research

November 9, 2023

[ ]:
```
# !pip install pandas -q
# !pip install numpy -q
# !pip install sklearn -q
```

[1]:
```
import pandas as pd
import numpy as np
import os

pd.set_option('display.max_colwidth', None)
```

[2]:
```
DATA_FILE_PATH = os.path.join('static', 'dataset', 'train.csv')
DATA_FILE_PATH
```

[2]:
```
'static\\dataset\\train.csv'
```

[3]:
```
df = pd.read_csv(DATA_FILE_PATH, nrows = 2900, delimiter = ',')
df.dropna(inplace = True)
df.reset_index(drop = True, inplace = True)
df.head()
```

[3]:
```
       headline  \
0

1      :                                ,

2
:
3                                          ,

4                                                 ,
      …

                                         summary  \
0
                  ( )
          2,000
1
```

.\r\n
.\r\n .
2
, -
' '
3
\r\n
.\r\n15 .
4
\r\n \r\n ' '

article
0
( )
2,000
\r‘ ’

2,000
\n\r‘ ’

2,000

1

.
9/11
.
.\r
.

.\r
9/11 15
" ." \r
. ‘‘
.’’ 9/11
. \r

.(,
,
)\n\r
.

.\r
9/11 15 "
." \r

. ‘‘
.’’ 9/11
. \r

.(,
,
)\n\r
9/11 15 "
." \r
. ‘‘
.’’ 9/11
. \r

.(,
,
)\n\r . ‘‘
.’’ 9/11

. \r

.(,
,)\n\r

.(,
,
)\n(, ,
)
2
, -
' ' \r
-20 -

\r 619

,
\r
,
,
\r ,

, ( ) ,
\r
, , 400

\r
,

\r ,

,
,
\r
‘ ,
,

,
\r ,
,
\n\r -20
-

\r 619

, \r
,

,
\r ,

,
( ) ,
\r , ,
400

\r
,

\r ,

,
,

\r ‘
,
,

, \r
, ,
\n\r 619

,
\r
,
,
\r ,

, ( ) ,
\r
, , 400

\r
,

\r ,

,
,
\r
‘ ,
,

,
\r ,
,
\n\r
,
,
\r ,

, ( ) ,
\r
, , 400

\r

,

\r ,

,
,
\r
‘ ,
,

,
\r ,
,
\n\r ,

, ( ) ,
\r ,
, 400

\r
,

\r ,

,
,
\r ‘
, ,

,
\r , ,
\n\r
, , 400

\r
,

\r
,

,
,

\r ‘
,
,

, \r
, ,
\n\r
,

\r
,

,
,

\r ‘
,
,

, \r
, ,
\n\r ,

,
,
\r
‘ ,
,

,
\r ,
,
\n\r ‘
, ,

, \r
, ,
\n\r
, ,

3
'

. .'\r

'
… …'\n\r
'
…
…'
4
' '
(Meezaan Jaffery) .
(Meezaan Jaffery) (Navya Naveli Nanda)
.
(Navya Naveli Nanda) (Meezaan Jaffery)
. (Meezaan Jaffery) (Navya
Naveli Nanda) . ,
(Meezaan Jaffery) (Navya
Naveli Nanda) .\n\nBlur But Beautiful ????
@meezaanj and @navyananda . #meezaan #navyanavelinanda #sanjayleelabhansali
#vimalfilmfareawards2019 #zeecineawards #bollywood #debutante #malaal #malal
#slb #inshallah #salmankhan #aliabhatt\nA post shared by meezaan's world
(@meezaan_jaffery) on Mar 26, 2019 at 4:37am PDT\n
(Meezaan Jaffery) ,
(Navya Naveli Nanda) .
(Meezaan Jaffery) " ."
' .
.
.'
(Alaviaa Jaaferi) .
. \n\nA post shared by Alaviaa
Jaaferi (@alaviaajaaferi) on Oct 13, 2018 at 1:18pm PDT\n\n
(Meezaan Jaffery)
' ' (Malaal) . (Meezaan

```
Jaffery)        5                  .
                           .                   ,
                              .                 (Navya
Naveli Nanda)

                    .
                                  .
                           .
```

[4]:
```
df.shape
```

[4]:
```
(202, 3)
```

[5]:
```
df
```

[5]:
```
          headline   \
0

1         :                                ,

2
   :
3                                                ,

4
   ,        …
..
…
197           PAKvsWI :     -
56
198                                       24                 ,
12
199         Om Birla:                                       ,
    10
200
  ,
201                                                       -
       : US

                                              summary   \
0
                       (  )
          2,000
1
          .\r\n
 .\r\n                              .
2
```

, -
' '
3
\r\n
.\r\n15 .
4
\r\n \r\n ' '

..
…
197
289 \r\n 346 \r\n
116
198
\r\n40
\r\n
199 4
1962 . \r\n
.\r\n .
200
\r\n \r\n

201

article
0
( )
2,000
\r‘ ’

2,000
\n\r‘ ’

2,000

1

.
9/11
.
.\r
.

.\r

9/11 15
" ." \r
. ‘‘
.’’ 9/11
. \r

.(,
,
)\n\r
.

