

# Report template for the project in the course DD2380 at KTH

## GROUP31

Akash Patel	Christoph Kaiser
921119-3256	920416-T336
19/11/92	16/04/92
akash@kth.se	ckai@kth.se

Lisa Schmitz	Nikolaos Tatarakis
920606-T203	
06/06/92	BIRTHDATE4
lschmitz@kth.se	MAIL4@kth.se

## Abstract

This paper explores the potential accuracy of the analysis of song lyrics. Different text analysers were tested for their ability to categorize lyrics as *negative* or *positive*. The focus lies on the comparison of different feature extraction methods and classifier. The identification of emotion in lyrics is a problem which has no satisfying solution yet.

## NOTE

Related Works, Experimental Results, Discussions, Summary are sections that MUST be contained.

The section *Contributions* is a place to express any difference in contributions. The default assumption is that you all agree that all of you had an equal part to play in the project.

## 1 Introduction (1–2 pages)

Music has a great impact on people. Everyone knows the phenomenon that a song can influence our mood, it can make us sad and it can make us happy. This amazing control over people's feelings is something which can be used for many different purposes. For example music provider like Spotify offer playlists labelled with a certain mood. But industries are not the only area of application. Researchers see a use for it in edutainment and even psychological therapy [?]. Unfortunately, the task of predicting the correct assoziated mood is not an easy one due to the comlexity of how emotion is transferred in songs. Obviously emotion is encoded both in the audio and the lyrics of a song [?]. This paper compares methods to identify emotion by analysing the text of song lyrics. In order to do this different variants of text analysers were tested. The modification was conducted by using different categories of emotions and classifiers.

### 1.1 Contribution

### 1.2 Outline

Since our work deals with different approaches of categorising and classifying song lyrics, previous work should be taken into count. The related work is therefore presented in Section ???. We based our text analyser in the results of this previous studies. Section ??? explains the method we used to realise and implement the analyser in detail. We used different variations of our text analyser, modifying the the categorisation and the classifier. The results we were able to gather are described in Section ???. Moreover, problems that came up during the research are mentioned in this section. The results are sumarized in Section ?? and possible further research areas are given eventually.

## 2 Related work

Different bag-of-words model classifiers are explained by Li et al. [?].

Bla bla  
bla  
bla bla bla bla bla bla bla bla bla bla bla bla bla

The basic model we used for our text analyser is the bag-of-words model [?]. The functionality is illustrated in the figure ?? . In this model a database with labeled data such as lyrics are stored. The next step would be to preprocess the lyrics and create a dictionary with  $n$ -grams based on it. The database is divided into two parts: a training and a test set. Using this data, a classifier is trained with a training set and its accuracy can then be evaluated with the classification of the test set. In the following abstracts the individual parts of the bag-of-words model will be explained in detail.

3

Even though this prelabeling of the songs was helpful, with over ... emotional labels it were simply too many categories to use. Therefore we had to find supercategories for the existing labels. Downie et al. suggest in [?] to use five clusters of emotions. Whereas Yang and Lee only suggests a binary distinction into positive and negative emotions [?]. Both variants were tested during our research.

**Pre-processing** The lyrics have been pre-processed the same way for each of the variants of the text analyser. Stop-words are filtered out using the Natural Language Toolkit and special character are deleted. In a next step, every word of the lyric is stemmed which means that it is transferred into its basic form. For example "wait" and "waiting" would be interpreted as the same word after the stemming has been done. The pre-processing is performed due to avoid misclassifications.

**Feature extraction** The feature extraction is conducted for all lyrics. The procedure consists of transferring the lyrics into  $n$ -gram representation. We considered only 1-, 2- and 3-grams as reasonable parameters which is why only these have been tested in our study.

**Classification** The Natural Language Toolkit and the SciKit-Learn library provide the most commonly used text classifiers. In the following section the tested classifiers will be introduced.

- *Naive Bayes* Let  $\mathcal{C} = (c_1, \dots, c_n)$  be the categories and  $n$  the number of categories. Given a document  $\mathcal{D}$  and its word list  $\mathcal{W} = (w_1, \dots, w_m)$  with  $m$  being the number of words of a certain document, the Naive Bayes classifier determines the category of a document as follows:

$$c_{NB}^* = \underset{c_j \in \mathcal{C}}{\operatorname{argmax}} P(c_j) \prod_{i=1}^d P(w_i | c_j)$$

with  $P(c_j)$  being the *a priori* probability of a class  $c_j$  and  $P(w_i | c_j)$  being the conditional probability of the word  $w_i$  given class  $c_j$ .

Naive Bayes will only work well for large enough datasets.

- *Multinomial Naive Bayes* Multinomial Naive Bayes extends the Naive Bayes algorithm to handle multinomial distributed data. In text analysis this is particularly useful where events represent the occurrence of a word in a document.

A feature vector  $w = (w_1, \dots, w_n)$  is given where  $w_i$  counting the number of times a word  $i$  was observed in a particular instance. The likelihood of observing a histogram  $\mathbf{w}$  is then given by,

$$P(\mathbf{w}|c_j) = \frac{(\sum_i w_i)!}{\prod_i w_i!} \prod_i p_{ji}^{w_i}$$

Here  $p_{ki}$  is part of a multinomial distribution  $p_k$  and the probability that a word  $i$  is observed.

