Semester Project Report

Department Of Data Science and Engineering

Emotion Patterns in Music Playlists

Authors:
Sara Giammusso
Mario Guerriero

Supervisors:
Raphael Troncy
Pasquale Lisena
Enrico Palumbo

May 27, 2018

Contents

Abstract

Music streaming services such as Spotify are revolutionizing the music world, enabling a transition from artist-created bundles of songs (CDs) to user-created playlists.

Different logics may be applied in the generation of a playlist: they can contain songs of a similar genre (e.g. “Rock playlist”), fit to a particular occasion (e.g. “New year’s eve party”), to a particular context (e.g. “Gym”), to a particular mood (e.g.“Happy”) and so on.

The goal of this semester project is to unravel the emotion patterns underlying the sequences of songs in a playlist using automatic approaches of Emotion Detection on the lyrics.

Chapter 1

Introduction

1.1 Background

In the last few years the online music streaming services such as Spotify, Apple Music and Deezer introduced, among others, the possibility to create playlists thus opening new challenges on music recommendation.

One of the new possible task a Recommender System should perform is the automatic playlist continuation. By suggesting appropriate songs to add to a playlist, a Recommender System can increase user engagement by making playlist creation easier, as well as extending listening beyond the end of existing playlists.

More precisely the task of automatic playlist continuation consists in: given a set of playlist features, the system shall generate a list of recommended tracks that can be added to that playlist, thereby "continuing" the playlist.

1.2 Project scope

In light of this one of the possible feature to consider in developing a system for the automatic playlist continuation task is the emotion expressed in each song contained in the playlist and the more frequent transition patterns from one emotion to the other. Thus, not only the emotion of each song lyrics must be detected, but also the transitions between emotions must be analyzed.

Emotion Detection is a novel and promising field of study of Natural Language Understanding, which is able to automatically infer what are the emotions expressed in a text. It can be considered a Sentiment Analysis task, the computational treatment of opinions, sentiments and subjectivity of text.

1.3 Report outline

The report is structured as follow:

- *Chapter 2*:

Chapter 2

Sentiment Analysis

2.1 Sentiment Analysis

Sentiment Analysis (SA) is the computational study of people's opinions, attitudes and emotions toward an entity, the entity being an individual, event or topic.[1]

Sentiment Analysis can be considered a classification task as illustrated in Fig 4.1.

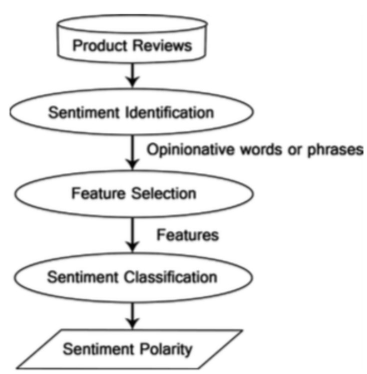


Figure 2.1: Sentiment analysis process on product reviews

There are three main classification levels in SA:

- Document-level

- Sentence-level
- Aspect-level

While document-level SA aims to classify an opinion document a expressing a positive or negative opinion or sentiment by considering the whole document as the basic information unit, sentence-level SA aims to classify sentiment expressed in each sentence.

In both cases the first step is to identify whether the sentence/document is subjective or objective and if it is subjective determine whether the sentence expresses positive or negative opinions.

Classifying text at the document level or at the sentence level does not provide the necessary detail needed on opinions on all aspects of the entity which is needed in many applications. Aspect-level SA instead aims to classify the sentiment with respect to the specific aspects of entities. The first step is to identify the entities and their aspects. The opinion holders can give different opinions for different aspects of the same entity, e.g. "This chair is ugly but it is comfortable".

Sentiment Analysis task is considered a sentiment classification problem. The first step in the SC problem is to extract and select text features. Some of the current features are:

- **Terms presence and frequency**: These features are individual words or word n-grams and their frequency counts;
- **Parts of speech (POS)**: finding adjectives, pronouns, etc. as they are important indicators of opinions;
- **Opinion words and phrases**: these are words commonly used to express opinions including *good or bad, like or hate*. On the other hand, some phrases express opinions without using opinion words, e.g. *cost me and arm and a leg*;
- **Negations**: the appearance of negative words may change the opinion orientation like *not good* is equivalent to *bad*.

Sentiment Classification techniques can be roughly divided into machine learning approach, lexicon based approach and hybrid approach.

The *Machine Learning Approach (ML)* applies the famous ML algorithms and uses linguistic features. The *Lexicon-based Approach* relies on a sentiment lexicon, a collection of known and precomiled sentiment terms. It is

divided into dictionary-based approach and corpus-based approach which use statistical or semantic methods to find sentiment polarity.
The *hybrid Approach* combines both approaches. The various approaches and the most popular algorithms of SC are illustrated in Fig 4.2.

