Text classification using Machine Learning
CP-SC 881 Machine Learning

Hung VO H.M
Instructor: Professor Luo Feng

I. Abstract:

Documents automatically classification or text classification is of increasing interesting and applications. Examples of text classification applications are spam filter, knowledge management and retrieval, document in specific topics query, language guessing. This project is going to examine text classification machine learning methods and implement one of the methods, the Naïve Bayes method over twenty newsgroup categories. The Naïve Bayes method incorporating with TF-IDF methods are implemented to improve performance.

II. Introduction:

Text classification is to categorize electronic documents into appropriate classes. In another words, text classification is to assign each electronic document with an appropriate label. The task of text classification is divided into two kinds: supervised classification and unsupervised classification. Supervised classification uses some external mechanism such as human to support the task while unsupervised classification does not.

Text classification has many useful applications such as spam filter, knowledge management and retrieval, document in specific topics query, language guessing, topic spotting, email routing, webpage type classification, product review classification task… Spam filter is to determine whether an incoming email a spam mail, junk mail or a normal mail, or even a priority mail. Topic spotting is to determine topic of a text, while email routing is to forward an incoming email from general email address to specific email address based on content of received email.

Methods of text classification have been developed from time to time and become more and more powerful and accurate. Such methods are Naive Bayes classifier, Tf-idf, latent semantic indexing, support vector machines (SVM), artificial neural network, kNN, decision trees such as ID3 or C4.5, concept mining, Rough set based classifier, soft set based classifier… Every method has its own characteristic, has its own pros and cons. These methods can be used together so that they can complement each other. E.g., in this topic, Naïve Bayes and TF-IDF have been implemented to degrade their cons and improve the classification task performance.

In this project, a text classifier has been implemented from scratch based on Naïve Bayes algorithm and using TF-IDF as complement method to improve performance. Microsoft Visual C# 2008 has been used as programming environment and Microsoft .Net Framework 3.5 has been used to provide program user interface.

III. Basic text classification methods

Solutions for text classification problem can be human-engineered rule-base system or machine learning system. The former is easier to be implemented and more accuracy with small amount of data. There are several human-engineered rule-base systems such as CONSTRUE system which have precision of over 90% on 750 test cases [Hayes and Weinstein, 1991]. This is a good result, however, not sufficient for real world classification task. Therefore, we need machine learning system. An example of a machine learning system for the same task is a system based on Memory Based Reasoning [Masand et al., 1992], which employs nearest neighbor style classification and has a reported accuracy in the range of 70-80% on Dow Jones news stories.

For machine learning system, there are several solutions mentioned in introduction part. Here Decision tree and Naïve Bayes are provided as example text classification methods. Text classification is just such a domain with attributes are words, where number of attributes are large. More sophisticated model will take into account word pairs or words phrases.

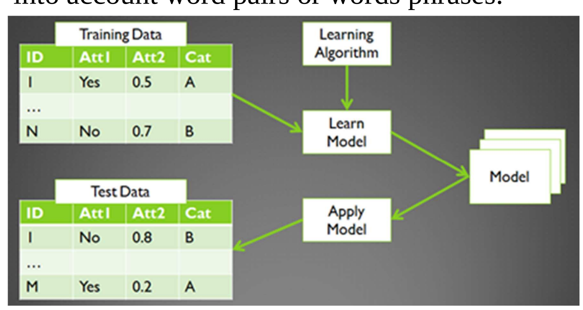


Figure 1-Machine learning approach for text classification

With Naïve Bayes model, the "Naïve Bayes assumption" is used such that all attributes of examples are independent of each other given the context of class. This makes the model easier to be implemented but does not much degrade the accurate rate. With decision tree model, a universe dictionary or local dictionary will be created by scanning the whole documents for words that appear over five times.. The top n (n<10000) frequent items in the dictionary will be chosen for finding patterns for specific topic. An induction rule such as Swap-1 [Chidanand Apte,1994] will be used for finding the patterns. The final step for decision tree is evaluating for choosing the best solution based on minimum classification error and cost.

