Natural Language Processing

Contents

Tokenization

- Tokenization is the process of tokenizing or splitting a string, text into a list of tokens.
- Example : a word is a token in a sentence, and a sentence is a token in a paragraph.

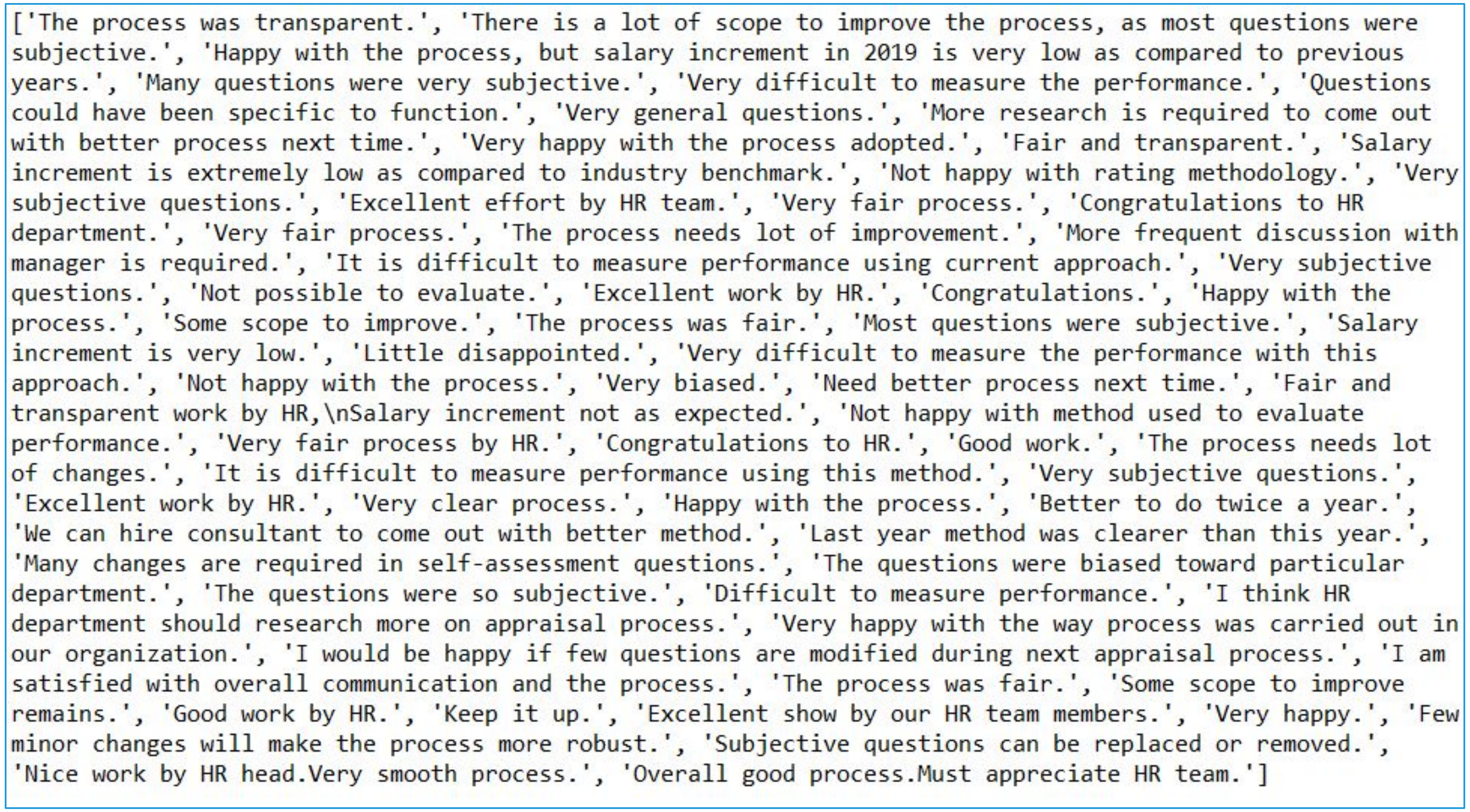
```
# Tokenization
# Import Tokenize from NLTK
# Import data as paragraph string object
from nltk.tokenize import sent_tokenize, word_tokenize
file_data = open('HR Appraisal process.txt', 'r')
data = file_data.read()

print(sent_tokenize(data))
print(word_tokenize(data))
```

❑ **open()** loads a text file in read only format.
❑ **read()** reads the text in the object file_data.
❑ **sent_tokenize()** tokenizes text data into sentences or sentence tokens.
❑ **word_tokenize()** tokenizes text data into word tokens.

