Reading the data

In [1]:
```
from nltk.tokenize import word_tokenize
raw_text = open('report.txt').read()
```

In [2]:
```
print(raw_text[0:1000])
```

```
-----BEGIN PRIVACY-ENHANCED MESSAGE-----
Proc-Type: 2001,MIC-CLEAR
Originator-Name: webmaster@www.sec.gov
Originator-Key-Asymmetric:
 MFgwCgYEVQgBAQICAf8DSgAwRwJAW2sNKK9AVtBzYZmr6aGjlWyK3XmZv3dTINen
 TWSM7vrzLADbmYQaionwg5sDW3P6oaM5D3tdezXMm7z1T+B+twIDAQAB
MIC-Info: RSA-MD5,RSA,
 EvPdKfnjzBIjWkEk2RgNCk1/52qXomHpN+LDwL/XTT/XBuAzk70AYYrsxlQbyiqr
 V5559QRyTgPe9PfVt0db9Q==

<SEC-DOCUMENT>0000950170-98-000413.txt : 19980309
<SEC-HEADER>0000950170-98-000413.hdr.sgml : 19980309
ACCESSION NUMBER:               0000950170-98-000413
CONFORMED SUBMISSION TYPE:      10-K405
PUBLIC DOCUMENT COUNT:          21
CONFORMED PERIOD OF REPORT:     19971228
FILED AS OF DATE:               19980306
SROS:                   NYSE

FILER:

        COMPANY DATA:
                COMPANY CONFORMED NAME:                 SUNBEAM CORP/FL/
                CENTRAL INDEX KEY:                      0000003662
                STANDARD INDUSTRIAL CLASSIFICATION:     ELECTRIC HOUSEWARES & FANS [3634]
                IRS NUMBER:                             251638266
                STATE OF INCORPORATION:                 DE
                FISCAL YEAR END:                        1229

        FILING VALUES:
                FORM TYPE:              10-K405
                SEC ACT:
                SEC FILE NUMBER:        001-00052
                F
```

Cleaning Data

In [5]:
```
import re
text = re.sub("[^a-zA-Z]"," ",str(raw_text)) #removing puncutuations
text1 = text.lower()                         #lower case letters
text2 = " ".join(text1.split())              #removing white-space
print(text2[0:1000])
```

```
begin privacy enhanced message proc type mic clear originator name webmaster www sec gov originator key asymmetric mfgwcgyevqgbaqicaf dsgawrwjaw snkk avtbzyzmr agjlwyk xmzv dtinen twsm vrzladbmyqaionwg sdw p oam d tdezxmm z t b twidaqab mic info
rsa md rsa evpdkfnjzbijwkek rgnck qxomhpn ldwl xtt xbuazk ayyrsxlqbyiqr v qrytgpe pfvt db q sec document txt sec header hdr sgml accession number conformed submission type k public document count conformed period of report filed as of date sros n
yse filer company data company conformed name sunbeam corp fl central index key standard industrial classification electric housewares fans irs number state of incorporation de fiscal year end filing values form type k sec act sec file number fil
m number business address street south congress avenue street suite city delray beach state fl zip business phone mail address street south congress avenue street suite city delray beach state fl zip former company former conformed name sunbeam o
ster company inc
```