There are two different classifiers but have the same name "Naïve Bayes" that both use the "Naïve Bayes assumption". One of them is called "Multi-variate Bernoulli Model" and the other one is called "Multi nominal event model". The former one is suitable for tasks that have fixed numbers of attributes. In this case, the documents can be considered as events while absence and presence of words are attributes of events. In case of the later one, the word occurrences are events while the document is the collection of word event. The "Multinomial event model" takes into account number of times each word occurrence in the document but the Multi-variate Bernoulli Model does not.

IV. Implementation

There are two main phases in text classifications. That are training phase and testing phase. In former phase, Naïve Bayes in cooperating with TF-IDF will be used to build the text classification model. The later phase will involve in using Naive Bayes for applying text classification task.

The most important attributes of text classification model are words and probability of words in documents and in category that the documents belong to. Therefore, important words from collection of documents need to be extracted first to build up the model. Important word extraction is solved by following steps: tokenizing, removing stop words, stemming and getting top most important words by applying TF-IDF weighting factor.

Each document in each category of training dataset will be read sequentially and tokenized into many terms using following regular expression: [^a-zA-Z]. Each term will then be removed if they are in stop word list. Stop words not only appear a lot compare to other words in a document but also appear in almost every document. Therefore, stop words are unimportant; they cannot help to distinguish contents between documents. The list of 450 stop words has been use in this project.

Stemming is process of remove a word prefix, suffix, and turn it to original or turn a set

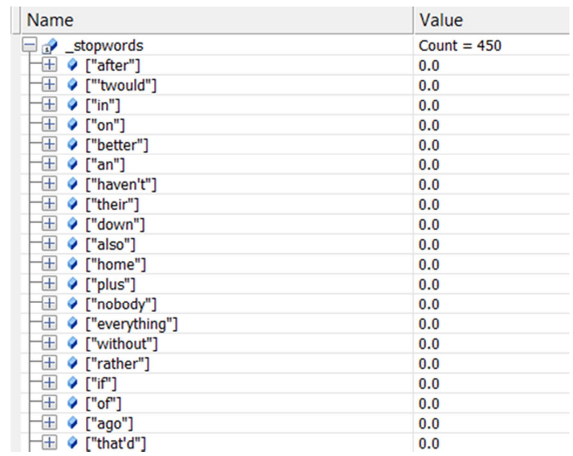


Figure 2 - Stop words list

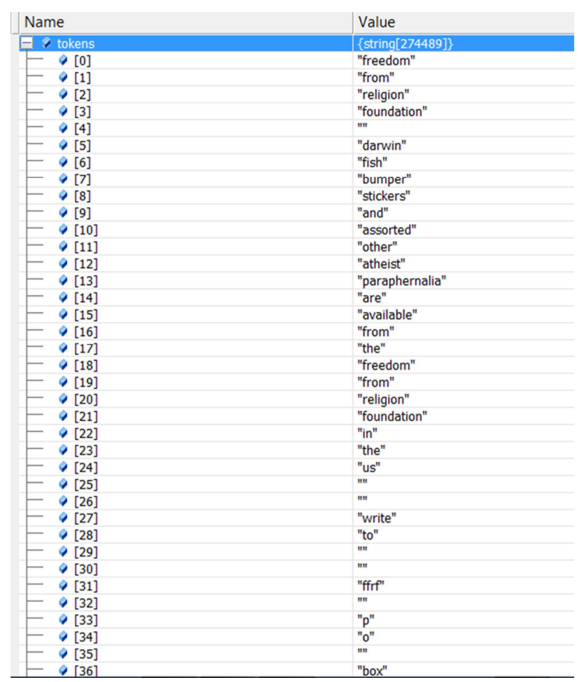
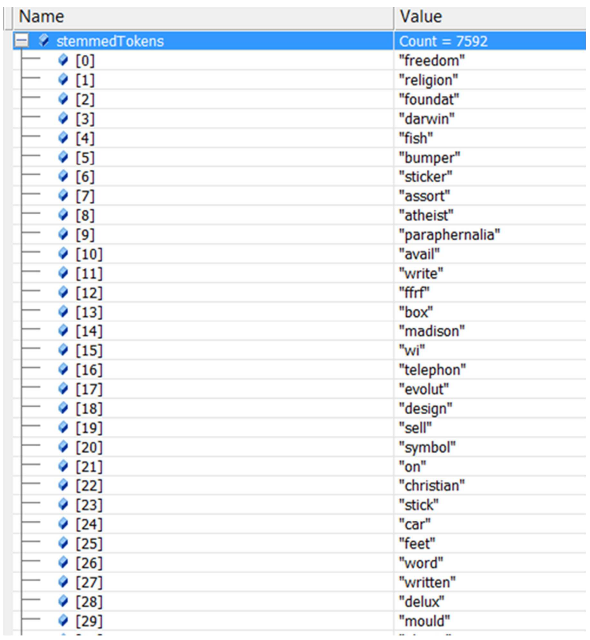