- *Bernoulli Naive Bayes* In contrast to the Multinomial NB the Bernoulli NB uses binary term occurrence rather than term frequencies which allows it to penalize non-occurring terms. Although there may be multiple features, each one is assumed to be a binary valued variable. Thus samples are represented as binary feature vectors. The likelihood of a feature vector given a category is given by,

$$P(\mathbf{w}|c_j) = \prod_i p_{ji}^{w_i} (1 - p_{ji})^{1-w_i}$$

The model is particularly useful for shorter texts where the lack of certain features can tell you more than solely analyzing occurring features.

- *Logistic Regression* Logistic regression is a linear model for classification where the probabilities describing the possible outcomes of a single event are modeled using a logistic function. This model is then used to classify new input data.
- *Support Vector Machine* If input data are overlap and have similar features, SVMs can be used. They essentially perform non-linear separations by transforming input data to higher dimensions with so called Kernels. This is then combined with a Lagrange constraint to find hyperplanes which separates classes with the largest distance.
- *Linear Support Vector Machine* The main difference with linear SVM compared to kernel SVMs is that the input data is linearly separable and thus a non-linear kernel to transform data to an higher dimension is not necessary. They are particularly useful for data which have a lot of features and thus easier and faster to separate with a hyperplane.
- *Stochastic gradient descent* This classifier combines multiple binary classifiers to separate features in input data. These classifiers are then

updated using a version of gradient descent to update the parameters and minimize the error. In contrast to normal gradient descent the stochastic version updates the classification parameters for every training example it encounters with respect to the gradient of the error. Traditionally one has to run through the entire training set before calculating the gradient and updating the parameters. This method allows for a faster learning rate compared to batch gradient descent.

- *Combination of multiple classifiers* We combined all introduced classifier into one classifier. It basically classifies documents according to how many classifiers resulted into a certain category. The category, to which most of the classification results belong to, gets selected as the resulting category of the song.

## 4 Experimental results

The goal of our experiment was to compare the performance of different variants of the bag-of-words model classification for emotion in lyrics.

### 4.1 Experimentl setup

In order to compare the performance of the different classifier variants we changed various parameters. The effect of different amounts of emotional categories have been tested, as well as variations of the  $n$  of the  $n$ -grams, the size of the trainingsset and classifiers.

### 4.2 Experiment ...

**Drawbacks** We were aware of the difficulty of the task of designing a sufficient lyrics classifier. Especially in the last years a lot of research has been conducted in this area but there is still a lack of a reliable lyrics analyser. But during our research we were able to gain a better inside into the nature of this difficulty. Feelings and emotions are something very personal, and the perception of the emotion which is transferred by a certain songtext highly depends on the person who is reading the lyrics. It is not unusual that two different people might disagree on the general mood of a lyrical text [?]. This is why it is of supreme importance to select a representative model of categories.

Clusters	Mood Adjectives
Cluster 1	passionate, rousing, confident, boisterous, rowdy
Cluster 2	rollicking, cheerful, fun, sweet, amiable/good natured
Cluster 3	literate, poignant, wistful, bittersweet, autumnal, brooding
Cluster 4	humorous, silly, campy, quirky, whimsical, witty, wry
Cluster 5	aggressive, fiery, tense/anxious, intense, volatile, visceral

Table 1: Clusters of mood adjectives used in the MIREX Audio Mood Classification task [?].

**Variations in categorisation** During our research we came to the conclusion that the use of only two categories - positive and negative - seem to have led to an overgeneralisation of emotions. Even the categories themselves which were provided by Allmusic.com could not clearly be mapped to one of these two categories. To illustrate the difficulty we discovered, a research has been done. We used a questionnaire where we asked people to evaluate the categories from Allmusic.com as either negative or positive and to do the same with some randomly picked song lyrics. The results are shown in figure ???. It can be seen that there is a significant discrepancy of opinion across participants. This emphasizes the personal perception of emotion. Due to this, it cannot be expected to gain a higher accuracy by a classifier in emotion categorisation.

A second model of categorisation used in this research has been proposed by Downie et al. [?]. They used five clusters of emotion to categorise lyrics as shown in table ???.

**Variations in n-grams** Another parameter we modified was the assignment of  $n$  for the  $n$ -grams. The effect on the accuracy of the classifiers is illustrated by the figures ?? and ??.

### Variations in the size of the training set

**Variations in classifiers** We tested the different classifiers described in ???. The performance of different classifiers for the two mood variants are shown in ?? and for the cluster variant are presented in ??. As it can be seen in the figures, the ... classifier yields the best average results for both categorisation variants. The reason why the linear classifier is working well is that the tested data is highly individual. There is a significant variance

between song lyrics and this is why a linear distinction is the best choice for the data in the present case.

[illegible]

We the members of project groupXX unanimously declare that we have all equally contributed toward the completion of this project. (PLEASE CHANGE THIS SUITABLY WITH DETAILS, IF IT IS NOT TRUE)