Emotion detection in song lyrics

Simone Quadrelli

Università Statale degli studi di Milano, simone.quadrelli@studenti.unimi.it

1 Introduction

The hereby paper addresses the challenge of detecting emotions and their intensity from the text of tweets via machine learning and aims to transfer the prediction capabilities to predict songs emotions and their intensity from the lyrics. These information allows to fathom the existence of correlations between musical genres and sentiments and between authors and sentiments.

Emotion detection has a widespread applications in many fields: it is used in marketing to tune and evaluate advertisement, it is exploited to monitor elections and the preference of voters and it is part of music recommendation systems in many streaming applications. Despite emotion detection being increasingly used in many field, still some remarkable issues remain. There is disagreement about the way sentiment should be measured: some suggest to use valence, arousal, and dominance features, others prefer to use simple categorical variables. Moreover, there are few reliable datasets for emotion detection and just some of them have a suitable dimension for supervised learning. Indeed, a very small dataset, consisting in less than two hundred songs has been constructed by R. Malheiro et al. in [1]. Another issues concerns the emotions to be used: if the number of emotions (or moods) is too big then the subjectivity of annotators may be too much and they should be clustered into wider emotions, if just positive or negative moods are involved, as in [2], they may be too few to be significant. In the end, for music recommendation systems using just lyrics may not be enough and the melody should be equally taken into account.

2 Research question and methodology

The objective of this work is two-fold: to develop predictors of sentiment and intensity and to study whether or not there exist a correlation between genre and sentiments in song lyrics. Achieving this objectives is not straightforward and requires to transfer the prediction capability of predictors trained on labelled twits to unlabelled songs. Moreover, for the delight of the user, it is possible to create a playlist given a chosen genre and sentiment.

Formally, the first objective consists in finding a transform $\phi : \mathcal{D} \to \mathcal{F}$ to map documents into numerical features suitable for machine learning. Then it

is required to select the best classifier $h : \mathcal{F} \to \mathcal{Y}$ that maps the features $\mathcal{F}$ into sentiments $\mathcal{Y}$ and the best predictor for intensity $h' : \mathcal{F} \to \mathcal{I}$ that maps features into intensity $\mathcal{I}$. Then exploiting the features extracted from the song lyrics via ϕ it is possible to use h and h' to assign sentiment and intensity to songs.
In the end, the correlation between the unconditional distribution of the sentiment and conditional distribution of the sentiment given the genre, is tested by χ^2 test.

To clarify the overall description of the process that leads to the computation of the correlations and the creation of playlists the reader may refer to Figure 1.

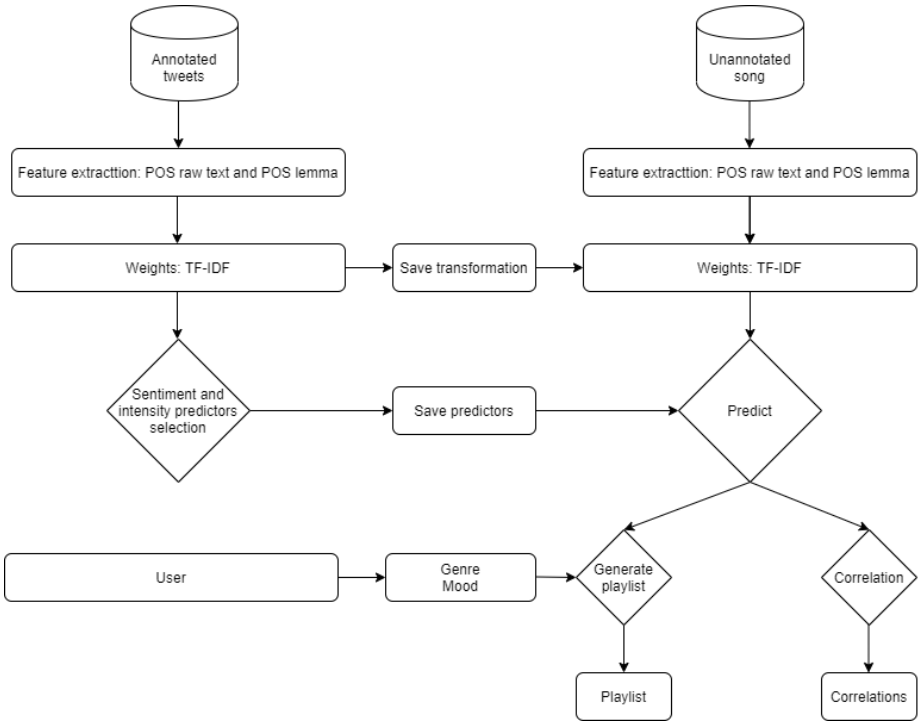

Fig. 1: Schema of the project

A more in-depth description of the process along with some experiments and their results will be detailed in the following section.

