README - IR Project 2 - Steven Van Damme

General Remarks:

The complete code can be found in the "src" folder of the submitted zip file.
The package vsteven.irproject2 contains the code I implemented myself to complete the tasks. It contains the following files: Main, NaiveBayes, LogisticRegression, SVM and PrecisionRecallF1.

The Main file is used to run the project. Depending on its input argument, it will run one of the three classifiers. NaiveBayes, LogisticRegression and SVM contain the code for those three classifiers respectively. The PrecisionRecallF1 file is used to evaluate the solution of the labelled test set.

The 6 output files are named as requested and also contained in the zip file.

Approaches:

In all three approaches, I first remove the stop words and do stemming on the tokens and I compute for each topic in topic_codes.txt a one-vs-all classifier.

1) NaiveBayes

The classifier is represented by the formula from slide 18 of lecture 6.

$c(d) = argmax[log\,P(c) + \sum_w (tf(w;d) * log\,P(w|c))]$

P(c) is computed with the formula from slide 17: $P(c) = \#\{d : d \in c\}/\#\{d\}$),
while for P(w|c) I use Laplace Smoothing and use the formula from slide 20 with $\alpha = 1$:

$P(w|c) = (\sum_{d \in c} tf(w;d) + \alpha)/(\sum_{d \in c} len(d) + \alpha\#\{w\})$

In the train method, we only compute the numerator and the denominator. This idea is inspired from the code provided on slides 21 and 22.

In the classifier functions, I then compute c(d) for each document and each topic. At this point I combine the numerator and
denominator of P(w|c) computed before. Finally I got the best results by selecting the 4 topics with highest score.

2) Logistic Regression

The classifier formula is represented by the top formula on slide 26 of lecture 6, if you remove the b (for simplicity).
$P(c = 1|d;\theta) \ = \ 1 / (1 \ + \ exp[-< x_d, \theta >])$
As it's stated there, for this classifier to work, theta needs to be given. Theta is obtained by training it on the training data set. There is a different theta for each topic and each theta is initially 0. The update step used during the training for theta can be found on slide 29 of lecture 6. One parameter of this update step is η_t. I found the best possible value for that parameter is $\eta_t \ = \ 0.1 / t$, with t being the iteration number.

Once the theta is trained the classifiers just need to compute the above classification rule. The implementation is inspired by the implementation provided on slide 30 of lecture 6. (logistic function) Here we pick for each document the 3 topics with the highest value.

3) SVM

The classifier formula is $c(d) \ = \ < \theta, x_d >$ which can be found in the section of SVM in the slides 34-38. The goal is to maximize this function.
Here again a theta is needed. This time it is again initialized with 0 and updated on each step of our training function. The update algorithm used is Pegasos as suggested in the lecture slides. The formulas for the update step can be found on slide 37. The parameter lambda was set to 0.2, this reached the best result. More testing was not possible since the computation became longer and longer the smaller lambda got. I assume the F1 score will even get better with smaller lambda.

With the trained theta, the test data can again be classified, we take again the 3 topics for each document that reach the highest
value with the classification formula.

Instructions:

To run the project, one just needs to import the src folder into a Scala project.
In the Main.scala file, the values of the paths of the different input files need
to be changed.

Line 23: the path to the topic_codes.txt file
(e.g. D:/Projects/Scala/Assignment2/topic_codes.txt)

Line 24: the path to the folder containing the zip files of the training set
(e.g. D:/Projects/Scala/Assignment2/data/training/train)

Line 25: the path to the folder containing the zip files of the labelled test set
(e.g. D:/Projects/Scala/Assignment2/data/test-with-labels/test-with-labels)

Line 26: the path to the folder containing the zip files of the unlabelled test set
(e.g. D:/Projects/Scala/Assignment2/data/test-without-labels/test-without-labels)

As arguments to the VM, the same as for the last project were used:
-Xss400m -Xms4g -Xmx4g -XX:-UseGCOverheadLimit

Program arguments should be either 0, 1 or 2 with 0 = NaiveBayes, 1 = LogisticRegression and 2 = SVM