* Note : Here we continue using previous data, 'HR Appraisal process' data.

Tokenization

```
# Output :
```



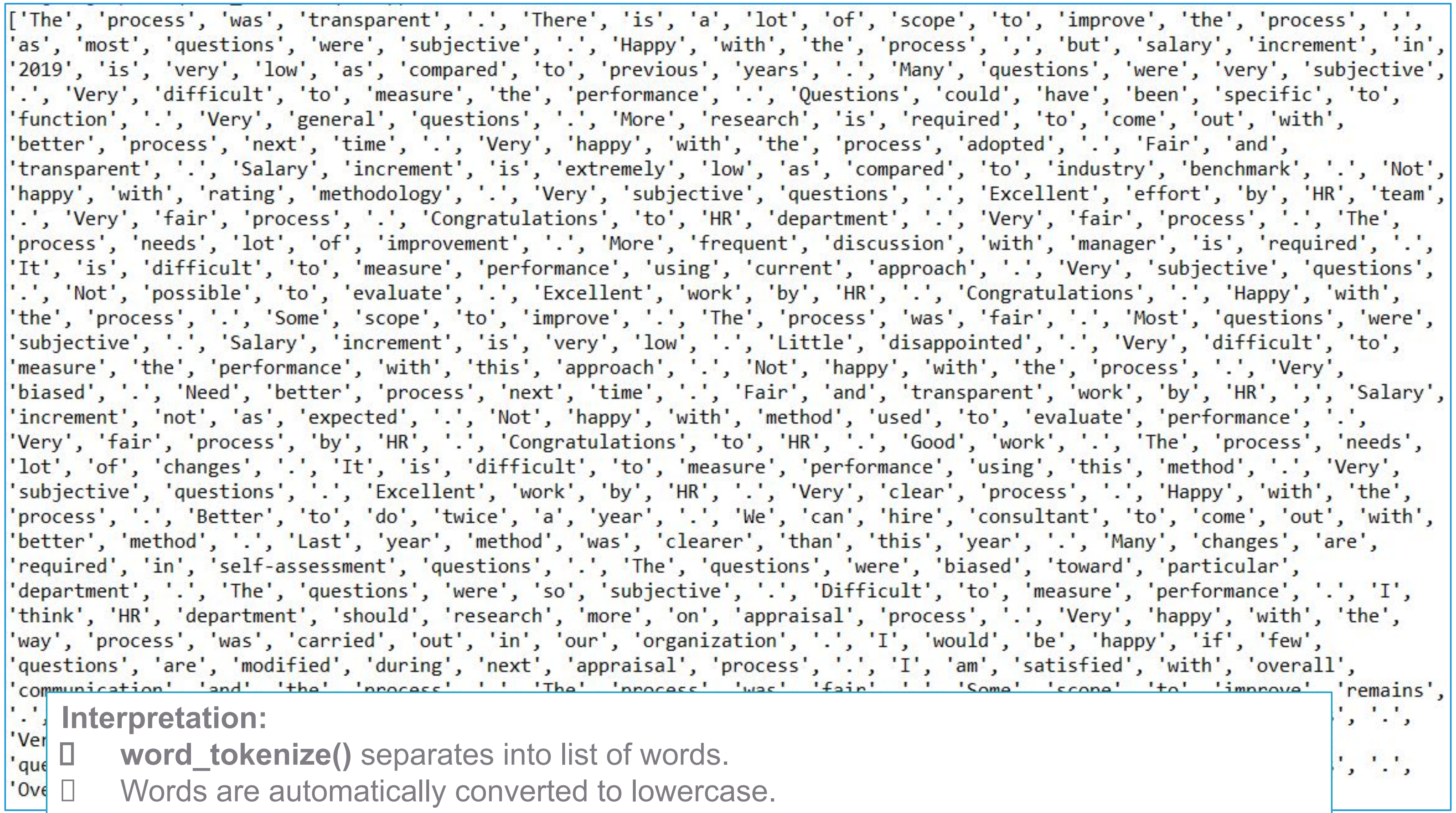
```
['The process was transparent.', 'There is a lot of scope to improve the process, as most questions were
subjective.', 'Happy with the process, but salary increment in 2019 is very low as compared to previous
years.', 'Many questions were very subjective.', 'Very difficult to measure the performance.', 'Questions
could have been specific to function.', 'Very general questions.', 'More research is required to come out
with better process next time.', 'Very happy with the process adopted.', 'Fair and transparent.', 'Salary
increment is extremely low as compared to industry benchmark.', 'Not happy with rating methodology.', 'Very
subjective questions.', 'Excellent effort by HR team.', 'Very fair process.', 'Congratulations to HR
department.', 'Very fair process.', 'The process needs lot of improvement.', 'More frequent discussion with
manager is required.', 'It is difficult to measure performance using current approach.', 'Very subjective
questions.', 'Not possible to evaluate.', 'Excellent work by HR.', 'Congratulations.', 'Happy with the
process.', 'Some scope to improve.', 'The process was fair.', 'Most questions were subjective.', 'Salary
increment is very low.', 'Little disappointed.', 'Very difficult to measure the performance with this
approach.', 'Not happy with the process.', 'Very biased.', 'Need better process next time.', 'Fair and
transparent work by HR,\nSalary increment not as expected.', 'Not happy with method used to evaluate
performance.', 'Very fair process by HR.', 'Congratulations to HR.', 'Good work.', 'The process needs lot
of changes.', 'It is difficult to measure performance using this method.', 'Very subjective questions.',
'Excellent work by HR.', 'Very clear process.', 'Happy with the process.', 'Better to do twice a year.',
'We can hire consultant to come out with better method.', 'Last year method was clearer than this year.',
'Many changes are required in self-assessment questions.', 'The questions were biased toward particular
department.', 'The questions were so subjective.', 'Difficult to measure performance.', 'I think HR
department should research more on appraisal process.', 'Very happy with the way process was carried out in
our organization.', 'I would be happy if few questions are modified during next appraisal process.', 'I am
satisfied with overall communication and the process.', 'The process was fair.', 'Some scope to improve
remains.', 'Good work by HR.', 'Keep it up.', 'Excellent show by our HR team members.', 'Very happy.', 'Few
minor changes will make the process more robust.', 'Subjective questions can be replaced or removed.',
'Nice work by HR head.Very smooth process.', 'Overall good process.Must appreciate HR team.']
```