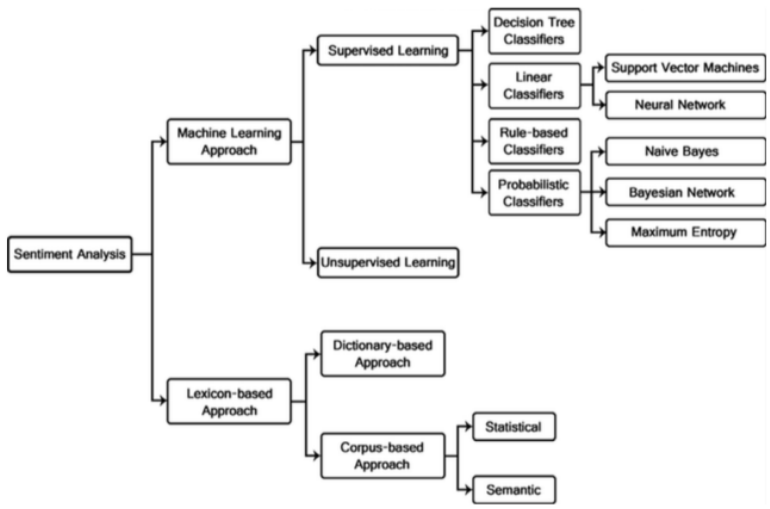


Figure 2.2: Sentiment classification techniques

The text classification methods using ML approach can be roughly divided into supervised and unsupervised learning methods. The supervised methods make use of a large number of labeled training documents. The unsupervised methods are used when it is difficult to find these labeled training training documents.

The lexicon-based approach depends on finding the opinion lexicon which is used to analyze the text. There are two methods in this approach.
In the Dictionary-based approach a small set of opinion words is collected manually with known orientations. Then, this set is grown by searching their synonyms and antonyms. The newly found words are added to the seed list then the next iteration starts. The iterative process stops when no new words are found.

The dictionary based approach has a major disadvantage which is the inability to find opinion words with domain and context specific orientation. This problem is solved by the corpus-based approach that depends on syntactic patterns or patterns that occur together along with a seed list of opinion words to find other opinion words in a large corpus. One of these methods is called *sentiment consistency*: it starts with a list of seed opinion adjectives, and used them along with a set of linguistic constraints to identify additional adjective opinion words and their orientations. The constraints being for example *AND, OR, BUT, EITHER-OR,...*; the conjunction *AND* for example says that conjoined adjectives usually have the same orientation.

2.1.1 Emotion Detection

Emotion detection (ED) is the process of identifying human emotions. It is a recent field of research that is closely related to Sentiment Analysis. Indeed, Sentiment Analysis aims to detect positive, neutral or negative feelings from text, whereas Emotion Analysis aims to detect and recognize feelings through the expression of texts.

Emotion is expressed as joy, sadness, anger, surprise, hate, fear and so on. Since there is not any standard emotion word hierarchy, focus is on the related research about emotion in cognitive psychology domain. In 2001, W. Gerrod Parrot, wrote a book named "Emotions In Social Psychology", in which he explained the emotion system and formally classified the human emotions through an emotion hierarchy in six classes at primary level which are *Love, Joy, Anger, Sadness, Fear and Surprise* [2].

Emotion detection may have useful applications, such as [3]:

- Measure citizens happiness;
- Pervasive computing: this may include suggesting help when anxiety is detected through speech, or to check the tone of an email;
- Improving perception of a customer to increase brand reputation and sales.

Some of the biggest challenges in determining emotion are:

- *Context-dependence of emotions*: people use different regulation strategies in different social contexts. A phrase can have element of *anger* without using the word "anger" or any of its synonyms, e.g. *"Shut up!"*
- *Word-sense disambiguation*: identifying which sense of a word (i.e. meaning) is used in a sentence, when the word has multiple meanings;

- *Co-reference resolution*: pronouns and other referring expressions must be connected to the right individuals;
- Lack of labelled emotion databases.

The main methods used for text based emotion detection are:

- *Keyword Spotting*
- *Lexical Affinity*
- *Learning-based*
- *Hybrid*

Keyword Spotting The keyword pattern matching problem can be described as the problem of finding occurrences of keywords from a given set as substrings in a given string. These words are classified into categories such as disgusted, sad, happy, angry, fearful, surprised, etc. Process of Keyword spotting method is shown in Fig 4.3.

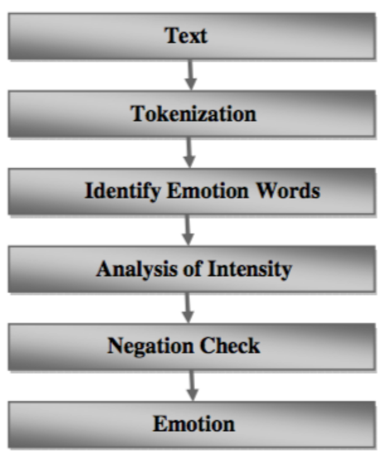


Figure 2.3: Keyword spotting technique

The first step is converting data into tokens, i.e. a sentence into words, then from these tokens emotion words are detected. The second step is analyzing the intensity of emotion words. Sentence, then, is checked whether negation is involved in it or not then finally an emotion class will be assigned.