Figure 3 - Category "alt.atheism" after tokenized

of words that have same original to same stem (this stem is not required to be root word or to be meaningful). For example, set of words: “learning”, “learnt”, “learned” should be stemmed to “learn” only. Purpose of stemming is to reduce the numbers of terms in our model so that performance will be enhanced. Figure 3 and figure 4 show the different between set of tokenized words and set of words after removing stop words and stemming. These words are extracted from the category “alt.atheism” in the training dataset.



Name	Value
stemmedTokens	Count = 7592
[0]	"freedom"
[1]	"religion"
[2]	"foundat"
[3]	"darwin"
[4]	"fish"
[5]	"bumper"
[6]	"sticker"
[7]	"assort"
[8]	"atheist"
[9]	"paraphernalia"
[10]	"avail"
[11]	"write"
[12]	"ffrf"
[13]	"box"
[14]	"madison"
[15]	"wi"
[16]	"telephon"
[17]	"evolut"
[18]	"design"
[19]	"sell"
[20]	"symbol"
[21]	"on"
[22]	"christian"
[23]	"stick"
[24]	"car"
[25]	"feet"
[26]	"word"
[27]	"written"
[28]	"delux"
[29]	"mould"

Figure 4 - Category "alt.atheism" after stop words removal and stemming

In order to enhance more performance, the list of tokens/ words in the model can be reduced by either using threshold method or TF-IDF method. The former one is much easier than the later one. In the former one, what we need to do is just to specify an upper and a lower threshold value so that every word that appear more than upper threshold or less than lower threshold value will be removed. The recommend upper threshold should be 100 and lower threshold should be 10, in my opinion. The reason that makes threshold work is that unimportant words that not appear quite often or appear a lot in almost documents over the whole document collection should be removed.

TF-IDF method is more complicated but can help removing a lot of unimportant words with confident. In TF-IDF method, we define weight of term is (TF*IDF) where TF refers to Term Frequency and IDF refers to Inverse Document Frequency. Weight is a measure of how important a word is to a document in a collection. TF tells us how often the word appears in a document compare to other words. In the other hand, DF (document frequency) shows us how many documents in a collection contain the word. IDF is inverted of DF; this means the higher DF, the smaller IDF. Consequently, the higher TF is as well as the higher IDF is, the more important the word is. In other words, the higher weight of term (TF*IDF), the more important the term is. TF and IDF formula is given as following:

$$tf_{i,j} = \frac{n_{i,j}}{\Sigma_k n_{k,j}}$$

$$i\,df_i = \log \frac{|D|}{|\{d: t_i \in d\}|}$$

Where $n_{i,j}$ is count of word t_i in the document d_j. |D| is total number of documents in the collection. $|\{d: t_i \in d\}|$ is number of documents that contain word t_i. In order to avoid division by zero, $1 + |\{d: t_i \in d\}|$ should be used instead of $|\{d: t_i \in d\}|$.