.\r
9/11 15 "
." \r
. ‘‘
.’’ 9/11
. \r

.(,
,
)\n\r
9/11 15 "
." \r
. ‘‘
.’’ 9/11
. \r

.(,
,
)\n\r . ‘‘
.’’ 9/11

. \r

.(,
,)\n\r

.(,
,
)\n(, ,
)
2
, -
' ' \r
-20 -

\r 619

,
\r
,
,
\r ,

, ( ) ,
\r
, , 400

\r
,

\r ,

,
,
\r
‘ ,
,

,
\r ,
,
\n\r -20
-

\r 619

, \r
,

,
\r ,
,
( ) ,
\r , ,
400

\r

,

\r ,

,
,

\r ‘
, ,

, \r
, ,
\n\r 619

,
\r
,

,
\r ,

, ( ) ,
\r
, , 400

\r
,

\r ,

,
,
\r
‘ ,
,

,
`\r` ,
,
`\n\r`
,
,
`\r` ,

, ( ) ,
`\r`
, , 400

`\r`
,

`\r` ,

,
,
`\r`
‘ ,
,

,
`\r` ,
,
`\n\r` ,

, ( ) ,
`\r` ,
, 400

`\r`
,

`\r` ,

,
,
\r ‘
, ,

,
\r , ,
\n\r
, , 400

\r
,

\r
,

,
,

\r ‘
, ,

, \r
, ,
\n\r
,

\r
,

,
,

\r ‘

, ,
, \r
, ,
\n\r ,
,
,
\r
‘ ,
,
,
\r ,
,
\n\r ‘
, ,
, \r
, ,
\n\r
, ,
3
'
. .'\r
'
… …'\n\r
'
…
…'
4
' '
(Meezaan Jaffery) .
(Meezaan Jaffery) (Navya Naveli Nanda)
.
(Navya Naveli Nanda) (Meezaan Jaffery)
. (Meezaan Jaffery) (Navya
Naveli Nanda) . ,
(Meezaan Jaffery) (Navya
Naveli Nanda) .\n\nBlur But Beautiful ????
@meezaanj and @navyananda . #meezaan #navyanavelinanda #sanjayleelabhansali

#vimalfilmfareawards2019 #zeecineawards #bollywood #debutante #malaal #malal
#slb #inshallah #salmankhan #aliabhatt\nA post shared by meezaan's world
(@meezaan_jaffery) on Mar 26, 2019 at 4:37am PDT\n
(Meezaan Jaffery) ,
(Navya Naveli Nanda) .
(Meezaan Jaffery) " ."
' .
.
.'
(Alaviaa Jaaferi) .
. \n\nA post shared by Alaviaa
Jaaferi (@alaviaajaaferi) on Oct 13, 2018 at 1:18pm PDT\n\n
(Meezaan Jaffery)
' ' (Malaal) . (Meezaan
Jaffery) 5 .
. ,
. (Navya
Naveli Nanda)

.
.
.
..
…
197
-
56 1-0
. 116
. 289 .
346 .
.\r
. 224

.\r
95 . (04)
.
(15) 126 .
\r
.
(00) .

289 .(,
,)\n\r
. 224

.\r
95 . (04)

.
(15) 126 .
\r
.
(00) .

289 .(,
,)\n\r
95 .
(04) .
(15) 126
. \r
.
(00) .

289 .(,
,)\n\r

.
(00) .
289
.(, ,
)\n(,
,)
198
24 ,
12 74 .
.
16 .
, 600 120
.\r 12 .
.
.\r
,
. \r ,
.\r
200 , 40 20
.\r 74
, 20 .
,
.\r

.
.\r , '
. 29
'.\r
. \r ,

.
.
.
.\r( )\n\r 12
. .
.\r
,
. \r ,
.\r
200 , 40 20
.\r 74
, 20 .
,
.\r

.
.\r , '
. 29
'.\r
. \r ,

.
. .
.\r( )\n\r ,
. \r ,

.\r 200 , 40
20 .\r
74 , 20 .
,
.\r

.
.\r
, ' . 29
'.\r

. \r ,
.
. .
.\r( )\n\r ,

.\r 200
, 40 20 .\r
74 , 20
. ,

.\r
.
.\r
, ' .
29
'.\r
. \r ,
.
.
. .\r(
)\n\r 200 , 40
20 .\r 74
, 20 .
,
.\r

.
.\r ,
' . 29
'.\r

. \r ,
.
. .
.\r( )\n\r
74 , 20
. ,

.\r
.
.\r
, ' .
29
'.\r
. \r ,
.
.
. .\r(
)\n\r
.
.\r
, ' .
29
'.\r
. \r ,
.
.

. .\r(
)\n\r , '
. 29
'.\r
. \r ,
.
.
.
.\r( )\n\r
. \r ,

.
. .
.\r( )\n\r ,
.
. .
.\r( )\n\r( )
199
(OM Birla) .
(Om Birla)
. 2.5
. 4 1962
. 17
. (BJP MP Om Birla) . 2003,
2008 2013 12 , 13 14 .
10 ..\r \n
,
\n

200
,
.
–
. 6
.
( )
.
. \r
,
.
.
.
–
.\r \r\r .\r \r\r

.
.

