4.- Supervised Methods – Maximum Entropy Classifiers

4.1

a) *Use the encoded corpus* corpus/efe/f50/train.0 *to learn a Maximum Entropy Model using the* megam_i686.opt *executable:*

   ./megam_i686.opt -quiet -fvals multiclass corpus/efe/f50/train.f0 > f50.mem

b) *Test the performance of the module on the corpus* corpus/efe/f50/test.f0 *using megam test mode:*

   ./megam_i686.opt -fvals -predict f50.mem multiclass corpus/efe/f50/test.f0 >out

c) *Complete the program* classifier.py *to compute the probability of each class for each input example, and produce the same output than* megam *test mode. Use the correct answer in the test files to compute the accuracy statistics.*

4.2

a) *Modify the program* classifier.py *to output not only the most likely class, but all classes with a probability over a given threshold. Modifiy the evaluation to compute also precision, recall, and F1. Check how results vary depending on the given threshold.*

b) *Train and test a classifier using the corpus* corpus/efe/f100/train.f0 *for training and the corpus* corpus/efe/f100/test.f0 *for testing. Compare the performance of this classifier with that of the classifier obtained in the previous exercise using corpus* f50. *Perform a hypothesis test to find out whether the difference is statistically significant.*

c) *Perform a cross-validation evaluation for the same cases above, using corpus* corpus/efe/f50/train.* *and* corpus/efe/f50/test.* *to train and test five folds of one classifier, and* corpus/efe/f100/train.* *and* corpus/efe/f100/test.* *for the other. Discuss the changes in the statistical significance of the difference between both models.*