Natural Language Processing
(UCS633 L)
Lab-evaluation 2

Application

Text Classification: Topic Identification

Submitted by:

Razat Aggarwal(101510063)

Cml3

Submitted to:

Prof Jasmeet Singh

1.What is **Text Classification**?

Text classification is one of the important and typical task in *supervised* machine learning (ML). Assigning categories to documents, which can be a web page, library book, media articles, gallery etc. has many applications like e.g. spam filtering, email routing, sentiment analysis etc.

Classification can occur at various scales such as images, videos, features, text, etc. But when the input to the classification algorithm is a textual data or a document, then that type of classification is called text classification.

Various examples of text classification includes:-
_ **Topic Identification:** Is the news article about Politics, Sports, or Technology?
_ **Spam Detection:** Is the email a spam or not?
_ **Sentiment Analysis:** Is the movie review positive or negative? , Are the customers satisfied from the product or not?
_ **Spelling correction:** To find the correct spelling of a word i.e. whether or weather, color or colour.

2.Tools Used:

2.1 Scikit-learn (for ML classifiers)

2.2 20Newsgroups (as dataset)

3.About 20Newsgroups text Dataset

The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. The data is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other (e.g. **comp.sys.ibm.pc.hardware / comp.sys.mac.hardware**), while others are highly unrelated (e.g **misc.forsale / soc.religion.christian**).

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	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

4.Work flow

4.1 *Loading the dataset using sklearn*
The data set is the famous "20 Newsgoup" data set.

4.2 *Extracting features from textfiles*

Text files are actually series of words (ordered). In order to run machine learning algorithms, we need to convert the text files into numerical feature vectors i.e. we will be creating a document-matrix.

4.2.1 Removal of stop-words

4.2.2 Replacing count by "TF-IDF" as weight of term

Just counting the number of words in each document has 1 issue, it will give more weightage to longer documents than shorter documents. To avoid this, we use TF-IDF. Moreover, A high weight in TF–IDF is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents.

4.3 *Training Classifiers*

4.3.1 Naïve Bayes classifier

4.3.2 Support Vector Machine classifier

4.4 *Testing Classifiers*

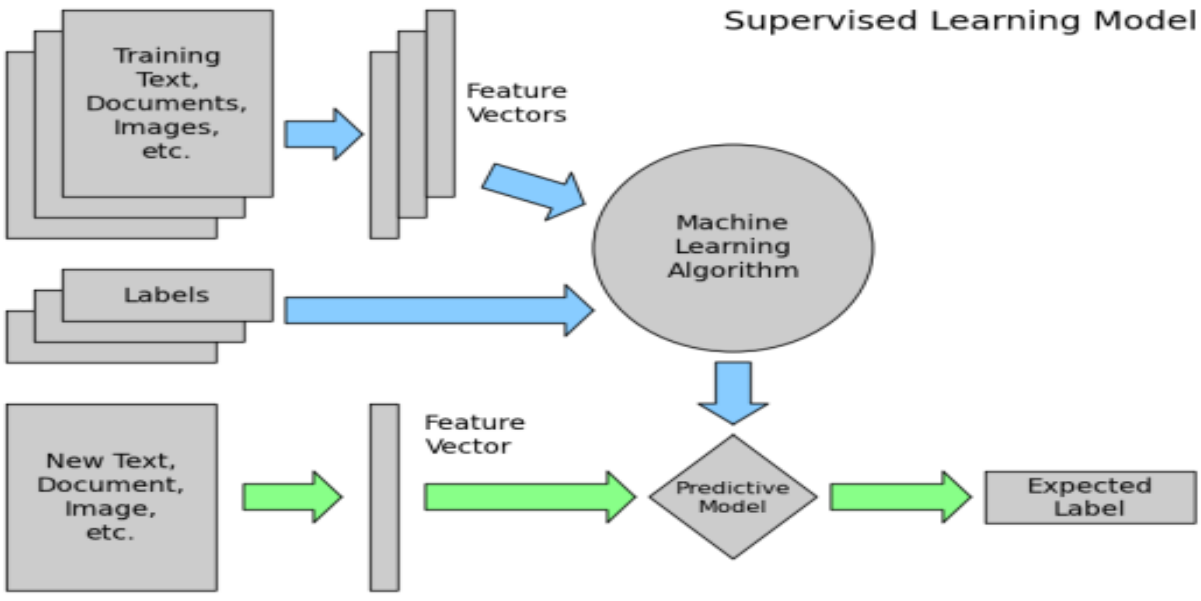

Fig2: Illustrating the basic workflow of a text classification

5. Code:

Text Classification

step1:Loading the dataset i.e. 20Newsgroups

In [3]:
```
from sklearn.datasets import fetch_20newsgroups
twenty_train = fetch_20newsgroups(subset='train', shuffle=True)
```

In [8]:
```
#checking the 20 categories in the dataset
twenty_train.target_names
```

Out[8]:
```
['alt.atheism',
 'comp.graphics',
 'comp.os.ms-windows.misc',
 'comp.sys.ibm.pc.hardware',
 'comp.sys.mac.hardware',
 'comp.windows.x',
 'misc.forsale',
 'rec.autos',
 'rec.motorcycles',
 'rec.sport.baseball',
 'rec.sport.hockey',
 'sci.crypt',
 'sci.electronics',
 'sci.med',
 'sci.space',
 'soc.religion.christian',
 'talk.politics.guns',
 'talk.politics.mideast',
 'talk.politics.misc',
 'talk.religion.misc']
```

step2:Extracting features from text files

In [28]:
```
#2.1building feature vector and removing stop words using CountVectorizer
from sklearn.feature_extraction.text import CountVectorizer
#2.2 Transforming the values in document-matrix into TF-IDF
from sklearn.feature_extraction.text import TfidfTransformer
#building the pipeline for the above process so as to simplify our code
from sklearn.pipeline import Pipeline
#for nb classifier
from sklearn.naive_bayes import MultinomialNB
text_nb_clf = Pipeline([('vect', CountVectorizer(stop_words='english')), ('tfidf', TfidfTransformer()),
                  ('clf', MultinomialNB())])
#for svm classifier
from sklearn.linear_model import SGDClassifier
text_svm_clf = Pipeline([('vect', CountVectorizer(stop_words='english')), ('tfidf', TfidfTransformer()),
                      ('clf-svm', SGDClassifier(loss='hinge', penalty='l2',alpha=1e-3, n_iter=5, random_state=42))])
```

step3: training classifiers

In [29]:
```
#Naive bayes classifier
#now fitting our model on the training dataset
text_nb_clf = text_nb_clf.fit(twenty_train.data, twenty_train.target)
```

In [37]:
```
#svm classifier
#now fiiting our model on the training dataset
text_svm_clf = text_svm_clf.fit(twenty_train.data, twenty_train.target)
```

step4 : testing classifiers

In [34]:
```
#fetching the test data
twenty_test = fetch_20newsgroups(subset='test', shuffle=True)
import numpy as np
#naive bayes
predicted_nb = text_nb_clf.predict(twenty_test.data)
accuracy_nb=np.mean(predicted_nb == twenty_test.target)*100
```

In [35]:
```
accuracy_nb
```

Out[35]: 81.6914498141264

In [38]:
```
#support vector machine
predicted_svm = text_svm_clf.predict(twenty_test.data)
accuracy_svm=np.mean(predicted_svm == twenty_test.target)*100
```

In [39]:
```
accuracy_svm
```

Out[39]: 82.249070631970255

step5: analysing Results ¶

In [12]:
```
#test case
text1=twenty_test.data[0]
print text1
```

```
From: v064mb9k@ubvmsd.cc.buffalo.edu (NEIL B. GANDLER)
Subject: Need info on 88-89 Bonneville
Organization: University at Buffalo
Lines: 10
News-Software: VAX/VMS VNEWS 1.41
Nntp-Posting-Host: ubvmsd.cc.buffalo.edu


 I am a little confused on all of the models of the 88-89 bonnevilles.
I have heard of the LE SE LSE SSE SSEI. Could someone tell me the
differences are far as features or performance. I am also curious to
know what the book value is for prefereably the 89 model. And how much
less than book value can you usually get them for. In other words how
much are they in demand this time of year. I have heard that the mid-spring
early summer is the best time to buy.

                                Neil Gandler
```

In [15]:
```
expected=twenty_test.target_names[twenty_test.target[0]]
print expected
```

```
rec.autos
```

In [20]:
```
observed=twenty_test.target_names[predicted_svm[0]]
print observed
```

```
rec.autos
```

6.Conclusion

Here, I was able to successfully classify the **20newsgroup.test** dataset with 81.69% accuracy using Naïve Bayes and 82.25% accuracy using Support Vector Machine.

Below fig shows the labels of top 20 documents in target and their predicted target values by Naïve Bayes and SVM respectively.

```
In [29]: 1 twenty_test.target[0:20]
Out[29]: array([ 7,  5,  0, 17, 19, 13, 15, 15,  5,  1,  2,  5, 17,  8,  0,  2,  4,
                 1,  6, 16])

In [30]: 1 predicted_nb[0:20]
Out[30]: array([ 7,  1,  0, 17,  0, 13, 15,  2,  5,  1,  2,  5, 17,  8, 15,  3,  2,
                 1,  6, 16])

In [31]: 1 predicted_svm[0:20]
Out[31]: array([ 7,  1,  0, 17,  0, 13, 15,  2,  5,  1,  2,  5, 17,  8, 15,  3,  2,
                 1,  6, 16])
```

7. References

6.1 http://scikit-learn.org/stable/datasets/index.html

6.2 http://qwone.com/~jason/20Newsgroups/