News Similarity Checking

2022/08/11

—

Assignment 1

—

News Content Similarity
W.P Pallewatta
Index No: - 18001149
Reg No:- 2018/CS/114

Table of Content

1) Introduction

What is document similarity Checking?

According to **MarketingProfs**, more than 2 million articles are published daily on the web. However, Online News websites have also disseminated editorial material that determines which articles to show on their homepages and which articles to promote, e.g., large font size for major news stories.

Text categorization and **text analytics** are essential applications of **Natural Language Processing**. This requires the development of a classifier. The trouble with text data, however, is that computers cannot directly comprehend natural language. Computers cannot simply accept text input and comprehend its context.

Many of the articles posted on a news website are quite similar to those provided on several other news websites. The selective reporting of prominent news headlines and the comparability of news across multiple news outlets are well-identified but seldom quantified.

Python makes **TF-IDF analysis** implementation conveniently. Computers can comprehend numbers but not the sense of a sentence. The link between the words and the numbers may be understood by converting the words to numbers.

This concept is used for the identifying the given **text files content** regrading about the news content. Target is to identify content similarity **when title is given**.

2) Methodologies

A) Term Frequency (TF)

The term is a quantifier for the occurrences of a given word **w** in some text **d**. As a percentage, it is equal to the number of times the word w appears in document d expressed as a percentage of the total number of words in the document. Term frequency is a measurement used to establish how often a certain word appears in each document relative to the total number of words. Consistency in the denominator is guaranteed.

$$Term\ Frequency = \frac{number\ of\ instances\ of\ word\ w\ in\ document\ d}{total\ number\ of\ words\ in\ document\ d}$$

Figure 2-1 Term Frequency

B) Inverse Document Frequency (IDF)

The significance of a word is quantified by this metric. Inverse In a text corpus D, the frequency of a word w is calculated as **N / (the number of documents containing w)**

$$IDF = log\ \left(\frac{total\ number\ of\ documents\ (N)\ in\ text\ corpus\ D}{number\ of\ documents\ containing\ w}\right)$$

Figure 2-2 Inverse Document Frequency Equation (IDF)

C) Bag-of-Words mechanism

A bag-of-words (BoW) model is a technique for extracting characteristics from text for use in modelling, such as using machine learning techniques.
The method is straightforward and versatile, and it may be used in a variety of ways to extract characteristics from texts.

A bag-of-words is a textual representation that represents the frequency of words inside a document. It includes two elements:

- A collection of recognized terms.
- A measurement of the frequency of recognized words.

D) Cosine Similarity

Even if two comparable texts are separated by a large Euclidean distance (due to the document's length), the cosine similarity increases the likelihood that they are still orientated in a way that is beneficial to the user. When comparing cosine similarity, a smaller angle is preferable.

E) RapidFuzz

RapidFuzz is an alternative **string-matching** library that does more than just compute string differences. C++ was mostly used to speed up the text matching process. There are three primary components:

- Fuzz Module
- String Metric Module
- Process Module

From all the available methodologies I am doing the **string matching using 3 main** functionalities. Which are

1) **TF-IDF Methodology (For identify words similarity based of the frequency)**
2) **Cosine Similarity**
3) **RapidFuzz (String comparison package that computes the differences between strings)**

From **these three analyzations** will help me to segregate the text data resides in each given files in order to identify the news topic in accurate way

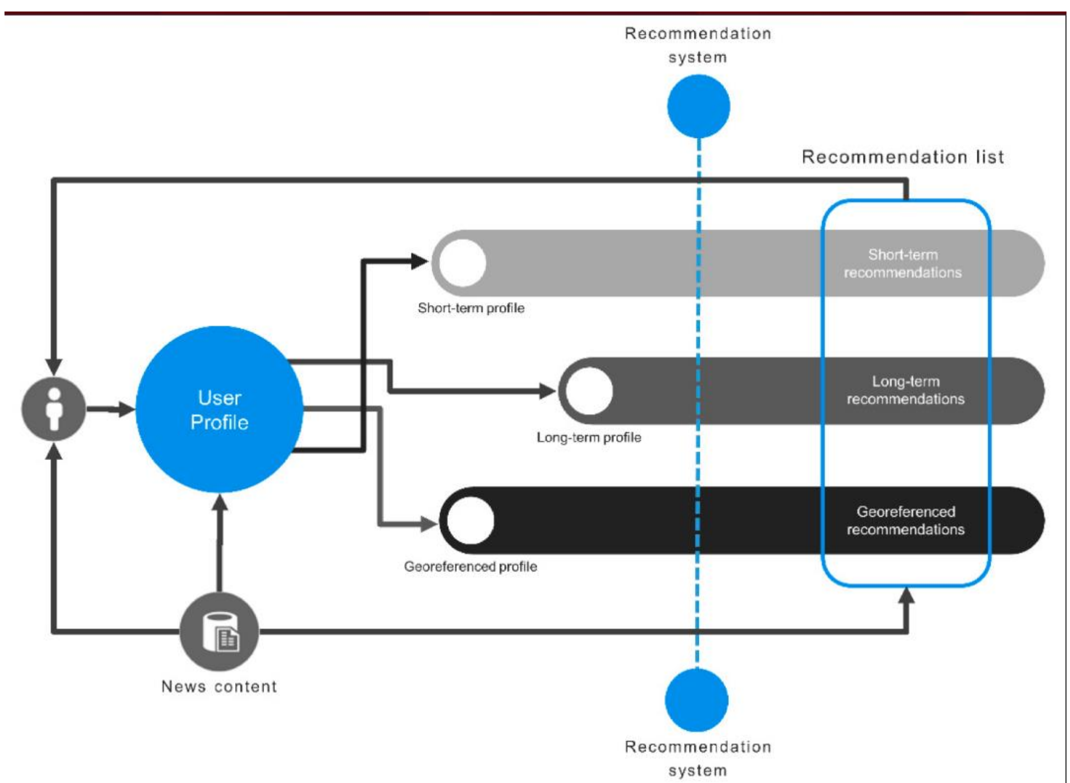

Figure 2-0-1 Component diagram for the News Recommendation System

3) Preprocessing

A) For preprocessing, **first text files read separately and stored into pandas' data frame**.

```
In [4]: files_path="D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files"
        read_files=glob.glob(os.path.join(files_path,"*.txt"))

In [5]: read_files

Out[5]: ['D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 1.txt',
         'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 2.txt',
         'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 3.txt',
         'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 4.txt',
         'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 5.txt',
         'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 6.txt',
         'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 7.txt',
         'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 8.txt']
```

Figure 3-1 Text files Reading

B) Importing necessary libraries for the **Headline Similarity Analyzation**

```
In [2]: # Below libraries are for text processing using NLTK
        from nltk.corpus import stopwords
        from nltk.tokenize import word_tokenize
        from nltk.stem import WordNetLemmatizer

        # Below libraries are for feature representation using sklearn
        from sklearn.feature_extraction.text import CountVectorizer
        from sklearn.feature_extraction.text import TfidfVectorizer

        # Below libraries are for similarity matrices using sklearn
        from sklearn.metrics.pairwise import cosine_similarity
        from sklearn.metrics import pairwise_distances

        from sklearn.metrics.pairwise import cosine_similarity, cosine_distances
```

Figure 3-2 Import NLTK Libraries

C) **Skleran Libraries for** similarity identification in texts

```
[3]: # Below libraries are for similarity matrices using sklearn
     from sklearn.metrics.pairwise import cosine_similarity

     from sklearn.metrics import pairwise_distances
     import copy
     from IPython.display import clear_output

     import warnings

     from re import sub
     import plotly
     import plotly.express as px
     import matplotlib.pyplot as plt
     import seaborn as sns
     from wordcloud import WordCloud
     plotly.offline.init_notebook_mode (connected = True)

     import random
```

Figure 3-3 Sklearn Similarity Libraries

sklearn cosine similarity :-
The cosine similarity module will be imported from the **sklearn.metrics.pairwise package**. Here will also import the **NumPy array** construction library.

D) Headline Storing in Pandas for Accessing

Headlines Storing in a Pandas dataset

```
In [112]: News_head1 = ["Hurricane Gilbert Heads Toward Dominican Coast","IRA terrorist attack","McDonald's Opens First Restaurant in China
          # News_head2 = ["IRA terrorist attack"]
          # News_head3= ["McDonald's Opens First Restaurant in China"]

In [113]: Head= pd.DataFrame(News_head1, columns=['Headline'])

In [114]: Head['Headline']=Head['Headline'].astype('string')
          Head.dtypes

Out[114]: Headline    string
          dtype: object
```

Figure 3-4 Headlines Storing

E) Using Vector Space Model for Implementing TD-IDF method

The **vector space model for text similarity** is rather simple: It produces a vector space in which each dimension corresponds to a single word. **Words are extracted** from all texts under consideration.

A single document is a vector in the vector space. Each dimension of a document vector reflects the frequency with which a certain word occurs in the text.

Proceeding to the TF-IDF transformation.

```
In [37]: #tfidf
         tf_idf_vector=tfidf_transformer.transform(word_count1)
         feature_names = count1.get_feature_names()

In [38]: first_document_vector=tf_idf_vector[1]
         df_tfifd= pd.DataFrame(first_document_vector.T.todense(), index=feature_names, columns=["TF-Idf"])

In [39]: df_tfifd.sort_values(by=["TF-Idf"],ascending=False).head(20)
Out[39]:
```

	TF-Idf
the	0.217551
registers	0.213023
rang	0.213023
breaking	0.213023
cash	0.213023
food	0.213023
27	0.213023
30	0.213023
meals	0.213023
largest	0.213023
landmark	0.213023
worldwide	0.213023

Figure 3-5 Vector Space Model for Text Similarity in Each text document for Words identification

Here are the preprocessing part of the text files reading and model building.
In TF-IDF model I have firstly stores all the text data in separate csv files and **concat into numpy** array for the building Vector model for Similarity Checking.
Example I have done when reading text file1:-

1)For the First Document

```
In [16]: new_data1.describe()
Out[16]:
```

	Information
count	25
unique	25
top	The previous opening-day record for sales was ...
freq	1

```
In [17]: new_data1
```

7	The official exchange rate is 1.59 dollar per ...
8	Half the day's sales were donated to the Sovie...
9	The restaurant, built by the company in a join...
10	The previous opening-day record for sales was ...
11	Soviets got a first-hand look at such alien co...
12	Accordions played folk songs and women in trad...
13	One Muscovite, accustomed to clerks who snarl ...
14	For most customers, it was their first experie...
15	They tried them one-handed.They picked their s...
16	"It tasted great!" a 14 years old boy said.
17	It's a lot different from a stolovaya," he co...
18	Under the sign of the golden arches, accented ...
19	Publicity conscious managers had the staff sho...
20	McDonald's of Canada Chairman George Cohon, th...

```
In [18]: info1=new_data1['Information'].to_numpy()

In [19]: info1_len=len(info1)
```

Figure 3-6 Document 1 Preprocess for Numpy Array

This mechanism continues for the **whole 8 text** files, and you can have a better understanding by going through the code. **(At the end of the Document)**

4) Results Evaluation

A) Document 1 (doc 1.txt)

```
In [38]: first_document_vector=tf_idf_vector[1]
         df_tfifd= pd.DataFrame(first_document_vector.T.todense(), index=feature_names, columns=["TF-Idf"])

In [39]: df_tfifd.sort_values(by=["TF-Idf"],ascending=False).head(20)
Out[39]:
```

	TF-Idf
the	0.217551
registers	0.213023
rang	0.213023
breaking	0.213023
cash	0.213023
food	0.213023
27	0.213023
30	0.213023
meals	0.213023
largest	0.213023
landmark	0.213023
worldwide	0.213023
world	0.213023
version	0.213023
record	0.188795
000	0.188795
american	0.188795
chain	0.188795
officials	0.188795
fast	0.188795

Figure 4-1 Doc 1 Tf-IDF result

```
In [36]: tfidf_transformer=TfidfTransformer(smooth_idf=True,use_idf=True)
         tfidf_transformer.fit(word_count1)
         df_idf = pd.DataFrame(tfidf_transformer.idf_, index=count1.get_feature_names(),columns=["IDF_Weights"])

         #inverse document frequency
         df_idf.sort_values(by=['IDF_Weights'])
Out[36]:
```

	IDF_Weights
the	1.213574
and	1.424883
to	1.619039
of	1.693147
for	1.693147
...	...
food	3.564949
folk	3.564949
flags	3.564949
gamburgers	3.564949
youthful	3.564949

Figure 4-2 Doc1 IDF-Weights Results

According the TF-IDF and other two analysis It's clear that this content belongs to

⇨ *McDonald's Opens First Restaurant in China*

Also the **RapidFuzz results are higher in "***McDonald's Opens First Restaurant in China"* **Compared to other two** Topics. (Explained in the Code Clarity)

B) Document 2 (doc 2.txt)

```
In [52]: #tfidf
tf_idf_vector2=tfidf_transformer2.transform(word_count2)
feature_names2 = count2.get_feature_names()

In [53]: second_document_vector=tf_idf_vector2[1]
df_tfifd2= pd.DataFrame(second_document_vector.T.todense(), index=feature_names2, columns=["TF-Idf"])

In [54]: df_tfifd2.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[54]:

	TF-Idf
no	0.445322
casualties	0.445322
immediate	0.445322
there	0.397848
reports	0.397848
were	0.235531
of	0.181726
000	0.000000
power	0.000000
preparation	0.000000
prensa	0.000000

Figure 4-3 Doc 2 Tf-IDF result

```
In [211]: tfidf_transformer2=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer2.fit(word_count2)
df_idf2 = pd.DataFrame(tfidf_transformer2.idf_, index=count2.get_feature_names(),columns=["IDF_Weights"])

#inverse document frequency
df_idf2.sort_values(by=['IDF_Weights'],ascending=False).head(20)
```

order	3.80336
only	3.80336
office	3.80336
next	3.80336
ocho	3.80336
ocean	3.80336
now	3.80336
northwest	3.80336
northward	3.80336
northeast	3.80336
normally	3.80336
packed	3.80336
packing	3.80336

Figure 4-4 Doc2 IDF-Weights Results

According the TF-IDF and other two analysis It's clear that this content belongs to

⇨ *Hurricane Gilbert Heads Toward Dominican Coast*

Also the **RapidFuzz results are higher in "***Hurricane Gilbert Heads Toward Dominican Coast"* **Compared to other two** Topics. (Explained in the Code Clarity)

C) Document 3 (doc 3.txt)

```
In [62]: #tfidf
tf_idf_vector3=tfidf_transformer3.transform(word_count3)
feature_names3 = count3.get_feature_names()
```

```
In [63]: third_document_vector=tf_idf_vector3[1]
df_tfifd3= pd.DataFrame(third_document_vector.T.todense(), index=feature_names3, columns=["TF-Idf"])
```

```
In [64]: df_tfifd3.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[64]:

	TF-Idf
mph	0.450933
gusting	0.287878
75	0.287878
sustained	0.287878
approaching	0.287878
92	0.287878
southeast	0.225467
from	0.225467
with	0.225467
the	0.225336
was	0.205375

Figure 4-5 Doc 3 Tf-IDF result

```
tfidf_transformer3=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer3.fit(word_count3)
df_idf3 = pd.DataFrame(tfidf_transformer3.idf_, index=count3.get_feature_names(),columns=["IDF_Weights"])
```

```
In [61]: tfidf_transformer3=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer3.fit(word_count3)
df_idf3 = pd.DataFrame(tfidf_transformer3.idf_, index=count3.get_feature_names(),columns=["IDF_Weights"])

#inverse document frequency
df_idf3.sort_values(by=['IDF_Weights'])
```

	IDF_Weights
the	1.251314
and	1.587787
of	1.693147
hurricane	1.944462
to	2.098612
...	...
happy	3.197225
had	3.197225
gusts	3.197225
gulf	3.197225
000	3.197225

167 rows × 1 columns

Figure 4-6 Doc3 IDF-Weights Results

According the TF-IDF and other two analysis It's clear that this content belongs to

⇨ *Hurricane Gilbert Heads Toward Dominican Coast*

Also the **RapidFuzz results are higher in "***Hurricane Gilbert Heads Toward Dominican Coast"* **Compared to other two** Topics. (Explained in the Code Clarity)

D) Document 4 (doc 4.txt)

```
In [73]: #tfidf
         tf_idf_vector4=tfidf_transformer4.transform(word_count4)
         feature_names4 = count4.get_feature_names()

In [74]: fourth_document_vector=tf_idf_vector4[1]
         df_tfifd4= pd.DataFrame(fourth_document_vector.T.todense(), index=feature_names4, columns=["TF-Idf"])

In [75]: df_tfifd4.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[75]:

	TF-Idf
claimed	0.502906
blast	0.452781
responsibility	0.452781
for	0.389631
ira	0.316967
the	0.291177
100	0.000000
plaster	0.000000
province	0.000000
provided	0.000000
prime	0.000000

Figure 4-7 Doc 4 Tf-IDF result

```
In [72]: tfidf_transformer4=TfidfTransformer(smooth_idf=True,use_idf=True)
         tfidf_transformer4.fit(word_count4)
         df_idf4 = pd.DataFrame(tfidf_transformer4.idf_, index=count4.get_feature_names(),columns=["IDF_Weights"])