3 Experimental results

As stated before, two datasets are involved in the analysis: one containing 3613 tweets[1] and the other one concerning 362237 songs form MetroLyrics[2]. The dataset of tweets is composed by three variables: text, sentiment and intensity, where labels in sentiment belongs to the set {anger, fear, joy, sadness} and intensity $\in [0, 1]$. The song dataset contains the variable: title, lyrics, year, artist and genre.

Among all the possible metric to evaluate the performance of classifiers F1-score $\in [0, 1]$ seems a suitable choice:

$$\text{F1-score} = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{1}$$

It is the armonic mean of precision ($precision = \frac{tp}{tp+fp}$) and recall ($recall = \frac{tp}{tp+fn}$), which can be defined starting from the notions of true positive (tp), false positives (fp) and false negative (fn). Since there are four labels and F1-score works just for binary labels, it is computed class-wise and then scores are averaged.

For the intensity estimation, a suitable measure is the mean absolute error (MAE) between the predicted intensity $\hat{y}$ and the ground truth y computed on N predictions as

$$\text{MAE}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{2}$$

Bearing in mind the descriptions of the datasets and the knowledge of the metrics it is possible to dig deeper into the analysis descirbed in Figure 1.
The very first preprocessing step to clean the corpus consists in removing punctuation and other sequence of characters that carry no information, for instance, the various new line symbols in song lyrics. Even after this preliminary step, it is possible to manipulate the documents in the corpus to map similar words into the same basic expression that must be a word of the vocabulary. This process is known as lemmatization and the basic expressions retrieved are called lemmas. Even if lemmas are extracted, a lot of useless information remains, for instance: pronouns, articles and very frequent verbs. It is possible to select only some interesting part of speech (POS) elements such as nouns and adjective to use what are expected to be very informative words or lemmas. At the end of this first processing nouns and adjective both in lemmatized and not lemmatized form were selected, since they are more likely to convey emotional information.

[1] https://saifmohammad.com/WebPages/EmotionIntensity-SharedTask.html
[2] https://www.kaggle.com/gyani95/380000-lyrics-from-metrolyrics

To be able to train classifiers data should be converted into some suitable numerical form by ϕ. To overcome the issue, it is possible to use Term frequency - inverse document frequency schema to give higher weights to presumably more important words. This weighting schema can be applied when the data have been vectorized. Indeed it is possible to embed the text into a numeric matrix $M_{N\times W}$, where $N = |D|$ is the number of documents in the corpus D and W is the total number of unique words (or lemmas) in it. The term frequency $ft(t, d)$ of a terms t in a document d is

$$tf(t, d) = \frac{f(t, d)}{\sum_{t' \in d} f_{t',d}}, \tag{3}$$

where $tf(t, d) = \mathbb{1}\{t \in d\}$.

The inverse document frequency $idf(t, D)$ is the logarithm of the ratio between the total number of documents and the number of time the term t appears in the set of documents D

$$idf(t, D) = \log\left(\frac{N}{\sum_{d=1}^{D} \mathbb{1}\{t \in d\}}\right) \tag{4}$$

From this two quantities it is possible to obtain tf-idf weights as

$$\text{tf-idf}(t, d, D) = tf(t, d) \cdot idf(t, D) \tag{5}$$

After the weights have been computed the python object that computes the transformation is saved to be later used to extract the weights from song lyrics.

