Project Proposal

Project Topic: Text Analyser

Specific Problem Formulation

The aim is to design a text analyser to classify texts according to their transferred emotion. In a first step the analyser should be able to categorise song lyrics correctly. In an additional step the text analyser should be made applicable to other kinds of texts such as poems.

The task of identifying emotions is of deep interest to many researchers. It cannot only support research in other fields like human computer interaction and computer linguistic, but can also be useful for market analyses or educational games [Strapparava, C., & Mihalcea, R. (2008, March). Learning to identify emotions in text. In *Proceedings of the 2008 ACM symposium on Applied computing* (pp. 1556-1560). ACM.].

Project timeline

When	Who	What	Time spent
Week 39	Akash and Lisa	Find datasets, do research on different kinds of dataset	12h
	Chris and Lisa	Find and compare libraries for feature extraction	12h
	Niko and Chris	Find classifier libraries	12h
	Akash and Niko	Create code skeleton (interfaces)	16h
Week 40	Niko and Chris	Define Tests	12h
	Akash and Chris	Construct a dictionary	16h
	Niko and Lisa	Implement bag of word	16h
	Akash and Lisa	Integrate all components	16h
Milestone: Implemented a Prototype			
Week 41	Akash and Niko	Benchmarking	8h
	Akash and Lisa	Test different datasets	16h
	Niko and Chris	Compare classifiers	16h
	Chris and Lisa	Compare feature extraction	16h
Week 42	Akash and Niko and Lisa and Chris	Refactoring and optimization	32h
		Backup Time	24h
Sum			224h

Project setup sketch

Brief description of sub systems

A) **Dataset**: Includes the main corpus of digital text that we are going to work with. Typically, a data set consists of a Training set along with its labels. Additionally, it contains a Test set where the evaluation of the proposed system will be performed and its robustness will be assessed. Finally, the “Dataset” procedure also includes all these actions that needs to be taken (from programing perspective) in order to assure that each document from the training will be read and stored correctly in a repository along with its own label. Same procedure for Test set, however there are no labels here.

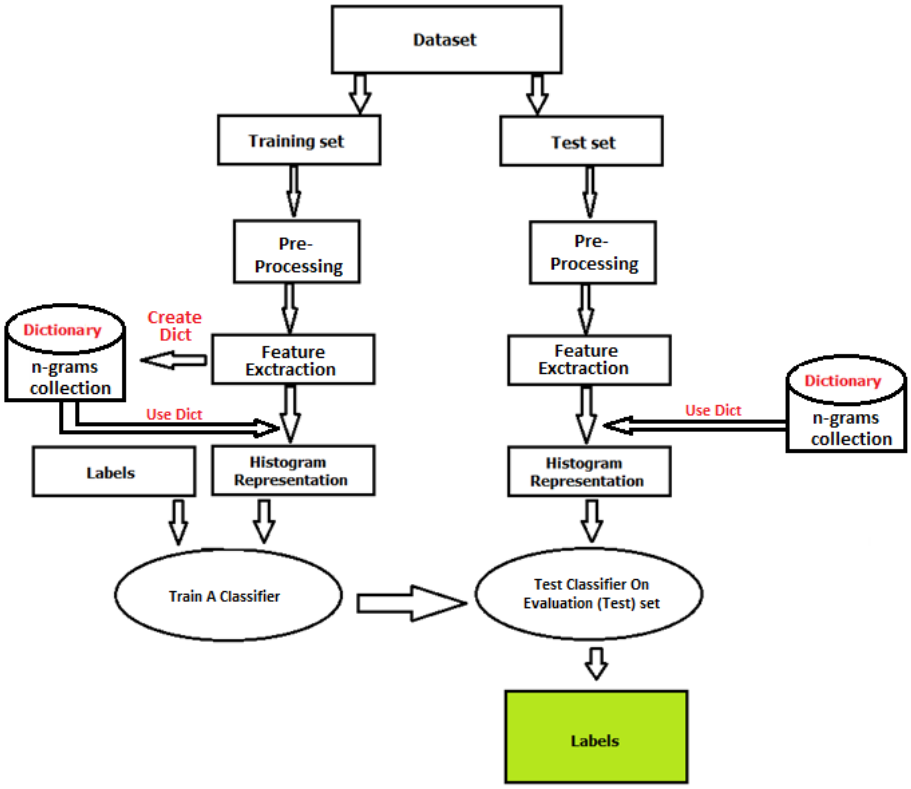

Figure 1. Diagram of the Suggested System

B) **Pre-Processing**: Common words like “is, this, it, a, the” etc. or numbers should be filtered out n order to avoid misclassifications and poor overall performance. This also includes any other actions that prepare our raw text data in a more appropriate form for the next step so that we can minimize the classification error.

C) **Feature extraction**: In this step we will have to extract features from each document. We will have to choose an n-gram model for this procedure (1, 2, 3-gram etc.). This is basically a transformation of our textual data into feature vectors that machine learning algorithms can understand, i.e. sequences of numbers extracted from textual features (words).

D) **Dictionary**: This contains a collection of n-gram features extracted from the training set (previous step). Contents of the dictionary are unique, no duplicates.

E) **Histogram Representation**: It is also called the Bag of Words model. This is basically a statistical representation of our initial corpus. It counts the occurrences of each word on the dictionary found in a given text.

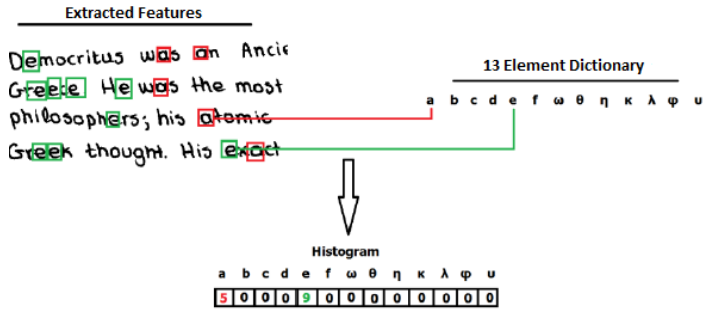

Figure 2. Simplistic Histogram Representation of a given text on letter level. In a real world situation we will have to deal with texts on word(s) level features, n-grams.

F) Classification: This is the final step. We are going to use the histograms from the training set with their own labels and feed it to a classifier. This is the training step. Once training is done we will use the test histograms to the trained classifier in order to predict their labels and score the system.