Interpretation:

- **sent_tokenize()** converts the text into separate sentences.

Tokenization

```
# Output :
```



```
['The', 'process', 'was', 'transparent', '.', 'There', 'is', 'a', 'lot', 'of', 'scope', 'to', 'improve', 'the', 'process', ',',
'as', 'most', 'questions', 'were', 'subjective', '.', 'Happy', 'with', 'the', 'process', ',', 'but', 'salary', 'increment', 'in',
'2019', 'is', 'very', 'low', 'as', 'compared', 'to', 'previous', 'years', '.', 'Many', 'questions', 'were', 'very', 'subjective',
'.', 'Very', 'difficult', 'to', 'measure', 'the', 'performance', '.', 'Questions', 'could', 'have', 'been', 'specific', 'to',
'function', '.', 'Very', 'general', 'questions', '.', 'More', 'research', 'is', 'required', 'to', 'come', 'out', 'with',
'better', 'process', 'next', 'time', '.', 'Very', 'happy', 'with', 'the', 'process', 'adopted', '.', 'Fair', 'and',
'transparent', '.', 'Salary', 'increment', 'is', 'extremely', 'low', 'as', 'compared', 'to', 'industry', 'benchmark', '.', 'Not',
'happy', 'with', 'rating', 'methodology', '.', 'Very', 'subjective', 'questions', '.', 'Excellent', 'effort', 'by', 'HR', 'team',
'.', 'Very', 'fair', 'process', '.', 'Congratulations', 'to', 'HR', 'department', '.', 'Very', 'fair', 'process', '.', 'The',
'process', 'needs', 'lot', 'of', 'improvement', '.', 'More', 'frequent', 'discussion', 'with', 'manager', 'is', 'required', '.',
'It', 'is', 'difficult', 'to', 'measure', 'performance', 'using', 'current', 'approach', '.', 'Very', 'subjective', 'questions',
'.', 'Not', 'possible', 'to', 'evaluate', '.', 'Excellent', 'work', 'by', 'HR', '.', 'Congratulations', '.', 'Happy', 'with',
'the', 'process', '.', 'Some', 'scope', 'to', 'improve', '.', 'The', 'process', 'was', 'fair', '.', 'Most', 'questions', 'were',
'subjective', '.', 'Salary', 'increment', 'is', 'very', 'low', '.', 'Little', 'disappointed', '.', 'Very', 'difficult', 'to',
'measure', 'the', 'performance', 'with', 'this', 'approach', '.', 'Not', 'happy', 'with', 'the', 'process', '.', 'Very',
'biased', '.', 'Need', 'better', 'process', 'next', 'time', '.', 'Fair', 'and', 'transparent', 'work', 'by', 'HR', '.', 'Salary',
'increment', 'not', 'as', 'expected', '.', 'Not', 'happy', 'with', 'method', 'used', 'to', 'evaluate', 'performance', '.',
'Very', 'fair', 'process', 'by', 'HR', '.', 'Congratulations', 'to', 'HR', '.', 'Good', 'work', '.', 'The', 'process', 'needs',
'lot', 'of', 'changes', '.', 'It', 'is', 'difficult', 'to', 'measure', 'performance', 'using', 'this', 'method', '.', 'Very',
'subjective', 'questions', '.', 'Excellent', 'work', 'by', 'HR', '.', 'Very', 'clear', 'process', '.', 'Happy', 'with', 'the',
'process', '.', 'Better', 'to', 'do', 'twice', 'a', 'year', '.', 'We', 'can', 'hire', 'consultant', 'to', 'come', 'out', 'with',
'better', 'method', '.', 'Last', 'year', 'method', 'was', 'clearer', 'than', 'this', 'year', '.', 'Many', 'changes', 'are',
'required', 'in', 'self-assessment', 'questions', '.', 'The', 'questions', 'were', 'biased', 'toward', 'particular',
'department', '.', 'The', 'questions', 'were', 'so', 'subjective', '.', 'Difficult', 'to', 'measure', 'performance', '.', 'I',
