Python Script of measuring Text Similarity using Lexical and Semantic Approach

First we measure Text similarity using Lexical method

In [17]:

```
# Imported libaries which are used in this program.
import numpy as np
import seaborn as sns
import pandas as pd
```

In [18]:

```
#We have taken two strings and saved them into list named text_data
doc_A = "Data is the oil of the digital economy"
doc_B = "Data is a new oil"
text_data = [doc_A, doc_B]
print(text_data)
```

```
['Data is the oil of the digital economy', 'Data is a new oil']
```

In [19]:

```
# here we have imported countvectorizer from sci-kit learn libary
# then created vectorizer
#then we have transform the input data and stored it a variable
from sklearn.feature_extraction.text import CountVectorizer
count_vectorizer = CountVectorizer()
matrix = count_vectorizer.fit_transform(text_data)
```

In [20]:

```
#we have stored feature names into variable
tokens = count_vectorizer.get_feature_names()
tokens
```

Out[20]:

```
['data', 'digital', 'economy', 'is', 'new', 'of', 'oil', 'the']
```

In [21]:

```
matrix.toarray()
```

Out[21]:

```
array([[1, 1, 1, 1, 0, 1, 1, 2],
       [1, 0, 0, 1, 1, 0, 1, 0]], dtype=int64)
```

In [22]:

```
#we have created dataframe of given data to present in in form of rows and columns
df = pd.DataFrame(data=matrix.toarray(), index=['doc_1','doc_2'], columns=tokens)
df
```

Out[22]:

	data	digital	economy	is	new	of	oil	the
doc_1	1	1	1	1	0	1	1	2
doc_2	1	0	0	1	1	0	1	0

In [23]:

```
# here we have import cosine_similarity from sci-kit learn library
from sklearn.metrics.pairwise import cosine_similarity

cosine_similarity_matrix = cosine_similarity(matrix)
cosine_similarity_matrix
```

Out[23]:

```
array([[1.        , 0.47434165],
       [0.47434165, 1.        ]])
```

In [24]:

```
#here we have present the result in table format having index and column names
df = pd.DataFrame(data=cosine_similarity_matrix, index=['doc_1','doc_2'], columns=['doc_1','doc_2'])
df
```

Out[24]:

	doc_1	doc_2
doc_1	1.000000	0.474342
doc_2	0.474342	1.000000

This result shows that document_12 is 0.47 % similar to document_1, which means that document are almost 50% same.

Now Measuring Semantic Similarity

In [25]:

```
## Imported libaries which are used in this program.
import pandas as pd
from sentence_transformers import SentenceTransformer, util
import numpy as np
```

In [26]:

```
# we have initialize the sentence transform model
model = SentenceTransformer('all-MiniLM-L6-v2')
```

In [27]:

```
#these are two input sentences
sentence1 = "how old are you"
sentence2 = "what is your age"
print("Sentence 1:", sentence1)
print("Sentence 2:", sentence2)
```

```
Sentence 1: how old are you
Sentence 2: what is your age
```

In [28]:

```
# we have encoded both sentences using sentence tramsform model
embedding1 = model.encode(sentence1)
embedding2 = model.encode(sentence2)
```

In [29]:

```
# we have used cos similarity function to measure similarity between them
sim_score = util.cos_sim(embedding1, embedding2)
```

In [30]:

```
# we have printed sentances and similarity Scores.
print("Sentence 1:", sentence1)
print("Sentence 2:", sentence2)
print("Similarity score:", sim_score.item())
```

```
Sentence 1: how old are you
Sentence 2: what is your age
Similarity score: 0.7851502299308777
```

Similarity Score of these two sentances is 0.785. which means that there is semantic similarity between these sentences.