         #inverse document frequency
         df_idf4.sort_values(by=['IDF_Weights'])
```

Out[72]:

	IDF_Weights
the	1.177681
and	1.870828
was	1.927987
of	1.927987
said	1.988611
...	...
frantically	4.068053
four	4.068053
force	4.068053
heavy	4.068053
killing	4.068053

387 rows × 1 columns

Figure 4-8 Doc4 IDF-Weights Results

According the TF-IDF and other two analysis It's clear that this content belongs to

⇨ *IRA terrorist attack*

Also the **RapidFuzz results are higher in "*IRA terrorist attack"* Compared to other two** Topics. (Explained in the Code Clarity)

E) Document 5 (doc 5.txt)

```
In [82]: #tfidf
         tf_idf_vector5=tfidf_transformer5.transform(word_count5)
         feature_names5 = count5.get_feature_names()

In [83]: five_document_vector=tf_idf_vector5[1]
         df_tfifd5= pd.DataFrame(five_document_vector.T.todense(), index=feature_names5, columns=["TF-Idf"])

In [84]: df_tfifd5.sort_values(by=["TF-Idf"],ascending=False).head(20)

Out[84]:
```

	TF-Idf
crash	0.390942
blitz	0.390942
terrific	0.390942
reminded	0.390942
which	0.348568
me	0.318504
there	0.276130
of	0.195327
was	0.187582
the	0.126987
resident	0.000000

Figure 4-9 Doc 5 Tf-IDF result

```
In [80]: # info2_len=len(info2)
         # info2
         tfidf_transformer5=TfidfTransformer(smooth_idf=True,use_idf=True)
         tfidf_transformer5.fit(word_count5)
         df_idf5 = pd.DataFrame(tfidf_transformer5.idf_, index=count5.get_feature_names(),columns=["IDF_Weights"])

In [81]: tfidf_transformer5=TfidfTransformer(smooth_idf=True,use_idf=True)
         tfidf_transformer5.fit(word_count5)
         df_idf5 = pd.DataFrame(tfidf_transformer5.idf_, index=count5.get_feature_names(),columns=["IDF_Weights"])

         #inverse document frequency
         df_idf5.sort_values(by=['IDF_Weights'])

Out[81]:
```

	IDF_Weights
the	1.215111
was	1.794930
of	1.869038
and	1.869038
said	2.131402
...	...
home	3.740840
homes	3.740840
horrible	3.740840
has	3.740840
years	3.740840

Figure 4-10 Doc5 IDF-Weights Results

According the TF-IDF and other two analysis It's clear that this content belongs to

⇨ *IRA terrorist attack*

Also the **RapidFuzz results are higher in "***IRA terrorist attack"* **Compared to other two** Topics. (Explained in the Code Clarity)

F) Document 6 (doc 6.txt)

```
In [91]: #tfidf
         tf_idf_vector6=tfidf_transformer6.transform(word_count6)
         feature_names6 = count6.get_feature_names()
```

```
In [92]: sixth_document_vector=tf_idf_vector6[1]
         df_tfifd6= pd.DataFrame(sixth_document_vector.T.todense(), index=feature_names6, columns=["TF-Idf"])
```

```
In [93]: sixth_document_vector

Out[93]: <1x230 sparse matrix of type '<class 'numpy.float64'>'
                 with 14 stored elements in Compressed Sparse Row format>
```

```
In [94]: df_tfifd6.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

feared	0.307714
rubble	0.307714
thirty	0.307714
up	0.307714
18	0.270486
missing	0.270486
trapped	0.244071
injured	0.244071
people	0.223583
to	0.151189
in	0.135850
the	0.116786
platts	0.000000

Figure 4-11 Doc 6 Tf-IDF result

```
In [90]: tfidf_transformer6=TfidfTransformer(smooth_idf=True,use_idf=True)
         tfidf_transformer6.fit(word_count6)
         df_idf6 = pd.DataFrame(tfidf_transformer6.idf_, index=count6.get_feature_names(),columns=["IDF_Weights"])

         #inverse document frequency
         df_idf6.sort_values(by=['IDF_Weights'],ascending=False).head(20)
```

no	3.351375
northern	3.351375
nothing	3.351375
occurred	3.351375
officials	3.351375
park	3.351375
idea	3.351375
part	3.351375
past	3.351375
platts	3.351375
playing	3.351375
port	3.351375
precautionary	3.351375

Figure 4-12 Doc6 IDF-Weights Results

According the TF-IDF and other two analysis It's clear that this content belongs to

⇨ *IRA terrorist attack*

Also the **RapidFuzz results are higher in "***IRA terrorist attack"* **Compared to other two** Topics. (Explained in the Code Clarity)

G) Document 7 (doc 7.txt)

In [102]:
```
seven_document_vector=tf_idf_vector7[1]
df_tfifd7= pd.DataFrame(seven_document_vector.T.todense(), index=feature_names7, columns=["TF-Idf"])
```

In [103]:
```
df_tfifd7.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[103]:

	TF-Idf
serious	0.248001
hit	0.248001
immediately	0.248001
750	0.248001
noon	0.248001
force	0.248001
by	0.248001
the	0.233869
000	0.221121
no	0.221121
full	0.221121
which	0.221121
around	0.221121
injuries	0.221121
city	0.202049
of	0.184231
people	0.175168

Figure 4-13 Doc 7 Tf-IDF result

In [98]:
```
tfidf_transformer7=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer7.fit(word_count7)
df_idf7 = pd.DataFrame(tfidf_transformer7.idf_, index=count7.get_feature_names(),columns=["IDF_Weights"])
```

In [99]:
```
tfidf_transformer7=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer7.fit(word_count7)
df_idf7 = pd.DataFrame(tfidf_transformer7.idf_, index=count7.get_feature_names(),columns=["IDF_Weights"])

#inverse document frequency
df_idf7.sort_values(by=['IDF_Weights'],ascending=False)
```

man	3.740840
neighboring	3.740840
noon	3.740840
north	3.740840
not	3.740840
...	...
to	1.794930
hurricane	1.794930
and	1.661398
of	1.389465
the	1.175891

323 rows × 1 columns

Figure 4-14 Doc7 IDF-Weights Results

According the TF-IDF and other two analysis It's clear that this content belongs to

⇨ *Hurricane Gilbert Heads Toward Dominican Coast*

Also the **RapidFuzz results are higher in "***Hurricane Gilbert Heads Toward Dominican Coast"* **Compared to other two** Topics. (Explained in the Code Clarity)

H) Document 8(doc 8.txt)

```
In [110]: eight_document_vector=tf_idf_vector8[1]
          df_tfifd8= pd.DataFrame(eight_document_vector.T.todense(), index=feature_names8, columns=["TF-Idf"])
```

```
In [111]: df_tfifd8.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[111]:

	TF-Idf
taste	0.444788
student	0.222394
three	0.222394
waited	0.222394
genuine	0.222394
school	0.222394
wanted	0.222394
nikolic	0.222394
to	0.208651
milica	0.196818
just	0.196818
hamburgers	0.196818
hours	0.196818
high	0.196818
her	0.196818
american	0.178672
big	0.164597
first	0.164597
for	0.164597
mac	0.164597

Figure 4-15 Doc 8 Tf-IDF result

```
In [108]: tfidf_transformer8=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer8.fit(word_count8)
df_idf8 = pd.DataFrame(tfidf_transformer8.idf_, index=count8.get_feature_names(),columns=["IDF_Weights"])

#inverse document frequency
df_idf8.sort_values(by=['IDF_Weights'],ascending=False).head(20)
```

Out[108]:

	IDF_Weights
110	3.525729
numerous	3.525729
market	3.525729
meals	3.525729
media	3.525729
milk	3.525729
million	3.525729
milosevic	3.525729
modern	3.525729
month	3.525729
more	3.525729
nicer	3.525729
nicest	3.525729
nikolic	3.525729
number	3.525729
official	3.525729
management	3.525729
onions	3.525729
only	3.525729
or	3.525729

Figure 4-16 Doc8 IDF-Weights Results

According the TF-IDF and other two analysis It's clear that this content belongs to

⇨ *McDonald's Opens First Restaurant in China*

Also the **RapidFuzz results are higher in "***McDonald's Opens First Restaurant in China"* **Compared to other two** Topics. (Explained in the Code Clarity)

For text similarity checking used **Rapid fuzzy** for predicting the content is relevant to the selected title and it's being proved and validate by **cosine-similarity checking.**

In my code all the explanation is given clearly and relevantly.

```
In [251]: News_df.dtypes
Out[251]: 000           float64
          10            float64
          100           float64
          11            float64
          110           float64
                          ...
          youthful      float64
          yugoslav      float64
          yugoslavia    float64
          yugoslavs     float64
          zone          float64
          Length: 1411, dtype: object
```

```
In [246]: News_df.head(20)
Out[246]:
```

	000	10	100	11	110	115	12	125	14	140	...	year	years	yet	you	young	yo
0	0.052621	0.060016	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.055325	0.000000	...	0.000000	0.035081	0.000000	0.023184	0.046367	0.0
1	0.015023	0.017134	0.034269	0.000000	0.059569	0.000000	0.000000	0.000000	0.000000	0.039713	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
2	0.061567	0.000000	0.035110	0.000000	0.000000	0.000000	0.048548	0.048548	0.000000	0.040687	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
3	0.000000	0.000000	0.017594	0.035188	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.015426	0.024329	0.000000	0.040778	0.0
4	0.000000	0.000000	0.000000	0.027831	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.024401	0.000000	0.000000	0.000000	0.0
5	0.000000	0.000000	0.000000	0.031240	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.043197	0.000000	0.000000	0.036202	0.000000	0.0
6	0.033987	0.019382	0.000000	0.000000	0.000000	0.053601	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.0
7	0.000000	0.000000	0.000000	0.000000	0.025177	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.019049	0.000000	0.000000	0.000000	0.0

8 rows × 1411 columns

Figure 4-17 Feature Name identification in all the documents

```
In [252]: # get cosine similarity matrix by using created dataframe
          print(cosine_similarity(News_df.values, News_df.values))

[[1.         0.54082435 0.42279101 0.56849572 0.50821448 0.48433927
  0.5472568  0.59938462]
 [0.54082435 1.         0.68250002 0.61184088 0.53099415 0.53330571
  0.84194483 0.50997942]
 [0.42279101 0.68250002 1.         0.4670546  0.40854181 0.41944037
  0.65355313 0.3833241 ]
 [0.56849572 0.61184088 0.4670546  1.         0.74806744 0.7313034
  0.61104358 0.54061195]
 [0.50821448 0.53099415 0.40854181 0.74806744 1.         0.55023635
  0.53310933 0.45222212]
 [0.48433927 0.53330571 0.41944037 0.7313034  0.55023635 1.
  0.52849343 0.46656382]
 [0.5472568  0.84194483 0.65355313 0.61104358 0.53310933 0.52849343
  1.         0.51410949]
 [0.59938462 0.50997942 0.3833241  0.54061195 0.45222212 0.46656382
  0.51410949 1.        ]]
```

Figure 4-18 Cosine Similarity of the Content respective to Article Titles

5) Final Predictions

List of .txt documents related to each news topic

Examining each algorithm and concept. I've been able to deduce the pertinent titles of the eight papers using specified approaches. Here is a table with the final results**.**

News Topics	Respective Documents
Hurricane Gilbert Heads Toward Dominican Coast	⇨ Doc 2.txt ⇨ Doc 3.txt ⇨ Doc 7.txt
McDonald's Opens First Restaurant in China	⇨ Doc 1.txt ⇨ Doc 8.txt
IRA terrorist attack	⇨ Doc 4.txt ⇨ Doc 5.txt ⇨ Doc 6.txt

Figure 5-1 News Title related Text files

Github Repo :-
https://github.com/Pandula1234/PythonDeepSource/tree/main/News%20Similarity%20Processing

6) References

[1] Brownlee, J. 2017. A Gentle Introduction to the Bag-of-Words Model - Machine Learning Mastery. Available at: https://machinelearningmastery.com/gentle-introduction-bag-words-model/ [Accessed: 02nd August 2022].

[2] Cosine Similarity – Understanding the math and how it works (with python codes). 2018. Available at: https://www.machinelearningplus.com/nlp/cosine-similarity/ [Accessed: 05th August 2022].

[3] Shah, P. 2021. All about RapidFuzz — String Similarity and Matching. Available at: https://medium.com/mlearning-ai/all-about-rapidfuzz-string-similarity-and-matching-cd26fdc963d8 [Accessed: 06th August 2022].

[4] sklearn.metrics.pairwise.cosine_similarity. 2000. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html [Accessed: 01st August 2022].

7) List of Figures

8) Full-code Implementation

(Down Below)

Headline based similarity on Articles

> **Index :- 18001149**
>
> I am Pandu, like **Data Analysis**

Generally, we assess **similarity** based on **distance**. If the **distance** is minimum then high **similarity** and if it is maximum then low **similarity**. To calculate the **distance**, we need to represent the headline as a **d-dimensional** vector. Then we can find out the **similarity** based on the **distance** between vectors.

There are multiple methods to represent a **text** as **d-dimensional** vector like **Bag of words**, **TF-IDF method**, **Word2Vec embedding** etc. Each method has its own advantages and disadvantages.

Let's see the feature representation of headline through all the methods one by one.

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

In [2]:
```
# Below libraries are for text processing using NLTK
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

# Below libraries are for feature representation using sklearn
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

# Below libraries are for similarity matrices using sklearn
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics import pairwise_distances

from sklearn.metrics.pairwise import cosine_similarity, cosine_distances
```

In [3]:
```
# Below libraries are for similarity matrices using sklearn
from sklearn.metrics.pairwise import cosine_similarity

from sklearn.metrics import pairwise_distances
import copy
from IPython.display import clear_output

import warnings

from re import sub
import plotly
import plotly.express as px
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
plotly.offline.init_notebook_mode (connected = True)

import random
warnings.filterwarnings("ignore")
```

In [10]:
```
files_path="D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files"
read_files=glob.glob(os.path.join(files_path,"*.txt"))
```

In [7]:
```
read_files
```

Out[7]:
```
['D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 1.txt',
 'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 2.txt',
 'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 3.txt',
 'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 4.txt',
 'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 5.txt',
 'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 6.txt',
 'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 7.txt',
 'D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 8.txt']
```

In [11]:
```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.decomposition import TruncatedSVD
from sklearn.pipeline import Pipeline, make_pipeline

# Below libraries are for feature representation using sklearn
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [12]:
```
# myfile = open("D:\\DC Universe\\Ucsc\\Fourth Year\\SCS4204 Data Analytics\\Assignments\\News Files\\doc 1.txt", "r")
# myline = myfile.readline()
# print(myline)
```

In [13]:
```
#Data Extracted into csv files for further analyzation

new_data1=pd.read_csv(read_files[0],error_bad_lines=False,header=None,delimiter = ' \t ')

new_data2=pd.read_csv(read_files[1],error_bad_lines=False,header=None,delimiter = ' \t ')

new_data3=pd.read_csv(read_files[2],error_bad_lines=False,header=None,delimiter = ' \t ')

new_data4=pd.read_csv(read_files[3],error_bad_lines=False,header=None,delimiter = ' \t ')

new_data5=pd.read_csv(read_files[4],error_bad_lines=False,header=None,delimiter = ' \t ')

new_data6=pd.read_csv(read_files[5],error_bad_lines=False,header=None,delimiter = ' \t ')

new_data7=pd.read_csv(read_files[6],error_bad_lines=False,header=None,delimiter = ' \t ')

new_data8=pd.read_csv(read_files[7],error_bad_lines=False,header=None,delimiter = ' \t ')
```

In [14]:
```
new_data1.set_axis(["Information"],axis=1,inplace=True)
```

In [15]:
```
# new_data1.dtypes
new_data1["Information"]=new_data1["Information"].astype('string')
```

Headlines Storing in a Pandas dataset

In [112…
```
News_head1 = ["Hurricane Gilbert Heads Toward Dominican Coast","IRA terrorist attack","McDonald's Opens First Restaurant in China"
# News_head2 = ["IRA terrorist attack"]
# News_head3= ["McDonald's Opens First Restaurant in China"]
```

In [113…
```
Head= pd.DataFrame(News_head1, columns=['Headline'])
```

In [114…
```
Head['Headline']=Head['Headline'].astype('string')
Head.dtypes
```

Out[114…
```
Headline    string
dtype: object
```

> **Stopword tokenization usinh NLTK library**

In [212…
```
# This function is to remove stopwords from a particular column and to tokenize it
def Stopword_tokenize(data,name):

    def getting(sen):
        example_sent = sen

        stop_words = set(stopwords.words('english'))

        word_tokens = word_tokenize(example_sent)

        filtered_sentence = [w for w in word_tokens if not w in stop_words]

        filtered_sentence = []

        for w in word_tokens:
            if w not in stop_words:
                filtered_sentence.append(w)
        return filtered_sentence
    x=[]
    for i in data[name].values:
        x.append(getting(i))
    data[name]=x
```

A)Using TF-IDF method

TF-IDF method is a weighted measure which gives more importance to less frequent words in a corpus. It assigns a weight to each term(word) in a document based on **Term frequency(TF)** and **inverse document frequency(IDF)**.

TF(i,j) = (# times word i appears in document j) / (# words in document j)

IDF(i,D) = log_e(#documents in the corpus D) / (#documents containing word i)

weight(i,j) = **TF(i,j)** x **IDF(i,D)**

So if a word occurs more number of times in a document but less number of times in all other documents then its **TF-IDF** value will be high.

1)For the First Document

In [16]:
```
new_data1.describe()
```

Out[16]:

	Information
count	25
unique	25
top	The previous opening-day record for sales was ...
freq	1

In [17]:
```
new_data1
```

Out[17]:

	Information
0	Thousands of queue-hardened Soviets on Wednesd...
1	The world's largest version of the landmark Am...
2	The Soviets, bundled in fur coats and hats, se...
3	The crush of customers was so intense the comp...
4	I only waited an hour and I think they served ...
5	And it was only 10 rubles for all this, she sa...
6	Big Macs were priced at 3.75 rubles and double...
7	The official exchange rate is 1.59 dollar per ...
8	Half the day's sales were donated to the Sovie...
9	The restaurant, built by the company in a join...
10	The previous opening-day record for sales was ...
11	Soviets got a first-hand look at such alien co...
12	Accordions played folk songs and women in trad...
13	One Muscovite, accustomed to clerks who snarl ...
14	For most customers, it was their first experie...
15	They tried them one-handed.They picked their s...
16	''It tasted great!'' a 14 years old boy said.
17	It's a lot different from a stolovaya,'' he co...
18	Under the sign of the golden arches, accented ...
19	Publicity conscious managers had the staff sho...
20	McDonald's of Canada Chairman George Cohon, th...
21	The restaurant limited purchases to 10 Big Mac...
22	McDonald's built its own factory, including ba...
23	One McDonald's associate said the company woun...
24	They found you need a permit to buy nails.

In [18]:
```
info1=new_data1['Information'].to_numpy()
```

In [19]:
```
info1_len=len(info1)
```

In [20]:
```
info1
```

Out[20]:
```
array(["Thousands of queue-hardened Soviets on Wednesday cheerfully lined up to get a taste of ''gamburgers'', ''chizburgers'' and ''Filay-o-feesh'' sandwiches as McDonald's opened in the land of Lenin for the first time.",
       "The world's largest version of the landmark American fast-food chain rang up 30,000 meals on 27 cash registers, breaking the opening-day record for McDonald's worldwide, officials said.",
       'The Soviets, bundled in fur coats and hats, seemed unfazed, lining up before dawn outside the 700 seat restaurant, the first of 20 planned across the Soviet Union.',
       'The crush of customers was so intense the company stayed open until midnight, two hours later than planned.',
       'I only waited an hour and I think they served thousands before me, said a happy middle-aged woman who works at an aluminum plant.',
       "And it was only 10 rubles for all this, she said. I'm taking it back for the girls at the factory to try.",
       "Big Macs were priced at 3.75 rubles and double cheeseburgers at 3 rubles about two hours' pay for a starting McDonald's staffer or the average Soviet, but much cheaper than other private restaurants that have sprung up recently.",
       'The official exchange rate is 1.59 dollar per ruble but foreign visitors can buy rubles for 16 cents each, about what the currency is worth on the black market.',
       "Half the day's sales were donated to the Soviet Children's Fund, which provides medical care and assistance to orphans and disadvantaged children, Gary Reinblatt, senior vice president of McDonald's Canada, said from Toronto.",
```

