A REPORT ON

MOVIE REVIEWS AND CLASSIFICATION

By:

POOJA R

SUPRITHA C

NITHYA S JOIS

NEHA S PRASAD

ABSTRACT

Internet today contains a huge quantity of textual data, which is growing every day. The text is prevalent data format on the web, since it is easy to generate and publish. What is hard nowadays is not availability of useful information but rather extracting it in the proper context from the the vast ocean of content.

The entertainment industry requires new and better ways to target specific users with certain features. This project is exploring the possibility of classifying the movie review corpus as positive or negative to help enable make better decisions for target users, using data mining techniques.

It is now beyond human power and time to seed through it manually; therefore, the research problem of automatic categorization and organizing data is apparent.

The main aim of this project is to identify the underlying sentiment of a movie review on the basis of its textual information. In this project, we try to classify whether a person liked the movie or not based on the review they give for the movie.

This project is exploring the possibility of classifying the movie review corpus as "positive" and "negative" to help enable make better decisions for target user, using data mining techniques.

REQUIREMENTS

HARDWARE:

1) Operating System-WINDOWS/LINUX(UBUNTU)

2) Minimum 4GB RAM

3)500GB Hard disk

SOFTWARE:

1) PYTHON: http://www.python.org/downloads/

2) NUMPY: http://sourceforge.net/projects/numpy/files/Numpy/

3) NLTK: http://pypi.python.org/pypi/nltk

DESIGN

Suitable tasks are performed for implementing the project as listed below:

1. Selective files are subjected to extraction.
2. Choose proper training technique for validation.
3. Identify the test data.
4. Creation of the test module.
5. Next training the module with respective training data.
6. After training the module it is subjected to validation.
7. Verification of the test data.
8. Final step is calculate the accuracy of classification.

CODE

```
#!/usr/bin/python

import os, sys
#import shutil
import nltk
import random
file_paths1=[]
file_paths=[]
count = {}

DIR =r"C:\Users\DELL\AppData\Local\Programs\Python\Python35\mrc\pos"
for root,directories,files in os.walk(DIR):
for filename in files:
filepath=os.path.join(root,filename)
file_paths.append(filepath)

all_words=[]
lnames=[]
lpos=[[[],'pos']]
for p in file_paths:
```

```
lnames=open(p,'r').read().split()
lpos.append([lnames,'pos'])
for w in lnames:
all_words.append(w)
#print(lpos[1])


DIR1=r"C:\Users\DELL\AppData\Local\Programs\Python\Python35\mrc\neg”
for root,directories,files in os.walk(DIR1):
for filename in files:
filepath1=os.path.join(root,filename)
file_paths1.append(filepath1)

for q in file_paths1:
lnames=open(q,'r').read().split()
lpos.append([lnames,'neg'])
for w in lnames:
all_words.append(w)
```

CONCLUSION AND CHALLENGES

One of the major improvements that can be incorporated as we move ahead in this project is to merge words with similar meanings before training the classifiers [3]. Another point of improvement can be to model this problem as a multi-class classification problem where we classify the sentiments of reviewer in more than binary fashion like “Happy”, “Bored”, “Afraid”, etc. This problem can be further remodeled as a regression problem where we can predict the degree of affinity for the movie instead of complete like/dislike. Working on this project has been a good learning experience for us.

The future work includes:

Performing larger scale experiments using our techniques, we could benefit from having larger data set but unfortunately it requires manual work of tagging sentences with labels. Analyzing correspondence between number of Diggs (measure of popularity) and the comments.

REFERENCES

- http://wiki.python.org/moin/BeginnersGuide

- https://docs.python.org/3/tutorial/

- https://developers.google.com/edu/python/