Lexical Affinity method The Lexical Affinity approach is an extension of keyword spotting technique: apart from picking up emotional keywords it assigns *probabilistic affinity* for a particular emotion to arbitrary words. This technique has the main disadvantage of missing out emotion content that resides deeper than the word level.
For example the word 'accident', having been assigned a high probability of indicating a negative emotion, would not contribute correctly to the emotional assessment of phrases like *"I avoided an accident"* or *"I met my girlfriend by accident"*.

Learning-based methods Learning-based methods change the focus from "determining emotions" to "classify the input texts into different emotions". Indeed, learning-based methods try to detect emotions based on a previously trained classifier, which apply various theories of machine learning such as SVM.

Hybrid Methods Since keyword-based methods and naive learning-based methods could not acquire satisfactory results, some systems use hybrid approach by combining both keyword spotting technique and learning based method, which help to improve accuracy.

However all these methods have some major limitations:

- *Ambiguity in Keyword Definitions*: words can have multiple and vague meanings that can change according to different usages and contexts. Moreover emotion labels could have different emotions in some extreme cases such as ironic or cynical sentences;

- *Lack of Linguistic Information*: these methods totally ignore syntax structures and semantics that also have influences on expressed emotions. For example the sentences *"He laughed at me"* or *"I laughed at him"* express two totally different meanings;

- *Incapability of Recognizing Sentences without Keywords*: sentences without any keyword would imply that they do not contain any emotion at all, which is obviously wrong.

There are mainly two possible way to label data [3]:

1. The label is one between the set of emotions, e.g. *anger, disgust, sad, happy, surpise, fear, neutral*;

2. *Slider approach*: the label is composed of percentages for each emotion, as described in Fig 4.4.

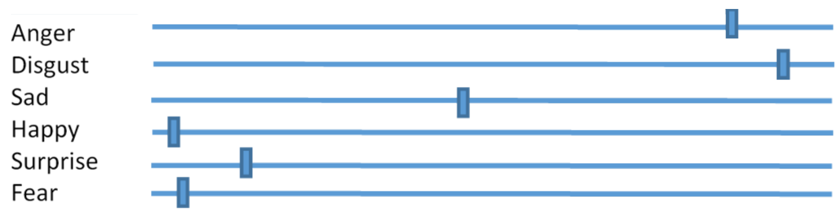


Figure 2.4: Slider approach

The design n.2 offers more information but it comes with complication. Labels produced by design n.2 can be turned into distinct labels of single emotion, just like those produced by design n.1, but not vice versa.

Chapter 3

Problem Pre-processing

3.1 The problem

The goal of the semester project is to unravel the emotion pattern underlying the sequences of songs in a playlist using automatic approaches of Emotion Detection on the lyrics.

The problem can be divided into two main parts:

1. Classify emotions for each song based on the lyrics
2. Analyze emotion patterns in the playlists

3.2 Related work

Emotion detection domain has already attracted many researchers from computer science, psychology, cognitive science and so on.

Before building our own emotion detection system we start analyzing some already existent classifier.

IBM Watson Natural Language Understanding [4] Watson is a question answering computer system capable of answering questions posed in natural language, developed by IBM.

Natural Language Understanding is a collection of APIs that allows to:

- Recognize the overall sentiment, in a scale from negative to positive [-1, 1];

- Detect the emotion percentage between: joy, anger, disgust, sadness, fear;
- Determine keywords ranked by relevance;
- Extract entities: people, companies, organizations, cities and other information;
- Classify content into hierarchical categories;
- Identify general concepts that may not be directly referenced in the text;

Results obtained analyzing Oasis - Wonderwall are illustrated in Fig.

IBM Watson Tone Analyzer [5] It uses linguistic analysis to detect joy, fear, sadness, anger, analytical, confident, and tentative tones found in text. It allows to select different sources: tweets, online reviews, email messages, or other text. It uses both:

- the document level: to get a sense of the overall tone;
- the sentence level: to identify specific areas where tones are the strongest.

QEmotion [6] Qemotion detects the main emotion of the speech and define the corresponding emotion in terms of temperature.

- From $31°$ to $40° \rightarrow$ Happiness
- From $21°$ to $30° \rightarrow$ Surprise
- From $11°$ to $20° \rightarrow$ Calm
- From $6°$ to $10° \rightarrow$ Fear
- From $-5°$ to $5° \rightarrow$ Sadness
- From $-14°$ to $-6° \rightarrow$ Anger
- From $-20°$ to $-15° \rightarrow$ Disgust

3.3 NLP libraries

In order to select the best Natural Language Processing library for our purpose we also analyzed pros and cons of the main Natural Language Processing libraries, i.e. NLTK, TextBlob, Standord's CoreNLP and SpaCy.