Before proceeding to the model selection, there is issues tackle: the dataset is unbalanced; indeed, the class *fear* has higher frequency than the other ones. To prevent classifiers from predicting too many instances of class *fear*, an oversample procedure is applied. Hence, the final dataset is perfectly balanced.
There is also choice to make when deciding how to handle the prediction task: either train predictors that output sentiment and intensity together or a classifier for the sentiment and an independent predictor for the intensity. Since the correlation between sentiment and emotion, computed by Spearman's correlation test, is very close to zero, the latter approach seems a valid and much simpler option. Hence, there two different problems arise: it is necessary to select the best classification model for the sentiment and the best regression model for the intensity. For classification, the best model was selected accordingly to the cross-validated F1-score among K-nearest neighbours (KNN), random forests (RF) and SVMs. Analogously the best regression model was selected accordingly to the cross-validated mean absolute error among the previous classes modified for regression models. This procedure is performed two times using as features POS lemma and POS raw text.

Table 1: Performance of predictors

(a) Classification performance

Predictor	Features	F1-score
KNN	POS lemma	0.66
KNN	POS raw	0.61
RF	POS lemma	0.70
RF	POS raw	0.70
SVM	POS lemma	0.87
SVM	POS raw	0.89

(b) Regression performance

Predictor	Features	MAE
KNN	POS lemma	0.14
KNN	POS raw	0.14
RF	POS lemma	0.15
RF	POS raw	0.15
SVM	POS lemma	0.11
SVM	POS raw	0.11

As can be seen from Table 1, the best models for sentiment and intensity is the SVM for classification and the SVM for regression with POS raw text features and indeed they are applied to songs to assign them sentiment and intensity. Then the intensity is used to select the songs that have at least intensity higher that 0.5, assuming that if the intensity is too low, then the predicted sentiment may not be relevant.

As previously stated, one objective of this work is to understand whether or not there is a correlation between genre and sentiment. In order to understand it is possible to run the χ^2 test for independence between the unconditional distribution of sentiment (see Table 2) and the distribution of sentiment within each class (see Figure 3).

Anger	Fear	Joy	Sadness
0.17	0.32	0.26	0.25

Table 2: Unconditional distribution of the sentiments

As Figure 2 shows, there is correlation between sentiments and genres, indeed the p-values are 0 and the test scores are remarkably high and therefore the independence hypothesis should be rejected.

```
Power_divergenceResult(statistic=59236142.0970242, pvalue=0.0)
Power_divergenceResult(statistic=11203428.310288608, pvalue=0.0)
Power_divergenceResult(statistic=869650.506747599, pvalue=0.0)
Power_divergenceResult(statistic=221392022.56083325, pvalue=0.0)
Power_divergenceResult(statistic=2198070.1291956482, pvalue=0.0)
Power_divergenceResult(statistic=12938333.560142726, pvalue=0.0)
Power_divergenceResult(statistic=159398734.2544964, pvalue=0.0)
Power_divergenceResult(statistic=84988573.01230262, pvalue=0.0)
Power_divergenceResult(statistic=4968962.380629653, pvalue=0.0)
Power_divergenceResult(statistic=310096282.6461709, pvalue=0.0)
Power_divergenceResult(statistic=2732474.551130772, pvalue=0.0)
Power_divergenceResult(statistic=2563877910.467037, pvalue=0.0)
```

Fig. 2: χ^2 test results

Moreover, from the confusion matrix (see Figure 3) of genres and sentiment, normalized by columns, it is possible to appreciate the difference in the distribution of sentiments in each genre.

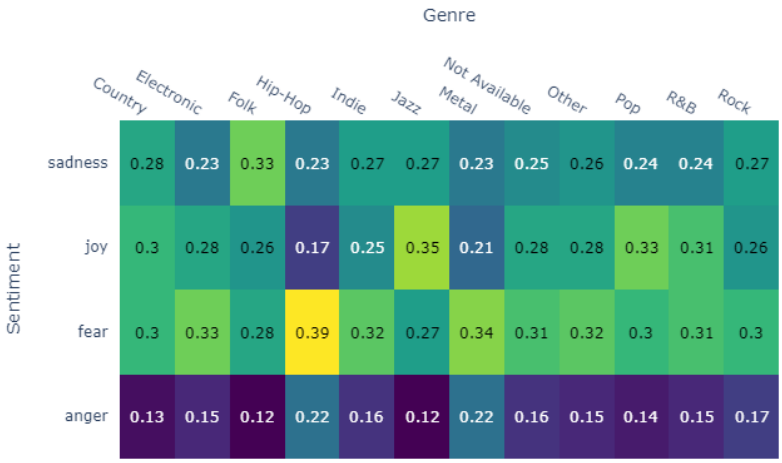

Fig. 3: Correlation between genre and sentiment of songs whose intensity is greater than 0.5

It is also possible to analyze the sentiment of different authors for which a relevant number of songs is present as Figure 4 shows.