Based on weight (TF*IDF), top 1000, 5000, 10000… words with top weight can be chosen for text classification model confidentially. The number of top weighted words is chosen depending on purpose of classification task whether correctness or speed is higher priority. The lower number of chosen words, the faster classification task will be.

After tokenizing, removing stop words, stemming and getting top most important words by applying TF-IDF weighting factor, it is time to calculate parameters for our model. Naïve Bayes is key algorithm for this task. Considering our model now contains following information:

- **D:** Set of documents
- **N:** number of documents in **D**
- **V:** Set of vocabulary/tokens/terms
- **C:** set of Categories

Besides of these parameters, in order to complete our model, we need 2 more parameters that are *prior* and *condprob*. *prior* and *conprob* are calculated by steps shown below. *prior* tells us how many documents in a category compared to other categories in the collection. *condprob* shows us how important a word compared to other words in specific category.

- **Foreach** *category c* in **C**
  $\mathbf{N_c}$ = Number of documents in c
  Prior = N_c/N
  $text_c$ = All text in category c
  foreach *t* in **V**
  do T_{ct}= *countTokens*(*t*,$text_c$)
  foreach *t* in **V**

$$condprob[t,c] = \frac{Tct + 1}{\Sigma t'(Tct' + 1)}$$

- **return** V,*prior*, *condprob*

Now, our text classification model is completed and can be applied to classification task. Let *d* is document to be classified and *W* is extracted tokens/words form (**V,**d). Result returned from below function is category of document *d*.

- **Foreach** *c* in **C**
  score[*c*] = *log*(*prior*[*c*])
  foreach *t* in *W*
  score[*c*]+= *log*(*condprob*[*t*,*c*])
  return $\text{argmax}_{c \text{ in } \mathbf{C}}$(*score*[*c*])

Finally, a classifier has been completely implemented throughout this section. From training data, the classifier is able to build up a model and apply that model for classification task. Another quick note is this classifier using multi-nomial Naïve Bayes algorithm.

V. Testing and results

In this section, the classifier implemented in previous section will do training task and testing task over several datasets. All of the datasets contain following twenty categories:

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey
talk.politics.misc talk.politics.guns talk.politics.mideast	misc.forsale
talk.religion.misc alt.atheism soc.religion.christian	sci.crypt sci.electronics sci.med sci.space

The first dataset using for both training task and testing task contains 19997 articles. Without TF-IDF, total 95432 terms were found from this dataset. After TF-IDF, total terms were reduced to 7681, 4780, 1343 terms with 89.62%, 88.82%, 88.29% correctness alternatively. Details about classification correctness of each category can be found in figure 5, 6, 7.

The second dataset is a little bit different from previous one. This dataset is revised version of original dataset (19997 documents). The documents are sorted by date and divided into training (60%) and test (40%) sets. Cross-posts (duplicates) and newsgroup-identifying headers are removed. Training set is described as following:

Total documents	**11314**
alt.atheism	480
comp.graphics	584
comp.os.ms-windows.misc	591
comp.sys.ibm.pc.hardware	590
comp.sys.mac.hardware	578
comp.windows.x	593
misc.forsale	585
rec.autos	594
rec.motorcycles	598
rec.sport.baseball	597
rec.sport.hockey	600
sci.crypt	595
sci.electronics	591
sci.med	594
sci.space	593
soc.religion.christian	599
talk.politics.guns	546
talk.politics.mideast	564
talk.politics.misc	465
talk.religion.misc	377

Without TF-IDF, total 69604 terms were found from this dataset. After TF-IDF, total terms were reduced to 8291, 5017, 1380 terms with 78.44%, 77.28%, 72.69% correctness alternatively. Details can be found in figure 8, 9, 10.

