Name: Abiyaz Chowdhury

NLP Spring 2016 Project I Write-Up

* Instructions explaining how to use your system (including how to compile the code, if necessary).

Put the nlp1.cpp file in your TC\_provided directory, then compile and run the program. Enter the corpora files assuming you are in TC\_provided directory, and include only the filename (exclude dot slash).

* Which basic machine learning method did you use?

Naïve Bayes with unigrams (words) as the individual features.

* How does your system tokenize training and test files?

Converts all whitespace delimited strings to lowercase, then discards all punctuation other than hyphens. The punctuation is discarded without splitting the word, so the word “can’t“ would become “cant“. The punctuation discarded are: commas, periods, question marks, backslash, forward slash, apostrophe, parenthesis, single and double quotation marks.

* What weighting scheme, if any, is used for tokens?

None

* If you used Naïve Bayes, what method of smoothing is used?

Laplace. The default smoothing factor is 0.01. So to measure each unigram probability, 0.01 is added to the numerator, and 0.01\*V is added to the denominator where V is the vocabulary size of that category.

* Which optional parameters or features did you experiment with (e.g., possibilities might include case sensitivity, POS tagging, stop lists, etc.)? Which parameters or features made a significant difference, and how are they set in your final system?

The Laplace smoothing factor was varied. Optimal performance for the three documents was obtained in the range 0.01 to 0.1

The value used in the final program is 0.01 but you are welcome to adjust this yourself by accessing the program and adjusting the named floating-point variable “laplace”. No other crazy techniques were used.

* How did you evaluate your system's performance for the second and third data sets?

I used two-fold cross-validation and ran the program several times with different seed values for the RNG. To measure performance, I measured overall accuracy and eyeballed the contingency table. Using half the training set for training and the other half for testing (for both corpora 2 and 3), I obtained results of about 80% and 87% respectively. Using the full training set to train for the first corpus, I obtained results of 89%.