CSE 586 Information Retrieval , Spring 2019

Assignment #2 – Spam E-mail Filtering

In this assignment you will implement a Multinomial Naive Bayes classifier for spam email filtering. You will use a subset of the Ling-Spam corpus[1] to train and test your system. The provided training and the test sets (included in the *dataset.zip file*) each contain 240 spam and 240 legitimate email messages. Each email message is provided as a separate file. All files start with a "subject:" heading. Stopword removal and lemmatization have already been performed.

Preprocess the files by extracting the individual words from them. Assume that a word consists of letters from the English alphabet. Discard all tokens that contain different characters such as digits, punctuation marks, or other special symbols (e.g. $).

Perform feature selection by using the Document Frequency (DF) thresholding approach. That is, compute the DF score for each word in the training set (i.e., the number of documents that contain the word) separately for the spam and the legitimate classes. Then, select the top 200 words for the spam class and the top 200 words for the legitimate class and use these as features with your Naive Bayes classifier. Test your classifier by using the provided test set.

Submission: You should submit a *".zip"* file named as YourNameSurname.zip containing the following files.

1. **Report**:

   (a) Briefly describe your work (i.e., the steps in your pipeline such as preprocessing, learning the parameters, smoothing, testing).

   (b) List the top 200 words and their DF scores for each class.

   (c) Examine the lists of words and discuss the following: Do you think the selected words make sense, given the task. Are there words which are common to both classes. Are there words which you think are not good features, and should not have been selected. Are there words which should have been included as features, but were missed.

   (d) Report the *macro-averaged* and *micro-averaged* precision, recall, and F-measure values obtained by your system on the test set, as well as the performance values obtained for *each class separately without using smoothing*

   (e) Report the *macro-averaged* and *micro-averaged* precision, recall, and F-measure values obtained by your system on the test set, as well as the performance values obtained for *each class separately* by using *Laplace smoothing* with α = 1.

2. **Commented source code and readme**: You may use any programming language of your choice. However, I need to be able to test your code. Submit a readme file containing the instructions for how to run your code.

[1] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, George Paliouras, and C.D. Spyropoulos, "An Evaluation of Naive Bayesian Anti-Spam Filtering". In Potamias, G., Moustakis, V. and van Someren, M. (Eds.), Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML 2000), Barcelona, Spain, pp. 9-17, 2000.