```
       "The restaurant, built by the company in a joint venture with the city of Moscow that began 14 years ago, brought to 52 the
number of countries where McDonald's operates.",
       "The previous opening-day record for sales was in Budapest, company officials said. Besides its restaurants in the United S
tates, the leading number of McDonald's are in Canada and Japan, the officials said.",
       'Soviets got a first-hand look at such alien concepts as efficiency and fast, friendly service. Normally dour citizens brok
e into grins as they caught the infectious cheerful mood from youthful Soviet staffers hired for their ability to smile and work h
ard.',
       'Accordions played folk songs and women in traditional costumes danced with cartoon characters, including Mickey Mouse and
Baba Yaga, a witch of Russian fairy tales.',
       'One Muscovite, accustomed to clerks who snarl if they say anything at all, asked for a straw and was startled when a smili
ng young Soviet woman found him one and popped it straight into his drink.',
       "For most customers, it was their first experience with a hamburger. Sandwiches were served in the familiar bag marked with
the golden arches, but were packed in wrappers bearing Cyrillic letters, approximating ``gamburger.''",
       "They tried them one-handed.They picked their sandwiches apart to examine the contents. One young woman finally squashed he
r ``Beeg Mak'' to fit her lips around it.",
       "''It tasted great!'' a 14 years old boy said.",
       "It's a lot different from a stolovaya,'' he continued with a smile, referring to the much cheaper but run down dirty cafet
erias that slop rice and fat or boiled sausage.",
       "Under the sign of the golden arches, accented by the Soviet hammer and sickle flag, hundreds lined up for the long awaited
grand opening at 10 am on Pushkin Square, reaching out excitedly for McDonald's flags and pins as the hamburger chain's army fulfi
lled the Soviet penchant for souvenirs with Western logos.",
       "Publicity conscious managers had the staff shout ''Good morning, America!'' in English and Russian, for an American TV net
work.",
       "McDonald's of Canada Chairman George Cohon, the man behind the deal, said many people were buying multiple orders and the
restaurant served 15,000 to 20,000 people in just the first five hours of operation.",
       'The restaurant limited purchases to 10 Big Macs per customer in hopes of preventing burger scalping.',
       "McDonald's built its own factory, including bakery, dairy, meat processing plant and even potato storage yard, to provide
its own guaranteed supplies in a country where up to 25 percent of the harvest rots en route to the consumer.",
       "One McDonald's associate said the company wound up importing wooden crates from Finland for storing potatoes because when
they went to build crates, they found there was no wood, and no nails.",
       'They found you need a permit to buy nails.'], dtype=object)
```

In [32]:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [27]:

```
count1 = CountVectorizer()
word_count1=count1.fit_transform(info1)
print(word_count1)
```

```
  (0, 362)	1
  (0, 245)	3
  (0, 290)	1
  (0, 175)	1
  (0, 331)	1
  (0, 249)	1
  (0, 383)	1
  (0, 83)	1
  (0, 209)	1
  (0, 376)	1
  (0, 364)	1
  (0, 158)	1
  (0, 351)	1
  (0, 155)	1
  (0, 86)	1
  (0, 28)	1
  (0, 136)	1
  (0, 135)	1
  (0, 310)	1
  (0, 36)	1
  (0, 223)	1
  (0, 253)	1
  (0, 190)	1
  (0, 355)	2
  (0, 201)	1
  :	:
  (23, 388)	1
  (23, 148)	1
  (23, 39)	1
  (23, 403)	1
  (23, 189)	1
  (23, 397)	1
  (23, 101)	2
  (23, 138)	1
  (23, 344)	1
  (23, 278)	1
  (23, 48)	1
  (23, 384)	1
  (23, 62)	1
  (23, 358)	1
  (23, 242)	2
  (23, 396)	1
  (23, 239)	1
  (24, 364)	1
  (24, 359)	1
  (24, 67)	1
  (24, 148)	1
  (24, 239)	1
  (24, 408)	1
  (24, 240)	1
  (24, 270)	1
```

In [28]:

```
word_count1.shape
```

Out[28]: (25, 411)

In [29]:
```
print(word_count1.toarray())
```

```
[[0 0 0 ... 0 0 0]
 [1 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 1 0 0]]
```

In [30]:
```
tfidf_transformer=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer.fit(word_count1)
df_idf = pd.DataFrame(tfidf_transformer.idf_, index=count1.get_feature_names(),columns=["IDF_Weights"])
```

In [36]:
```
tfidf_transformer=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer.fit(word_count1)
df_idf = pd.DataFrame(tfidf_transformer.idf_, index=count1.get_feature_names(),columns=["IDF_Weights"])

#inverse document frequency
df_idf.sort_values(by=['IDF_Weights'])
```

Out[36]:

	IDF_Weights
the	1.213574
and	1.424883
to	1.619039
of	1.693147
for	1.693147
...	...
food	3.564949
folk	3.564949
flags	3.564949
gamburgers	3.564949
youthful	3.564949

411 rows × 1 columns

Proceeding to the TF-IDF transformation.

In [37]:
```
#tfidf
tf_idf_vector=tfidf_transformer.transform(word_count1)
feature_names = count1.get_feature_names()
```

In [38]:
```
first_document_vector=tf_idf_vector[1]
df_tfifd= pd.DataFrame(first_document_vector.T.todense(), index=feature_names, columns=["TF-Idf"])
```

In [39]:
```
df_tfifd.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[39]:

	TF-Idf
the	0.217551
registers	0.213023
rang	0.213023
breaking	0.213023
cash	0.213023
food	0.213023
27	0.213023
30	0.213023
meals	0.213023
largest	0.213023
landmark	0.213023
worldwide	0.213023
world	0.213023
version	0.213023
record	0.188795

	TF-Idf
000	0.188795
american	0.188795
chain	0.188795
officials	0.188795
fast	0.188795

2)For the Second Document

In [40]:
```
new_data2.set_axis(["Information"],axis=1,inplace=True)
# new_data1.dtypes
new_data2["Information"]=new_data2["Information"].astype('string')
```

In [43]:
```
info2=new_data2['Information'].to_numpy()
info2
```

Out[43]:
```
array(['Hurricane Gilbert, packing 110 mph winds and torrential rain, moved over this capital city today after skirting Puerto Ric
o, Haiti and the Dominican Republic.',
       'There were no immediate reports of casualties.',
       'Telephone communications were affected.',
       "Right now it's actually moving over Jamaica,said Bob Sheets, director of the National Hurricane Center in Miami.",
       "We've already had reports of 110 mph winds on the eastern tip.",
       "It looks like the eye is going to move lengthwise across that island, and they're going to bear the full brunt of this pow
erful hurricane, Sheets said.",
       'Forecasters say Gilbert was expected to lash Jamaica throughout the day and was on track to later strike the Cayman Island
s, a small British dependency northwest of Jamaica.',
       "Meanwhile, Havana Radio reported today that 25,000 people were evacuated from Guantanamo Province on Cuba's southeastern c
oast as strong winds fanning out from Gilbert began brushing the island.",
       'All Jamaica-bound flights were canceled at Miami International Airport, while flights from Grand Cayman, the main island o
f the three-island chain, arrived packed with frightened travelers.',
       'People were running around in the main lobby of our hotel (on Grand Cayman) like chickens with their heads cut off, said o
ne vacationer who was returning home to California through Miami.',
       'Hurricane warnings were posted for the Cayman Islands, Cuba and Haiti.',
       'Warnings were discontinued for the Dominican Republic.',
       'All interests in the Western Caribbean should continue to monitor the progress of this dangerous hurricane, the service sa
id, adding, Little change in strength is expected for the next several hours as the hurricane moves westward over Jamaica.',
       'The Associated Press Caribbean headquarters in San Juan, Puerto Rico, was unable to get phone calls through to Kingston, w
here high winds and heavy rain preceding the storm drenched the capital overnight, toppling trees, causing local flooding and litt
ering streets with branches.',
       'Most Jamaicans stayed home, boarding up windows in preparation for the hurricane.',
       'Some companies broadcast appeals for technicians and electricians to report to work.',
       "The weather bureau predicted Gilbert's center, 140 miles southeast of Kingston before dawn, would pass south of Kingston a
nd hit the southern parish of Clarendon.",
       'Flash flood warnings were issued for the parishes of Portland on the northeast and St. Mary on the north.',
       'The north coast tourist region from Montego Bay on the west and Ocho Rios on the east, far from the southern impact zone a
nd separated by mountains, was expected only to receive heavy rain.',
       'Officials urged residents in the higher risk areas along the south coast to seek higher ground.',
       "It's certainly one of the larger systems we've seen in the Caribbean for a long time, said Hal Gerrish, forecaster at the
National Hurricane Center.",
       'Forecasters at the center said the eye of Gilbert was 140 miles southeast of Kingston at dawn today.',
       'Maximum sustained winds were near 110 mph, with tropical-storm force winds extending up to 250 miles to the north and 100
miles to the south.',
       'Prime Minister Edward Seaga of Jamaica alerted all government agencies, saying Sunday night: Hurricane Gilbert appears to
be a real threat and everyone should follow the instructions and hurricane precautions issued by the Office of Disaster Preparedne
ss in order to minimize the danger.',
       'Forecasters said the hurricane had been gaining strength as it passed over the ocean after it dumped 5 to 10 inches of rai
n on the Dominican Republic and Haiti, which share the island of Hispaniola.',
       "We should know within about 72 hours whether it's going to be a major threat to the United States,'' said Martin Nelson, a
nother meteorologist at the center.",
       "It's moving at about 17 mph to the west and normally hurricanes take a northward turn after they pass central Cuba.",
       "Cuba's official Prensa Latina news agency said a state of alert was declared at midday in the Cuban provinces of Guantanam
o, Holguin, Santiago de Cuba and Granma.",
       'In the report from Havana received in Mexico City, Prensa Latina said civil defense officials were broadcasting bulletins
on national radio and television recommending emergency measures and providing information on the storm.',
       "Heavy rain and stiff winds downed power lines and caused flooding in the Dominican Republic on Sunday night as the hurrica
ne's center passed just south of the Barahona peninsula, then less than 100 miles from neighboring Haiti.",
       "The storm ripped the roofs off houses and flooded coastal areas of southwestern Puerto Rico after reaching hurricane stren
gth off the island's southeast Saturday night.",
       'Flights were canceled Sunday in the Dominican Republic, where civil defense director Eugenio Cabral reported some flooding
in parts of the capital of Santo Domingo and power outages there and in other southern areas.'],
      dtype=object)
```

In [44]:
```
count2 = CountVectorizer()
word_count2=count2.fit_transform(info2)
print(word_count2)
```

```
  (0, 147)	1
  (0, 124)	1
  (0, 237)	1
  (0, 3)	1
  (0, 204)	1
  (0, 370)	1
  (0, 23)	2
  (0, 343)	1
  (0, 264)	1
  (0, 201)	1
  (0, 234)	1
  (0, 333)	1
  (0, 56)	1
```

```
  (0, 68)       1
  (0, 341)      1
  (0, 14)       1
  (0, 301)      1
  (0, 262)      1
  (0, 278)      1
  (0, 132)      1
  (0, 328)      1
  (0, 91)       1
  (0, 275)      1
  (1, 331)      1
  (1, 360)      1
  :     :
  (31, 275)     1
  (31, 331)     1
  (31, 360)     1
  (31, 221)     2
  (31, 87)      1
  (31, 151)     3
  (31, 273)     1
  (31, 109)     1
  (31, 55)      1
  (31, 364)     1
  (31, 112)     1
  (31, 303)     1
  (31, 307)     1
  (31, 27)      1
  (31, 319)     1
  (31, 69)      1
  (31, 85)      1
  (31, 248)     1
  (31, 100)     1
  (31, 52)      1
  (31, 240)     1
  (31, 288)     1
  (31, 90)      1
  (31, 233)     1
  (31, 230)     1
```

In [45]:

```
word_count2.shape
print(word_count2.toarray())
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 1 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

In [46]:

```
# info2_len=len(info2)
# info2
tfidf_transformer2=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer2.fit(word_count2)
df_idf2 = pd.DataFrame(tfidf_transformer2.idf_, index=count2.get_feature_names(),columns=["IDF_Weights"])
```

In [211…

```
tfidf_transformer2=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer2.fit(word_count2)
df_idf2 = pd.DataFrame(tfidf_transformer2.idf_, index=count2.get_feature_names(),columns=["IDF_Weights"])

#inverse document frequency
df_idf2.sort_values(by=['IDF_Weights'],ascending=False).head(20)
```

Out[211…

	IDF_Weights
000	3.80336
official	3.80336
overnight	3.80336
outages	3.80336
out	3.80336
our	3.80336
other	3.80336
order	3.80336
only	3.80336
office	3.80336
next	3.80336
ocho	3.80336
ocean	3.80336
now	3.80336
northwest	3.80336
northward	3.80336

	IDF_Weights
northeast	3.80336
normally	3.80336
packed	3.80336
packing	3.80336

In [52]:
```
#tfidf
tf_idf_vector2=tfidf_transformer2.transform(word_count2)
feature_names2 = count2.get_feature_names()
```

In [53]:
```
second_document_vector=tf_idf_vector2[1]
df_tfifd2= pd.DataFrame(second_document_vector.T.todense(), index=feature_names2, columns=["TF-Idf"])
```

In [54]:
```
df_tfifd2.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[54]:

	TF-Idf
no	0.445322
casualties	0.445322
immediate	0.445322
there	0.397848
reports	0.397848
were	0.235531
of	0.181726
000	0.000000
power	0.000000
preparation	0.000000
prensa	0.000000
predicted	0.000000
preceding	0.000000
precautions	0.000000
powerful	0.000000
portland	0.000000
posted	0.000000
press	0.000000
phone	0.000000
people	0.000000

3)For the Third Document

In [55]:
```
new_data3.set_axis(["Information"],axis=1,inplace=True)
# new_data1.dtypes
new_data3["Information"]=new_data3["Information"].astype('string')
```

In [56]:
```
info3=new_data3['Information'].to_numpy()
info3
```

Out[56]:
```
array(['Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south co
ast to prepare for high winds, heavy rains and high seas.',
       'The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.',
       'There is no need for alarm, Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Satur
day.',
       "Cabral said residents of the province of Barahona should closely follow Gilbert's movement.",
       'An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo
Domingo.',
       'Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night.',
       'The National Hurricane Center in Miami reported its position at 2 a.m.',
       'Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast o
f Santo Domingo.',
       'The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a broad area of clo
udiness and heavy weather rotating around the center of the storm.',
       'The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.',
       "Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet feet to Puerto
Rico's south coast.",
       'There were no reports of casualties.',
       'San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.',
       'On Saturday, Hurricane Florence was downgraded to a tropical storm and its remnants pushed inland from the Gulf Coast.',
       'Residents returned home, happy to find little damage from 80 mph winds and sheets of rain.',
```

```
 'Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.',
 'The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.'],
dtype=object)
```

