Spam Content - Report

Group 26:

- Samarjoy Pandit
- Saksham
- Vishal
- Akanksha Jojwan
- Saurabh

Source Code:

1. Count articles (a, an, the)

```
text = open('sample.txt').read()
a_count = text.count('a') + text.count('A')
an_count = text.count('an') + text.count('An')
the_count = text.count('the') + text.count('The')
print('Number of a:', a_count)
print('Number of an:', an_count)
print('Number of the:', the_count);
```

Screenshot:

```
Average length of words in the file: 5.413990982374641
```

2. Average length of words

```
s = open('sample.txt').read()
numlist = list(map(len, s.split()))
sum = 0
for x in numlist:
    sum = sum + x
print('Average length of words in the file:',sum/len(numlist))
```

Screenshot:

```
Number of a: 3178
Number of an: 566
Number of the: 469
```

3. tri-grams

```python
import nltk
def get_words(string):
    tokenizer = nltk.tokenize.RegexpTokenizer(r'\w+')
    return tokenizer.tokenize(string)
string = open('sample.txt').read()
words = get_words(string)
finder = nltk.collocations.TrigramCollocationFinder.from_words(words)
scored = finder.score_ngrams(nltk.collocations.TrigramAssocMeasures().raw_freq)
trigrams = finder.ngram_fd.items()
for x in trigrams:
    print(x+"\n")
```

Screenshots:

```
(('Banner', 'logo', 'This'), 1)

(('logo', 'This', 'November'), 1)

(('This', 'November', 'is'), 1)

(('November', 'is', 'the'), 1)

(('is', 'the', 'Wikipedia'), 1)

(('the', 'Wikipedia', 'Asian'), 1)

(('Wikipedia', 'Asian', 'Month'), 1)

(('Asian', 'Month', 'Come'), 1)

(('Month', 'Come', 'join'), 1)

(('Come', 'join', 'us'), 1)

(('Search', 'engine', 'optimization'), 6)

(('engine', 'optimization', 'From'), 2)

(('optimization', 'From', 'Wikipedia'), 2)

(('From', 'Wikipedia', 'the'), 2)

(('Wikipedia', 'the', 'free'), 2)

(('the', 'free', 'encyclopedia'), 2)

(('free', 'encyclopedia', 'SEO'), 2)

(('encyclopedia', 'SEO', 'redirects'), 2)

(('SEO', 'redirects', 'here'), 2)
```