.\r \r 6 .\r
.\r \r A post
shared by Amrita Arora (@amuaroraofficial) on Jun 17, 2017 at 1:08am PDT \r A
post shared by Amrita Arora (@amuaroraofficial) on Jun 13, 2017 at 1:00am
PDT \r A post shared by Amrita Arora (@amuaroraofficial) on May 29, 2017 at
12:09am PDT A post shared by Amrita Arora (@amuaroraofficial) on May 29, 2017 at
1:02am PDT\r 20
. .
' '
.\n\r ,
.
.
.
- .\r \r\r .\r \r\r
.
.
.\r \r 6
.\r
.\r \r A post shared by Amrita Arora (@amuaroraofficial) on Jun 17, 2017 at
1:08am PDT \r A post shared by Amrita Arora (@amuaroraofficial) on Jun 13, 2017
at 1:00am PDT \r A post shared by Amrita Arora (@amuaroraofficial) on May 29,
2017 at 12:09am PDT A post shared by Amrita Arora (@amuaroraofficial) on May 29,
2017 at 1:02am PDT\r 20
. .
' '
.\n\r
.
.
.\r \r
6 .\r
.\r \r A post shared by Amrita Arora
(@amuaroraofficial) on Jun 17, 2017 at 1:08am PDT \r A post shared by Amrita
Arora (@amuaroraofficial) on Jun 13, 2017 at 1:00am PDT \r A post shared by
Amrita Arora (@amuaroraofficial) on May 29, 2017 at 12:09am PDT A post shared by
Amrita Arora (@amuaroraofficial) on May 29, 2017 at 1:02am PDT\r
20 .
.
' ' .\n
6 .\nA post shared by Amrita Arora
(@amuaroraofficial) on Jun 17, 2017 at 1:08am PDT\nA post shared by Amrita Arora
(@amuaroraofficial) on Jun 13, 2017 at 1:00am PDT\nA post shared by Amrita Arora
(@amuaroraofficial) on May 29, 2017 at 12:09am PDT\nA post shared by Amrita
Arora (@amuaroraofficial) on May 29, 2017 at 1:02am PDT
201

```
                    \r          ,

     \r
  ,
               ,

                    ?                 ?
                          \r      ,

 \n\r          ,
                                    \r
                               ,

    ,

     ?                  ?
         \r      ,
                          \n\r
                       ,
                                 ,

                                      ?
             ?
\r      ,
                        \n\r      ,



[202 rows x 3 columns]
```

[6]:
```python
NEW_CSV_FILE = os.path.join('static', 'dataset', 'train_1.csv')
df.to_csv(NEW_CSV_FILE, index = False)
```

1 Split csv to 10 Kb and 100 Kb file

[7]:
```python
from sklearn.model_selection import train_test_split
```

[8]:
```python
df = pd.read_csv(NEW_CSV_FILE)
df.head()
```

[8]:
```
       headline  \
0

1       :                                ,
```

2
:
3 ,

4 ,
…

summary \
0
( )
2,000
1
.\r\n
.\r\n .
2
, -
' '
3
\r\n
.\r\n15 .
4
\r\n \r\n ' '

article
0
( )
2,000
\r‘ ’

2,000
\n\r‘ ’

2,000

1

.
9/11
.
.\r
.

.\r
9/11 15
" ." \r

. ‘‘
.’’ 9/11
. \r

.(,
,
)\n\r
.

.\r
9/11 15 "
." \r
. ‘‘
.’’ 9/11
. \r

.(,
,
)\n\r
9/11 15 "
." \r
. ‘‘
.’’ 9/11
. \r

.(,
,
)\n\r . ‘‘
.’’ 9/11

. \r

.(,
,)\n\r

.(,
,
)\n(, ,
)
2
, -
' ' \r
-20 -

\r 619

,

\r
,
,
\r ,

, ( ) ,
\r
, , 400

\r
,

\r ,

,
,
\r
‘ ,
,

,
\r ,
,
\n\r -20
-

\r 619

, \r
,

,
\r ,
,
( ) ,
\r , ,
400

\r
,

\r ,

,
,

\r ‘
, ,

, \r
, ,
\n\r 619

,
\r
,

,
\r ,

, ( ) ,
\r
, , 400

\r
,

\r ,

,
,
\r
‘ ,
,

,
\r ,

,
\n\r
,
,
\r ,

, ( ) ,
\r
, , 400

\r
,

\r ,

,
,
\r
‘ ,
,

,
\r ,
,
\n\r ,

, ( ) ,
\r ,
, 400

\r
,

\r ,

,
,
\r ‘
,
,

,
\r , ,
\n\r
, , 400

\r
,

\r
,

,
,

\r ‘
,
,

, \r
, ,
\n\r
,

\r
,

,
,

\r ‘
,
,

, \r
, ,
\n\r ,

,
,
\r
‘ ,
,

,
\r ,
,
\n\r ‘
, ,

, \r
, ,
\n\r
, ,

3
'

. .'\r

'
… …'\n\r
'

…
…'
4
' '
(Meezaan Jaffery) .
(Meezaan Jaffery) (Navya Naveli Nanda)
.
(Navya Naveli Nanda) (Meezaan Jaffery)
. (Meezaan Jaffery) (Navya
Naveli Nanda) . ,
(Meezaan Jaffery) (Navya
Naveli Nanda) .\n\nBlur But Beautiful ????
@meezaanj and @navyananda . #meezaan #navyanavelinanda #sanjayleelabhansali
#vimalfilmfareawards2019 #zeecineawards #bollywood #debutante #malaal #malal
#slb #inshallah #salmankhan #aliabhatt\nA post shared by meezaan's world

```
(@meezaan_jaffery) on Mar 26, 2019 at 4:37am PDT\n
               (Meezaan Jaffery)         ,
      (Navya Naveli Nanda)                      .
       (Meezaan Jaffery)     "          ."
                 '                    .
                         .
   .'
    (Alaviaa Jaaferi)              .
                          . \n\nA post shared by Alaviaa
Jaaferi (@alaviaajaaferi) on Oct 13, 2018 at 1:18pm PDT\n\n
                     (Meezaan Jaffery)
         '  ' (Malaal)               .        (Meezaan
Jaffery)        5                   .
                      .                  ,
                       .              (Navya
Naveli Nanda)

                .
                              .
                      .
```