Threshold method was also examined instead of TF-IDF in removing unimportant words. If lower threshold is equal to ten, no upper threshold is used, total terms remained will be 12881, and correctness is 74.49%. If upper threshold equal to 100 is added, 9860 terms will remain and the correctness achieved is 71.52%.

```
Test with 7681 terms
category              alt.atheism :  894 True |  106 False |  1000 Total |  89.40 % correct
category            comp.graphics :  922 True |   78 False |  1000 Total |  92.20 % correct
category  comp.os.ms-windows.misc :  122 True |  878 False |  1000 Total |  12.20 % correct
category comp.sys.ibm.pc.hardware :  951 True |   49 False |  1000 Total |  95.10 % correct
category    comp.sys.mac.hardware :  978 True |   22 False |  1000 Total |  97.80 % correct
category           comp.windows.x :  892 True |  108 False |  1000 Total |  89.20 % correct
category             misc.forsale :  956 True |   44 False |  1000 Total |  95.60 % correct
category                rec.autos :  970 True |   30 False |  1000 Total |  97.00 % correct
category          rec.motorcycles :  984 True |   16 False |  1000 Total |  98.40 % correct
category       rec.sport.baseball :  991 True |    9 False |  1000 Total |  99.10 % correct
category         rec.sport.hockey :  988 True |   12 False |  1000 Total |  98.80 % correct
category                sci.crypt :  983 True |   17 False |  1000 Total |  98.30 % correct
category          sci.electronics :  965 True |   35 False |  1000 Total |  96.50 % correct
category                  sci.med :  971 True |   29 False |  1000 Total |  97.10 % correct
category                sci.space :  970 True |   30 False |  1000 Total |  97.00 % correct
category   soc.religion.christian :  996 True |    1 False |   997 Total |  99.90 % correct
category       talk.politics.guns :  958 True |   42 False |  1000 Total |  95.80 % correct
category    talk.politics.mideast :  957 True |   43 False |  1000 Total |  95.70 % correct
category       talk.politics.misc :  810 True |  190 False |  1000 Total |  81.00 % correct
category       talk.religion.misc :  664 True |  336 False |  1000 Total |  66.40 % correct
In total,  17922 True |   2075 False |  19997 in Total |  89.62 % correct
```

Figure 5 – Applying full twenty newsgroups dataset for both training and testing - 7681 terms used after TF-IDF

```
Test with 4780 terms
category              alt.atheism :  886 True |  114 False |  1000 Total |  88.60 % correct
category            comp.graphics :  907 True |   93 False |  1000 Total |  90.70 % correct
category  comp.os.ms-windows.misc :   80 True |  920 False |  1000 Total |   8.00 % correct
category comp.sys.ibm.pc.hardware :  946 True |   54 False |  1000 Total |  94.60 % correct
category    comp.sys.mac.hardware :  973 True |   27 False |  1000 Total |  97.30 % correct
category           comp.windows.x :  878 True |  122 False |  1000 Total |  87.80 % correct
category             misc.forsale :  955 True |   45 False |  1000 Total |  95.50 % correct
category                rec.autos :  964 True |   36 False |  1000 Total |  96.40 % correct
category          rec.motorcycles :  984 True |   16 False |  1000 Total |  98.40 % correct
category       rec.sport.baseball :  988 True |   12 False |  1000 Total |  98.80 % correct
category         rec.sport.hockey :  989 True |   11 False |  1000 Total |  98.90 % correct
category                sci.crypt :  981 True |   19 False |  1000 Total |  98.10 % correct
category          sci.electronics :  961 True |   39 False |  1000 Total |  96.10 % correct
category                  sci.med :  968 True |   32 False |  1000 Total |  96.80 % correct
category                sci.space :  962 True |   38 False |  1000 Total |  96.20 % correct
category   soc.religion.christian :  996 True |    1 False |   997 Total |  99.90 % correct
category       talk.politics.guns :  960 True |   40 False |  1000 Total |  96.00 % correct
category    talk.politics.mideast :  944 True |   56 False |  1000 Total |  94.40 % correct
category       talk.politics.misc :  792 True |  208 False |  1000 Total |  79.20 % correct
category       talk.religion.misc :  648 True |  352 False |  1000 Total |  64.80 % correct
In total,  17762 True |   2235 False |  19997 in Total |  88.82 % correct
```

