LING572 Hw7: TBL
Due: 11pm on Feb 27, 2019

The example files are under dropbox/18-19/572/hw7/examples/.

Q1 (35 points): Write a TBL trainer, **TBL_train.sh**, for the text classification task.

- The command line is: TBL_train.sh train_data model_file min_gain
- The initial annotator simply tags each document with the **first** class in the training data (e.g., if the training data is **train2.txt**, the first class would be **"talk.politics.guns"**).
- train_data has the same format as before (see **train2.txt**)
- model_file has the default classname (i.e., the first class in the training data) in the first line, followed by a list of transformations (one transformation per line). The transformation line has the format "featName from_class to_class net_gain".
- min_gain should be a positive integer. If it is not, the code should print out an error message and exit.
  - If the net gain of the best transformation for the current iteration is less than min_gain, the TBL training will stop.
  - For instance, if min_gain is 1, the trainer will not stop until the best transformation in the current iteration cannnot provide a positive gain. In this case, the model file contains all the transformations with positive gains.
- In order to find the best transformation, you need to go over all the instances **including** the ones whose current class labels are correct.

  If your implementation is efficent, for every iteration of training, you need to go over the training data only once to find the best transformation. The trick is that for each training instance, determine what transformations would be triggered by the instance and update their net gains accordingly. See the slides for hw7.

Q2 (25 points): Write a TBL decoder, **TBL_classify.sh**, that uses a TBL model to classify test instances.

- The command line is: TBL_classify.sh test_data model_file sys_output N
- test_data has the same format as before (see **test2.txt**)
- model_file is the model created by TBL_train.sh
- The format of sys_output is "instanceName trueLabel sysLabel transformation1 transformation2 ...":
  - trueLabel is the label in the gold standard

  - sysLabel is the label produced by the TBL classifier
  - each transformation has the format “featName from_class to_class”
  - transformation1 is the first one applied to the instance, tranformation2 is the second, and so on.

- N is the number of transformations in the model_file that will be used. For instance, suppose the model file has 1,000 transformations and N is 10, then only the first 10 transformations in the model file will be used for decoding, and the rest will be totally ignored as if they were not in the file.

Q3 (15 points): Run **TBL_train.sh** with **train2.txt** as the training data, and run **TBL_classify.sh** with **train2.txt** and **test2.txt** as the test data for training accuracy and test accuracy, respectively.

(a) Fill out Table 1. N is the number of transformations used by TBL_classify.sh.

(b) For Table 1, you only need to run **TBL_train.sh** with min_gain=1, and use the model file for every row in Table 1. Let us call that model file **model_file**.

(c) When you run TBL_classify.sh on **test2.txt** with a value N, name the sys_output file as **sys_output_N**. You need to submit **model_file**, **sys_output_20**, **sys_output_50**, and **sys_output_100**.

Table 1: The classification results

N	Training Accuracy	Test accuracy
1		
5		
10		
20		
50		
100		
200		

Submission: Submit the following to Canvas:

- Your note file *readme.(txt | pdf)* that includes Table 1, and any notes that you want the TA to read.

- hw.tar.gz that includes all the files specified in dropbox/18-19/572/hw7/submit-file-list, plus any source code (and binary code) used by the shell scripts.

- Make sure that you run **check_hw7.sh** before submitting your hw.tar.gz.