Lang & Stats: Semantic Coherence Project, Rose Hoberman

May 7, 2005

2. Final Performance on Development Set

My classification accuracy rate on the development set was 94%. I misclassified 12 documents out of 200; only 2 of these were actually fake, the majority of errors was on real sentences which I mistook for fake. The average probability I gave to the correct class was .87, and the average of the logs of the probabilities was -.34.

3. Method

3.1 Vocabulary and Stoplist

In all of my experiments I used only a subset of the words present. I defined my vocabulary as any of the word types present in the 100MW corpus, with the exception of words that occurred only once or twice, and a list of around 650 non-content words that I obtained from a standard stoplist. Any word in the vocabulary was considered a content word.

3.2 Classifier and Confidence Scores

For my experiments I tried various types of features, and I combined them into a single classifier using an implementation of Adaboost. (I did not try varying the learner because of some previous experiments I did last semester.) Adaboost outputs a score which is *not* a probability. I used the score/class pairs from the training data to learn the parameters for a logistic function, which transforms the score into a probability that the sentence is real.

3.3 Baseline: Repetition and Stemming

My initial baseline system achieved 75% accuracy. This included only some simple features which looked for repetition: the number of types that occurred more than once in the document, the number of tokens whose types occurred more than once, the maximum number of times a word type occurred, and these values as percentages of the number of words and types in the document

I used the Porter stemmer to stem the corpus, in order to conflate morphologically related word forms. This increased the baseline to around 85% accuracy. I continued using the stemmed forms for the remainder of my experiments.

3.4 More Features

I then added additional features: the number of (non-stop) types present in the document, the length of the longest repeated n-gram, the percent of the words which were stop words, the length of the longest sequence of all stop words, and the length of the longest sequence of all content words.

From the 100MW corpus I collected wordpair counts (the number of sentences in which both words occurred). From these counts I derived some new simple features: the number and percent of wordpairs which were never seen in the 100MW corpus

3.5 Correlations

From the wordpair counts and the number of sentences in which each individual word appeared I derived a measure of correlation between word pair using Yule's Q statistic. I also briefly tried using the Chi statistic, but I didn't detect any improvements, and possibly a slight degradation in performance, so I didn't pursue this.

From these wordpair correlations I derived some simple features: the mean, median, minimum, maximum, variance, and range of correlation values between wordpairs in the document, the same features but only for wordpairs at least 5 words apart (to exclude trigram effects), the percentage of wordpairs with correlations above a thresholded value.

I also looked at the correlation values as a function of distance between wordpairs. I defined features that measured the minimum and maximum correlation for words below or above a thresholded distance, and conversely the minimum and maximum distance at which I could find a correlation above or below some thresholded value.

3.6 Clustering

Finally, I tried grouping (clustering) the words in each document, to get an idea of how many different topics were discussed. I used a simple greedy agglomerative clustering algorithm. The closeness of two clusters was defined by the average-link measure: specifically, the similarity was the weighted average of the correlation between the word types in those clusters, where more frequent types were weighted more heavily.

This method seemed suboptimal because of problems arriving through multiple word senses. For example if the bigram "drag racing" is present in a document, but so is the word "queen", the greedy algorithm would merge "drag" and "queen". Since "racing" isn't correlated with "queen" it would now be orphaned in its own cluster. I tried to remedy this problem by first stepping linearly through the document and merging neighboring stopwords which had correlations above some threshold. After this step I continued to merge clusters based only on the average-link measure.

From these clusters I derived features, such as the number of clusters, the number of orphans (clusters of size 1), the maximum cluster size, as well as these same numbers normalized by the document length.

3.7 Impact of Feature Sets

Determining the impact of individual features was problematic. In general if I removed any small set of features there was little impact on performance on the development set. But this does not mean that these sets are useless. I think it is just that there is a substantial amount of overlap among my features, and boosting can learn similar "concepts" from many different features. This makes it very difficult to report the impact of any one feature set. I could report the performance obtained from each feature individually, but these numbers are generally poor and do not reflect the benefit obtained from the interaction among features.

One thing I can report is the first features selected by Adaboost, which correspond to the features whose have a split which best reduce the entropy of the training set. The top features and the best values at which to split them are listed below:

The sentence is likely to be true if:

1. The percent of pairs with correlation greater than .4 was $> .2$
2. The number of words whose type occurred more than once, normalized by the document length was $> .062$
3. The median Q value at distance 5. was $> .163$
4. The maximum Q value at distance 4 was $> .982$
5. The longest sequence of consecutive stop words was > 4
6. The maximum number of times a word type was repeated, normalized by document length was $> .0138$
7. The minimum Q value at distance 1 was $<$ -.312

5. Comments and Suggestions

I would have been liked to have been able to see the actual "UNK" words, to see if stemming would have been able to reduce them to a recognizable form.