[9]:
```
train_1ookb, test_1ookb = train_test_split(df, test_size=0.10,␣
  ↪random_state=104, shuffle=True)
```

[10]:
```
train_1ookb.shape
```

[10]: `(181, 3)`

[11]:
```
test_1ookb.shape
```

[11]: `(21, 3)`

[12]:
```
train_1ookb_CSV_FILE = os.path.join('static', 'dataset', 'train_100kb.csv')
train_1ookb.to_csv(train_1ookb_CSV_FILE, index = False)

test_1ookb_CSV_FILE = os.path.join('static', 'dataset', 'test_100kb.csv')
test_1ookb.to_csv(test_1ookb_CSV_FILE, index = False)
```

[13]:
```
train_1okb, test_1okb = train_test_split(df, test_size=0.01, random_state=104,␣
  ↪shuffle=True)
```

[14]:
```
train_1okb.shape
```

[14]: `(199, 3)`

[15]:
```
test_1okb.shape
```

[15]:

```
(3, 3)
```

[16]:

```
train_1okb_CSV_FILE = os.path.join('static', 'dataset', 'train_10kb.csv')
train_1okb.to_csv(train_1okb_CSV_FILE, index = False)

test_1okb_CSV_FILE = os.path.join('static', 'dataset', 'test_10kb.csv')
test_1okb.to_csv(test_1okb_CSV_FILE, index = False)
```

2 Find Time and space complexity

3 predict and find similarity test

[17]:

```
from TextSummarization_utils import *
import nltk
from nltk.util import ngrams
from nltk.metrics import precision, recall, f_measure
import time
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\DELL\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\DELL\AppData\Roaming\nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\DELL\AppData\Roaming\nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
C:\Users\DELL\AppData\Local\Programs\Python\Python310\lib\site-
packages\tqdm\auto.py:21: TqdmWarning: IProgress not found. Please update
jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
C:\Project_Work\Basic_Python\Suresh Interview Project\Text Summarization using
Deep Learning\Text Summarization in Hindi
Deploy_Hindi\TextSummarization_utils.py:44: FutureWarning: Possible nested set
at position 1
  patternEnglish = re.compile(r'[[A-Z]|[a-z]]*')
```

[18]:

```
data_10kb_CSV_FILE = os.path.join('static', 'dataset', 'test_10kb.csv')
data_100kb_CSV_FILE = os.path.join('static', 'dataset', 'test_100kb.csv')
data_1mb_CSV_FILE = os.path.join('static', 'dataset', 'train_100kb.csv')

data_files = [data_10kb_CSV_FILE, data_100kb_CSV_FILE, data_1mb_CSV_FILE]

model_list = ['Term frequency weighting', 'TFIDF sentence weighting', 'Rule␣
  ↪based',
```

```
               'Bidirectional Encoder Representations from Transformers (BERT)' ]
```

```
[19]: # Function to calculate ROUGE scores for n-grams
def rouge_n_score(machine_tokens, reference_tokens, n):
    machine_ngrams = list(ngrams(machine_tokens, n))
    reference_ngrams = list(ngrams(reference_tokens, n))

    # Calculate precision, recall, and F1-score
    precision_score = precision(set(machine_ngrams), set(reference_ngrams))
    recall_score = recall(set(machine_ngrams), set(reference_ngrams))
    f1_score = f_measure(set(machine_ngrams), set(reference_ngrams))

    return precision_score, recall_score, f1_score
```

