Project Outline

Project Topic: Text Analyser for lyrics

Related Work

- Two-dimensional lyric identifier using the valence-arousal model.
  [Van Zaanen, M., & Kanters, P. (2010, August). Automatic Mood Classification Using TF* IDF Based on Lyrics. In *ISMIR* (pp. 75-80).]
- Overview of current methods for music identifying methods. Different category models and classifier are introduced.
  [Kim, Y. E., Schmidt, E. M., Migneco, R., Morton, B. G., Richardson, P., Scott, J., ... & Turnbull, D. (2010, August). Music emotion recognition: A state of the art review. In *Proc. ISMIR* (pp. 255-266).]
- Uses Allmusic.com as lyrics database. Allmusic already classifies songs according to their emotion. This paper gives a reference which of the emotions are positive and which ones are negative. Our classification is based on this paper.
  [Yang, D., & Lee, W. S. (2009, December). Music emotion identification from lyrics. In *Multimedia, 2009. ISM'09. 11th IEEE International Symposium on* (pp. 624-629). IEEE.]

Problem Definition

The approached problem in our project work is to categories lyrics according to their mood. A lot of research has been done in this area and since lyrics are very complex it is difficult to find appropriate categories. Some research paper addressed the problem by using only two categories: positive and negative. Therefore it seems legitimate to start with these two categories. Further category models like the valance-arousal model might be implemented in a later state of the project.

Implementation

The following abstract will present the implementation details of the different parts of the text analyser, that have been introduced in the project proposal. The implementation of the analyser itself will be done in Python.

A. Dataset: A ready to use dataset could not be found on the Internet due to copyright issues. Therefore it is necessary to get the data manually or by a program. Moreover, appropriate emotion categories for the lyrics were needed. Since labelling them manually would neither very efficient nor objective, the database should already provide labels.
This is the reason why we implemented a Java program which successfully extracts lyrics from the website **Allmusic.com**. This websites provides music categorised by its mood. Moreover, this website has been used by one of the papers mentioned above. Since more than 40 emotion categories are used by the website, these had to be put into the two super-categories positive and negative.
The result is a dataset of a few thousands labelled songs which can be used for our text analyser.

For the following steps, the **Natural Language Toolkit (NLTK)** for Python is used. It provides stop-words of different languages, stemmer and lemmatizer and methods for feature extraction, as well as appropriate data structures for the dictionary.

B. Pre-Processing: Using toolkits like SciKit-learn or NLTK we will filter common words that would end up in creating confusing histogram representations for the classifier.

C. Feature Extraction: We will evaluate our system for different features/ different n-gram models.

D. Histogram Representation: There are some ready to use toolkits that actually computes the histograms, like vectorizer() from SciKit learn given a dictionairy.

E. Classification: NLTK provides also a very simple naïve bayes classifier. But more than one classifier should be tested and therefore it is reasonable to choose another library for this specific problem. One choice would be the **SciKit-Learn** library for Python. Some of the provided classifiers are:

- Multinomial naïve Bayes
- Gaussian naïve Bayes
- Bernoulli naïve Bayes
- Logistic regression
- Stochastic gradient descent
- Support vector machine
- Linear support vector machine
- Nu-support vector machine

Usually, the classification results on a task like this are given in the form of a confusion matrix (especially using SVMs):

Confusion Matrix:

$$M_{i,j} = \frac{\left|\{I_k \in C_j : h(I_k) = i\}\right|}{\left|C_j\right|}$$

where $i, j \in \{1, \dots, N_c\}$, C_j represents the total number of test songs from class j, $h(I_k)$ is the category with the highest success for the song I_k . For example:

		Predicted Class		
Some Data Set		**Emotion1**	**Emotion2**	**Emotion3**
Actual Class	**Emotion1**	**0.525**	0.35	0.125
	Emotion2	0.3	**0.55**	0.15
	Emotion3	0.175	0.125	**0.70**
Mean Diagonal Accuracy : 59.16%				

Table 1. Confusion Matrix Example