Figure 6 – Applying full twenty newsgroups dataset for both training and testing - 4780 terms used after TF-IDF

```
Test with 1343 terms
category              alt.atheism :  854 True |  146 False |  1000 Total |  85.40 % correct
category            comp.graphics :  913 True |   87 False |  1000 Total |  91.30 % correct
category  comp.os.ms-windows.misc :   66 True |  934 False |  1000 Total |   6.60 % correct
category comp.sys.ibm.pc.hardware :  940 True |   60 False |  1000 Total |  94.00 % correct
category    comp.sys.mac.hardware :  968 True |   32 False |  1000 Total |  96.80 % correct
category           comp.windows.x :  877 True |  123 False |  1000 Total |  87.70 % correct
category             misc.forsale :  965 True |   35 False |  1000 Total |  96.50 % correct
category                rec.autos :  963 True |   37 False |  1000 Total |  96.30 % correct
category          rec.motorcycles :  981 True |   19 False |  1000 Total |  98.10 % correct
category       rec.sport.baseball :  989 True |   11 False |  1000 Total |  98.90 % correct
category         rec.sport.hockey :  985 True |   15 False |  1000 Total |  98.50 % correct
category                sci.crypt :  983 True |   17 False |  1000 Total |  98.30 % correct
category          sci.electronics :  956 True |   44 False |  1000 Total |  95.60 % correct
category                  sci.med :  980 True |   20 False |  1000 Total |  98.00 % correct
category                sci.space :  959 True |   41 False |  1000 Total |  95.90 % correct
category   soc.religion.christian :  996 True |    1 False |   997 Total |  99.90 % correct
category       talk.politics.guns :  940 True |   60 False |  1000 Total |  94.00 % correct
category    talk.politics.mideast :  924 True |   76 False |  1000 Total |  92.40 % correct
category       talk.politics.misc :  785 True |  215 False |  1000 Total |  78.50 % correct
category       talk.religion.misc :  631 True |  369 False |  1000 Total |  63.10 % correct
In total,  17655 True |   2342 False |  19997 in Total |  88.29 % correct
```

Figure 7 - Applying full twenty newsgroups dataset for both training and testing - 1343 terms used after TF-IDF

```
Test with 8291 terms
category                alt.atheism :  258 True |    61 False |   319 Total |  80.88 % correct
category              comp.graphics :  301 True |    88 False |   389 Total |  77.38 % correct
category    comp.os.ms-windows.misc :    1 True |   393 False |   394 Total |   0.25 % correct
category   comp.sys.ibm.pc.hardware :  282 True |   110 False |   392 Total |  71.94 % correct
category      comp.sys.mac.hardware :  330 True |    55 False |   385 Total |  85.71 % correct
category             comp.windows.x :  283 True |   112 False |   395 Total |  71.65 % correct
category               misc.forsale :  317 True |    73 False |   390 Total |  81.28 % correct
category                  rec.autos :  355 True |    41 False |   396 Total |  89.65 % correct
category            rec.motorcycles :  372 True |    26 False |   398 Total |  93.47 % correct
category         rec.sport.baseball :  366 True |    31 False |   397 Total |  92.19 % correct
category           rec.sport.hockey :  376 True |    23 False |   399 Total |  94.24 % correct
category                  sci.crypt :  354 True |    42 False |   396 Total |  89.39 % correct
category            sci.electronics :  277 True |   116 False |   393 Total |  70.48 % correct
category                    sci.med :  327 True |    69 False |   396 Total |  82.58 % correct
category                  sci.space :  355 True |    39 False |   394 Total |  90.10 % correct
category     soc.religion.christian :  363 True |    35 False |   398 Total |  91.21 % correct
category         talk.politics.guns :  335 True |    29 False |   364 Total |  92.03 % correct
category      talk.politics.mideast :  314 True |    62 False |   376 Total |  83.51 % correct
category         talk.politics.misc :  188 True |   122 False |   310 Total |  60.65 % correct
category         talk.religion.misc :  154 True |    97 False |   251 Total |  61.35 % correct
In total,   5908 True |   1624 False |   7532 in Total |  78.44 % correct
```