NLTK: Natural Language Toolkit It is recommended only as an education and research tool.
Pros:

- its modularized structure makes it excellent for learning and exploring NLP concepts;
- over 50 corpora and lexicons, 9 stemmers, and a dozens of algorithms to choose from (this can also be considered as a con).

Cons:

- Heavy library with a steep learning curve;
- Slow and not production-ready.

TextBlob Built on top on NLTK.
Pros:

- More intuitive;
- Gentle learning curve.

Stanford's CoreNLP Java library with Python wrappers.
Pros:

- fast;
- support for several major languages.

SpaCy It is a new NLP library designed to be fast, streamlined and production-ready.
Pros:

- minimal number of options;
- its philosophy is to only present the best algorithm for each purpose.

Cons:

- it is new, so its support community is not as large as other libraries, but it is growing very fast.

3.4 Word embedding techniques

Word embeddings are a set of feature learning techniques mapping words or phrases from the vocabulary to vectors or real numbers.

These techniques map sparse word vectors into continuous space based on the surrounding context. For example if *"salt"* and *"seasoning"* appear within the same context, the model will indicate that *"salt"* is conceptually closer to *"seasoning"*, than another word, say *"chair"*.

There are two main embedding libraries: Word2vec and FastText. While Word2vec treats each word in corpus like an atomic entity generating a vector for each word, FastText treats each word as composed of character ngrams, so the vector for a word is made of the sum of this character n grams.

For example, the word vector *"apple"* is a sum of the vectors of the n-grams "ap", "app", "appl", "apple", "ppl', "pple", "ple", "le" assuming 3 and 6 as minimum and maximum ngrams size.
The difference between Word2vec and FastText manifests as follows:

1. *Rare words*: even if words are rare, their character n-grams are still shared with other words - hence the embeddings with FastText can still be good;
2. *Out of vocabulary words*: FastText can construct the vector for a word from its character n-grams even if word does not appear in training corpus;
3. *Hyperparameters choice*: FastText requires to choose the minimum and maximum n-grams sizes, and this directly impacts the computation time and the memory requirements.

3.5 Public datasets

A big challenge in emotion detection is the lack of a labelled emotion database to enable active innovation. Currently, few publicly accessible databases are available.

MoodyLyrics[7] contains around 2500 songs manually annotated through Amazon Mechanical Turk with 4 different emotion, i.e., happy, sad, angry and relaxed.

EmoInt[8] contains manually annotated tweets classified according to the intensities of anger, fear, joy and sadness. *EmoBank*[9] instead contains 10.000 sentences, each of which has been annotated according to both the emotion expressed by the writer and the emotion perceived by the reader.

Chapter 4

Sentiment Analysis

4.1 Sentiment Analysis

Sentiment Analysis (SA) is the computational study of people's opinions, attitudes and emotions toward an entity, the entity being an individual, event or topic.[1]

Sentiment Analysis can be considered a classification task as illustrated in Fig 4.1.

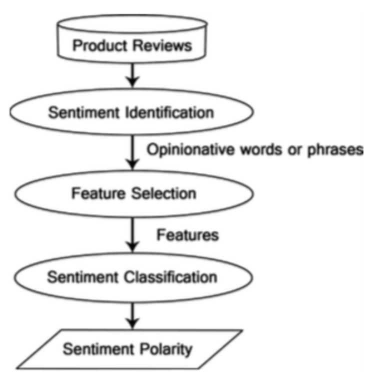


Figure 4.1: Sentiment analysis process on product reviews

There are three main classification levels in SA:

- Document-level

- Sentence-level
- Aspect-level

While document-level SA aims to classify an opinion document a expressing a positive or negative opinion or sentiment by considering the whole document as the basic information unit, sentence-level SA aims to classify sentiment expressed in each sentence.

In both cases the first step is to identify whether the sentence/document is subjective or objective and if it is subjective determine whether the sentence expresses positive or negative opinions.

Classifying text at the document level or at the sentence level does not provide the necessary detail needed on opinions on all aspects of the entity which is needed in many applications. Aspect-level SA instead aims to classify the sentiment with respect to the specific aspects of entities. The first step is to identify the entities and their aspects. The opinion holders can give different opinions for different aspects of the same entity, e.g. "This chair is ugly but it is comfortable".

Sentiment Analysis task is considered a sentiment classification problem. The first step in the SC problem is to extract and select text features. Some of the current features are:

- **Terms presence and frequency**: These features are individual words or word n-grams and their frequency counts;
- **Parts of speech (POS)**: finding adjectives, pronouns, etc. as they are important indicators of opinions;
- **Opinion words and phrases**: these are words commonly used to express opinions including *good or bad, like or hate*. On the other hand, some phrases express opinions without using opinion words, e.g. *cost me and arm and a leg*;
- **Negations**: the appearance of negative words may change the opinion orientation like *not good* is equivalent to *bad*.

Sentiment Classification techniques can be roughly divided into machine learning approach, lexicon based approach and hybrid approach.