'think', 'HR', 'department', 'should', 'research', 'more', 'on', 'appraisal', 'process', '.', 'Very', 'happy', 'with', 'the',
'way', 'process', 'was', 'carried', 'out', 'in', 'our', 'organization', '.', 'I', 'would', 'be', 'happy', 'if', 'few',
'questions', 'are', 'modified', 'during', 'next', 'appraisal', 'process', '.', 'I', 'am', 'satisfied', 'with', 'overall',
'communication', 'and', 'the', 'process', '.', 'The', 'process', 'was', 'fair', '.', 'Some', 'scope', 'to', 'improve', 'remains',
'.',                                                                                                                      ', '.',
'Ver
'que                                                                                                                      ', '.',
'Ove
```

Interpretation:

- **word_tokenize()** separates into list of words.
- Words are automatically converted to lowercase.
- Punctuations are also treated as separate tokens in word tokenization.

What Is NLP?

- **Natural Language Processing (NLP)** is the ability of a computer to analyze and process natural language data.
- NLP is used to extract relevant information from a piece of text which is then used for various purposes.
- NLP works on four levels - lexical, syntactic, semantic, pragmatic.

  - **Lexical**-pre-processing of the text, such as removal of stop words, making all text lowercase etc.

  - **Syntax analysis**-It analyses the 'structure' of text, 'correctness' of a sentence in terms of the grammar of the language of origin.

  - **Semantic assessment**-attempts to study the 'meaning' of the text.

  - **Pragmatic** -the analysis is aimed at deciphering the 'intended' meaning of the text.

Applications of NLP

Some of the most notable applications of NLP are:

- Search algorithms - when you search for “What is the population of India”, the top result shows the actual answer.
- General websites - Pop-up windows on websites offering ‘chat with their representative’ These are chatbots trained to correctly answer commonly asked questions.
- Retail - Assessment of product feedback using text summarization and sentiment analysis; Query resolution with automated responses.
- Personalized services - Email apps predicting next word(s) in an email, tagging emails as important, personal etc.
- Translation Apps

Sentiment Analysis

- Sentiment Analysis is the **process of determining whether a piece of writing is positive, negative or neutral**.
- It's also known as opinion mining, deriving the opinion or attitude of a speaker.
- Sentiment analysis is performed using natural language processing, text analysis, computational linguistics and, sometimes, biometrics to systematically identify, extract, quantify, and study affective states and subjective information.
- Basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level—whether the expressed opinion is positive, negative, or neutral. Advanced - "beyond polarity" - sentiment classification looks at emotional states, for instance, "angry", "sad", and "happy".