Figure 8- Applying revised twenty newsgroups dataset - 8291 terms after TF-IDF

```
Test with 5017 terms
category                alt.atheism :  251 True |    68 False |   319 Total |  78.68 % correct
category              comp.graphics :  300 True |    89 False |   389 Total |  77.12 % correct
category    comp.os.ms-windows.misc :    1 True |   393 False |   394 Total |   0.25 % correct
category   comp.sys.ibm.pc.hardware :  273 True |   119 False |   392 Total |  69.64 % correct
category      comp.sys.mac.hardware :  329 True |    56 False |   385 Total |  85.45 % correct
category             comp.windows.x :  274 True |   121 False |   395 Total |  69.37 % correct
category               misc.forsale :  321 True |    69 False |   390 Total |  82.31 % correct
category                  rec.autos :  351 True |    45 False |   396 Total |  88.64 % correct
category            rec.motorcycles :  369 True |    29 False |   398 Total |  92.71 % correct
category         rec.sport.baseball :  366 True |    31 False |   397 Total |  92.19 % correct
category           rec.sport.hockey :  371 True |    28 False |   399 Total |  92.98 % correct
category                  sci.crypt :  347 True |    49 False |   396 Total |  87.63 % correct
category            sci.electronics :  266 True |   127 False |   393 Total |  67.68 % correct
category                    sci.med :  319 True |    77 False |   396 Total |  80.56 % correct
category                  sci.space :  352 True |    42 False |   394 Total |  89.34 % correct
category     soc.religion.christian :  355 True |    43 False |   398 Total |  89.20 % correct
category         talk.politics.guns :  331 True |    33 False |   364 Total |  90.93 % correct
category      talk.politics.mideast :  308 True |    68 False |   376 Total |  81.91 % correct
category         talk.politics.misc :  185 True |   125 False |   310 Total |  59.68 % correct
category         talk.religion.misc :  152 True |    99 False |   251 Total |  60.56 % correct
In total,   5821 True |   1711 False |   7532 in Total |  77.28 % correct
```

Figure 9 - Applying revised twenty newsgroups dataset - 5017 terms after TF-IDF

```
Test with 1380 terms
category                alt.atheism :  213 True |   106 False |   319 Total |  66.77 % correct
category              comp.graphics :  308 True |    81 False |   389 Total |  79.18 % correct
category    comp.os.ms-windows.misc :    1 True |   393 False |   394 Total |   0.25 % correct
category   comp.sys.ibm.pc.hardware :  250 True |   142 False |   392 Total |  63.78 % correct
category      comp.sys.mac.hardware :  297 True |    88 False |   385 Total |  77.14 % correct
category             comp.windows.x :  262 True |   133 False |   395 Total |  66.33 % correct
category               misc.forsale :  306 True |    84 False |   390 Total |  78.46 % correct
category                  rec.autos :  344 True |    52 False |   396 Total |  86.87 % correct
category            rec.motorcycles :  366 True |    32 False |   398 Total |  91.96 % correct
category         rec.sport.baseball :  354 True |    43 False |   397 Total |  89.17 % correct
category           rec.sport.hockey :  370 True |    29 False |   399 Total |  92.73 % correct
category                  sci.crypt :  340 True |    56 False |   396 Total |  85.86 % correct
category            sci.electronics :  230 True |   163 False |   393 Total |  58.52 % correct
category                    sci.med :  284 True |   112 False |   396 Total |  71.72 % correct
category                  sci.space :  336 True |    58 False |   394 Total |  85.28 % correct
category     soc.religion.christian :  340 True |    58 False |   398 Total |  85.43 % correct
category         talk.politics.guns :  312 True |    52 False |   364 Total |  85.71 % correct
category      talk.politics.mideast :  278 True |    98 False |   376 Total |  73.94 % correct
category         talk.politics.misc :  157 True |   153 False |   310 Total |  50.65 % correct
category         talk.religion.misc :  127 True |   124 False |   251 Total |  50.60 % correct
In total,   5475 True |   2057 False |   7532 in Total |  72.69 % correct
```

