RunningPerceptron

Running the Perceptron to generate scores

Generating scores is a two step process: model validation and score generation. To prepare your environment for running perceptron, execute the following after setting CORPUS:

```
cd masses
rm -rf ORIG NSBASE SPBASE ham-validate.log spam-validate.log ham.log spam.log
mkdir ORIG
for CLASS in ham spam ; do
  cat $CORPUS/submit/$CLASS*.log > ORIG/$CLASS.log
  for I in 0 1 2 3 ; do
    ln -s $CLASS.log ORIG/$CLASS-set$I.log
  done
done
```

Model validation

Before generating the final set of scores, you need to pick a configuration for the training program. In order to do this, you run a series of "ten-fold cross validations" and use "Student's t-test" to compare their results. Should you be so inclined, you can also use ANOVA to compare result sets. This is left as an exercise to the reader.

In the masses directory, you will find a file called "config." As its name suggests, this is the configuration for the validate-model (and runGA) script. It consists of 5 fields: SCORESET, HAM_PREFERENCE, THRESHOLD, EPOCHS and NOTE. SCORESET is an integer between 0 and 3. Set 0 is for the ruleset with bayes and network tests disabled. Set 1 is for the ruleset with network tests enabled. Set 2 is for the ruleset with bayes enabled. Set 3 is for the ruleset with network tests and bayes enabled. HAM_PREFERENCE, THRESHOLD and EPOCHS correspond to options passed to perceptron. See its documentation. NOTE is appended to the name of the directory containing the result sets.

To refine your parameters, do an iterative process of editing the config file and then running validate-model. To compare the results of two runs using Student's t-test, use the "compare-models" script. Each result set will be stored in a directory of the form "vm-$NAME-$HAM_PREFERENCE-$THRESHOLD-$EPOCHS-$NOTE" and contains a file called "validate" which contains the aggregated results from the cross-validation. Run compare-models like so: ./compare-models vm-set0-2.0-5.0-100-before/validate vm-set0-2.0-5.0-100-after/validate

To speed things up, validate-models caches most of the compiled files. If you change your logs or any of the scripts that are used as part of compilation, you will need to rm -rf vm-cache.

Score generation

When you are happy with your configuration, set it in your config file and execute the runGA script. You will find your results in a directory of the form "gen-$NAME-$HAM_PREFERENCE-$THRESHOLD-$EPOCHS-$NOTE" runGA uses a randomly selected corpus with 90% being used for training and 10% being used for testing.