Sentiment Analysis Using “TextBlob"

```
# Library for sentimental analysis
pip install textblob
```

```
# Calculate sentiment score for the overall feedback
from textblob import TextBlob
sentiment_analysis = list()
```

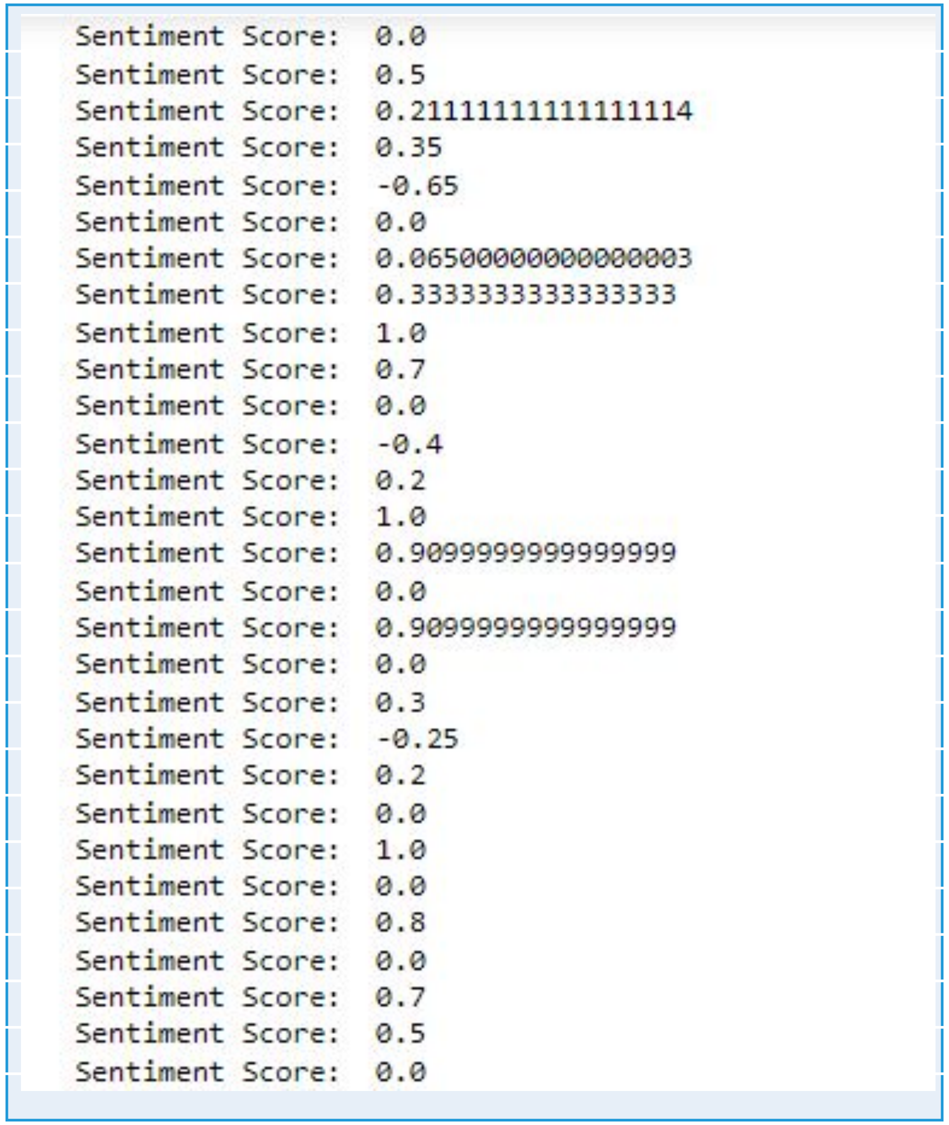
TextBlob() loads text and calculates the score of each sentence on the basis of the presence of words having positive or negative sentiment and presence of negation.

```
# Display the summary of sentiment score of all the documents
import re
sentences = re.split(r' *[\.\?!][\'"\)\]]* *', data)
for i in range(0,len(sentences)):
    sentiment = TextBlob(sentences[i])
    print("Sentiment Score: ", sentiment.sentiment.polarity)
    sentiment_analysis.append(sentiment)
```

sentiment.sentiment.polarity gives probability of sentiment analysis. Polarity is in float which lies in the range of [-1,1] where 1 indicates positive statement and -1 negative.

Sentiment Analysis Using “TextBlob"

```
# Output
```

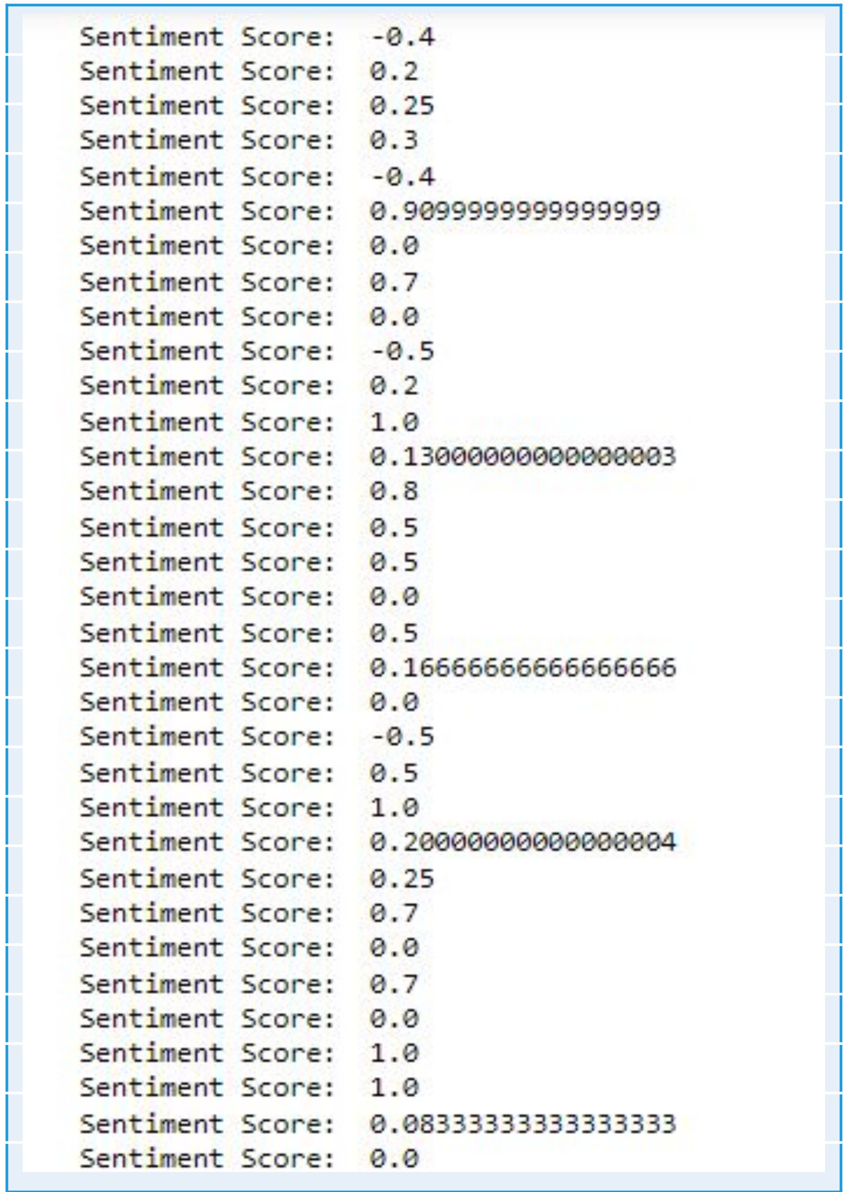


```
Sentiment Score:  0.0
Sentiment Score:  0.5
Sentiment Score:  0.21111111111111114
Sentiment Score:  0.35
Sentiment Score:  -0.65
Sentiment Score:  0.0
Sentiment Score:  0.06500000000000003
Sentiment Score:  0.3333333333333333
Sentiment Score:  1.0
Sentiment Score:  0.7
Sentiment Score:  0.0
Sentiment Score:  -0.4
Sentiment Score:  0.2
Sentiment Score:  1.0
Sentiment Score:  0.9099999999999999
Sentiment Score:  0.0
Sentiment Score:  0.9099999999999999
Sentiment Score:  0.0
Sentiment Score:  0.3
Sentiment Score:  -0.25
Sentiment Score:  0.2
Sentiment Score:  0.0
Sentiment Score:  1.0
Sentiment Score:  0.0
Sentiment Score:  0.8
Sentiment Score:  0.0
Sentiment Score:  0.7
Sentiment Score:  0.5
Sentiment Score:  0.0
```

Interpretation:

- TextBlob sentiment score varies from -1 to 1. 0 indicates neutral, between 0 to 1 it is positive and 0 to -1 is negative.

Sentiment Analysis Using “TextBlob"



```
Sentiment Score:  -0.4
Sentiment Score:  0.2
Sentiment Score:  0.25
Sentiment Score:  0.3
Sentiment Score:  -0.4
Sentiment Score:  0.9099999999999999
Sentiment Score:  0.0
Sentiment Score:  0.7
Sentiment Score:  0.0
Sentiment Score:  -0.5
Sentiment Score:  0.2
Sentiment Score:  1.0
Sentiment Score:  0.13000000000000003
Sentiment Score:  0.8
Sentiment Score:  0.5
Sentiment Score:  0.5
Sentiment Score:  0.0
Sentiment Score:  0.5
Sentiment Score:  0.16666666666666666
Sentiment Score:  0.0
Sentiment Score:  -0.5
Sentiment Score:  0.5
Sentiment Score:  1.0
Sentiment Score:  0.20000000000000004
Sentiment Score:  0.25
Sentiment Score:  0.7
Sentiment Score:  0.0
Sentiment Score:  0.7
Sentiment Score:  0.0
Sentiment Score:  1.0
Sentiment Score:  1.0
Sentiment Score:  0.08333333333333333
Sentiment Score:  0.0
```

Sentiment Analysis Using “vader"

- **VADER (Valence Aware Dictionary and sEntiment Reasoner)** is a lexicon and rule-based sentiment analysis tool in python. It is used mostly for analyzing sentiments on social media.
- In this approach, each of the words in the lexicon are rated as to whether it is positive or negative, and in many cases, how positive or negative.
- While analysing, **VADER checks the text data it finds if any of the words are present in the lexicon.** For example, in the sentence “Happy with the process, but salary increment in 2019 is very low as compared to previous years.” has two words in the lexicon (**happy and low**).
- VADER produces **four sentiment categories**. The first three categories positive, negative, neutral gives proportion of the same. The compound score gives the sum of all the lexicon ratings which are standardised between -1 and 1.

Sentiment Analysis Using “vader"

```
# Download “vader_lexicon” from “nltk”
```

```
nltk.download('vader_lexicon')
```

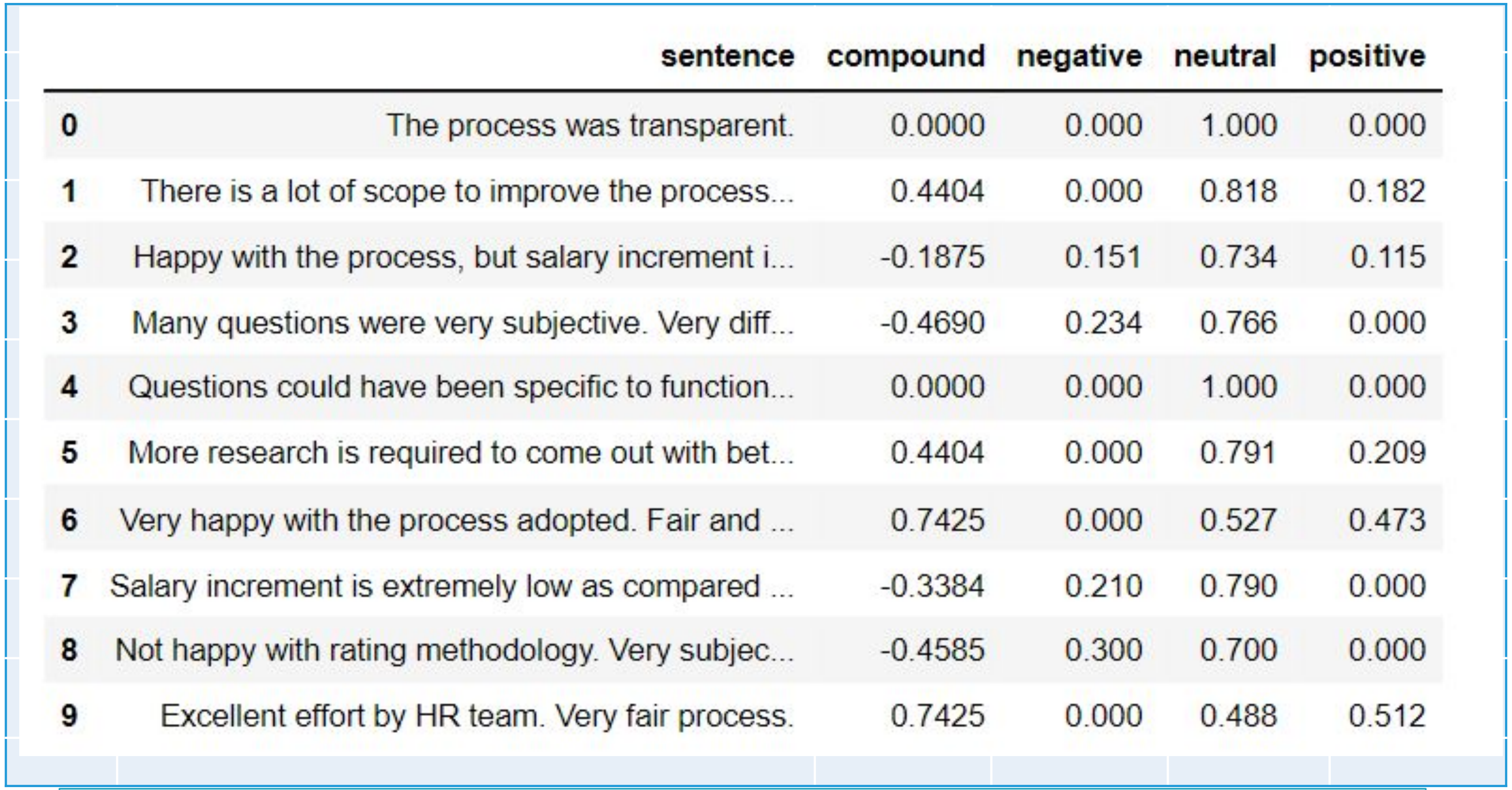
```
# Calculate Sentiment Values
```

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import pandas as pd
sent_analysis = pd.DataFrame(columns =
['sentence','compound','negative', 'neutral','positive'])
sid = SentimentIntensityAnalyzer()
for i in range(0,len(text)):
    ss =  sid.polarity_scores(text[i])
    compound = ss['compound']
    negative = ss['neg']
    neutral = ss['neu']
    positive = ss['pos']
    sent_analysis = sent_analysis.append({"sentence": text[i],
"compound": compound,"negative":  negative,"neutral":
neutral,"positive":  positive}, ignore_index=True)
sent_analysis.head(10)
```

- **SentimentIntensityAnalyzer()** is used in sentiment analysis using Python nltk. It has four values ‘compound’, ‘neg’, ‘neu’, ‘pos’.

* Note : When imported nltk, nltk.download() will download the required libraries from NLTK for text mining. Run nltk.download() only for the first time

Sentiment Analysis Using “vader"



	sentence	compound	negative	neutral	positive
0	The process was transparent.	0.0000	0.000	1.000	0.000
1	There is a lot of scope to improve the process...	0.4404	0.000	0.818	0.182
2	Happy with the process, but salary increment i...	-0.1875	0.151	0.734	0.115
3	Many questions were very subjective. Very diff...	-0.4690	0.234	0.766	0.000
4	Questions could have been specific to function...	0.0000	0.000	1.000	0.000
5	More research is required to come out with bet...	0.4404	0.000	0.791	0.209
6	Very happy with the process adopted. Fair and ...	0.7425	0.000	0.527	0.473
7	Salary increment is extremely low as compared ...	-0.3384	0.210	0.790	0.000
8	Not happy with rating methodology. Very subjec...	-0.4585	0.300	0.700	0.000
9	Excellent effort by HR team. Very fair process.	0.7425	0.000	0.488	0.512

Interpretation:

- Negative compound score indicates negative sentiments. Compound score gives the sum of all the lexicons standardized between -1 and 1.

Quick Recap

In this session, we continued to learn about **Text Mining & NLP in Python** :

Tokenization	• Tokenization is the process of tokenizing or splitting a string, text into a list of tokens. • **`sent_tokenize()`** tokenizes text data into sentences or sentence tokens. • **`word_tokenize()`** tokenizes text data into word tokens.
NLP	• Natural Language Processing (NLP) is the ability of a computer to analyze and process natural language data.
Sentiment Analysis	• Sentiment Analysis is the process of determining whether a piece of writing is positive, negative or neutral. • Download '**vader_lexicon**' from NLTK for sentiment analysis.