|  |  |  |
| --- | --- | --- |
| **Assignment 1** | **Assigned:**  **1/4/2021** | **Due to:**  **16/4/2021 23.59** |

**Exercise 01 [3/100]:**

1. Write a program (**counter.py**) that prints the 50 most frequent bigrams of an English corpus, omitting bigrams that contain stopwords and punctuation. A variable “corpus\_path” should specify the corpus; i.e. the path to a folder that contains text files.
2. The program should work for at least a corpus of 20 different files that each contains at least 500 words.
3. The program should contain comments that explain how is works

**Exercise 02 [2/100]:**

1. Create a text file and put some English text. At least 500 words
2. Write a **compare\_stemmers.py** that
   1. Loads the contents of the text file and tokenizes it
   2. Uses the Porter Stemmer on each word (ignore punctuation)
   3. Uses the Lancaster Stemmer on each word (ignore punctuation)
   4. For each word print the word, the Porter stem and the Lancaster stem in the same line (ignore punctuation)
   5. See if you observe any differences. Write your observations in a text file (**report.txt**)
3. The program should contain comments that explain how it works

**Exercise 03 [2.5/100]:**

1. Copy the python files **train.py, dataset.py** of lecture 04. The train.py generates TF-IDF vectors of N features for the phrases of a dataset and trains a K-NN model from them.
2. Create a function (is a separate **utils.py**) that takes as parameters, a matrix of TF-IDF vectors and the respective phrases. The function should POS tag each phrase count how many ADJ and ADV occur (universal tagset) and add two features in each vector. Use position N and N+1 for the two POS features. The new vectors should be returned by the function.
3. Use the function and train K-NN models with N=100+2, 1000+2, 5000+2, 10000+2; i.e., the TF-IDF features and the 2 POS tagged features. Save the model using pickle. Use as input the training part of the Kaggle dataset that we have used in lecture 4.
4. Copy **predict**.**py** and extend it so that it creates features vectors for the test part of the Kaggle dataset (the same that we have used for lecture 4). Compute the predictions for the test vectors (N=100+2, 1000+2, 5000+2, 10000+2) and calculate accuracy, precision, recall, F1. Compare with the respective models that just uses TF-IDF features. Write your observations in **report.docx.**
5. The program should contain comments that explain how it works

All the code and text files should be put in one project folder and sent via eclass as a zip. The assignment is to be done in groups of two. In the zip include a students.txt file with your names.