TOKENIZING

In [7]:
```
text3 = re.sub("[^\w]", " ",  text2).split()
print(text3[0:100])
```

```
['begin', 'privacy', 'enhanced', 'message', 'proc', 'type', 'mic', 'clear', 'originator', 'name', 'webmaster', 'www', 'sec', 'gov', 'originator', 'key', 'asymmetric', 'mfgwcgyevqgbaqicaf', 'dsgawrwjaw', 'snkk', 'avtbzyzmr', 'agjlwyk', 'xmzv', 'dt
inen', 'twsm', 'vrzladbmyqaionwg', 'sdw', 'p', 'oam', 'd', 'tdezxmm', 'z', 't', 'b', 'twidaqab', 'mic', 'info', 'rsa', 'md', 'rsa', 'evpdkfnjzbijwkek', 'rgnck', 'qxomhpn', 'ldwl', 'xtt', 'xbuazk', 'ayyrsxlqbyiqr', 'v', 'qrytgpe', 'pfvt', 'db',
'q', 'sec', 'document', 'txt', 'sec', 'header', 'hdr', 'sgml', 'accession', 'number', 'conformed', 'submission', 'type', 'k', 'public', 'document', 'count', 'conformed', 'period', 'of', 'report', 'filed', 'as', 'of', 'date', 'sros', 'nyse', 'file
r', 'company', 'data', 'company', 'conformed', 'name', 'sunbeam', 'corp', 'fl', 'central', 'index', 'key', 'standard', 'industrial', 'classification', 'electric', 'housewares', 'fans', 'irs', 'number', 'state', 'of']
```

REMOVING STPOWORDS

In [9]:
```
import nltk
import string
from nltk.corpus import stopwords
stopwords = nltk.corpus.stopwords.words('english')
words_new = [i for i in text3 if i not in stopwords]
print(words_new[0:100])
```

```
['begin', 'privacy', 'enhanced', 'message', 'proc', 'type', 'mic', 'clear', 'originator', 'name', 'webmaster', 'www', 'sec', 'gov', 'originator', 'key', 'asymmetric', 'mfgwcgyevqgbaqicaf', 'dsgawrwjaw', 'snkk', 'avtbzyzmr', 'agjlwyk', 'xmzv', 'dt
inen', 'twsm', 'vrzladbmyqaionwg', 'sdw', 'p', 'oam', 'tdezxmm', 'z', 'b', 'twidaqab', 'mic', 'info', 'rsa', 'md', 'rsa', 'evpdkfnjzbijwkek', 'rgnck', 'qxomhpn', 'ldwl', 'xtt', 'xbuazk', 'ayyrsxlqbyiqr', 'v', 'qrytgpe', 'pfvt', 'db', 'q', 'sec',
'document', 'txt', 'sec', 'header', 'hdr', 'sgml', 'accession', 'number', 'conformed', 'submission', 'type', 'k', 'public', 'document', 'count', 'conformed', 'period', 'report', 'filed', 'date', 'sros', 'nyse', 'filer', 'company', 'data', 'compan
y', 'conformed', 'name', 'sunbeam', 'corp', 'fl', 'central', 'index', 'key', 'standard', 'industrial', 'classification', 'electric', 'housewares', 'fans', 'irs', 'number', 'state', 'incorporation', 'de', 'fiscal', 'year', 'end', 'filing']
```

Lemmatization & Stemming

In [11]:
```
from nltk.stem import PorterStemmer
steam = nltk.PorterStemmer()
text_words = [steam.stem(word) for word in words_new]
print(text_words[0:500])
```