The *Machine Learning Approach (ML)* applies the famous ML algorithms and uses linguistic features. The *Lexicon-based Approach* relies on a sentiment lexicon, a collection of known and precomiled sentiment terms. It is

divided into dictionary-based approach and corpus-based approach which use statistical or semantic methods to find sentiment polarity.
The *hybrid Approach* combines both approaches. The various approaches and the most popular algorithms of SC are illustrated in Fig 4.2.

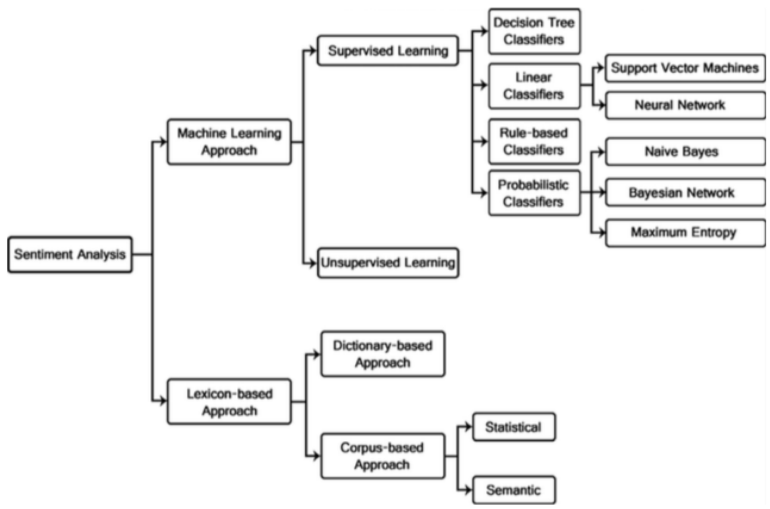


Figure 4.2: Sentiment classification techniques

The text classification methods using ML approach can be roughly divided into supervised and unsupervised learning methods. The supervised methods make use of a large number of labeled training documents. The unsupervised methods are used when it is difficult to find these labeled training training documents.

The lexicon-based approach depends on finding the opinion lexicon which is used to analyze the text. There are two methods in this approach.
In the Dictionary-based approach a small set of opinion words is collected manually with known orientations. Then, this set is grown by searching their synonyms and antonyms. The newly found words are added to the seed list then the next iteration starts. The iterative process stops when no new words are found.

The dictionary based approach has a major disadvantage which is the inability to find opinion words with domain and context specific orientation. This problem is solved by the corpus-based approach that depends on syntactic patterns or patterns that occur together along with a seed list of opinion words to find other opinion words in a large corpus. One of these methods is called *sentiment consistency*: it starts with a list of seed opinion adjectives, and used them along with a set of linguistic constraints to identify additional adjective opinion words and their orientations. The constraints being for example *AND, OR, BUT, EITHER-OR,...*; the conjunction *AND* for example says that conjoined adjectives usually have the same orientation.

4.1.1 Emotion Detection

Emotion detection (ED) is the process of identifying human emotions. It is a recent field of research that is closely related to Sentiment Analysis. Indeed, Sentiment Analysis aims to detect positive, neutral or negative feelings from text, whereas Emotion Analysis aims to detect and recognize feelings through the expression of texts.

Emotion is expressed as joy, sadness, anger, surprise, hate, fear and so on. Since there is not any standard emotion word hierarchy, focus is on the related research about emotion in cognitive psychology domain. In 2001, W. Gerrod Parrot, wrote a book named "Emotions In Social Psychology", in which he explained the emotion system and formally classified the human emotions through an emotion hierarchy in six classes at primary level which are *Love, Joy, Anger, Sadness, Fear and Surprise* [2].

Emotion detection may have useful applications, such as [3]:

- Measure citizens happiness;
- Pervasive computing: this may include suggesting help when anxiety is detected through speech, or to check the tone of an email;
- Improving perception of a customer to increase brand reputation and sales.

Some of the biggest challenges in determining emotion are:

- *Context-dependence of emotions*: people use different regulation strategies in different social contexts. A phrase can have element of *anger* without using the word "anger" or any of its synonyms, e.g. *"Shut up!"*
- *Word-sense disambiguation*: identifying which sense of a word (i.e. meaning) is used in a sentence, when the word has multiple meanings;

- *Co-reference resolution*: pronouns and other referring expressions must be connected to the right individuals;
- Lack of labelled emotion databases.

The main methods used for text based emotion detection are:

- *Keyword Spotting*
- *Lexical Affinity*
- *Learning-based*
- *Hybrid*

Keyword Spotting The keyword pattern matching problem can be described as the problem of finding occurrences of keywords from a given set as substrings in a given string. These words are classified into categories such as disgusted, sad, happy, angry, fearful, surprised, etc. Process of Keyword spotting method is shown in Fig 4.3.