In [57]:

```
count3 = CountVectorizer()
word_count3=count3.fit_transform(info3)
print(word_count3)
```

```
  (0, 75)	1
  (0, 64)	1
  (0, 148)	1
  (0, 154)	1
  (0, 150)	2
  (0, 47)	1
  (0, 122)	1
  (0, 146)	1
  (0, 19)	2
  (0, 37)	1
  (0, 44)	1
  (0, 17)	1
  (0, 83)	1
  (0, 70)	1
  (0, 110)	1
  (0, 139)	1
  (0, 40)	1
  (0, 153)	1
  (0, 112)	1
  (0, 61)	1
  (0, 72)	2
  (0, 165)	1
  (0, 71)	1
  (0, 117)	1
  (0, 131)	1
  :	:
  (15, 150)	3
  (15, 141)	2
  (15, 159)	1
  (15, 106)	1
  (15, 59)	1
  (15, 138)	1
  (15, 100)	1
  (15, 7)	1
  (15, 25)	1
  (15, 132)	1
  (15, 133)	1
  (16, 75)	1
  (16, 150)	2
  (16, 40)	1
  (16, 27)	1
  (16, 55)	1
  (16, 43)	1
  (16, 118)	1
  (16, 95)	1
  (16, 142)	1
  (16, 28)	1
  (16, 73)	1
  (16, 91)	1
  (16, 85)	1
  (16, 96)	1
```

In [59]:

```
word_count3.shape
print(word_count3.toarray())
```

```
[[0 0 0 ... 0 1 0]
 [0 0 0 ... 0 1 1]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

In [60]:

```
# info2_len=len(info2)
# info2
tfidf_transformer3=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer3.fit(word_count3)
df_idf3 = pd.DataFrame(tfidf_transformer3.idf_, index=count3.get_feature_names(),columns=["IDF_Weights"])
```

In [61]:

```
tfidf_transformer3=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer3.fit(word_count3)
df_idf3 = pd.DataFrame(tfidf_transformer3.idf_, index=count3.get_feature_names(),columns=["IDF_Weights"])

#inverse document frequency
df_idf3.sort_values(by=['IDF_Weights'])
```

Out[61]:

	IDF_Weights
the	1.251314
and	1.587787
of	1.693147
hurricane	1.944462
to	2.098612

	IDF_Weights
...	...
happy	3.197225
had	3.197225
gusts	3.197225
gulf	3.197225
000	3.197225

167 rows × 1 columns

In [62]:
```
#tfidf
tf_idf_vector3=tfidf_transformer3.transform(word_count3)
feature_names3 = count3.get_feature_names()
```

In [63]:
```
third_document_vector=tf_idf_vector3[1]
df_tfifd3= pd.DataFrame(third_document_vector.T.todense(), index=feature_names3, columns=["TF-Idf"])
```

In [64]:
```
df_tfifd3.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[64]:

	TF-Idf
mph	0.450933
gusting	0.287878
75	0.287878
sustained	0.287878
approaching	0.287878
92	0.287878
southeast	0.225467
from	0.225467
with	0.225467
the	0.225336
was	0.205375
winds	0.205375
storm	0.188959
to	0.188959
of	0.152451
ponce	0.000000
on	0.000000
reported	0.000000
remnants	0.000000
people	0.000000

4)For the Fourth Document

In [65]:
```
new_data4.set_axis(["Information"],axis=1,inplace=True)
# new_data1.dtypes
new_data4["Information"]=new_data4["Information"].astype('string')
```

In [66]:
```
new_data4.shape
```

Out[66]:
```
(42, 1)
```

In [67]:
```
info4=new_data4['Information'].to_numpy()
info4
```

Out[67]:
```
array(['An explosion today flattened a military barracks and tore through nearby homes, killing 11 people and injuring 22, police said.',
       'The IRA claimed responsibility for the blast.',
       'More than 100 rescue workers frantically dug through the rubble of a three-story building that collapsed at the Royal Marines School of Music near Deal.',
       'Stunned neighbors gathered outside homes that were damaged or destroyed.',
       'Chief Police Inspector Alan Butterfield of Kent, who who provided the casualty figures and coordinated the rescue effort, first reported that one person was missing but later said everyone was accounted for.',
       'He said many of the injured were seriously hurt.',
```

```
       'There was a terrific crash which reminded me of the Blitz.',
       'After that, the ceiling started to fall down around me, said pensioner Joan Betteridge.',
       'Defense Secretary Tom King, inspecting the wreckage, said, It is not yet absolutely confirmed that it is a bomb, but all t
he evidence is quite clearly that this is an IRA atrocity.',
       "British military installations are a frequent bombing target of the Irish Republican Army in its campaign to rid Northern
Ireland of British rule, but today's explosion in the coastal town 70 miles southeast of London was the worst IRA attack on the Br
itish mainland in more than seven years.",
       'The explosion occurred at at 8:26 a.m. in a lounge in thebarracks.',
       'One of the bands had just stopped playing on the parade ground, said a ministry spokesman, speaking anonymously in keeping
with British custom.',
       'Dozens of homes near the school were damaged, including four that were destroyed. Witnesses reported hearing the explosion
two miles away.',
       'The Defense Ministry would not say how many servicemen and civilians were included in the casualty figures.',
       'However, King told reporters the attack was directed against unarmed bandsmen.',
       'Firefighters used heavy lifting equipment and thermal cameras to search through the debris, said Kent Fire Brigade spokesm
an Kevin Simmons.',
       'Ten doctors were giving emergency treatment at the scene and 11 ambulances were taking the injured to two hospitals, the a
mbulance service said.',
       "A statement telephoned to Ireland International, a Dublin news agency, said, we have visited the Royal Marines in Kent in
response to Prime Minister Margaret Thatcher's visit to Northern Ireland nine days ago.",
       'The IRA said Mrs Thatcher went to the British province with a message of war,but we still want peace and we want the Briti
sh government to leave our country.',
       "It was signed P. O'Neill, a nom de guerre the IRA usually uses to claim responsibility for actions outside Northern Irelan
d.",
       'Irish Prime Minister Charles Haughey issued a statement in Dublin condemning the attack, calling it an outrage.',
       'The last IRA bomb attempt on the British mainland was in February when about 60 soldiers were evacuated from their barrack
s in Shropshire, western England, just before a bomb exploded.',
       'One soldier was killed and nine wounded in an IRA bomb attack on an army barracks in north London in August 1988.',
       "In July 1982, eight soldiers died in IRA bombings near the Household Cavalry barracks in central London and at a bandstand
in the capital's Regent's Park where an army band was playing.",
       'Three people died later and a total of 51 were injured in the bombings.',
       'The music school is the training center for young recruits who want to play in the seven Royal Marines bands.',
       'Up to 250 young men, most between 16 and 20, are based at the school, where they receive military and musical training.',
       "The roof of Janet Minnock's house was torn off by the force of the blast and all the back windows were shattered.",
       'The house has been blown to bits, she said.',
       'We are all shaken up.',
       "Mrs Minnock's next-door neighbor, Heather Hackett, said she was standing at her kitchen window facing the barracks at the
time of the explosion.",
       'She was holding her 4 months old son Luke in her arms with her other boys, Ben and Joshua at her side.',
       'I looked up from the sink and I just saw the whole building explode,she said.',
       'I told the boys to run and as Joshua turned a slither of glass embedded itself in his back.',
       'The whole window was blown across the kitchen.',
       'I just screamed and ran out of the room.',
       'The bang was so loud I thought the whole house was coming in.',
       'Sean Minnock said, I was asleep but woke up with a hell of a jolt.',
       'As workers tried to patch holes in his roof, he said: The bedroom ceiling fell in on me.',
       'I woke to find huge slabs of plaster on the bed and floor.',
       'I wondered what it was.',
       'As soon as I got up I looked out of what was left of the window and knew it was the barracks.'],
      dtype=object)
```

In [69]:

```python
count4 = CountVectorizer()
word_count4=count4.fit_transform(info4)
print(word_count4)
```

```
  (0, 25)	1
  (0, 119)	1
  (0, 341)	1
  (0, 129)	1
  (0, 215)	1
  (0, 46)	1
  (0, 26)	2
  (0, 344)	1
  (0, 338)	1
  (0, 227)	1
  (0, 158)	1
  (0, 193)	1
  (0, 1)	1
  (0, 255)	1
  (0, 170)	1
  (0, 6)	1
  (0, 260)	1
  (0, 283)	1
  (1, 329)	2
  (1, 175)	1
  (1, 82)	1
  (1, 275)	1
  (1, 131)	1
  (1, 56)	1
  (2, 338)	1
  :	:
  (39, 164)	1
  (39, 304)	1
  (39, 257)	1
  (39, 48)	1
  (39, 130)	1
  (40, 362)	1
  (40, 180)	1
  (40, 378)	1
  (40, 367)	1
  (41, 46)	1
  (41, 26)	1
  (41, 329)	2
  (41, 239)	2
  (41, 362)	2
  (41, 180)	1
  (41, 354)	1
  (41, 373)	1
  (41, 203)	1
```

```
  (41, 32)	2
  (41, 247)	1
  (41, 367)	1
  (41, 310)	1
  (41, 140)	1
  (41, 200)	1
  (41, 196)	1
```

In [70]:

```
# info2_len=len(info2)
# info2
tfidf_transformer4=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer4.fit(word_count4)
df_idf4 = pd.DataFrame(tfidf_transformer4.idf_, index=count4.get_feature_names(),columns=["IDF_Weights"])
```

In [72]:

```
tfidf_transformer4=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer4.fit(word_count4)
df_idf4 = pd.DataFrame(tfidf_transformer4.idf_, index=count4.get_feature_names(),columns=["IDF_Weights"])

#inverse document frequency
df_idf4.sort_values(by=['IDF_Weights'])
```

Out[72]:

	IDF_Weights
the	1.177681
and	1.870828
was	1.927987
of	1.927987
said	1.988611
...	...
frantically	4.068053
four	4.068053
force	4.068053
heavy	4.068053
killing	4.068053

387 rows × 1 columns

In [73]:

```
#tfidf
tf_idf_vector4=tfidf_transformer4.transform(word_count4)
feature_names4 = count4.get_feature_names()
```

In [74]:

```
fourth_document_vector=tf_idf_vector4[1]
df_tfifd4= pd.DataFrame(fourth_document_vector.T.todense(), index=feature_names4, columns=["TF-Idf"])
```

In [75]:

```
df_tfifd4.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[75]:

	TF-Idf
claimed	0.502906
blast	0.452781
responsibility	0.452781
for	0.389631
ira	0.316967
the	0.291177
100	0.000000
plaster	0.000000
province	0.000000
provided	0.000000
prime	0.000000
police	0.000000
playing	0.000000
play	0.000000
person	0.000000
ran	0.000000
people	0.000000
pensioner	0.000000

	TF-Idf
peace	0.000000
patch	0.000000

5)For the Fifth Document

In [76]:
```
new_data5.set_axis(["Information"],axis=1,inplace=True)
# new_data1.dtypes
new_data5["Information"]=new_data5["Information"].astype('string')
```

In [78]:
```
info5=new_data5['Information'].to_numpy()
info5
```

Out[78]:
```
array(['Neighbors were breakfasting, heading to work or asleep in bed when an explosion at a military barracks turned their homes
to rubble and they were confronted with the sight of  bodies being carried away.',
       'There was a terrific crash which reminded me of the Blitz.',
       'After that, the ceiling started to fall down around me, said Joan Betteridge, a pensioner in the southern England town of
Deal, where the blast at the Royal Marines School of Music occurred.',
       'The Irish Republican Army claimed reponsibilty for the explosion, which police said killed 11 people and injured 22.',
       'Nearby resident Sean Minnock said, I was asleep but woke up with a hell of a jolt, the bedroom ceiling fell in on me.',
       'I woke to find huge slabs of plaster on the bed and floor.',
       'From the wrecked, smoke-clouded barracks, I could hear terrified screams of agony.',
       'People started rushing about all over the place.',
       'It was horrible to watch and listen to, said Minnock.',
       'I knew people had been seriously hurt. I saw the rescuers pull out two bodies.',
       'I knew they were dead when they put them on the floor and put bed blankets right over them.',
       "Minnock's wife, Janet, said the roof of their house was torn off and all the back windows were shattered.",
       'The house has been blown to bits, she said.',
       'Mrs. Minnock was feeding her 2 years old son Thomas his breakfast when the explosion wrecked four terraced houses in the s
treet backing onto the barracks.',
       'Her next-door neighbor, Heather Hackett, was standing at her kitchen window facing the barracks, holding her 4-month-old s
on Luke in her arms.',
       'Her other boys, Ben and Joshua were at her side.',
       'I looked up from the sink and I just saw the whole building explode,she said.',
       'I told the boys to run and as Joshua turned a sliver of glass embedded itself in his back.',
       'The whole window was blown across the kitchen.',
       'I just screamed and ran out of the room.',
       'The bang was so loud I thought the whole house was coming in.',
       'At first I thought for sure Joshua had been seriously injured.',
       'There was blood coming out of his back.',
       'Doctors removed the glass and sent him home.',
       'College student Simon Mitford, narrowly escaped being injured in the explosion because he got up earlier than usual.',
       'His room was completely wrecked by the blast, his brother Alex said.',
       'Of the barracks, he said, I heard music playing and then it went bang and there was glass everywhere.',
       'It was a two-story building but now 90 percent of it is rubble.',
       'I heard a marine scream out, The band is under there.',
       'I was scared there was going to be a second explosion.'],
      dtype=object)
```

In [79]:
```
count5 = CountVectorizer()
word_count5=count5.fit_transform(info5)
print(word_count5)
```

```
  (0, 130)	1
  (0, 220)	2
  (0, 40)	1
  (0, 84)	1
  (0, 208)	2
  (0, 230)	1
  (0, 139)	1
  (0, 15)	1
  (0, 100)	1
  (0, 25)	1
  (0, 221)	1
  (0, 9)	1
  (0, 66)	1
  (0, 16)	1
  (0, 121)	1
  (0, 22)	1
  (0, 212)	1
  (0, 201)	1
  (0, 94)	1
  (0, 163)	1
  (0, 10)	1
  (0, 205)	1
  (0, 52)	1
  (0, 228)	1
  (0, 200)	1
  :	:
  (27, 104)	2
  (27, 213)	1
  (27, 42)	1
  (27, 191)	1
  (27, 132)	1
  (27, 2)	1
  (27, 145)	1
  (27, 103)	1
  (28, 200)	1
  (28, 204)	1
  (28, 141)	1
  (28, 86)	1
```

```
  (28, 103)	1
  (28, 118)	1
  (28, 170)	1
  (28, 20)	1
  (28, 214)	1
  (29, 208)	1
  (29, 66)	1
  (29, 204)	1
  (29, 217)	2
  (29, 168)	1
  (29, 78)	1
  (29, 23)	1
  (29, 174)	1
```

In [80]:

```
# info2_len=len(info2)
# info2
tfidf_transformer5=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer5.fit(word_count5)
df_idf5 = pd.DataFrame(tfidf_transformer5.idf_, index=count5.get_feature_names(),columns=["IDF_Weights"])
```

In [81]:

```
tfidf_transformer5=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer5.fit(word_count5)
df_idf5 = pd.DataFrame(tfidf_transformer5.idf_, index=count5.get_feature_names(),columns=["IDF_Weights"])

#inverse document frequency
df_idf5.sort_values(by=['IDF_Weights'])
```

Out[81]:

	IDF_Weights
the	1.215111
was	1.794930
of	1.869038
and	1.869038
said	2.131402
...	...
home	3.740840
homes	3.740840
horrible	3.740840
has	3.740840
years	3.740840

233 rows × 1 columns

In [82]:

```
#tfidf
tf_idf_vector5=tfidf_transformer5.transform(word_count5)
feature_names5 = count5.get_feature_names()
```

In [83]:

```
five_document_vector=tf_idf_vector5[1]
df_tfifd5= pd.DataFrame(five_document_vector.T.todense(), index=feature_names5, columns=["TF-Idf"])
```

In [84]:

```
df_tfifd5.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[84]:

	TF-Idf
crash	0.390942
blitz	0.390942
terrific	0.390942
reminded	0.390942
which	0.348568
me	0.318504
there	0.276130
of	0.195327
was	0.187582
the	0.126987
resident	0.000000
ran	0.000000
put	0.000000
removed	0.000000
reponsibilty	0.000000

	TF-Idf
republican	0.000000
pull	0.000000
police	0.000000
playing	0.000000
rescuers	0.000000

6)For the Sixth Document

In [85]:
```
new_data6.set_axis(["Information"],axis=1,inplace=True)
# new_data1.dtypes
new_data6["Information"]=new_data6["Information"].astype('string')
```