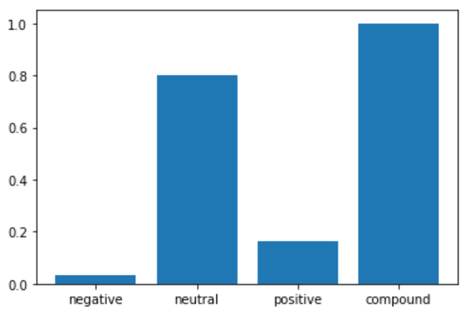
```
['begin', 'privaci', 'enhanc', 'messag', 'proc', 'type', 'mic', 'clear', 'origin', 'name', 'webmast', 'www', 'sec', 'gov', 'origin', 'key', 'asymmetr', 'mfgwcgyevqgbaqicaf', 'dsgawrwjaw', 'snkk', 'avtbzyzmr', 'agjlwyk', 'xmzv', 'dtinen', 'twsm',
'vrzladbmyqaionwg', 'sdw', 'p', 'oam', 'tdezxmm', 'z', 'b', 'twidaqab', 'mic', 'info', 'rsa', 'md', 'rsa', 'evpdkfnjzbijwkek', 'rgnck', 'qxomhpn', 'ldwl', 'xtt', 'xbuazk', 'ayyrsxlqbyiqr', 'v', 'qrytgp', 'pfvt', 'db', 'q', 'sec', 'document', 'tx
t', 'sec', 'header', 'hdr', 'sgml', 'access', 'number', 'conform', 'submiss', 'type', 'k', 'public', 'document', 'count', 'conform', 'period', 'report', 'file', 'date', 'sro', 'nyse', 'filer', 'compani', 'data', 'compani', 'conform', 'name', 'sun
beam', 'corp', 'fl', 'central', 'index', 'key', 'standard', 'industri', 'classif', 'electr', 'housewar', 'fan', 'ir', 'number', 'state', 'incorpor', 'de', 'fiscal', 'year', 'end', 'file', 'valu', 'form', 'type', 'k', 'sec', 'act', 'sec', 'file',
'number', 'film', 'number', 'busi', 'address', 'street', 'south', 'congress', 'avenu', 'street', 'suit', 'citi', 'delray', 'beach', 'state', 'fl', 'zip', 'busi', 'phone', 'mail', 'address', 'street', 'south', 'congress', 'avenu', 'street', 'sui
t', 'citi', 'delray', 'beach', 'state', 'fl', 'zip', 'former', 'compani', 'former', 'conform', 'name', 'sunbeam', 'oster', 'compani', 'inc', 'de', 'date', 'name', 'chang', 'sec', 'header', 'document', 'type', 'k', 'sequenc', 'text', 'unit', 'stat
e', 'secur', 'exchang', 'commiss', 'washington', 'c', 'form', 'k', 'mark', 'one', 'x', 'annual', 'report', 'pursuant', 'section', 'secur', 'exchang', 'act', 'fiscal', 'year', 'end', 'decemb', 'transit', 'report', 'pursuant', 'section', 'secur',
'exchang', 'act', 'fee', 'requir', 'transit', 'period', 'commiss', 'file', 'number', 'sunbeam', 'logo', 'sunbeam', 'corpor', 'exact', 'name', 'registr', 'specifi', 'charter', 'tabl', 'c', 'delawar', 'state', 'jurisdict', 'r', 'employ', 'identif',
'number', 'incorpor', 'organ', 'congress', 'avenu', 'suit', 'delray', 'beach', 'florida', 'address', 'princip', 'execut', 'offic', 'zip', 'code', 'tabl', 'registr', 'telephon', 'number', 'includ', 'area', 'code', 'secur', 'regist', 'pursuant', 's
ection', 'b', 'act', 'titl', 'class', 'name', 'exchang', 'regist', 'common', 'stock', 'par', 'valu', 'new', 'york', 'stock', 'exchang', 'secur', 'regist', 'pursuant', 'section', 'g', 'act', 'none', 'indic', 'check', 'mark', 'whether', 'registr',
'file', 'report', 'requir', 'file', 'section', 'secur', 'exchang', 'act', 'preced', 'month', 'shorter', 'period', 'registr', 'requir', 'file', 'report', 'subject', 'file', 'requir', 'past', 'day', 'ye', 'x', 'indic', 'check', 'mark', 'disclosur',
'delinqu', 'filer', 'pursuant', 'item', 'regul', 'k', 'section', 'chapter', 'contain', 'herein', 'contain', 'best', 'registr', 'knowledg', 'definit', 'proxi', 'inform', 'statement', 'incorpor', 'refer', 'part', 'iii', 'form', 'k', 'amend', 'for
m', 'k', 'x', 'aggreg', 'market', 'valu', 'class', 'registr', 'vote', 'stock', 'held', 'non', 'affili', 'februari', 'approxim', 'februari', 'share', 'registr', 'common', 'stock', 'outstand', 'document', 'incorpor', 'refer', 'portion', 'proxi', 's
tatement', 'annual', 'meet', 'sharehold', 'incorpor', 'refer', 'part', 'iii', 'hereof', 'page', 'sunbeam', 'corpor', 'subsidiari', 'annual', 'report', 'form', 'k', 'tabl', 'content', 'tabl', 'caption', 'page', 'c', 'c', 'part', 'item', 'busi', 'g
ener', 'restructur', 'growth', 'plan', 'pend', 'acquisit', 'product', 'competit', 'strength', 'custom', 'patent', 'trademark', 'employe', 'season', 'raw', 'materi', 'environment', 'matter', 'cautionari', 'statement', 'item', 'properti', 'item',
'legal', 'proceed', 'item', 'submiss', 'matter', 'vote', 'secur', 'holder', 'execut', 'offic', 'registr', 'part', 'ii', 'item', 'market', 'registr', 'common', 'equiti', 'relat', 'stockhold', 'matter', 'item', 'select', 'financi', 'data', 'item',
'manag', 'discuss', 'analysi', 'financi', 'condit', 'result', 'oper', 'item', 'financi', 'statement', 'supplementari', 'data', 'item', 'chang', 'disagr', 'account', 'account', 'financi', 'disclosur', 'part', 'iii', 'item', 'director', 'execut',
'offic', 'registr', 'item', 'execut', 'compens', 'item', 'secur', 'ownership', 'certain', 'benefici', 'owner', 'manag', 'item', 'certain', 'relationship', 'relat', 'transact', 'part', 'iv', 'item', 'exhibit', 'financi', 'statement', 'schedul', 'r
eport', 'form', 'k', 'signatur', 'tabl', 'page', 'part', 'item', 'busi', 'gener', 'sunbeam', 'corpor', 'collect', 'subsidiari', 'compani', 'sunbeam', 'lead', 'design', 'manufactur', 'market', 'brand', 'consum', 'product', 'compani', 'primari', 'b
usi', 'manufactur', 'market', 'distribut', 'durabl']
```