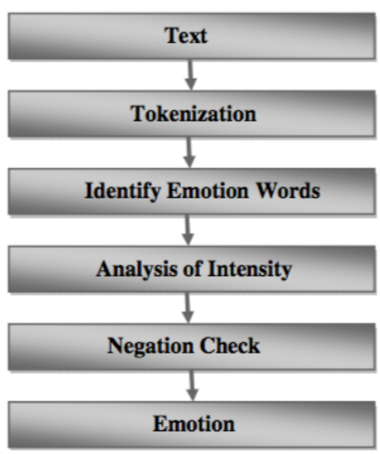


Figure 4.3: Keyword spotting technique

The first step is converting data into tokens, i.e. a sentence into words, then from these tokens emotion words are detected. The second step is analyzing the intensity of emotion words. Sentence, then, is checked whether negation is involved in it or not then finally an emotion class will be assigned.

Lexical Affinity method The Lexical Affinity approach is an extension of keyword spotting technique: apart from picking up emotional keywords it assigns *probabilistic affinity* for a particular emotion to arbitrary words. This technique has the main disadvantage of missing out emotion content that resides deeper than the word level.
For example the word 'accident', having been assigned a high probability of indicating a negative emotion, would not contribute correctly to the emotional assessment of phrases like *"I avoided an accident"* or *"I met my girlfriend by accident"*.

Learning-based methods Learning-based methods change the focus from "determining emotions" to "classify the input texts into different emotions". Indeed, learning-based methods try to detect emotions based on a previously trained classifier, which apply various theories of machine learning such as SVM.

Hybrid Methods Since keyword-based methods and naive learning-based methods could not acquire satisfactory results, some systems use hybrid approach by combining both keyword spotting technique and learning based method, which help to improve accuracy.

However all these methods have some major limitations:

- *Ambiguity in Keyword Definitions*: words can have multiple and vague meanings that can change according to different usages and contexts. Moreover emotion labels could have different emotions in some extreme cases such as ironic or cynical sentences;

- *Lack of Linguistic Information*: these methods totally ignore syntax structures and semantics that also have influences on expressed emotions. For example the sentences *"He laughed at me"* or *"I laughed at him"* express two totally different meanings;

- *Incapability of Recognizing Sentences without Keywords*: sentences without any keyword would imply that they do not contain any emotion at all, which is obviously wrong.

There are mainly two possible way to label data [3]:

1. The label is one between the set of emotions, e.g. *anger, disgust, sad, happy, surpise, fear, neutral*;

2. *Slider approach*: the label is composed of percentages for each emotion, as described in Fig 4.4.

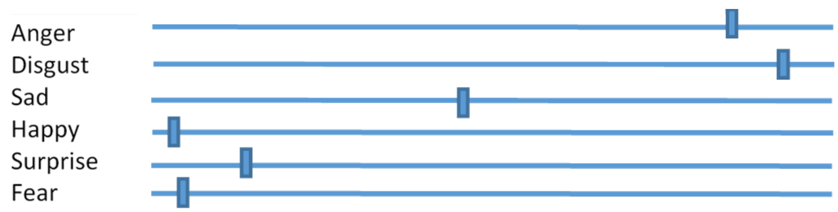


Figure 4.4: Slider approach

The design n.2 offers more information but it comes with complication. Labels produced by design n.2 can be turned into distinct labels of single emotion, just like those produced by design n.1, but not vice versa.

Chapter 5

Model improvements

5.1 Feature Engineering

In order to improve our model performances, we focused our attention on feature engineering. Specifically we tried to extract stylometric, structural, orientation and vocabulary based features[10]. Apart from this we also generated a word embedding vector of the words contained in each song's lyric by using SpaCy's[11] pre-trained language model based on word2vec[12].

Here is a comprehensive list of the features we extracted from our dataset, followed by a brief description:

Title_vector: word embedding vector of the song's title

Lyric_vector: word embedding vector of the lyric content

%Rhymes: defined as the percentage of the number of rhymes over the number of total lines. A rhyme is defined as a rhyme between two following lines

Line_count: number of lines in the lyric

Word_count: number of words in the lyric

%Past_tense_verbs: defined as the the percentage of the number of past tense verbs over the total number of verbs

%Present_tense_verbs: defined as the the percentage of the number of present tense verbs over the total number of verbs

%Future_tense_verbs: defined as the the percentage of the number of future tense verbs over the total number of verbs, where future is just will + base form

%ADJ: percentage of adjectives over the total number of words

%ADP: percentage of adpositions (e.g. in, to, during) over the total number of words

%ADV: percentage of adverbs (e.g. very, tomorrow, down, where, there) over the total number of words

%AUX: percentage of auxiliaries (e.g. is, has (done), will (do), should (do)) over the total number of words

%INTJ: percentage of interjections (e.g. psst, ouch, bravo, hello) over the total number of words

%NOUN: percentage of nouns over the total number of words

%NUM: percentage of numerals over the total number of words

%PRON: percentage of pronouns (e.g. I, you, he, she, myself, themselves, somebody,...) over the total number of words

%PROPN: percentage of proper nouns (e.g. Mary, John) over the total number of words

%PUNCT: percentage of puntuctuation (e.g. ., (,), ?) over the total number of words

%VERB: percentage of verbs over the total number of words

Selfish_degree: percentage of 'I' pronouns over the total number of pronouns

%Echoism: percentage of echoism over the total number of words, where an echoism is either a sequence of two subsequent repeated words or the repetition of a vowel in a word

%Duplicate_Lines: number of lines duplicated across the lyric text

isTitleInLyric: boolean, true if the title string is also a substring of the lyric

Sentiment: sentiment between -1 and 1

Subjectivity_degree: degree of subjectivity of the text

Since the word embedding vectors we generated had length 300, at the end we were able to obtain 623 distinct numerical features for each of the songs in our dataset.