In [86]:
```
info6=new_data6['Information'].to_numpy()
info6
```

Out[86]:
```
array(['An explosion rocked the Royal Marines School of Music in a southeastern coastal town today, causing one building to collap
se and killing eight people, officials said.',
       'Thirty people were injured and up to 18 were missing and feared trapped in the rubble.',
       'The blast occurred at at 8:26 a.m. in a lounge in the barracks near Deal, about 70 miles southeast of London, the Defense
Ministry said.',
       'The building has collapsed, said a ministry spokesman, speaking anonymously in keeping with British custom.',
       "We've no idea of the cause of the blast at the moment.",
       'It is too early to tell.',
       'Scotland Yard said a forensic team from its antiterrorist squad had been called in to help investigate.',
       'Firefighters used heavy lifting equipment and thermal cameras to search for those trapped in the debris, said Kent Fire Br
igade spokesman Kevin Simmons.',
       'Kent police said 17 or 18 people were trapped.',
       'The Defense Ministry said seven were missing.',
       'Ten doctors gave emergency treatment at the scene and 11 ambulances took the injured to two hospitals, the ambulance servi
ce said.',
       'They are suffering from flash burns to their head and arms, fractures, and the sort of injuries you would expect after an
explosion, said a spokesman for Buckland Hospital in Dover, 20 miles south of Deal.',
       'South Eastern British Gas sent investigators to the scene but said there was nothing to indicate the explosion was caused
by a gas leak.',
       'Gas supplies to the barracks were cut as a precautionary measure, a spokesman said.',
       'Guy Platts, who owns a bookstore in Deal, located 20 miles north of the English Channel port of Dover, said he heard a mas
sive explosion.',
       'There are dozens of ambulances, police and fire brigade making their way there.',
       'Military targets on the British mainland have been attacked several times by the Irish Republican Army in the past year as
part of its campaign to rid Northern Ireland of British rule.',
       'One soldier was killed and nine wounded in an IRA attack on an army barracks in north London in August 1988. About 60 sold
iers narrowly escaped death or injury in February when they were evacuated from their barracks in Shropshire, western England, jus
t before a bomb exploded.',
       "In July 1982, eight soldiers died in IRA bombings near the Household Cavalry barracks at Knightsbridge in central London a
nd at a bandstand in the capital's Regent's Park where an army band was playing.",
       'Three people died later and a total of 51 were injured in the bombings.'],
      dtype=object)
```

In [88]:
```
count6 = CountVectorizer()
word_count6=count6.fit_transform(info6)
print(word_count6)
```

```
  (0, 14)	1
  (0, 76)	1
  (0, 165)	1
  (0, 195)	1
  (0, 166)	1
  (0, 130)	1
  (0, 171)	1
  (0, 147)	1
  (0, 138)	1
  (0, 101)	1
  (0, 185)	1
  (0, 52)	1
  (0, 209)	1
  (0, 205)	1
  (0, 48)	1
  (0, 150)	1
  (0, 38)	1
  (0, 204)	1
  (0, 53)	1
  (0, 15)	1
  (0, 120)	1
  (0, 67)	1
  (0, 156)	1
  (0, 148)	1
  (0, 169)	1
  :	:
  (18, 33)	1
  (18, 99)	1
  (18, 49)	1
  (18, 121)	1
  (18, 50)	1
  (18, 27)	1
  (18, 45)	1
  (18, 162)	1
  (18, 153)	1
  (18, 222)	1
```

```
  (18, 26)	1
  (18, 158)	1
  (19, 195)	1
  (19, 147)	1
  (19, 101)	1
  (19, 15)	1
  (19, 156)	1
  (19, 219)	1
  (19, 103)	1
  (19, 61)	1
  (19, 33)	1
  (19, 202)	1
  (19, 122)	1
  (19, 208)	1
  (19, 7)	1
```

In [89]:

```
# info2_len=len(info2)
# info2
tfidf_transformer6=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer6.fit(word_count6)
df_idf6 = pd.DataFrame(tfidf_transformer6.idf_, index=count6.get_feature_names(),columns=["IDF_Weights"])
```

In [90]:

```
tfidf_transformer6=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer6.fit(word_count6)
df_idf6 = pd.DataFrame(tfidf_transformer6.idf_, index=count6.get_feature_names(),columns=["IDF_Weights"])

#inverse document frequency
df_idf6.sort_values(by=['IDF_Weights'],ascending=False).head(20)
```

Out[90]:

	IDF_Weights
11	3.351375
owns	3.351375
military	3.351375
moment	3.351375
music	3.351375
narrowly	3.351375
nine	3.351375
no	3.351375
northern	3.351375
nothing	3.351375
occurred	3.351375
officials	3.351375
park	3.351375
idea	3.351375
part	3.351375
past	3.351375
platts	3.351375
playing	3.351375
port	3.351375
precautionary	3.351375

In [91]:

```
#tfidf
tf_idf_vector6=tfidf_transformer6.transform(word_count6)
feature_names6 = count6.get_feature_names()
```

In [92]:

```
sixth_document_vector=tf_idf_vector6[1]
df_tfifd6= pd.DataFrame(sixth_document_vector.T.todense(), index=feature_names6, columns=["TF-Idf"])
```

In [93]:

```
sixth_document_vector
```

Out[93]:

```
<1x230 sparse matrix of type '<class 'numpy.float64'>'
	with 14 stored elements in Compressed Sparse Row format>
```

In [94]:

```
df_tfifd6.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[94]:

	TF-Idf
were	0.385378
and	0.319880
feared	0.307714

	TF-Idf
rubble	0.307714
thirty	0.307714
up	0.307714
18	0.270486
missing	0.270486
trapped	0.244071
injured	0.244071
people	0.223583
to	0.151189
in	0.135850
the	0.116786
platts	0.000000
playing	0.000000
occurred	0.000000
of	0.000000
royal	0.000000
officials	0.000000

7)For the Seventh Document

In [95]:

```
new_data7.set_axis(["Information"],axis=1,inplace=True)
# new_data1.dtypes
new_data7["Information"]=new_data7["Information"].astype('string')
```

In [96]:

```
info7=new_data7['Information'].to_numpy()
info7
```

Out[96]:

```
array(['Hurricane Gilbert slammed into Kingston on Monday with torrential rains and 115 mph winds that ripped roofs off homes and
buildings, uprooted trees and downed power lines.',
       'No serious injuries were immediately reported in the city of 750,000 people, which was hit by the full force of the hurric
ane around noon.',
       'For half an hour, the hurricane lashed the city, tearing branches from trees, blowing down fences and whipping paper throu
gh the air.',
       "The National Weather Service reported heavy damage to Kingston's airport and aircraft parked on its fields.",
       'The first shock let up as the eye of the storm moved across the city.',
       'Skies brightened, the winds died down and people waited for an hour before the second blow of the hurricane arrived.',
       'All Jamaica-bound flights were canceled at Miami International Airport.',
       'Flights from the Cayman Islands, reportedly next in the path of the hurricane, arrived in Miami packed with travelers cutt
ing short their vacations.',
       'People were running around in the main lobby of our hotel (on Grand Cayman Island) like chickens with their heads cut off
said one man.',
       'A National Weather Service report said the hurricane was moving west at 17 mph with maximum sustained winds of 115 mph.',
       'It said Jamaica would receive up to 10 inches of rain that would cause flash floods and mud slides.',
       "Right now it's actually moving over Jamaica, said Bob Sheets, director of the National Hurricane Center in Miami.",
       "It looks like the eye is going to move lengthwise across that island, and they're going to bear the full brunt of this pow
erful hurricane, he said.",
       'Gilbert reached Jamaica after skirting southern Puerto Rico, Haiti and the Dominican Republic.',
       'Hurricane warnings were issued Monday for the south coast of Cuba east of Camaguey, the Cayman Islands, and Haiti, while w
arnings were discontinued for the Dominican Republic.',
       'High winds and heavy rain preceding the storm drenched Kingston overnight, toppling trees, causing local flooding and litt
ering streets with branches.',
       "Most of Jamaica's 2.3 million people stayed home, boarding up windows in preparation for the hurricane.",
       'The popular north coast resort area, on the other side of the mountains, was expected to receive heavy rain but not as muc
h damage from the hurricane as the south coast, where officials urged residents to seek higher ground.',
       "Havana Radio, meanwhile, reported Monday that 25,000 people were evacuated from coastal areas in Guantanamo Province on th
e nation's southeastern coast as Gilbert's winds and rain began to brush the island.",
       'In Washington, the Navy reported its bases at Guantanamo Bay, Cuba, and Roosevelt Roads, Puerto Rico, had taken various pr
ecautionary steps but appeared to be safe from the brunt of the hurricane.',
       'Ken Ross, a spokesman, said the Navy station at Guantanamo reported that as of 2:30 p.m. EDT, the brunt of the storm appea
red to be passing southeastern Cuba.',
       'They have reported maximum winds of 25 knots and gusts up to 50 knots,said Ross.',
       'But there are no reports of injuries or damage.',
       'The spokesman said earlier in the day, Guantanamo had moved to Condition Two, meaning electrical power usage was cut back
to only essential uses and all non-essential personnel sent to their barracks.',
       'The storm also skirted Puerto Rico without causing any damage to military facilities, Ross said.',
       'Sheets said Gilbert was expected next to sweep over the Cayman Islands, on its westward track, and in two to three days ve
er northwest into the southern Gulf of Mexico.',
       'Residents of the neighboring Caymans, a British dependency to the northwest, were urged to rush all preparatory actions.',
       'The National Weather Service warned that the Caymans could expect high waters and large waves which may undermine building
s along the beaches.',
       'All interests in the Western Caribbean should continue to monitor the progress of this dangerous hurricane, the service ad
vised.',
       "Forecaster Hal Gerrish on Sunday described Gilbert certainly one of the larger systems we've seen in the Caribbean for a l
ong time."],
      dtype=object)
```

In [97]:

```
count7 = CountVectorizer()
word_count7=count7.fit_transform(info7)
```

```
print(word_count7)
```

```
  (0, 130)	1
  (0, 107)	1
  (0, 258)	1
  (0, 137)	1
  (0, 146)	1
  (0, 195)	1
  (0, 170)	1
  (0, 320)	1
  (0, 286)	1
  (0, 222)	1
  (0, 20)	3
  (0, 2)	1
  (0, 177)	1
  (0, 319)	1
  (0, 275)	1
  (0, 235)	1
  (0, 237)	1
  (0, 193)	1
  (0, 127)	1
  (0, 49)	1
  (0, 293)	1
  (0, 289)	1
  (0, 82)	1
  (0, 211)	1
  (0, 154)	1
  :	:
  (28, 171)	1
  (28, 217)	1
  (28, 72)	1
  (28, 11)	1
  (29, 107)	1
  (29, 195)	1
  (29, 132)	1
  (29, 276)	2
  (29, 192)	1
  (29, 101)	1
  (29, 196)	1
  (29, 54)	1
  (29, 103)	1
  (29, 116)	1
  (29, 106)	1
  (29, 269)	1
  (29, 76)	1
  (29, 60)	1
  (29, 149)	1
  (29, 272)	1
  (29, 308)	1
  (29, 299)	1
  (29, 246)	1
  (29, 158)	1
  (29, 283)	1
```

In [98]:

```
tfidf_transformer7=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer7.fit(word_count7)
df_idf7 = pd.DataFrame(tfidf_transformer7.idf_, index=count7.get_feature_names(),columns=["IDF_Weights"])
```

In [99]:

```
tfidf_transformer7=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer7.fit(word_count7)
df_idf7 = pd.DataFrame(tfidf_transformer7.idf_, index=count7.get_feature_names(),columns=["IDF_Weights"])

#inverse document frequency
df_idf7.sort_values(by=['IDF_Weights'],ascending=False)
```

Out[99]:

	IDF_Weights
man	3.740840
neighboring	3.740840
noon	3.740840
north	3.740840
not	3.740840
...	...
to	1.794930
hurricane	1.794930
and	1.661398
of	1.389465
the	1.175891

323 rows × 1 columns

In [101…

```
#tfidf
tf_idf_vector7=tfidf_transformer7.transform(word_count7)
feature_names7 = count7.get_feature_names()
```

In [102…
```
seven_document_vector=tf_idf_vector7[1]
df_tfifd7= pd.DataFrame(seven_document_vector.T.todense(), index=feature_names7, columns=["TF-Idf"])
```

In [103…
```
df_tfifd7.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[103…

	TF-Idf
serious	0.248001
hit	0.248001
immediately	0.248001
750	0.248001
noon	0.248001
force	0.248001
by	0.248001
the	0.233869
000	0.221121
no	0.221121
full	0.221121
which	0.221121
around	0.221121
injuries	0.221121
city	0.202049
of	0.184231
people	0.175168
was	0.175168
reported	0.164949
were	0.164949

8)For the eighth Document

In [104…
```
new_data8.set_axis(["Information"],axis=1,inplace=True)
# new_data1.dtypes
new_data8["Information"]=new_data8["Information"].astype('string')
```

In [105…
```
info8=new_data8['Information'].to_numpy()
info8
```

Out[105…
```
array(["Communism suffered its first Big Mac attack Thursday as McDonald's opened a restaurant in Yugoslavia, and police were call
ed in to keep customers who lined up for hours from getting too unruly under the golden arches.",
       'I just wanted to taste genuine American hamburgers, said Milica Nikolic, a high school student who waited for three hours
to taste her first Big Mac.',
       "People curiously examined the renovated restaurant's plush interior and the back-lit signs depicting the hamburgers, frenc
h fries, milk shakes and other fare more familiar in the West.",
       'It also featured amber-colored tables and floors, pastel-colored upholstery, modern art paintings and discreet illuminatio
n.',
       'The fast-food outlet, located on a downtown square, had drawn crowds in recent days, and they began gathering long before
it opened Thursday.',
       'Police kept watch on the lines of customers snaking around the block, and they regulated the number who came inside to avo
id overcrowding.',
       'No opening of a restaurant in Belgrade has created such a sensation as this one today, one policeman said.',
       'I think this restaurant has no competition in Belgrade, said Milica Danic, a housewife who treated her son to a cheeseburg
er.',
       'It is much cleaner, the service is faster, the interior is nicer and it is not too expensive.',
       "The Belgrade media have suggested that the success of McDonald's in Yugoslavia depends on its acceptance by citizens long
accustomed to a hamburger-like fast-food dish called the Pljeskavica: ground pork and onions on a bun.",
       "In fact, this is a clash between the Big Mac and Pljeskavica, said Vesna Milosevic, an official of Genex, a Yugoslav state
-run enterprise that has contracted a joint venture agreement with McDonald's.",
       "Our aim is not to destroy the Pljeskavica on the Yugoslav market, said Predrag Dostanic, managing director of the Genex-Mc
Donald's.",
       'We want to change customs of the local people used to completly different eating habits.',
       'He said that lounging at tables for a long time after a finished meal will draw a warning. Also, smoking is forbidden and
alcohol will not be served.',
       'This contrasts sharply with the Balkan and Yugoslav custom of sitting with a drink in smoke-filled restaurants and chattin
g with friends after the meal.',
       'The Big Mac meal, consisting of a hamburger, soft drink and french fries costs the equivalent of 2.57 dollar, or about as
much the similar meal would cost in numerous Pljeskavica joints around town.',
       "Sadik Seljami, a waiter in a small Pljeskavica outlet just a few hundred yards from the McDonald's, suggested that the Ame
rican restaurant wants to drive Yugoslav fast-food outlets out of business.",
       'However, we will not give up the fight even if we have to lower the prices, said Seljami.',
       "Glen Cook, an executive of the McDonald's Corp, said during the opening ceremonies, We are very excited about the opening
of this restaurant, not only because it is the first one in a communist country, but also because it is one of the nicest in Europ
e.",
       "McDonald's and Genex contribute $1 million each for the flagship restaurant.",
       'They will also share the profits equally even though it will be managed entirely by Yugoslavs.',
       'The restaurant has 350 seats and employs 110 people capable of serving 2,500 meals per hour. In an effort to keep a high l
evel of services, the management is entitled to fire any employees who fail to perform.',
```

```
 'The American corporation plans to open five additional restaurants Yugoslavia in the next five years.',
 "The next East European McDonald's, and the first in a Soviet bloc country, is to open next month in Budapest, Hungary."],
dtype=object)
```

In [106…

```
count8 = CountVectorizer()
word_count8=count8.fit_transform(info8)
print(word_count8)
```

```
  (0, 53)	1
  (0, 266)	1
  (0, 161)	1
  (0, 120)	1
  (0, 34)	1
  (0, 177)	1
  (0, 24)	1
  (0, 277)	1
  (0, 22)	1
  (0, 182)	1
  (0, 209)	1
  (0, 235)	1
  (0, 156)	2
  (0, 309)	1
  (0, 16)	1
  (0, 226)	1
  (0, 300)	1
  (0, 42)	1
  (0, 279)	1
  (0, 165)	1
  (0, 71)	1
  (0, 302)	1
  (0, 169)	1
  (0, 286)	1
  (0, 125)	1
  :	:
  (22, 236)	1
  (22, 63)	1
  (22, 223)	1
  (22, 208)	1
  (22, 121)	2
  (22, 7)	1
  (22, 194)	1
  (22, 307)	1
  (23, 120)	1
  (23, 182)	1
  (23, 156)	2
  (23, 16)	1
  (23, 279)	1
  (23, 271)	2
  (23, 159)	1
  (23, 66)	1
  (23, 208)	1
  (23, 194)	2
  (23, 91)	1
  (23, 102)	1
  (23, 260)	1
  (23, 35)	1
  (23, 191)	1
  (23, 37)	1
  (23, 153)	1
```

In [107…

```
tfidf_transformer8=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer8.fit(word_count8)
df_idf8 = pd.DataFrame(tfidf_transformer8.idf_, index=count8.get_feature_names(),columns=["IDF_Weights"])
```

In [108…

```
tfidf_transformer8=TfidfTransformer(smooth_idf=True,use_idf=True)
tfidf_transformer8.fit(word_count8)
df_idf8 = pd.DataFrame(tfidf_transformer8.idf_, index=count8.get_feature_names(),columns=["IDF_Weights"])

#inverse document frequency
df_idf8.sort_values(by=['IDF_Weights'],ascending=False).head(20)
```

Out[108…

	IDF_Weights
110	3.525729
numerous	3.525729
market	3.525729
meals	3.525729
media	3.525729
milk	3.525729
million	3.525729
milosevic	3.525729
modern	3.525729
month	3.525729
more	3.525729
nicer	3.525729

	IDF_Weights
nicest	3.525729
nikolic	3.525729
number	3.525729
official	3.525729
management	3.525729
onions	3.525729
only	3.525729
or	3.525729

In [109…

```
#tfidf
tf_idf_vector8=tfidf_transformer8.transform(word_count8)
feature_names8 = count8.get_feature_names()
```

In [110…

```
eight_document_vector=tf_idf_vector8[1]
df_tfifd8= pd.DataFrame(eight_document_vector.T.todense(), index=feature_names8, columns=["TF-Idf"])
```

In [111…

```
df_tfifd8.sort_values(by=["TF-Idf"],ascending=False).head(20)
```

Out[111…

	TF-Idf
taste	0.444788
student	0.222394
three	0.222394
waited	0.222394
genuine	0.222394
school	0.222394
wanted	0.222394
nikolic	0.222394
to	0.208651
milica	0.196818
just	0.196818
hamburgers	0.196818
hours	0.196818
high	0.196818
her	0.196818
american	0.178672
big	0.164597
first	0.164597
for	0.164597
mac	0.164597

Recalling the Headlines

In [117…

```
Head_len=len(Head)
Head_len
```

Out[117… 3

In [116…

```
# to select multiple rows
result = Head.iloc[[0,1,2]]
result
```

Out[116…

	Headline
0	Hurricane Gilbert Heads Toward Dominican Coast
1	IRA terrorist attack
2	McDonald's Opens First Restaurant in China

B) RapidFuzz

In [42]:
```
import rapidfuzz as rp
from rapidfuzz import process, fuzz
```

Document 1 Testing

Indentify as Title

> **McDonald's Opens First Restaurant in China**

Selecting Partial Ratio is provides the optimal results for the String Matching according to News Title

Partial Ratio: It finds the ratio similarity measure between the shorter string and every substring of length m of the longer string, and returns the maximum of those similarity measures. Basically, it searches for the optimal alignment of the shorter string in the longer string and returns the fuzz.ratio for this

Higher the Value Similaity of the text is increasing. Lower Score gives high chance of mismatches in the text.

In [179…
```
info1 #text will be extracted from the text files Respectivly for testing to verify our prediction is correct
```

Out[179…
```
array(["Thousands of queue-hardened Soviets on Wednesday cheerfully lined up to get a taste of ''gamburgers'', ''chizburgers'' and
''Filay-o-feesh'' sandwiches as McDonald's opened in the land of Lenin for the first time.",
       "The world's largest version of the landmark American fast-food chain rang up 30,000 meals on 27 cash registers, breaking t
he opening-day record for McDonald's worldwide, officials said.",
       'The Soviets, bundled in fur coats and hats, seemed unfazed, lining up before dawn outside the 700 seat restaurant, the fir
st of 20 planned across the Soviet Union.',
       'The crush of customers was so intense the company stayed open until midnight, two hours later than planned.',
       'I only waited an hour and I think they served thousands before me, said a happy middle-aged woman who works at an aluminum
plant.',
       "And it was only 10 rubles for all this, she said. I'm taking it back for the girls at the factory to try.",
       "Big Macs were priced at 3.75 rubles and double cheeseburgers at 3 rubles about two hours' pay for a starting McDonald's st
affer or the average Soviet, but much cheaper than other private restaurants that have sprung up recently.",
       'The official exchange rate is 1.59 dollar per ruble but foreign visitors can buy rubles for 16 cents each, about what the
currency is worth on the black market.',
       "Half the day's sales were donated to the Soviet Children's Fund, which provides medical care and assistance to orphans and
disadvantaged children, Gary Reinblatt, senior vice president of McDonald's Canada, said from Toronto.",
       "The restaurant, built by the company in a joint venture with the city of Moscow that began 14 years ago, brought to 52 the
number of countries where McDonald's operates.",
       "The previous opening-day record for sales was in Budapest, company officials said. Besides its restaurants in the United S
tates, the leading number of McDonald's are in Canada and Japan, the officials said.",
       'Soviets got a first-hand look at such alien concepts as efficiency and fast, friendly service. Normally dour citizens brok
e into grins as they caught the infectious cheerful mood from youthful Soviet staffers hired for their ability to smile and work h
ard.',
       'Accordions played folk songs and women in traditional costumes danced with cartoon characters, including Mickey Mouse and
Baba Yaga, a witch of Russian fairy tales.',
       'One Muscovite, accustomed to clerks who snarl if they say anything at all, asked for a straw and was startled when a smili
ng young Soviet woman found him one and popped it straight into his drink.',
       "For most customers, it was their first experience with a hamburger. Sandwiches were served in the familiar bag marked with
the golden arches, but were packed in wrappers bearing Cyrillic letters, approximating ``gamburger.''",
       "They tried them one-handed.They picked their sandwiches apart to examine the contents. One young woman finally squashed he
r ``Beeg Mak'' to fit her lips around it.",
       "''It tasted great!'' a 14 years old boy said.",
       "It's a lot different from a stolovaya,'' he continued with a smile, referring to the much cheaper but run down dirty cafet
erias that slop rice and fat or boiled sausage.",
       "Under the sign of the golden arches, accented by the Soviet hammer and sickle flag, hundreds lined up for the long awaited
grand opening at 10 am on Pushkin Square, reaching out excitedly for McDonald's flags and pins as the hamburger chain's army fulfi
lled the Soviet penchant for souvenirs with Western logos.",
       "Publicity conscious managers had the staff shout ''Good morning, America!'' in English and Russian, for an American TV net
work.",
       "McDonald's of Canada Chairman George Cohon, the man behind the deal, said many people were buying multiple orders and the
restaurant served 15,000 to 20,000 people in just the first five hours of operation.",
       'The restaurant limited purchases to 10 Big Macs per customer in hopes of preventing burger scalping.',
       "McDonald's built its own factory, including bakery, dairy, meat processing plant and even potato storage yard, to provide
its own guaranteed supplies in a country where up to 25 percent of the harvest rots en route to the consumer.",
       "One McDonald's associate said the company wound up importing wooden crates from Finland for storing potatoes because when
they went to build crates, they found there was no wood, and no nails.",
       'They found you need a permit to buy nails.'], dtype=object)
```

In [162…
```
 rp.fuzz.partial_ratio("McDonald's built its own factory, including bakery, dairy, meat processing plant and even potato storage y
```

Out[162…
```
55.072463768115945
```

> Compare Other Article Titles ==> **Which provides Lower Values**

In [163…
```
 rp.fuzz.partial_ratio("McDonald's built its own factory, including bakery, dairy, meat processing plant and even potato storage y
```

Out[163…
```
43.47826086956522
```

In [164…
```
rp.fuzz.partial_ratio("McDonald's built its own factory, including bakery, dairy, meat processing plant and even potato storage ya
```

Out[164…
```
44.99999999999999
```

Other titles provide less with text similarity score which means our Selected title is the Correct One

Document 2 Testing

Indentify as Title

> **Hurricane Gilbert Heads Toward Dominican Coast**

In [143…
```
info2 #text will be extracted from the text files Respectivly for testing to verify our prediction is correct
```

Out[143…
```
array(['Hurricane Gilbert, packing 110 mph winds and torrential rain, moved over this capital city today after skirting Puerto Ric
o, Haiti and the Dominican Republic.',
       'There were no immediate reports of casualties.',
       'Telephone communications were affected.',
       "Right now it's actually moving over Jamaica,said Bob Sheets, director of the National Hurricane Center in Miami.",
       "We've already had reports of 110 mph winds on the eastern tip.",
       "It looks like the eye is going to move lengthwise across that island, and they're going to bear the full brunt of this pow
erful hurricane, Sheets said.",
       'Forecasters say Gilbert was expected to lash Jamaica throughout the day and was on track to later strike the Cayman Island
s, a small British dependency northwest of Jamaica.',
       "Meanwhile, Havana Radio reported today that 25,000 people were evacuated from Guantanamo Province on Cuba's southeastern c
oast as strong winds fanning out from Gilbert began brushing the island.",
       'All Jamaica-bound flights were canceled at Miami International Airport, while flights from Grand Cayman, the main island o
f the three-island chain, arrived packed with frightened travelers.',
       'People were running around in the main lobby of our hotel (on Grand Cayman) like chickens with their heads cut off, said o
ne vacationer who was returning home to California through Miami.',
       'Hurricane warnings were posted for the Cayman Islands, Cuba and Haiti.',
       'Warnings were discontinued for the Dominican Republic.',
       'All interests in the Western Caribbean should continue to monitor the progress of this dangerous hurricane, the service sa
id, adding, Little change in strength is expected for the next several hours as the hurricane moves westward over Jamaica.',
       'The Associated Press Caribbean headquarters in San Juan, Puerto Rico, was unable to get phone calls through to Kingston, w
here high winds and heavy rain preceding the storm drenched the capital overnight, toppling trees, causing local flooding and litt
ering streets with branches.',
       'Most Jamaicans stayed home, boarding up windows in preparation for the hurricane.',
       'Some companies broadcast appeals for technicians and electricians to report to work.',
       "The weather bureau predicted Gilbert's center, 140 miles southeast of Kingston before dawn, would pass south of Kingston a
nd hit the southern parish of Clarendon.",
       'Flash flood warnings were issued for the parishes of Portland on the northeast and St. Mary on the north.',
       'The north coast tourist region from Montego Bay on the west and Ocho Rios on the east, far from the southern impact zone a
nd separated by mountains, was expected only to receive heavy rain.',
       'Officials urged residents in the higher risk areas along the south coast to seek higher ground.',
       "It's certainly one of the larger systems we've seen in the Caribbean for a long time, said Hal Gerrish, forecaster at the
National Hurricane Center.",
       'Forecasters at the center said the eye of Gilbert was 140 miles southeast of Kingston at dawn today.',
       'Maximum sustained winds were near 110 mph, with tropical-storm force winds extending up to 250 miles to the north and 100
miles to the south.',
       'Prime Minister Edward Seaga of Jamaica alerted all government agencies, saying Sunday night: Hurricane Gilbert appears to
be a real threat and everyone should follow the instructions and hurricane precautions issued by the Office of Disaster Preparedne
ss in order to minimize the danger.',
       'Forecasters said the hurricane had been gaining strength as it passed over the ocean after it dumped 5 to 10 inches of rai
n on the Dominican Republic and Haiti, which share the island of Hispaniola.',
       "We should know within about 72 hours whether it's going to be a major threat to the United States,'' said Martin Nelson, a
nother meteorologist at the center.",
       "It's moving at about 17 mph to the west and normally hurricanes take a northward turn after they pass central Cuba.",
       "Cuba's official Prensa Latina news agency said a state of alert was declared at midday in the Cuban provinces of Guantanam
o, Holguin, Santiago de Cuba and Granma.",
       'In the report from Havana received in Mexico City, Prensa Latina said civil defense officials were broadcasting bulletins
on national radio and television recommending emergency measures and providing information on the storm.',
       "Heavy rain and stiff winds downed power lines and caused flooding in the Dominican Republic on Sunday night as the hurrica
ne's center passed just south of the Barahona peninsula, then less than 100 miles from neighboring Haiti.",
       "The storm ripped the roofs off houses and flooded coastal areas of southwestern Puerto Rico after reaching hurricane stren
gth off the island's southeast Saturday night.",
       'Flights were canceled Sunday in the Dominican Republic, where civil defense director Eugenio Cabral reported some flooding
in parts of the capital of Santo Domingo and power outages there and in other southern areas.'],
      dtype=object)
```

In [161…
```
rp.fuzz.partial_ratio("Hurricane Gilbert, packing 110 mph winds and torrential rain, moved over this capital city today after ski
```

Out[161…
```
60.86956521739131
```

> Compare Other Article Titles ==> **Which provides Lower Values**

In [165…
```
rp.fuzz.partial_ratio("Hurricane Gilbert, packing 110 mph winds and torrential rain, moved over this capital city today after skir
```

Out[165…
```
50.0
```

In [166…
```
rp.fuzz.partial_ratio("Hurricane Gilbert, packing 110 mph winds and torrential rain, moved over this capital city today after ski
```

Out[166…
```
40.476190476190474
```

Document 3 Testing

Indentify as Title

> **Hurricane Gilbert Heads Toward Dominican Coast**

In [178…
```
info3 #text will be extracted from the text files Respectivly for testing to verify our prediction is correct
```

Out[178…
```
array(['Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south co
```

```
ast to prepare for high winds, heavy rains and high seas.',
       'The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.',
       'There is no need for alarm, Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Satur
day.',
       "Cabral said residents of the province of Barahona should closely follow Gilbert's movement.",
       'An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo
Domingo.',
       'Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night.',
       'The National Hurricane Center in Miami reported its position at 2 a.m.',
       'Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast o
f Santo Domingo.',
       'The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a broad area of clo
udiness and heavy weather rotating around the center of the storm.',
       'The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.',
       "Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet feet to Puerto
Rico's south coast.",
       'There were no reports of casualties.',
       'San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.',
       'On Saturday, Hurricane Florence was downgraded to a tropical storm and its remnants pushed inland from the Gulf Coast.',
       'Residents returned home, happy to find little damage from 80 mph winds and sheets of rain.',
       'Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.',
       'The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.'],
      dtype=object)
```

In [167…

```
rp.fuzz.partial_ratio("Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily po
```

Out[167… 79.12087912087912

> Compare Other Article Titles ==> **Which provides Lower Values**

In [168…

```
rp.fuzz.partial_ratio("Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily pop
```

Out[168… 40.0

In [169…

```
rp.fuzz.partial_ratio("Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily po
```

Out[169… 42.85714285714286

Document 4 Testing

Indentify as Title

> **IRA terrorist attack**

In [174…

```
info4 #text will be extracted from the text files Respectivly for testing to verify our prediction is correct
```

Out[174…

```
array(['An explosion today flattened a military barracks and tore through nearby homes, killing 11 people and injuring 22, police
said.',
       'The IRA claimed responsibility for the blast.',
       'More than 100 rescue workers frantically dug through the rubble of a three-story building that collapsed at the Royal Mari
nes School of Music near Deal.',
       'Stunned neighbors gathered outside homes that were damaged or destroyed.',
       'Chief Police Inspector Alan Butterfield of Kent, who who provided the casualty figures and coordinated the rescue effort,
first reported that one person was missing but later said everyone was accounted for.',
       'He said many of the injured were seriously hurt.',
       'There was a terrific crash which reminded me of the Blitz.',
       'After that, the ceiling started to fall down around me, said pensioner Joan Betteridge.',
       'Defense Secretary Tom King, inspecting the wreckage, said, It is not yet absolutely confirmed that it is a bomb, but all t
he evidence is quite clearly that this is an IRA atrocity.',
       "British military installations are a frequent bombing target of the Irish Republican Army in its campaign to rid Northern
Ireland of British rule, but today's explosion in the coastal town 70 miles southeast of London was the worst IRA attack on the Br
itish mainland in more than seven years.",
       'The explosion occurred at at 8:26 a.m. in a lounge in thebarracks.',
       'One of the bands had just stopped playing on the parade ground, said a ministry spokesman, speaking anonymously in keeping
with British custom.',
       'Dozens of homes near the school were damaged, including four that were destroyed. Witnesses reported hearing the explosion
two miles away.',
       'The Defense Ministry would not say how many servicemen and civilians were included in the casualty figures.',
       'However, King told reporters the attack was directed against unarmed bandsmen.',
       'Firefighters used heavy lifting equipment and thermal cameras to search through the debris, said Kent Fire Brigade spokesm
an Kevin Simmons.',
       'Ten doctors were giving emergency treatment at the scene and 11 ambulances were taking the injured to two hospitals, the a
mbulance service said.',
       "A statement telephoned to Ireland International, a Dublin news agency, said, we have visited the Royal Marines in Kent in
response to Prime Minister Margaret Thatcher's visit to Northern Ireland nine days ago.",
       'The IRA said Mrs Thatcher went to the British province with a message of war,but we still want peace and we want the Briti
sh government to leave our country.',
       "It was signed P. O'Neill, a nom de guerre the IRA usually uses to claim responsibility for actions outside Northern Irelan
d.",
       'Irish Prime Minister Charles Haughey issued a statement in Dublin condemning the attack, calling it an outrage.',
       'The last IRA bomb attempt on the British mainland was in February when about 60 soldiers were evacuated from their barrack
s in Shropshire, western England, just before a bomb exploded.',
       'One soldier was killed and nine wounded in an IRA bomb attack on an army barracks in north London in August 1988.',
       "In July 1982, eight soldiers died in IRA bombings near the Household Cavalry barracks in central London and at a bandstand
in the capital's Regent's Park where an army band was playing.",
       'Three people died later and a total of 51 were injured in the bombings.',
       'The music school is the training center for young recruits who want to play in the seven Royal Marines bands.',
       'Up to 250 young men, most between 16 and 20, are based at the school, where they receive military and musical training.',
       "The roof of Janet Minnock's house was torn off by the force of the blast and all the back windows were shattered.",
       'The house has been blown to bits, she said.',
```

```
        'We are all shaken up.',
        "Mrs Minnock's next-door neighbor, Heather Hackett, said she was standing at her kitchen window facing the barracks at the
time of the explosion.",
        'She was holding her 4 months old son Luke in her arms with her other boys, Ben and Joshua at her side.',
        'I looked up from the sink and I just saw the whole building explode,she said.',
        'I told the boys to run and as Joshua turned a slither of glass embedded itself in his back.',
        'The whole window was blown across the kitchen.',
        'I just screamed and ran out of the room.',
        'The bang was so loud I thought the whole house was coming in.',
        'Sean Minnock said, I was asleep but woke up with a hell of a jolt.',
        'As workers tried to patch holes in his roof, he said: The bedroom ceiling fell in on me.',
        'I woke to find huge slabs of plaster on the bed and floor.',
        'I wondered what it was.',
        'As soon as I got up I looked out of what was left of the window and knew it was the barracks.'],
       dtype=object)
```

