Capstone Project Report

Name: Sumit Patel

Course: AI and ML (Batch – AUG 2020)

Duration: 10 months

Applications in Natural Language Processing

Problem Statement:

Using NLP we can easily analyse any given text. The steps involved for such an analysis are tokenization, pre processing each word and then finally vectorising each of them. One of the most common and easy to implement vectorisation algorithm is BoW. Using BoW and NLTK for processing, implement a simple spam filter that marks all the spam texts as dangerous

Prerequisites

What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has latest version of python. The following url https://www.python.org/downloads/ can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-externalcommand/. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic. Second and easier option is to download anaconda and use its anaconda prompt to run the commands.

To install anaconda check this url https://www.anaconda.com/download/ You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6 then run below commands in command prompt/terminal to install these packages:

pip install numpy

pip install pandas

pip install sklearn

pip install tensorflow

pip install nltk

If you have chosen to install anaconda then run below commands in anaconda prompt to install these packages:

conda install -c anaconda numpy

conda install -c anaconda pandas

conda install -c anaconda sklearn

conda install -c anaconda tensorflow

conda install -c anaconda nltk

Importing the libraries and loading dataset.

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import string
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
```

Loading and reading data

```
data = pd.read_csv('../22. Applications in Natural Language Processing/spam.csv')
data.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

Data pre-processing

```
stop_words = nltk.corpus.stopwords.words('english')
stop_words[:5]
```

```
['i', 'me', 'my', 'myself', 'we']
```

```
def pre_process(text):
    text = text.translate(str.maketrans('', '', string.punctuation))
    text = [word for word in text.split() if word.lower() not in stopwords.words('english')]
    words = ""
    for i in text:
            words += (ps.stem(i))+" "
    return words
```

```
textFeatures = data['text'].copy()
textFeatures = textFeatures.apply(pre_process)
textFeatures[:5]
```

```
0    go jurong point crazi avail bugi n great world...
1                              ok lar joke wif u oni
2    free entri 2 wkli comp win fa cup final tkt 21...
3                  u dun say earli hor u c alreadi say
4            nah dont think goe usf live around though
Name: text, dtype: object
```

```
vectorizer = TfidfVectorizer(ngram_range=(1, 2))
features = vectorizer.fit_transform(textFeatures)
features.shape
```

```
(5572, 39213)
```

```
features_train, features_test, labels_train, labels_test = train_test_split(features, data['class'],
                                                                             test_size=0.3, random_state=111)
```

Training and evaluating the model

```
# Prediction using Support Vector Machine
svc = SVC(kernel='sigmoid', gamma=1.0)
svc.fit(features_train, labels_train)
prediction = svc.predict(features_test)
# accuracy_score(labels_test,prediction)
print(classification_report(labels_test, prediction, target_names = ['ham', 'spam']))
```

```
              precision    recall  f1-score   support

         ham       0.98      1.00      0.99      1440
        spam       0.99      0.85      0.91       232

    accuracy                           0.98      1672
   macro avg       0.98      0.93      0.95      1672
weighted avg       0.98      0.98      0.98      1672
```