TDT4310/2021S: Intelligent Text Analytics and Language Understanding Prof. Björn Gambäck

Lab Exercise 3

Lab Date: 19th of February 2021

1. **Guidelines**

   Deadline for submitting your solution: **23:59 4th March 2021**.
   Submission is a zipped folder with {*your name*} containing:

   (a) Source files (Python): format name as Lab{LabNumber}_{Exercise Number}.py.

   (b) A summary/report file formatted as a pdf which explains and presents the results with respect to the input values.

2. **Exercises**

   Exercise 1: This task is about gender detection, using the data from *NLTK corpus.names*. The data set contains 5000 male names, and 3000 female names. With the same training and test data (where you choose the ratio) and the same feature extractor, build three classifiers for the task.

   - Decision Tree
   - Naïve Bayes
   - Maximum Entropy

   Write code to compare the performance of the three classifiers on the task. How do you think that your results would change if you used a different feature extractor?

   Exercise 2: Tweet Classifier. With some Tweet Corpus from two Twitter accounts[1], vectorize the tweets (e.g. with Count Vectorizer or TF-IDF Vectorizer). Predict which account an input tweet is from and its probability.

   Exercise 3: Word features can be very useful for performing document classification, since the words that appear in a document give a strong indication of what its semantic content is. However, many words occur very infrequently, and some of the most informative words in a document may never have occurred in our training data. One solution is to make use of a lexicon, which describes how different words relate to each other. Using the *WordNet* lexicon, augment the movie review document classifier from the *NLTK book*[2] to use features that generalize the words that appear in a document, making it more likely that they will match words found in the training data.

   Exercise 4: Large websites are an ideal place to look for large corpora of natural language. In this exercise you're going to clean the textual data from the titles and descriptions of Youtube videos to classify their category.(i.e. Entertainment, Gaming, Anime etc)

   The task is solved by using the provided code skeleton. There are *TODO* comments to show which parts of the code you need to edit. The exercise consist of three tasks.

   Remove Noise

   Removal of stop words

   Normalize text

   If you want an extra challenge, there are two more tasks you can try. These are completely voluntary, and not mandatory to pass this lab.

   Collecting your own titles and descriptions from YT, to predict the most common content from your favorite channel [3]

   Performing data cleaning[4]

[1] If you have problems getting tweets, you can borrow from this repository.

[2] Movie review classifier from NLTK book.

[3] Google provides their own Youtube Data API that you can use to collect titles and descriptions from videos. It is also possible to webscape with f.ex selenium.

[4] The difference between textual cleaning and data cleaning is that data cleaning preceeds textual cleaning. You want to remove all "bad" data points, and after that you want to clean/normalize the text to improve the feature extraction.