In [175…

```
rp.fuzz.partial_ratio("The IRA said Mrs Thatcher went to the British province with a message of war,but we still want peace and w
```

Out[175… 50.0

> Compare Other Article Titles ==> **Which provides Lower Values**

In [176…

```
rp.fuzz.partial_ratio("The IRA said Mrs Thatcher went to the British province with a message of war,but we still want peace and we
```

Out[176… 41.30434782608695

In [177…

```
rp.fuzz.partial_ratio("The IRA said Mrs Thatcher went to the British province with a message of war,but we still want peace and w
```

Out[177… 40.476190476190474

Document 5 Testing

Indentify the as Title

> **IRA terrorist attack**

In [180…

```
info5 #text will be extracted from the text files Respectivly for testing to verify our prediction is correct
```

Out[180…

```
array(['Neighbors were breakfasting, heading to work or asleep in bed when an explosion at a military barracks turned their homes
to rubble and they were confronted with the sight of  bodies being carried away.',
        'There was a terrific crash which reminded me of the Blitz.',
        'After that, the ceiling started to fall down around me, said Joan Betteridge, a pensioner in the southern England town of
Deal, where the blast at the Royal Marines School of Music occurred.',
        'The Irish Republican Army claimed reponsibilty for the explosion, which police said killed 11 people and injured 22.',
        'Nearby resident Sean Minnock said, I was asleep but woke up with a hell of a jolt, the bedroom ceiling fell in on me.',
        'I woke to find huge slabs of plaster on the bed and floor.',
        'From the wrecked, smoke-clouded barracks, I could hear terrified screams of agony.',
        'People started rushing about all over the place.',
        'It was horrible to watch and listen to, said Minnock.',
        'I knew people had been seriously hurt. I saw the rescuers pull out two bodies.',
        'I knew they were dead when they put them on the floor and put bed blankets right over them.',
        "Minnock's wife, Janet, said the roof of their house was torn off and all the back windows were shattered.",
        'The house has been blown to bits, she said.',
        'Mrs. Minnock was feeding her 2 years old son Thomas his breakfast when the explosion wrecked four terraced houses in the s
treet backing onto the barracks.',
        'Her next-door neighbor, Heather Hackett, was standing at her kitchen window facing the barracks, holding her 4-month-old s
on Luke in her arms.',
        'Her other boys, Ben and Joshua were at her side.',
        'I looked up from the sink and I just saw the whole building explode,she said.',
        'I told the boys to run and as Joshua turned a sliver of glass embedded itself in his back.',
        'The whole window was blown across the kitchen.',
        'I just screamed and ran out of the room.',
        'The bang was so loud I thought the whole house was coming in.',
        'At first I thought for sure Joshua had been seriously injured.',
        'There was blood coming out of his back.',
        'Doctors removed the glass and sent him home.',
        'College student Simon Mitford, narrowly escaped being injured in the explosion because he got up earlier than usual.',
        'His room was completely wrecked by the blast, his brother Alex said.',
        'Of the barracks, he said, I heard music playing and then it went bang and there was glass everywhere.',
        'It was a two-story building but now 90 percent of it is rubble.',
        'I heard a marine scream out, The band is under there.',
        'I was scared there was going to be a second explosion.'],
       dtype=object)
```

In [193…

```
rp.fuzz.partial_ratio("The IRA claimed reponsibilty for the explosion, which police said killed 11 people and injured 22", "IRA t
```

Out[193… 40.0

> Compare Other Article Titles ==> **Which provides Lower Values**

In [192…

```
rp.fuzz.partial_ratio("The IRA claimed reponsibilty for the explosion, which police said killed 11 people and injured 22", "Hurric
```

Out[192… 37.2093023255814

In [196…

```
rp.fuzz.partial_ratio("The Irish Republican Army claimed reponsibilty for the explosion", "McDonald's Opens First Restaurant in C
```

Out[196… 37.5

Document 6 Testing

Indentify the as Title

> **IRA terrorist attack**

In [197…

```
info6 #text will be extracted from the text files Respectivly for testing to verify our prediction is correct
```

Out[197…

```
array(['An explosion rocked the Royal Marines School of Music in a southeastern coastal town today, causing one building to collap
se and killing eight people, officials said.',
       'Thirty people were injured and up to 18 were missing and feared trapped in the rubble.',
       'The blast occurred at at 8:26 a.m. in a lounge in the barracks near Deal, about 70 miles southeast of London, the Defense
Ministry said.',
       'The building has collapsed, said a ministry spokesman, speaking anonymously in keeping with British custom.',
       "We've no idea of the cause of the blast at the moment.",
       'It is too early to tell.',
       'Scotland Yard said a forensic team from its antiterrorist squad had been called in to help investigate.',
       'Firefighters used heavy lifting equipment and thermal cameras to search for those trapped in the debris, said Kent Fire Br
igade spokesman Kevin Simmons.',
       'Kent police said 17 or 18 people were trapped.',
       'The Defense Ministry said seven were missing.',
       'Ten doctors gave emergency treatment at the scene and 11 ambulances took the injured to two hospitals, the ambulance servi
ce said.',
       'They are suffering from flash burns to their head and arms, fractures, and the sort of injuries you would expect after an
explosion, said a spokesman for Buckland Hospital in Dover, 20 miles south of Deal.',
       'South Eastern British Gas sent investigators to the scene but said there was nothing to indicate the explosion was caused
by a gas leak.',
       'Gas supplies to the barracks were cut as a precautionary measure, a spokesman said.',
       'Guy Platts, who owns a bookstore in Deal, located 20 miles north of the English Channel port of Dover, said he heard a mas
sive explosion.',
       'There are dozens of ambulances, police and fire brigade making their way there.',
       'Military targets on the British mainland have been attacked several times by the Irish Republican Army in the past year as
part of its campaign to rid Northern Ireland of British rule.',
       'One soldier was killed and nine wounded in an IRA attack on an army barracks in north London in August 1988. About 60 sold
iers narrowly escaped death or injury in February when they were evacuated from their barracks in Shropshire, western England, jus
t before a bomb exploded.',
       "In July 1982, eight soldiers died in IRA bombings near the Household Cavalry barracks at Knightsbridge in central London a
nd at a bandstand in the capital's Regent's Park where an army band was playing.",
       'Three people died later and a total of 51 were injured in the bombings.'],
      dtype=object)
```

In [198…

```
rp.fuzz.partial_ratio("One soldier was killed and nine wounded in an IRA attack on an army barracks in north London in August 198
```

Out[198… 50.0

> Compare Other Article Titles ==> **Which provides Lower Values**

In [199…

```
rp.fuzz.partial_ratio("One soldier was killed and nine wounded in an IRA attack on an army barracks in north London in August 1988
```

Out[199… 44.99999999999999

In [200…

```
rp.fuzz.partial_ratio("One soldier was killed and nine wounded in an IRA attack on an army barracks in north London in August 198
```

Out[200… 42.85714285714286

Document 7 Testing

Indentify the as Title

> **Hurricane Gilbert Heads Toward Dominican Coast**

In [202…

```
info7 #text will be extracted from the text files Respectivly for testing to verify our prediction is correct
```

Out[202…

```
array(['Hurricane Gilbert slammed into Kingston on Monday with torrential rains and 115 mph winds that ripped roofs off homes and
buildings, uprooted trees and downed power lines.',
       'No serious injuries were immediately reported in the city of 750,000 people, which was hit by the full force of the hurric
ane around noon.',
       'For half an hour, the hurricane lashed the city, tearing branches from trees, blowing down fences and whipping paper throu
gh the air.',
       "The National Weather Service reported heavy damage to Kingston's airport and aircraft parked on its fields.",
       'The first shock let up as the eye of the storm moved across the city.',
       'Skies brightened, the winds died down and people waited for an hour before the second blow of the hurricane arrived.',
       'All Jamaica-bound flights were canceled at Miami International Airport.',
       'Flights from the Cayman Islands, reportedly next in the path of the hurricane, arrived in Miami packed with travelers cutt
ing short their vacations.',
       'People were running around in the main lobby of our hotel (on Grand Cayman Island) like chickens with their heads cut off
said one man.',
       'A National Weather Service report said the hurricane was moving west at 17 mph with maximum sustained winds of 115 mph.',
       'It said Jamaica would receive up to 10 inches of rain that would cause flash floods and mud slides.',
       "Right now it's actually moving over Jamaica, said Bob Sheets, director of the National Hurricane Center in Miami.",
       "It looks like the eye is going to move lengthwise across that island, and they're going to bear the full brunt of this pow
erful hurricane, he said.",
```

```
        'Gilbert reached Jamaica after skirting southern Puerto Rico, Haiti and the Dominican Republic.',
        'Hurricane warnings were issued Monday for the south coast of Cuba east of Camaguey, the Cayman Islands, and Haiti, while w
arnings were discontinued for the Dominican Republic.',
        'High winds and heavy rain preceding the storm drenched Kingston overnight, toppling trees, causing local flooding and litt
ering streets with branches.',
        "Most of Jamaica's 2.3 million people stayed home, boarding up windows in preparation for the hurricane.",
        'The popular north coast resort area, on the other side of the mountains, was expected to receive heavy rain but not as muc
h damage from the hurricane as the south coast, where officials urged residents to seek higher ground.',
        "Havana Radio, meanwhile, reported Monday that 25,000 people were evacuated from coastal areas in Guantanamo Province on th
e nation's southeastern coast as Gilbert's winds and rain began to brush the island.",
        'In Washington, the Navy reported its bases at Guantanamo Bay, Cuba, and Roosevelt Roads, Puerto Rico, had taken various pr
ecautionary steps but appeared to be safe from the brunt of the hurricane.',
        'Ken Ross, a spokesman, said the Navy station at Guantanamo reported that as of 2:30 p.m. EDT, the brunt of the storm appea
red to be passing southeastern Cuba.',
        'They have reported maximum winds of 25 knots and gusts up to 50 knots,said Ross.',
        'But there are no reports of injuries or damage.',
        'The spokesman said earlier in the day, Guantanamo had moved to Condition Two, meaning electrical power usage was cut back
to only essential uses and all non-essential personnel sent to their barracks.',
        'The storm also skirted Puerto Rico without causing any damage to military facilities, Ross said.',
        'Sheets said Gilbert was expected next to sweep over the Cayman Islands, on its westward track, and in two to three days ve
er northwest into the southern Gulf of Mexico.',
        'Residents of the neighboring Caymans, a British dependency to the northwest, were urged to rush all preparatory actions.',
        'The National Weather Service warned that the Caymans could expect high waters and large waves which may undermine building
s along the beaches.',
        'All interests in the Western Caribbean should continue to monitor the progress of this dangerous hurricane, the service ad
vised.',
        "Forecaster Hal Gerrish on Sunday described Gilbert certainly one of the larger systems we've seen in the Caribbean for a l
ong time."],
       dtype=object)
```

In [203…

```
rp.fuzz.partial_ratio("Hurricane Gilbert slammed into Kingston on Monday with torrential rains and 115 mph winds that ripped roofs
```

Out[203… 67.46987951807229

> Compare Other Article Titles ==> **Which provides Lower Values**

In [204…

```
rp.fuzz.partial_ratio("Hurricane Gilbert slammed into Kingston on Monday with torrential rains and 115 mph winds that ripped roof
```

Out[204… 44.99999999999999

In [205…

```
rp.fuzz.partial_ratio("Hurricane Gilbert slammed into Kingston on Monday with torrential rains and 115 mph winds that ripped roof
```

Out[205… 42.85714285714286

Document 8 Testing

Indentify the as Title

> **McDonald's Opens First Restaurant in China**

In [207…

```
info8 #text will be extracted from the text files Respectivly for testing to verify our prediction is correct
```

Out[207…

```
array(["Communism suffered its first Big Mac attack Thursday as McDonald's opened a restaurant in Yugoslavia, and police were call
ed in to keep customers who lined up for hours from getting too unruly under the golden arches.",
        'I just wanted to taste genuine American hamburgers, said Milica Nikolic, a high school student who waited for three hours
to taste her first Big Mac.',
        "People curiously examined the renovated restaurant's plush interior and the back-lit signs depicting the hamburgers, frenc
h fries, milk shakes and other fare more familiar in the West.",
        'It also featured amber-colored tables and floors, pastel-colored upholstery, modern art paintings and discreet illuminatio
n.',
        'The fast-food outlet, located on a downtown square, had drawn crowds in recent days, and they began gathering long before
it opened Thursday.',
        'Police kept watch on the lines of customers snaking around the block, and they regulated the number who came inside to avo
id overcrowding.',
        'No opening of a restaurant in Belgrade has created such a sensation as this one today, one policeman said.',
        'I think this restaurant has no competition in Belgrade, said Milica Danic, a housewife who treated her son to a cheeseburg
er.',
        'It is much cleaner, the service is faster, the interior is nicer and it is not too expensive.',
        "The Belgrade media have suggested that the success of McDonald's in Yugoslavia depends on its acceptance by citizens long
accustomed to a hamburger-like fast-food dish called the Pljeskavica: ground pork and onions on a bun.",
        "In fact, this is a clash between the Big Mac and Pljeskavica, said Vesna Milosevic, an official of Genex, a Yugoslav state
-run enterprise that has contracted a joint venture agreement with McDonald's.",
        "Our aim is not to destroy the Pljeskavica on the Yugoslav market, said Predrag Dostanic, managing director of the Genex-Mc
Donald's.",
        'We want to change customs of the local people used to completly different eating habits.',
        'He said that lounging at tables for a long time after a finished meal will draw a warning. Also, smoking is forbidden and
alcohol will not be served.',
        'This contrasts sharply with the Balkan and Yugoslav custom of sitting with a drink in smoke-filled restaurants and chattin
g with friends after the meal.',
        'The Big Mac meal, consisting of a hamburger, soft drink and french fries costs the equivalent of 2.57 dollar, or about as
much the similar meal would cost in numerous Pljeskavica joints around town.',
        "Sadik Seljami, a waiter in a small Pljeskavica outlet just a few hundred yards from the McDonald's, suggested that the Ame
rican restaurant wants to drive Yugoslav fast-food outlets out of business.",
        'However, we will not give up the fight even if we have to lower the prices, said Seljami.',
        "Glen Cook, an executive of the McDonald's Corp, said during the opening ceremonies, We are very excited about the opening
of this restaurant, not only because it is the first one in a communist country, but also because it is one of the nicest in Europ
e.",
        "McDonald's and Genex contribute $1 million each for the flagship restaurant.",
        'They will also share the profits equally even though it will be managed entirely by Yugoslavs.',
        'The restaurant has 350 seats and employs 110 people capable of serving 2,500 meals per hour. In an effort to keep a high l
```

```
evel of services, the management is entitled to fire any employees who fail to perform.',
       'The American corporation plans to open five additional restaurants Yugoslavia in the next five years.',
       "The next East European McDonald's, and the first in a Soviet bloc country, is to open next month in Budapest, Hungary."],
      dtype=object)
```

In [208…

```
rp.fuzz.partial_ratio("The next East European McDonald's, and the first in a Soviet bloc country, is to open next month in Budape
```

Out[208… 52.38095238095239

> Compare Other Article Titles ==> **Which provides Lower Values**

In [209…

```
rp.fuzz.partial_ratio("The next East European McDonald's, and the first in a Soviet bloc country, is to open next month in Budapes
```

Out[209… 41.860465116279066

In [210…

```
rp.fuzz.partial_ratio("The next East European McDonald's, and the first in a Soviet bloc country, is to open next month in Budape
```

Out[210… 40.0

> **According to this results Lets Validate the Prediction with Cosin-Similarity**

C) Finding Cosin Similarity

Topics that content needs to identify

Verifying Previous 2 methods results

> Hurricane Gilbert Heads Toward Dominican Coast =>> **Head.iloc[0]**
>
> IRA terrorist attack =>> **Head.iloc[1]**
>
> McDonald's Opens First Restaurant in China **Head.iloc[2]**

Creating Vectorize Vocabulary to identify common Words

In [233…

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics.pairwise import linear_kernel
```

In [236…

```
# opening the text files and copying to variables
def text_string(file_name):
    text = ''
    with open(file_name,"r") as provided_file:
        for line in provided_file:

            # reading each word
            for word in line.split():
                text = text + word + ' '

        return text.strip()
```

In [237…

```
news_1 = text_string(read_files[0])
news_2 = text_string(read_files[1])
news_3 = text_string(read_files[2])
news_4 = text_string(read_files[3])
news_5 = text_string(read_files[4])
news_6 = text_string(read_files[5])
news_7 = text_string(read_files[6])
news_8 = text_string(read_files[7])
```

In [239…

```
# create a corpus by using assigned variables
News_corpus = [news_1, news_2, news_3 ,news_4, news_5, news_6, news_7, news_8]
```

In [241…

```
News_vectorizer = TfidfVectorizer()
vectors = News_vectorizer.fit_transform(News_corpus)

feature_names = News_vectorizer.get_feature_names_out()
print(feature_names, len(feature_names))
```

```
['000' '10' '100' ... 'yugoslavia' 'yugoslavs' 'zone'] 1411
```

Creating Pandas frame for denselist

In [243…

```
for text in feature_names:
    print(text)
```

```
dense = vectors.todense()
denselist = dense.tolist()
News_df = pd.DataFrame(denselist, columns=feature_names)
```