4 Term frequency weighting

- data - 0 == 10kb
- data - 1 == 100kb
- data - 2 == 1mb

```
[20]: print('Term frequency weighting')
# run loop
for file in data_files: # [0:1]
    # to get time complexity
    start_time = time.time()

    print(f"data - {data_files.index(file)}")
    # load data
    df = pd.read_csv(file)

    ROUGE_1_Precision, ROUGE_1_Recall, ROUGE_1_F1_Score = [], [], []
    ROUGE_2_Precision, ROUGE_2_Recall, ROUGE_2_F1_Score = [], [], []
    ROUGE_3_Precision, ROUGE_3_Recall, ROUGE_3_F1_Score = [], [], []
    ROUGE_4_Precision, ROUGE_4_Recall, ROUGE_4_F1_Score = [], [], []

    for ind in range(len(df)):
        try:
            summary, article = df['summary'][ind], df['article'][ind]
            clean_sentences = get_clean_sentences(article)
            summary_pred =␣
 ↪summarise_term_frequency_sentence_weighing(clean_sentences)

            # Tokenize the sentences
            reference_tokens = nltk.word_tokenize(summary)
            machine_tokens = nltk.word_tokenize(summary_pred)

            # Calculate ROUGE-1, ROUGE-2, ROUGE-3, and ROUGE-4 scores
```

```
            rouge_1 = rouge_n_score(machine_tokens, reference_tokens, 1)
            rouge_2 = rouge_n_score(machine_tokens, reference_tokens, 2)
            rouge_3 = rouge_n_score(machine_tokens, reference_tokens, 3)
            rouge_4 = rouge_n_score(machine_tokens, reference_tokens, 4)

            ROUGE_1_Precision.append(float(rouge_1[0]))
            ROUGE_1_Recall.append(float(rouge_1[1]))
            ROUGE_1_F1_Score.append(float(rouge_1[2]))

            ROUGE_2_Precision.append(float(rouge_2[0]))
            ROUGE_2_Recall.append(float(rouge_2[1]))
            ROUGE_2_F1_Score.append(float(rouge_2[2]))

            ROUGE_3_Precision.append(float(rouge_3[0]))
            ROUGE_3_Recall.append(float(rouge_3[1]))
            ROUGE_3_F1_Score.append(float(rouge_3[2]))

            ROUGE_4_Precision.append(float(rouge_4[0]))
            ROUGE_4_Recall.append(float(rouge_4[1]))
            ROUGE_4_F1_Score.append(float(rouge_4[2]))

        except Exception as e:
            pass

    print("ROUGE-1 Precision:", np.mean(ROUGE_1_Precision))
    print("ROUGE-1 Recall:", np.mean(ROUGE_1_Recall))
    print("ROUGE-1 F1-Score:", np.mean(ROUGE_1_F1_Score))

    print("ROUGE-2 Precision:", np.mean(ROUGE_2_Precision))
    print("ROUGE-2 Recall:", np.mean(ROUGE_2_Recall))
    print("ROUGE-2 F1-Score:", np.mean(ROUGE_2_F1_Score))

    print("ROUGE-3 Precision:", np.mean(ROUGE_3_Precision))
    print("ROUGE-3 Recall:", np.mean(ROUGE_3_Recall))
    print("ROUGE-3 F1-Score:", np.mean(ROUGE_3_F1_Score))

    print("ROUGE-4 Precision:", np.mean(ROUGE_4_Precision))
    print("ROUGE-4 Recall:", np.mean(ROUGE_4_Recall))
    print("ROUGE-4 F1-Score:", np.mean(ROUGE_4_F1_Score))

    end_time = time.time()
    execution_time = end_time - start_time
    print(f"Execution time: {round(execution_time,2)} seconds")

    print('\n')
```

Term frequency weighting

```
data - 0
ROUGE-1 Precision: 0.5352564102564102
ROUGE-1 Recall: 0.09485053752569676
ROUGE-1 F1-Score: 0.16068139963167588
ROUGE-2 Precision: 0.3148604269293924
ROUGE-2 Recall: 0.04385615123320041
ROUGE-2 F1-Score: 0.07680502107385011
ROUGE-3 Precision: 0.23563218390804597
ROUGE-3 Recall: 0.030509717638430512
ROUGE-3 F1-Score: 0.053957636566332214
ROUGE-4 Precision: 0.18386243386243384
ROUGE-4 Recall: 0.023070913983161186
ROUGE-4 F1-Score: 0.040955250530074774
Execution time: 0.09 seconds


data - 1
ROUGE-1 Precision: 0.4346111974587397
ROUGE-1 Recall: 0.15495431980189023
ROUGE-1 F1-Score: 0.21393002394204041
ROUGE-2 Precision: 0.20021267408328183
ROUGE-2 Recall: 0.0619873260587549
ROUGE-2 F1-Score: 0.08804192030864041
ROUGE-3 Precision: 0.13132657375883977
ROUGE-3 Recall: 0.04251591269091361
ROUGE-3 F1-Score: 0.05962512731613724
ROUGE-4 Precision: 0.10030776102204672
ROUGE-4 Recall: 0.03392725301173105
ROUGE-4 F1-Score: 0.04727893428635939
Execution time: 0.18 seconds


data - 2
ROUGE-1 Precision: 0.4762101055103181
ROUGE-1 Recall: 0.21930734263219762
ROUGE-1 F1-Score: 0.27976129197782434
ROUGE-2 Precision: 0.23691836974891453
ROUGE-2 Recall: 0.10207732895557671
ROUGE-2 F1-Score: 0.13014602453423907
ROUGE-3 Precision: 0.16903520323641477
ROUGE-3 Recall: 0.07624689703420513
ROUGE-3 F1-Score: 0.09653932248826436
ROUGE-4 Precision: 0.1440867504247724
ROUGE-4 Recall: 0.0638705955490682
ROUGE-4 F1-Score: 0.0801899144717704
Execution time: 0.88 seconds
```

5 TFIDF sentence weighting

[21]:
```
print('TFIDF sentence weighting')
# run loop
for file in data_files: # [0:1]
    # to get time complexity
    start_time = time.time()

    print(f"data - {data_files.index(file)}")
    # load data
    df = pd.read_csv(file)

    ROUGE_1_Precision, ROUGE_1_Recall, ROUGE_1_F1_Score = [], [], []
    ROUGE_2_Precision, ROUGE_2_Recall, ROUGE_2_F1_Score = [], [], []
    ROUGE_3_Precision, ROUGE_3_Recall, ROUGE_3_F1_Score = [], [], []
    ROUGE_4_Precision, ROUGE_4_Recall, ROUGE_4_F1_Score = [], [], []

    for ind in range(len(df)):
        try:
            summary, article = df['summary'][ind], df['article'][ind]
            clean_sentences = get_clean_sentences(article)
            summary_pred = summarise_tf_idf_sentence_weighting(clean_sentences)

            # Tokenize the sentences
            reference_tokens = nltk.word_tokenize(summary)
            machine_tokens = nltk.word_tokenize(summary_pred)

            # Calculate ROUGE-1, ROUGE-2, ROUGE-3, and ROUGE-4 scores
            rouge_1 = rouge_n_score(machine_tokens, reference_tokens, 1)
            rouge_2 = rouge_n_score(machine_tokens, reference_tokens, 2)
            rouge_3 = rouge_n_score(machine_tokens, reference_tokens, 3)
            rouge_4 = rouge_n_score(machine_tokens, reference_tokens, 4)

            ROUGE_1_Precision.append(float(rouge_1[0]))
            ROUGE_1_Recall.append(float(rouge_1[1]))
            ROUGE_1_F1_Score.append(float(rouge_1[2]))

            ROUGE_2_Precision.append(float(rouge_2[0]))
            ROUGE_2_Recall.append(float(rouge_2[1]))
            ROUGE_2_F1_Score.append(float(rouge_2[2]))

            ROUGE_3_Precision.append(float(rouge_3[0]))
            ROUGE_3_Recall.append(float(rouge_3[1]))
            ROUGE_3_F1_Score.append(float(rouge_3[2]))

            ROUGE_4_Precision.append(float(rouge_4[0]))
            ROUGE_4_Recall.append(float(rouge_4[1]))
```

```
            ROUGE_4_F1_Score.append(float(rouge_4[2]))

        except Exception as e:
            pass

    print("ROUGE-1 Precision:", np.mean(ROUGE_1_Precision))
    print("ROUGE-1 Recall:", np.mean(ROUGE_1_Recall))
    print("ROUGE-1 F1-Score:", np.mean(ROUGE_1_F1_Score))

    print("ROUGE-2 Precision:", np.mean(ROUGE_2_Precision))
    print("ROUGE-2 Recall:", np.mean(ROUGE_2_Recall))
    print("ROUGE-2 F1-Score:", np.mean(ROUGE_2_F1_Score))

    print("ROUGE-3 Precision:", np.mean(ROUGE_3_Precision))
    print("ROUGE-3 Recall:", np.mean(ROUGE_3_Recall))
    print("ROUGE-3 F1-Score:", np.mean(ROUGE_3_F1_Score))

    print("ROUGE-4 Precision:", np.mean(ROUGE_4_Precision))
    print("ROUGE-4 Recall:", np.mean(ROUGE_4_Recall))
    print("ROUGE-4 F1-Score:", np.mean(ROUGE_4_F1_Score))

    end_time = time.time()
    execution_time = end_time - start_time
    print(f"Execution time: {round(execution_time,2)} seconds")

    print('\n')
```

```
TFIDF sentence weighting
data - 0
ROUGE-1 Precision: 0.422008547008547
ROUGE-1 Recall: 0.047809829059829057
ROUGE-1 F1-Score: 0.08532672800188723
ROUGE-2 Precision: 0.17323481116584563
ROUGE-2 Recall: 0.012824247468442226
ROUGE-2 F1-Score: 0.023748587766018767
ROUGE-3 Precision: 0.09523809523809523
ROUGE-3 Recall: 0.005544005544005544
ROUGE-3 F1-Score: 0.010478061558611656
ROUGE-4 Precision: 0.07407407407407407
ROUGE-4 Recall: 0.004056795131845842
ROUGE-4 F1-Score: 0.007692307692307692
Execution time: 0.02 seconds


data - 1
ROUGE-1 Precision: 0.6212870352751305
ROUGE-1 Recall: 0.18222823084127837
```

```
ROUGE-1 F1-Score: 0.25816412721912674
ROUGE-2 Precision: 0.4520077786128547
ROUGE-2 Recall: 0.10432817933776015
ROUGE-2 F1-Score: 0.15544463843028272
ROUGE-3 Precision: 0.40908167199697654
ROUGE-3 Recall: 0.09120781524934578
ROUGE-3 F1-Score: 0.13658918415098817
ROUGE-4 Precision: 0.38719099551903197
ROUGE-4 Recall: 0.08402830009343216
ROUGE-4 F1-Score: 0.12651134217626184
Execution time: 0.12 seconds


data - 2
ROUGE-1 Precision: 0.6143849368814263
ROUGE-1 Recall: 0.20030154684570334
ROUGE-1 F1-Score: 0.28254913765302586
ROUGE-2 Precision: 0.3850733499301939
ROUGE-2 Recall: 0.10866824183683549
ROUGE-2 F1-Score: 0.1573630240373342
ROUGE-3 Precision: 0.31222290912193723
ROUGE-3 Recall: 0.08734208631272233
ROUGE-3 F1-Score: 0.12736460858066836
ROUGE-4 Precision: 0.27881416860577624
ROUGE-4 Recall: 0.07700953207038186
ROUGE-4 F1-Score: 0.11246089910052871
Execution time: 0.8 seconds
```

6 Rule based

[22]:

```python
print('Rule based')
# run loop
for file in data_files: # [0:1]
    # to get time complexity
    start_time = time.time()

    print(f"data - {data_files.index(file)}")
    # load data
    df = pd.read_csv(file)

    ROUGE_1_Precision, ROUGE_1_Recall, ROUGE_1_F1_Score = [], [], []
    ROUGE_2_Precision, ROUGE_2_Recall, ROUGE_2_F1_Score = [], [], []
    ROUGE_3_Precision, ROUGE_3_Recall, ROUGE_3_F1_Score = [], [], []
    ROUGE_4_Precision, ROUGE_4_Recall, ROUGE_4_F1_Score = [], [], []
```

```
for ind in range(len(df)):
    try:
        summary, article = df['summary'][ind], df['article'][ind]
        clean_sentences = get_clean_sentences(article)
        summary_pred = generate_summary_rule_based(clean_sentences, 0.4)

        # Tokenize the sentences
        reference_tokens = nltk.word_tokenize(summary)
        machine_tokens = nltk.word_tokenize(summary_pred)

        # Calculate ROUGE-1, ROUGE-2, ROUGE-3, and ROUGE-4 scores
        rouge_1 = rouge_n_score(machine_tokens, reference_tokens, 1)
        rouge_2 = rouge_n_score(machine_tokens, reference_tokens, 2)
        rouge_3 = rouge_n_score(machine_tokens, reference_tokens, 3)
        rouge_4 = rouge_n_score(machine_tokens, reference_tokens, 4)

        ROUGE_1_Precision.append(float(rouge_1[0]))
        ROUGE_1_Recall.append(float(rouge_1[1]))
        ROUGE_1_F1_Score.append(float(rouge_1[2]))

        ROUGE_2_Precision.append(float(rouge_2[0]))
        ROUGE_2_Recall.append(float(rouge_2[1]))
        ROUGE_2_F1_Score.append(float(rouge_2[2]))

        ROUGE_3_Precision.append(float(rouge_3[0]))
        ROUGE_3_Recall.append(float(rouge_3[1]))
        ROUGE_3_F1_Score.append(float(rouge_3[2]))

        ROUGE_4_Precision.append(float(rouge_4[0]))
        ROUGE_4_Recall.append(float(rouge_4[1]))
        ROUGE_4_F1_Score.append(float(rouge_4[2]))

    except Exception as e:
        pass

print("ROUGE-1 Precision:", np.mean(ROUGE_1_Precision))
print("ROUGE-1 Recall:", np.mean(ROUGE_1_Recall))
print("ROUGE-1 F1-Score:", np.mean(ROUGE_1_F1_Score))

print("ROUGE-2 Precision:", np.mean(ROUGE_2_Precision))
print("ROUGE-2 Recall:", np.mean(ROUGE_2_Recall))
print("ROUGE-2 F1-Score:", np.mean(ROUGE_2_F1_Score))

print("ROUGE-3 Precision:", np.mean(ROUGE_3_Precision))
print("ROUGE-3 Recall:", np.mean(ROUGE_3_Recall))
print("ROUGE-3 F1-Score:", np.mean(ROUGE_3_F1_Score))
```

```
    print("ROUGE-4 Precision:", np.mean(ROUGE_4_Precision))
    print("ROUGE-4 Recall:", np.mean(ROUGE_4_Recall))
    print("ROUGE-4 F1-Score:", np.mean(ROUGE_4_F1_Score))

    end_time = time.time()
    execution_time = end_time - start_time
    print(f"Execution time: {round(execution_time,2)} seconds")

    print('\n')
```

```
Rule based
data - 0
ROUGE-1 Precision: 0.20512820512820515
ROUGE-1 Recall: 0.023494860499265788
ROUGE-1 F1-Score: 0.042160737812911735
ROUGE-2 Precision: 0.1264367816091954
ROUGE-2 Recall: 0.009623797025371828
ROUGE-2 F1-Score: 0.017886178861788615
ROUGE-3 Precision: 0.08333333333333333
ROUGE-3 Recall: 0.00532724505327245
ROUGE-3 F1-Score: 0.010014306151645206
ROUGE-4 Precision: 0.07407407407407407
ROUGE-4 Recall: 0.004329004329004329
ROUGE-4 F1-Score: 0.0081799591002045
Execution time: 0.07 seconds


data - 1
ROUGE-1 Precision: 0.5407960808068502
ROUGE-1 Recall: 0.17533678256761728
ROUGE-1 F1-Score: 0.25238116575261227
ROUGE-2 Precision: 0.40401080976593945
ROUGE-2 Recall: 0.09007273368122928
ROUGE-2 F1-Score: 0.1417294663777582
ROUGE-3 Precision: 0.359272291229385
ROUGE-3 Recall: 0.07633528438979711
ROUGE-3 F1-Score: 0.1212588797194494
ROUGE-4 Precision: 0.34146030843055314
ROUGE-4 Recall: 0.07064248662109938
ROUGE-4 F1-Score: 0.11287910313580544
Execution time: 0.27 seconds


data - 2
ROUGE-1 Precision: 0.5931180738946922
ROUGE-1 Recall: 0.2113868625075436
ROUGE-1 F1-Score: 0.2982179762123182
```

```
ROUGE-2 Precision: 0.3851614318357719
ROUGE-2 Recall: 0.11184795818495705
ROUGE-2 F1-Score: 0.1646668061825807
ROUGE-3 Precision: 0.3147622680616108
ROUGE-3 Recall: 0.08922022587300336
ROUGE-3 F1-Score: 0.13193020171011954
ROUGE-4 Precision: 0.281150352079331
ROUGE-4 Recall: 0.0784699103860547
ROUGE-4 F1-Score: 0.11622325167569258
Execution time: 2.13 seconds
```

