive model capable of determining genre, identifying relevant topics, and suggesting similar songs based solely on the textual content of a given song.

This work has several implications for the way we interact with music. First, it can help us to better understand the meaning and significance of song lyrics. Second, it can be used to personalize music recommendations, making it easier for people to find songs that they will enjoy. Third, it can be used to support music research, helping us to better understand the history, evolution, and impact of music.

References

[1] Genius Song Lyrics Dataset

[2] Genius Platform