5.1.1 Feature Selection

Having to deal with 623 different features for discriminating songs among 4 classes is probably enough and many features may be redundant or may not bring any useful information to our goal. Indeed, after running many experiments, we tried to keep our models as simple as possible by trying to select the fewer number of features possible.

In the end, we obtained the best results just by using the following features: *Lyric_vector*, *%Echoisms*, *%Duplicate_Lines*, *isTitleInLyrc*, *%Past_tense_verbs*, *%Present_tense_verbs*, *%Future_tense_verbs*, *%ADJ*, *%PUNCT*, *Sentiment* and *Subjectivity_degree*. This process of feature selection left us with just 310 distinct features per song.

5.2 MoodyLyrics duplicates bug

During the analysis of MoodyLyrics described in the previous chapter we detected the presence of duplicated songs inside the dataset. Moreover, sometimes different emotions were associated with the duplicated songs. Thus, to continue our analysis we eliminated duplicated rows and we chose as emotion label the most frequent emotion between all the duplicates.

After reporting the bug to MoodyLyrics owners we have been suggested to use MoodyLyrics4Q, that, according to the creators is a more accurate version of MoodyLyrics.

This advice opened us three possibilities: continue using MoodyLyrics, start using MoodyLyrics4Q or create a new dataset as the concatenation of the previous two. We decided to start using all these three models, in order to understand which one, at the end, will give us a better playlists classification.

The complete MoodyLyrics emotion classification analysis can be found at Notebook 1 while the MoodyLyrics4Q and the emotion detection analysis in the merged datasets can be found at Notebook 2.

5.3 MoodyLyrics4Q

MoodyLyrics4Q contains 2000 songs and has the same annotation schema as MoodyLyrics. Fig 5.1 shows the emotions distribution comparison between the two MoodyLyrics versions.

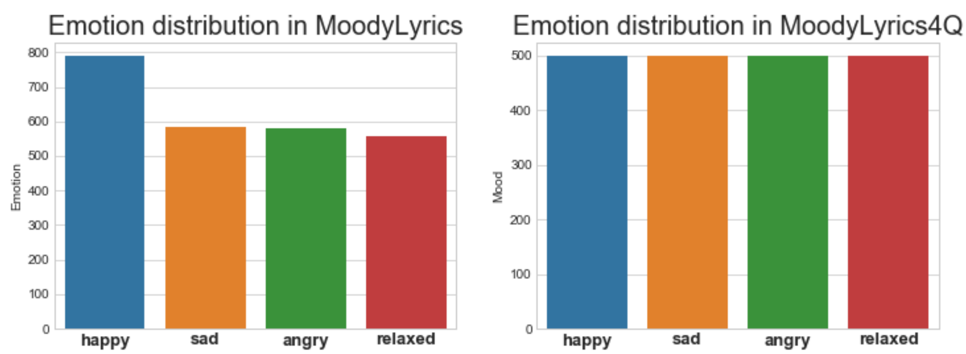


Figure 5.1: Emotions distribution comparison between MoodyLyrics and MoodyLyrics4Q

MoodyLyrics4Q classes are more much balanced, however MoodyLyrics4Q contains only 2000 songs instead of 2509.

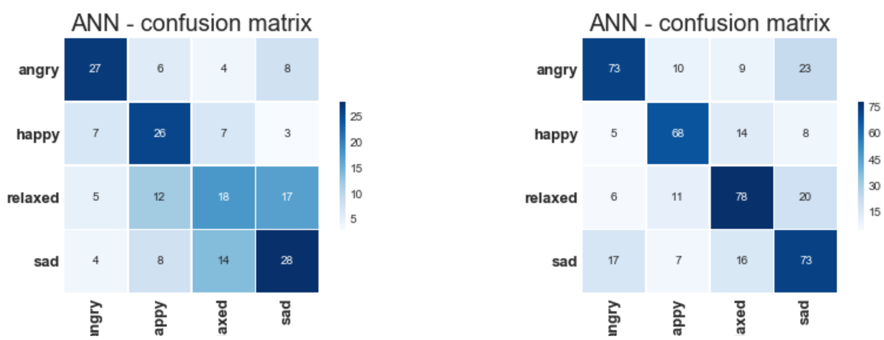
We studied the qualitative difference between the two version comparing the classification given to the songs contained in both datasets to establish what version, according to us, is more correct. The intersection between the two versions contains 47 songs, and 21 over 47 have been classified differently. We noticed that in 15 over this 21 songs the two datasets confuses *happy* with *relaxed* and *angry* with *sad*. Indeed, only 6 of 21 songs are classified totally differently, however reading the lyrics of each of this song we could not establish which version is the best one.