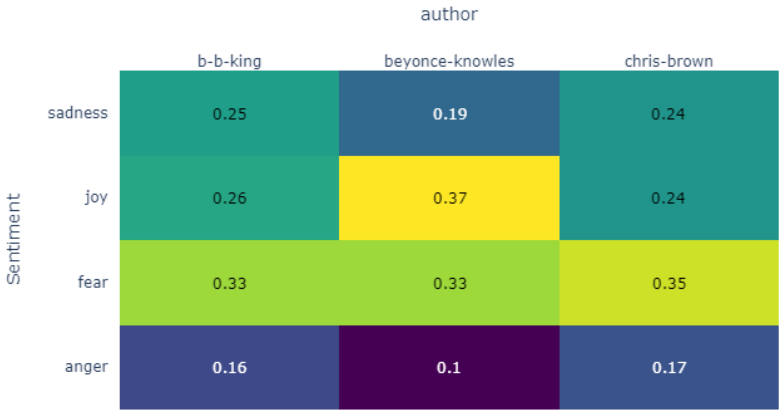

Fig. 4: BB King's, Beyoncé's, Chris Brown's songs sentiments

Moreover, given the genre and the sentiment provided by the user, the software outputs a personalized playlist. It is computed in two steps: first songs of the specified genre and sentiment are selected and then they are sorted by intensity: Figure 5 provides some example of playlists generated by the software.

Fig. 5: Example of playlists

```
Enter a gerne: Jazz
Enter a mood: joy
september song
body and soul
i am loved
body soul
sunny
i didn t know about you
it s a whistling kinda morning
what game shall we play today
```

(a) Playlist containing the most intense jazz song with respect to joy

```
Enter a gerne: Metal
Enter a mood: fear
nervous heart
unleashed upon mankind
from beyond the grave
seeds of mans destruction
intro unleashed upon mankind
stigmata
spellbound by the devil
```

(b) Playlist containing the most intense metal song with respect to fear

4 Concluding remarks

Overall, there is a correlation between genre and sentiments as the χ^2 tests clearly show (see Figure 2). Anger is the less frequent sentiment, even if the frequency of joy, sadness and anger is very similar in the original unbalanced dataset. On the contrary, the great amount of fear labels may come from the dataset which was originally unbalanced. The oversample procedure is such that some example of the less frequent sentiments are repeated but it may not be enough to completely remove the bias that comes from the original unbalancedness. Moreover, there are some interesting findings: a remarkable amount hip-hop and metal songs are labelled with fear but I would expect metal to be more related to anger while hip-hop to joy. On the contrary, jazz songs are strongly connected to joy and this seems a sound result. It is also worth noting that folk is the only genre whose songs are most frequently labelled with sadness.
The sentiment distribution of Beyoncé is very peculiar: indeed it has a very low number of anger labels and joy is remarkably high, even more if compared to pop sentiment distribution (see Figure 4). B.B. King, a jazz guitarist, has a distribution skewed toward fear, while the most frequent sentiment in jazz songs is joy (see Figure 4).
A last remark, even if the title may convey useful information about the sentiment it was not used as feature so that it can be exploited by the reader as an independent test to assess the precision of the model.

However, songs should not be categorized by the text alone, the music is at least as important as the lyrics. An interesting possibility is to analyze the music of these song to better predict the sentiment. Another possible future work may be consist in comparing the results of this work with those from an annotated and comparable dataset of song lyrics.

References

1. R. Malheiro, R. Panda, P. Gomes R. Paiva. Emotionally-relevant features for classification and regression of music lyrics. In: IEEE transactions on affective computing, (2016).
2. S. Raschka. MusicMood: Predicting the mood of music from song lyrics using machine learning. (2016).