CS486 Assignment 3

Justin Hu
20465661

Preprocessing

Attached table A presents the accuracy and time cost of different combinations of property settings. Blank entry is equivilent to value in the first row (the default value). The input has been resampled to 50% of the original size, and test split ratio is 66%.

a. Property Impact

Refers to row 2-17 of attached table A, where each row differs from another by a single property setting, and row 17 is the average outcome of these settings. Results better than average are highlighted.

- **IDFTransform & TFTransform**
  IDFTransform increases time needed to build the model.
  They have no effect on accuracy.
  (The reason that these properties have no effect on accuracy is because they need 'outputWordCounts' to set to true in order to get meaningful result, but we are asked to keep this value unchanged from its default)

- **lowerCaseTokens**
  Reduces time of model building probably because there are fewer attributes when all terms are converted into lowercase; however doing so decrease accuracy.

- **minTermFreq**
  Most terms have small frequencies. Discarding low frequency terms makes it faster to build a model, but results in lower accuracy.

- **stemmer**
  Lovins' algorithm demonstrates low accuracy with high speed, but Snowball does the opposite.

- **tokenizer**
  CharacterNGram is totally inaccurate. All other tokenizer are good in accuracy, but NGram appears to be slower than Alphabetic tokenizer.

- **useStopList**
  MultiStopWords, Rainbow, and WordsFromFile (file downloaded from xpo6.com/download-stop-word-list/) all yield better or same result than null list, and slightly more time cost. Rainbow yields particularly high accuracy.

In conclusion, Snowball stemmer, tokenizers and StopList would help increase accuracy of the classifier, whereas lowerCaseTokens, minTermFreq and (iterated and normal) Lovins stemmer decrease the time needed to build the model.

b. Recommended Joint Configuration

Judge by intuition, converting terms into lowercase should not affect accuracy but only reduce number of attributes created. Finding the word stem is not necessarily helpful, and word tokenizer would make the best sense. Stop list helps to reduce number of attributes by eliminating articles which have no effect on the document topic.

However, this is not true from experiments. Row 18-29 of the attached table A compares the outcomes of joint configurations of the properties that provide high accuracy in the above section. Snowball stemmer, NGram tokenizer and Rainbow stop list are the three factors that produce highest accuracy when applied seperatly. However, they do not produce promising result when combined together. The best result is same as using Rainbow stop list alone.

Thus, the only recommended configuration is to use nullStemmer, Rainbow stop list, WordTokenizer, false for all other properties and slightly adjust minTermFreq if time is sensitive.

Classification

Data to be found in attached table B.

a

Naive-Bayes is the fastest classifer.
SMO yields highest accuracy.
J48 is the slowest classifer but is more accurate than Naive-Bayes.

b

For Naive-Bayes the overfitting is 5%.
For SMO the overfitting is about 10%.
For J48 there is also 10% overfitting.

c

See attached table B.