7 Bidirectional Encoder Representations from Transformers (BERT)

[25]:

```
print('Bidirectional Encoder Representations from Transformers (BERT)')
# run loop
for file in data_files: # [0:1]
    # to get time complexity
    start_time = time.time()

    print(f"data - {data_files.index(file)}")
    # load data
    df = pd.read_csv(file)

    ROUGE_1_Precision, ROUGE_1_Recall, ROUGE_1_F1_Score = [], [], []
    ROUGE_2_Precision, ROUGE_2_Recall, ROUGE_2_F1_Score = [], [], []
    ROUGE_3_Precision, ROUGE_3_Recall, ROUGE_3_F1_Score = [], [], []
    ROUGE_4_Precision, ROUGE_4_Recall, ROUGE_4_F1_Score = [], [], []

    for ind in range(len(df)):
        try:
            summary, article = df['summary'][ind], df['article'][ind]
            clean_sentences = get_clean_sentences(article)
            summary_pred = summarise_bart_summary_generation(article)

            # Tokenize the sentences
            reference_tokens = nltk.word_tokenize(summary)
            machine_tokens = nltk.word_tokenize(summary_pred)

            # Calculate ROUGE-1, ROUGE-2, ROUGE-3, and ROUGE-4 scores
            rouge_1 = rouge_n_score(machine_tokens, reference_tokens, 1)
            rouge_2 = rouge_n_score(machine_tokens, reference_tokens, 2)
            rouge_3 = rouge_n_score(machine_tokens, reference_tokens, 3)
            rouge_4 = rouge_n_score(machine_tokens, reference_tokens, 4)
```

```
            ROUGE_1_Precision.append(float(rouge_1[0]))
            ROUGE_1_Recall.append(float(rouge_1[1]))
            ROUGE_1_F1_Score.append(float(rouge_1[2]))

            ROUGE_2_Precision.append(float(rouge_2[0]))
            ROUGE_2_Recall.append(float(rouge_2[1]))
            ROUGE_2_F1_Score.append(float(rouge_2[2]))

            ROUGE_3_Precision.append(float(rouge_3[0]))
            ROUGE_3_Recall.append(float(rouge_3[1]))
            ROUGE_3_F1_Score.append(float(rouge_3[2]))

            ROUGE_4_Precision.append(float(rouge_4[0]))
            ROUGE_4_Recall.append(float(rouge_4[1]))
            ROUGE_4_F1_Score.append(float(rouge_4[2]))

        except Exception as e:
            pass

    print("ROUGE-1 Precision:", np.mean(ROUGE_1_Precision))
    print("ROUGE-1 Recall:", np.mean(ROUGE_1_Recall))
    print("ROUGE-1 F1-Score:", np.mean(ROUGE_1_F1_Score))

    print("ROUGE-2 Precision:", np.mean(ROUGE_2_Precision))
    print("ROUGE-2 Recall:", np.mean(ROUGE_2_Recall))
    print("ROUGE-2 F1-Score:", np.mean(ROUGE_2_F1_Score))

    print("ROUGE-3 Precision:", np.mean(ROUGE_3_Precision))
    print("ROUGE-3 Recall:", np.mean(ROUGE_3_Recall))
    print("ROUGE-3 F1-Score:", np.mean(ROUGE_3_F1_Score))

    print("ROUGE-4 Precision:", np.mean(ROUGE_4_Precision))
    print("ROUGE-4 Recall:", np.mean(ROUGE_4_Recall))
    print("ROUGE-4 F1-Score:", np.mean(ROUGE_4_F1_Score))

    end_time = time.time()
    execution_time = end_time - start_time
    print(f"Execution time: {round(execution_time,2)} seconds")

    print('\n')
```

```
Bidirectional Encoder Representations from Transformers (BERT)
data - 0
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
```

```
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
ROUGE-1 Precision: 0.013888888888888888
ROUGE-1 Recall: 0.01075268817204301
ROUGE-1 F1-Score: 0.012121212121212121
ROUGE-2 Precision: 0.0
ROUGE-2 Recall: 0.0
ROUGE-2 F1-Score: 0.0
ROUGE-3 Precision: 0.0
ROUGE-3 Recall: 0.0
ROUGE-3 F1-Score: 0.0
ROUGE-4 Precision: 0.0
ROUGE-4 Recall: 0.0
ROUGE-4 F1-Score: 0.0
Execution time: 173.76 seconds


data - 1
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
```

```
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
ROUGE-1 Precision: 0.10013090649786702
ROUGE-1 Recall: 0.08470396263717238
ROUGE-1 F1-Score: 0.08832626300237087
ROUGE-2 Precision: 0.007318411235461926
ROUGE-2 Recall: 0.003594771241830065
ROUGE-2 F1-Score: 0.004799045116901623
ROUGE-3 Precision: 0.0
ROUGE-3 Recall: 0.0
ROUGE-3 F1-Score: 0.0
ROUGE-4 Precision: 0.0
ROUGE-4 Recall: 0.0
ROUGE-4 F1-Score: 0.0
Execution time: 998.22 seconds


data - 2
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
```

```
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
```

```
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
```

```
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
```

```
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
```

```
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
```

```
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
```

```
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
```

```
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
Type of translation : <class 'googletrans.models.Translated'>
Type of translation.text : <class 'str'>
ROUGE-1 Precision: 0.10074765311593348
ROUGE-1 Recall: 0.08348651471239342
ROUGE-1 F1-Score: 0.08440392633057518
ROUGE-2 Precision: 0.009519092739249077
ROUGE-2 Recall: 0.004938698953024103
ROUGE-2 F1-Score: 0.006350246782380094
ROUGE-3 Precision: 0.0007686550260485913
ROUGE-3 Recall: 0.00034301181680416277
ROUGE-3 F1-Score: 0.00047149503733671O3
ROUGE-4 Precision: 0.0
ROUGE-4 Recall: 0.0
ROUGE-4 F1-Score: 0.0
Execution time: 8820.18 seconds
```

```
[ ]:
```