Hw10

Hw10: Build a maxent tagger

- Q1: maxent_tagger.sh train_file test_file rare_thres feat_thres output_dir
- The format of train_file and test_file: w1/t1 … wn/tn

- Main steps:
  - Create feature vectors for train_file and test_file
  - Run “mallet import-file” to convert training feature vectors into binary format
  - Run “mallet train-classifier” to create a MaxEnt model
  - Run “mallet classify-file” to tag the test data

Creating feature vectors

- Features: Table 1 in (Ratnaparkhi, 1996)

- Use rare_thres to identify rare words in the training data and in the test data
- Remove low-frequency features from the feature vectors using feat_thres.
- Replace ``,'' with ``comma'' as Mallet treats ``,'' as a delimiter.

Q2: tagging results

Table 1: Tagging accuracy with different thresholds

Expt id	rare thres	feat thres	training accuracy	test accuracy	# of feats	# of kept feats	running time
1_1	1	1					
1_3	1	3					
2_3	2	3					
3_5	3	5					
5_10	5	10					

Output files

- Store under res_id/ (e.g., res_1_1/)
  - train_voc: “word freq”
  - init_feats: “featName freq”
  - kept_feats: “featName freq”
  - final_train.vectors.txt and final_test.vectors.txt
  - me_model: MaxEnt model
  - me_model.stdout and me_model.stderr: redirected stdout and stderr during training
  - sys_out: system out when running “mallet classify-file”

- Submission:
  - Submit res_id/ only for the 1st row and the last row.
  - If the compressed file is still too big, let the TA know the location of the tar file on patas, and make the file accessible. The timestamp of the file will be used to determine whether it is submitted on time.