POSITIVE AND NEGATIVE WORDS

In [12]:
```
from nltk.corpus import stopwords
from nltk.tokenize import sent_tokenize
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import matplotlib.pyplot as plt
def sentiment_analyzer(sentiment_text):
    score = SentimentIntensityAnalyzer().polarity_scores(sentiment_text)
    print(score)
    negative = score['neg']
    positive = score['pos']

    if negative > positive:
        print('Neagtive Sentiment')
    else:
        print('Positive sentiment')
    %matplotlib inline
    data = {'neg': 0.034,'neu': 0.803, 'pos': 0.164, 'compound': 1.0}
    names = list(data.keys())
    values = list(data.values())
    labels = ['negative', 'neutral', 'positive', 'compound']
    plt.bar(range(len(data)), values, tick_label = labels)
    plt.show()
sentiment_analysis = sentiment_analyzer(text2)
```

```
{'neg': 0.034, 'neu': 0.803, 'pos': 0.164, 'compound': 1.0}
Positive sentiment
```



Average Sentence Length

In [13]:
```
def Avg_length(text):
    sentence = text.split(".")
    words = text.split(" ")

    if(sentence[len(sentence) - 1] == ""):
        avg_sentence_length = len(words) / len(sentence) - 1
    else:
        avg_sentence_length = len(words) / len(sentence)
    return avg_sentence_length
average = Avg_length(text2)
print(average)
```

```
166116.0
```

Total Word Count

In [ ]:
```
import re
def word_count(text):
    frequency = {}
    pattern = re.findall(r'\b[a-z]{2,15}\b', text)
    for word in pattern:
        count = frequency.get(word, 0)
        frequency[word] = count + 1
        frequency_list = frequency.keys()
        for words in frequency_list:
            print(words, frequency[words])

    return text
total_count = word_count(text2)
print(total_count)
```

In [ ]: