CMPUT 501 Assignment 5

Afia Anjum and **Shehroze Khan**
Department of Computing Science
University of Alberta

1 Method and Implementation

Given a training file, consisting of texts and categories, and a 5 classification problem, we take the first step by splitting the given texts according to different categories or classes. Since we focus on training a Naive Bayes classifier, this split is necessary to calculate the prior probabilities of each of the classes. This classifier treats each of the words in a sentence as features and hence it is important to tokenize the given texts of our training sample at word level. We utilize the nltk tokenizer for tokenization. We make sure to remove any punctuation from our sentences so that the nltk tokenizer can accept plain texts as input and the Naive Bayes classifier can deal with words from this tokenization as features. We convert all the words into lowercase letters to prevent counting uppercase and lowercase instances of the same word separately. Going through the initial pre-processing steps, we apply Naive Bayes classification with 3 fold cross validation on our training samples. We look at the accuracy in each of the three folds, which in turn helps us in selecting the best parameters to train our model. With the best parameters in our hand, we train the Naive Bayes Classifier with our training data and finally obtain our prediction on the test data. We calculate the precision and recall for our classification and report the macro averaged and micro averaged precision and recall.

2 Naive Bayes Classifier and K Fold Cross Validation

We follow the pseudo code described in Figure 4.2 of the textbook to write two functions: one to train the Naive Bayes classifier, and the other to test new text files using the output from the train function.

During the training phase, we keep track of all the unique set of words that occurred in our training corpus. These unique set of words constitute our vocabulary. We also compute the likelihood for each word in our vocabulary during this phase. These two calculations are necessary in computing the maximum likelihood estimation during the test phase.

For cross-validation, we have four set of parameters that we use to train our model. These are listed as follows:

1. Don't handle stop words and don't ignore unknown words
2. Handle stop words, but don't ignore unknown words
3. Don't handle stop words, but ignore unknown words
4. Handle stop words and ignore unknown words

Handling stop words essentially means that we ignore the most frequent words in English (such as *a*, *it*, *the*, *and*, *he*, *she*, etc.) in our probability computations for both the training and the test data. And ignoring unknown words means that we do not compute probabilities for words in the test set that do not appear in our training set.

NLTK provides a corpus of stop words in English. If any word in our training and test data exists in this corpus list, we exclude it from the final word token list that is used in our models. In order to ignore unknown words in the test data,we simply do not compute their likelihood probability in the finals. If these words are considered, of course, the count of an unknown word is just one for the probability likelihood computation.

As a result of 3-fold cross-validation, we found that our model gives the best average accuracy (of around 97.0%) when we both handle stop words and ignore unknown words in our probability computations. This outcome is also intuitive, since removing stop words means that only

words most characteristic of a particular category are considered for likelihood probability computations. And ignoring unknown words means that we do not let their presence skew probability computations in favor of a certain class.

With these parameters, the final accuracy of our Naive Bayes classifier on the given test file is about 97.9%.

3 Output

The confusion matrix generated from our program is shown below. The horizontal axis shows the predicted labels, while the vertical axis shows the actual labels:

Category	Tech	Bus	Ent	Pol	Sport
Tech	106	0	0	1	0
Bus.	0	154	0	3	0
Ent.	3	1	112	1	0
Pol.	2	1	0	121	0
Sport	0	1	0	1	161

Precision and Recall for each of the categories as computed from the program:

Category	Precision	Recall
tech	0.954	0.990
business	0.980	0.980
entertainment	1.0	0.957
politics	0.952	0.975
sport	1.0	0.987

We also get the following aggregated precision from our program:

- Micro-averaged precision: 0.979
- Macro-averaged precision: 0.977

4 Error Analysis

We see 14 misclassifications on our test set. The most frequently misclassified texts actually belong to the entertainment category, which is incorrectly classfied five times. The business cateogry texts are next, and misclassified three times.

Out of its five misclassifications, the entertainment category texts are misclassified as tech three times. Upon closer inspection, we find that one of these texts is about fake dvds and piracy, another about forensic science as seen in the csi miami tv show, and the last one about walmart being sued because of lyrics on its cd. All three of these texts have some form of monetary valuation listed, which we think might have confused the classifier. However, the most obvious explanation is that for the text on forensic science, because it goes into detail using words to describe the science itself, rather than the tv show. Because individual words act as features for our model, we can understand this particular misclassification better.

The business category texts are misclassified as politics all three times. Upon closer inspection, we see that all three of these texts discuss (state / government) policy in one way or another. One of these talks about state pension policy, another about policy on retirement age, and the last one about British parliamentary elections and policy changes to tax credit rules and interest rates. Because all three texts contain words pertaining either to government policies or ideas of government officials, the classifier is tricked into misclassifying these texts as belonging to the politics category.

Looking at our model's classifications over the evaluation texts, we find that all of the assigned category labels generally correspond correctly to the texts. Upon closer inspection, however, we suspect that because two of the texts classified as politics talk about economic and tax policies, the classifier might have incorrectly classified them, with the correct category being business. Similarly, we find one text classified as tech talk about movies being released for the sony psp, which, similar to the entertainment misclassifications above, talks about the type of movies released, as well as their prices, and as such might actually belong to the entertainment category.

Of course, we cannot say the aforementioned for sure without having the true category labels, but our suspicions are based on the misclassifications described above.

References

NB Classifier
Stop Word Removal Guide
Writing CSV files
Speech and Language Processing book