Advanced Natural Language Processing and Information Retrieval

LAB 1: IR, Indexing, Frequencies and Text Categorization with the Text Classification Framework

Alessandro Moschitti

Department of Computer Science and Information Engineering
University of Trento
Email: moschitti@disi.unitn.it

Initialization

- Download TCF from
- http://disi.unitn.it/moschitti/TCF.tar.gz

- Set the path for executing TCF program
  - setenv PATH $PATH":bin"
  - setenv gamma 1

- Make directories needed for storing classifier partial and last results
  - mkdir temp // temp dir
  - mkdir CKB // classifier KB
  - mkdir CKB/cce // centroid for each category
  - mkdir CKB/splitClasses // file split (training set)
  - mkdir CKB/testdoc // file split test set
  - mkdir CKB/store // temporary directory
  - mkdir CKB/classes /category files

Building of Category counts

- ./bin/TCF UNI -RCclusteringCategories // learning file freq. Unification
- “clusteringCategories” is a directory containing the learning files, i.e.
  - 0000191 february 1 1
    0000191 in 1 1
    0000191 february 1 1
    0000263 alcan 1 1
    0000263 in 1 1
- output
  - 0000191 february 2 1
    0000191 in 1 1
    0000263 alcan 1 1
    0000263 in 1 1
- The output can be seen in the directory “classes”

Splitting and Centroid Building

- ./bin/TCF CCE -SP20 -SE1 #-DID/mnt/HD2/corpora/QC_testID.txt
  - Split of 20% with random seed 1
  - If you want to provide you own split -DID the path for a file containing in each line the numeric index of the document that you want put in the test-set
- The classes are split in *splitClasses* and *testdoc* directories
- Results in cce, e.g. for alumn.le.oce
  - about 16 9.000000
  - accelerate 1 1.000000
  - acceleration 1 1.000000
  - acceptance 1 1.000000

Global Centroid Building

- ./bin/TCF GCE -DF0 // sum the counts of all the centroids for each word
- The result is the file globalCentroid.le, e.g.
  - abandon 6
  - abandoned 5
  - abandoning 1
  - abated 1
  - abatements 1
- Moreover, if you specify DFx, only words with frequency greater than x will be used for later steps, i.e. in the Rocchio profile

Rocchio Profile Building

- IDF and TF are determined for each document
- Rocchio's formula is applied to the document of each category
- ./bin/TCF DIC -GA0
- GA is gamma where beta =1, so rho = gamma/1
- The profile weights are stored in the binary file Dict.Weight.le (which uses Dict.Offset.le to get the index)
- To watch the weight produced by Rocchio:
- Change Dir in CKB and run ../bin/printw x (where x is 0,..,n, i.e. the alphabetic position of the category)
  - wide: 0.00040450
  - widen: 0.00134680
  - widening: 0
  - wider: 0.00148100

Classification Step

- ./bin/TCF CLA -BP > BEP
- The document in testdoc are classified
- -BP means that the thresholds associated with the nearest BP are derived and the related performance computed.
- P, R, F1 for each category and the overall Micro/Macro evaluation for all categories are printed on the screen
- More over in the “thresholds” file we have this important data
  - 0.015625 0.015625 1.000000 0.928571
  - 0.004929 0.004929 1.000000 0.873950
  - 0.006836 0.006836 1.000000 0.914894
- Each line relates to a category (alphabetic order)
  - First and second columns are the minimum and max thresholds that produce the accuracy in the 4th column
  - The third column is the gamma used for the previous learning

Advanced Classification

- By providing the “thresholds” file you can use you own thresholds
- ./bin/TCF CLA
- In this case you can use your values in the second column

- To evaluate the Rocchio’s formula with a different gamma for each category we can use:
- ./bin/TCF DIC -GFgammaFileVector_medio