Figure 10 - Applying revised twenty newsgroups dataset - 1380 terms after TF-IDF

VI. Discussion:

Naïve Bayes methods for text classification is simple to implement compared to other algorithms. It has low variance and high bias. Naïve Bayes categorization is a simple probabilistic categorization based on Conditional Independence between features. Naïve Bayes classifies an unknown instance by computing the category which maximizes the posterior.

In cooperating with TF-IDF weighting, Naïve Bayes classification performance is improved incredibly. Only with less than 1400 terms left out of nearly 100000 terms in full twenty newsgroup dataset or out of nearly 70000 terms in the revised dataset was enough to achieve high correctness. Compared to threshold method to drop unimportant terms, TF-IDF is more efficient and precise. Moreover, if the stop words list was not used, those stop words should also be removed after TF-IDF.

Using same dataset for both training and testing purpose can result in really high correctness (almost 90% correctness in overall). Some category such as “soc.religion.christian” can even reaches 99.90% correctness. If training dataset and testing data set are separated, the result is not as good as in previous case. However, over 77% is still reliable result.

Despite of very excellent performance on independent categories such as “soc.religion.christian”, “misc.forsale”…, the “comp.os.ms-windows.misc” always gets worst performance. This proves that the assumption of Conditional Independence is violated by the real world data and Naïve Bayes has poor performance when the features are highly correlated, e.g. “comp.os.ms-windows.misc” is high correlated with “comp.windows.x” as well as other categories in “comp” parent category.

VII. Conclusion

Throughout this project, several text classification methods have been examined. A Naïve Bayes classifier in corporation with TF-IDF has been implemented and tested. High performance was shown by applying the twenty newsgroups dataset in several different ways. Despite of some strong points that Naïve Bayes and TF-IDF enhanced, there was still some weakness in classification high correlated dataset. These weaknesses should be overcome by other advanced classification methods. Future work of this project would be implementing more different classifier and comparing their performance as well as optimizing current classifier.

VIII. References

[1]. Tom M. Mitchell, “Machine Learning”, McGraw Hill, 1997.

[2]. C. Apte, F. Damerau , and S. M. Weiss , ”Automated Learning of Decision Rules for Text Categorization”, ACM Transactions on Information Systems, 1994,

[3]. McCallum, A. and Nigam K. "A Comparison of Event Models for Naive Bayes Text Classification", AAAI/ICML-98 Workshop on Learning for Text Categorization, 1998

[4]. Aditya Chainulu Karamcheti, “a comparative study on text classification”, MS thesis, University of Nevada, Las Vegas, 2010.

[5]. Twenty Newsgroups Dataset
http://people.csail.mit.edu/jrennie/20Newsgroups/

[6]. Stop words list
http://www.lextek.com/manuals/onix/stopwords1.html

[7]. Porter Stemmer Algorithm.
http://tartarus.org/~martin/PorterStemmer/

[8]. Term Frequency and Inverse Document Frequency – Wikipedia.
http://en.wikipedia.org/wiki/Term_frequency

[9]. Document classification –Wikipedia
http://en.wikipedia.org/wiki/Document_classification

[10]. Naïve Bayes Text Classification.
http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html