5.4 Results

In this section we present the result obtained while predicting one of the four emotion labels *relaxed*, *happy*, *sad*, *angry*, using an artificial neural network, a support vector machine, the logistic regression and xgboost. The accuracies have been computed with a 5-fold cross validation. All the implementation details can be found at Notebook 2.

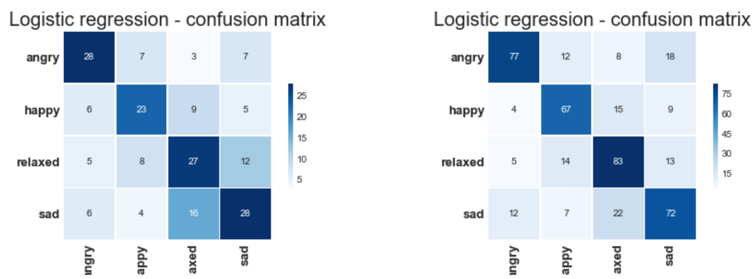
5-fold Cross Validation Accuracy				
Dataset	ANN	LR	SVM	xgboost
MoodyLyrics4Q	51%	55%	59%	56%
Both together	67%	68%	69%	63.7%

Table 5.1: Emotion detection accuracies



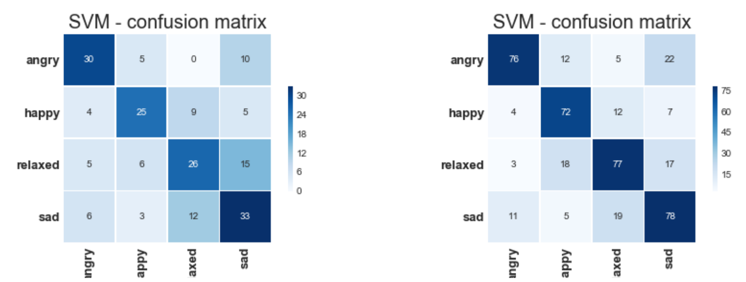
(a) ML4Q (b) ML + ML4Q

Figure 5.2: Artificial Neural Network - Confusion Matrix



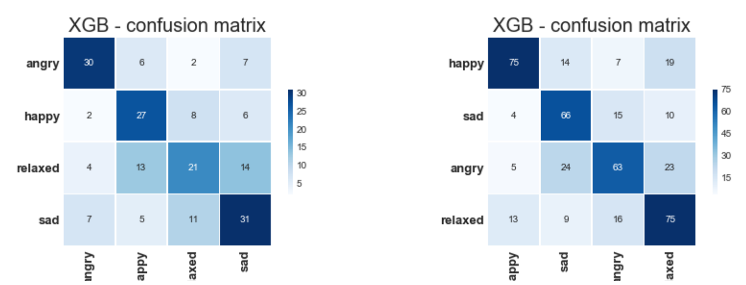
(a) ML4Q (b) ML + ML4Q

Figure 5.3: Logistic Regression - Confusion Matrix



(a) ML4Q (b) ML + ML4Q

Figure 5.4: Support Vector Machine - Confusion Matrix



(a) ML4Q (b) ML + ML4Q

Figure 5.5: Xgboost - Confusion Matrix

Bibliography

[1] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. *Sentiment Analysis Algorithms and Applications: A Survey*. Ain Shams Engineering Journal, 1994.

[2] Shiv Naresh Shivhare1 and Prof. Saritha Khethawat. *Emotion Detection From Text*, 2012.

[3] Emotion detection and recognition from text using Deep Learning, link

[4] IBM Watson: Natural Language Understanding APIs
`https://www.ibm.com/watson/services/natural-language-understanding/`

[5] IBM Watson: Tone Analyzer
`https://www.ibm.com/watson/services/tone-analyzer/`

[6] QEmotion
`http://www.qemotion.com/demo.php`

[7] MoodyLyrics: A Sentiment Annotated Lyrics Dataset
`https://www.researchgate.net/publication/317031495_MoodyLyrics_A_Sentiment_Annotated_Lyrics_Dataset`

[8] Shared Task on Emotion Intensity (EmoInt)
`http://saifmohammad.com/WebPages/EmotionIntensity-SharedTask.html`

[9] EmoBank
`https://github.com/JULIELab/EmoBank`

[10] Fell, Michael and Caroline Sporleder. "Lyrics-based Analysis and Classification of Music." COLING (2014).

[11] SpaCy - Industrial-Strength Natural Language Processing in Python
`https://spacy.io/`

[12] word2vec - Tool for computing continuous distributed representations of words
https://code.google.com/archive/p/word2vec/

[13] TextBlob: Simplified Text Processing
http://textblob.readthedocs.io/en/dev/index.html