```
000
10
100
11
110
115
12
125
14
140
15
16
17
18
1982
1988
20
200
22
25
250
26
27
30
350
50
500
51
52
57
59
60
67
70
700
72
75
750
80
90
92
ability
about
absolutely
accented
acceptance
accordions
accounted
accustomed
across
actions
actually
adding
additional
advised
affected
after
against
aged
agencies
agency
ago
agony
agreement
aim
air
aircraft
airport
alan
alarm
alcohol
alert
alerted
alex
alien
all
along
already
also
aluminum
am
amber
ambulance
ambulances
america
american
an
and
anonymously
another
antiterrorist
any
anything
apart
appeals
appeared
```

```
appears
approaching
approximating
arches
are
area
areas
arms
army
around
arrived
art
as
asked
asleep
assistance
associate
associated
at
atlantic
atrocity
attack
attacked
attempt
august
average
avoid
awaited
away
baba
back
backing
bag
bakery
balkan
band
bands
bandsmen
bandstand
bang
barahona
barracks
based
bases
bay
be
beaches
bear
bearing
because
bed
bedroom
beeg
been
before
began
behind
being
belgrade
ben
besides
betteridge
between
big
bits
black
blankets
blast
blitz
bloc
block
blood
blow
blowing
blown
boarding
bob
bodies
boiled
bomb
bombing
bombings
bookstore
bound
boy
boys
branches
breakfast
breakfasting
breaking
briefly
brigade
brightened
british
broad
broadcast
broadcasting
broke
brother
brought
```

```
brunt
brush
brushing
buckland
budapest
build
building
buildings
built
bulletins
bun
bundled
bureau
burger
burns
business
but
butterfield
buy
buying
by
cabral
cafeterias
california
called
calling
calls
camaguey
came
cameras
campaign
can
canada
canceled
capable
capital
care
caribbean
carried
cartoon
cash
casualties
casualty
caught
cause
caused
causing
cavalry
cayman
caymans
ceiling
center
central
cents
ceremonies
certainly
chain
chairman
change
channel
characters
charles
chatting
cheaper
cheerful
cheerfully
cheeseburger
cheeseburgers
chickens
chief
children
chizburgers
citizens
city
civil
civilians
claim
claimed
clarendon
clash
cleaner
clearly
clerks
closely
clouded
cloudiness
coast
coastal
coats
cohon
collapse
collapsed
college
colored
coming
communications
communism
communist
companies
company
```

```
competition
completely
completly
concepts
condemning
condition
confirmed
confronted
conscious
consisting
consumer
contents
continue
continued
contracted
contrasts
contribute
cook
coordinated
corp
corporation
cost
costs
costumes
could
countries
country
crash
crates
created
crowds
crush
cuba
cuban
curiously
currency
custom
customer
customers
customs
cut
cutting
cyrillic
dairy
damage
damaged
danced
danger
dangerous
danic
dawn
day
days
de
dead
deal
death
debby
debris
declared
defense
dependency
depends
depicting
described
destroy
destroyed
died
different
directed
director
dirty
disadvantaged
disaster
discontinued
discreet
dish
doctors
dollar
domingo
dominican
donated
door
dostanic
double
dour
dover
down
downed
downgraded
downtown
dozens
draw
drawn
drenched
drink
drive
dublin
dug
dumped
```

```
during
each
earlier
early
east
eastern
eating
edt
edward
efficiency
effort
eight
electrical
electricians
embedded
emergency
employees
employs
en
england
english
enterprise
entirely
entitled
equally
equipment
equivalent
escaped
essential
estimated
eugenio
europe
european
evacuated
even
everyone
everywhere
evidence
examine
examined
exchange
excited
excitedly
executive
expect
expected
expensive
experience
explode
exploded
explosion
extending
eye
facilities
facing
fact
factory
fail
fairy
fall
familiar
fanning
far
fare
fast
faster
fat
feared
featured
february
feeding
feesh
feet
fell
fences
few
fields
fight
figures
filay
filled
finally
find
finished
finland
fire
firefighters
first
fit
five
flag
flags
flagship
flash
flattened
flights
flood
flooded
flooding
floods
```

```
floor
floors
florence
folk
follow
food
for
forbidden
force
forecaster
forecasters
foreign
forensic
formed
found
four
fractures
frantically
french
frequent
friendly
friends
fries
frightened
from
fulfilled
full
fund
fur
gaining
gamburger
gamburgers
gary
gas
gathered
gathering
gave
genex
genuine
george
gerrish
get
getting
gilbert
girls
give
giving
glass
glen
going
golden
good
got
government
grand
granma
great
grins
ground
guantanamo
guaranteed
guerre
gulf
gusting
gusts
guy
habits
hackett
had
haiti
hal
half
hamburger
hamburgers
hammer
hand
handed
happy
hard
hardened
harvest
has
hats
haughey
havana
have
he
head
heading
headquarters
heads
hear
heard
hearing
heather
heavily
heavy
hell
help
her
```

```
high
higher
him
hired
his
hispaniola
hit
hitting
holding
holes
holguin
home
homes
hopes
horrible
hospital
hospitals
hotel
hour
hours
house
household
houses
housewife
how
however
huge
hundred
hundreds
hungary
hurricane
hurricanes
hurt
idea
if
illumination
immediate
immediately
impact
importing
in
inches
included
including
indicate
infectious
information
injured
injuries
injuring
injury
inland
inside
inspecting
inspector
installations
instructions
intense
interests
interior
international
into
investigate
investigators
ira
ireland
irish
is
island
islands
issued
it
its
itself
jamaica
jamaicans
janet
japan
joan
joint
joints
jolt
joshua
juan
july
just
keep
keeping
ken
kent
kept
kevin
killed
killing
king
kingston
kitchen
knew
knightsbridge
knots
```

```
know
land
landmark
large
larger
largest
lash
lashed
last
later
latina
latitude
leading
leak
least
leave
left
lengthwise
lenin
less
let
letters
level
lifting
like
limited
lined
lines
lining
lips
listen
lit
littering
little
live
lobby
local
located
logos
london
long
longitude
look
looked
looks
lot
loud
lounge
lounging
lower
luke
mac
macs
main
mainland
major
mak
making
man
managed
management
managers
managing
many
margaret
marine
marines
marked
market
martin
mary
massive
maximum
may
mcdonald
me
meal
meals
meaning
meanwhile
measure
measures
meat
media
medical
men
message
meteorologist
mexican
mexico
miami
mickey
midday
middle
midnight
miles
milica
military
milk
million
```

```
milosevic
minimal
minimize
minister
ministry
minnock
missing
mitford
modern
moment
monday
monitor
montego
month
months
mood
more
morning
moscow
most
mountains
mouse
move
moved
movement
moves
moving
mph
mrs
much
mud
multiple
muscovite
music
musical
nails
named
narrowly
nation
national
navy
near
nearby
need
neighbor
neighboring
neighbors
neill
nelson
network
news
next
nicer
nicest
night
nikolic
nine
no
nom
non
noon
normally
north
northeast
northern
northward
northwest
not
nothing
now
number
numerous
occurred
ocean
ocho
of
off
office
official
officials
old
on
one
onions
only
onto
open
opened
opening
operates
operation
or
order
orders
orphans
other
our
out
outages
outlet
```

```
outlets
outrage
outside
over
overcrowding
overnight
own
owns
packed
packing
paintings
paper
parade
parish
parishes
park
parked
part
parts
pass
passed
passing
past
pastel
patch
path
pay
peace
penchant
peninsula
pensioner
people
per
percent
perform
permit
person
personnel
phone
picked
pins
place
planned
plans
plant
plaster
platts
play
played
playing
pljeskavica
plush
police
policeman
ponce
popped
popular
populated
pork
port
portland
position
posted
potato
potatoes
power
powerful
precautionary
precautions
preceding
predicted
predrag
prensa
preparation
preparatory
prepare
preparedness
president
press
preventing
previous
priced
prices
prime
private
processing
profits
progress
provide
provided
provides
providing
province
provinces
publicity
puerto
pull
purchases
pushed
pushkin
```

```
put
queue
quite
radio
rain
rains
ran
rang
rate
re
reached
reaching
real
receive
received
recent
recently
recommending
record
recruits
referring
regent
region
registers
regulated
reinblatt
reminded
remnants
removed
renovated
reponsibilty
report
reported
reportedly
reporters
reports
republic
republican
rescue
rescuers
resident
residents
resort
response
responsibility
restaurant
restaurants
returned
returning
rice
rico
rid
right
rios
ripped
risk
roads
rocked
roof
roofs
room
roosevelt
ross
rotating
rots
route
royal
rubble
ruble
rubles
rule
run
running
rush
rushing
russian
sadik
safe
said
sales
san
sandwiches
santiago
santo
saturday
sausage
saw
say
saying
scalping
scared
scene
school
scotland
scream
screamed
screams
seaga
sean
search
```

```
seas
season
seat
seats
second
secretary
seek
seemed
seen
seljami
senior
sensation
sent
separated
serious
seriously
served
service
servicemen
services
serving
seven
several
shaken
shakes
share
sharply
shattered
she
sheets
shock
short
shortly
should
shout
shropshire
sickle
side
sight
sign
signed
signs
similar
simmons
simon
sink
sitting
sixth
skies
skirted
skirting
slabs
slammed
slides
slither
sliver
slop
small
smile
smiling
smoke
smoking
snaking
snarl
so
soft
soldier
soldiers
some
son
songs
soon
sort
south
southeast
southeastern
southern
southwestern
souvenirs
soviet
soviets
speaking
spokesman
sprung
squad
square
squashed
st
staff
staffer
staffers
standing
started
starting
startled
state
statement
states
station
stayed
```

```
steps
stiff
still
stolovaya
stopped
storage
storing
storm
story
straight
straw
street
streets
strength
strengthened
strike
strong
student
stunned
subsided
success
such
suffered
suffering
suggested
sunday
supplies
sure
sustained
sweep
swept
systems
tables
take
taken
taking
tales
target
targets
taste
tasted
team
tearing
technicians
telephone
telephoned
television
tell
ten
terraced
terrific
terrified
than
that
thatcher
the
thebarracks
their
them
then
there
thermal
they
think
thirty
this
thomas
those
though
thought
thousands
threat
three
through
throughout
thursday
time
times
tip
to
today
told
tom
too
took
toppling
tore
torn
toronto
torrential
total
tourist
toward
town
track
traditional
training
trapped
travelers
treated
```

```
treatment
trees
tried
tropical
try
turn
turned
tv
two
unable
unarmed
under
undermine
unfazed
union
united
unruly
until
up
upholstery
uprooted
urged
usage
used
uses
usual
usually
vacationer
vacations
various
ve
veer
venture
version
very
vesna
vice
virgin
visit
visited
visitors
waited
waiter
want
wanted
wants
war
warned
warning
warnings
was
washington
watch
waters
waves
way
we
weather
wednesday
went
were
west
western
westward
what
when
where
whether
which
while
whipping
who
whole
wife
will
window
windows
winds
witch
with
within
without
witnesses
woke
woman
women
wondered
wood
wooden
work
workers
works
world
worldwide
worst
worth
would
wound
wounded
wrappers
```

```
wreckage
wrecked
yaga
yard
yards
year
years
yet
you
young
youthful
yugoslav
yugoslavia
yugoslavs
zone
```

In [251…

```
News_df.dtypes
```

Out[251…

```
000           float64
10            float64
100           float64
11            float64
110           float64
               ...
youthful      float64
yugoslav      float64
yugoslavia    float64
yugoslavs     float64
zone          float64
Length: 1411, dtype: object
```

In [246…

```
News_df.head(20)
```

Out[246…

	000	10	100	11	110	115	12	125	14	140	...	year	years	yet	you	young
0	0.052621	0.060016	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.055325	0.000000	...	0.000000	0.035081	0.000000	0.023184	0.046367
1	0.015023	0.017134	0.034269	0.000000	0.059569	0.000000	0.000000	0.000000	0.000000	0.039713	...	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.061567	0.000000	0.035110	0.000000	0.000000	0.000000	0.048548	0.048548	0.000000	0.040687	...	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.017594	0.035188	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.015426	0.024329	0.000000	0.040778
4	0.000000	0.000000	0.000000	0.027831	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.024401	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.031240	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.043197	0.000000	0.000000	0.036202	0.000000
6	0.033987	0.019382	0.000000	0.000000	0.000000	0.053601	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000	0.025177	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.019049	0.000000	0.000000	0.000000

8 rows × 1411 columns

Count of Total Vocabulary

In [247…

```
print(len(News_vectorizer.vocabulary_))
```

```
1411
```

In [250…

```
News_vectorizer.vocabulary_
```

Out[250…

```
{'thousands': 1266,
 'of': 871,
 'queue': 997,
 'hardened': 575,
 'soviets': 1176,
 'on': 877,
 'wednesday': 1354,
 'cheerfully': 261,
 'lined': 722,
 'up': 1314,
 'to': 1275,
 'get': 537,
 'taste': 1235,
 'gamburgers': 527,
 'chizburgers': 267,
 'and': 87,
 'filay': 475,
 'feesh': 467,
 'sandwiches': 1077,
 'as': 108,
 'mcdonald': 770,
 'opened': 883,
 'in': 636,
 'the': 1251,
 'land': 697,
 'lenin': 714,
 'for': 502,
 'first': 483,
 'time': 1272,
 'world': 1388,
 'largest': 701,
```

```
'version': 1329,
'landmark': 698,
'american': 85,
'fast': 460,
'food': 501,
'chain': 252,
'rang': 1003,
'30': 23,
'000': 0,
'meals': 773,
'27': 22,
'cash': 236,
'registers': 1019,
'breaking': 185,
'opening': 884,
'day': 347,
'record': 1014,
'worldwide': 1389,
'officials': 875,
'said': 1074,
'bundled': 207,
'fur': 524,
'coats': 284,
'hats': 578,
'seemed': 1103,
'unfazed': 1309,
'lining': 724,
'before': 150,
'dawn': 346,
'outside': 898,
'700': 34,
'seat': 1098,
'restaurant': 1041,
'20': 16,
'planned': 938,
'across': 49,
'soviet': 1175,
'union': 1310,
'crush': 327,
'customers': 334,
'was': 1346,
'so': 1160,
'intense': 653,
'company': 295,
'stayed': 1195,
'open': 882,
'until': 1313,
'midnight': 790,
'two': 1304,
'hours': 615,
'later': 705,
'than': 1248,
'only': 880,
'waited': 1337,
'an': 86,
'hour': 614,
'think': 1259,
'they': 1258,
'served': 1112,
'me': 771,
'happy': 573,
'middle': 789,
'aged': 58,
'woman': 1380,
'who': 1367,
'works': 1387,
'at': 114,
'aluminum': 79,
'plant': 940,
'it': 667,
'10': 1,
'rubles': 1065,
'all': 75,
'this': 1261,
'she': 1124,
'taking': 1231,
'back': 126,
'girls': 540,
'factory': 452,
'try': 1300,
'big': 159,
'macs': 748,
'were': 1356,
'priced': 977,
'75': 36,
'double': 380,
'cheeseburgers': 263,
'about': 42,
'pay': 922,
'starting': 1189,
'staffer': 1185,
'or': 887,
'average': 121,
'but': 212,
'much': 825,
'cheaper': 259,
'other': 891,
'private': 980,
'restaurants': 1042,
'that': 1249,
```

'have': 581,
'sprung': 1179,
'recently': 1012,
'official': 874,
'exchange': 436,
'rate': 1004,
'is': 663,
'59': 30,
'dollar': 374,
'per': 928,
'ruble': 1064,
'foreign': 507,
'visitors': 1336,
'can': 227,
'buy': 214,
'16': 11,
'cents': 249,
'each': 397,
'what': 1360,
'currency': 331,
'worth': 1391,
'black': 161,
'market': 764,
'half': 567,
'sales': 1075,
'donated': 377,
'children': 266,
'fund': 523,
'which': 1364,
'provides': 986,
'medical': 780,
'care': 232,
'assistance': 111,
'orphans': 890,
'disadvantaged': 368,
'gary': 528,
'reinblatt': 1021,
'senior': 1106,
'vice': 1332,
'president': 973,
'canada': 228,
'from': 520,
'toronto': 1284,
'built': 204,
'by': 216,
'joint': 675,
'venture': 1328,
'with': 1375,
'city': 269,
'moscow': 814,
'began': 151,
'14': 8,
'years': 1402,
'ago': 61,
'brought': 195,
'52': 28,
'number': 866,
'countries': 321,
'where': 1362,
'operates': 885,
'previous': 976,
'budapest': 200,
'besides': 156,
'its': 668,
'united': 1311,
'states': 1193,
'leading': 708,
'are': 100,
'japan': 673,
'got': 548,
'hand': 571,
'look': 738,
'such': 1217,
'alien': 74,
'concepts': 299,
'efficiency': 405,
'friendly': 516,
'service': 1113,
'normally': 857,
'dour': 381,
'citizens': 268,
'broke': 193,
'into': 657,
'grins': 553,
'caught': 239,
'infectious': 641,
'cheerful': 260,
'mood': 811,
'youthful': 1406,
'staffers': 1186,
'hired': 599,
'their': 1253,
'ability': 41,
'smile': 1154,
'work': 1385,
'hard': 574,
'accordions': 46,
'played': 944,
'folk': 499,
'songs': 1166,

'women': 1381,
'traditional': 1291,
'costumes': 319,
'danced': 342,
'cartoon': 235,
'characters': 256,
'including': 639,
'mickey': 787,
'mouse': 817,
'baba': 125,
'yaga': 1398,
'witch': 1374,
'russian': 1071,
'fairy': 454,
'tales': 1232,
'one': 878,
'muscovite': 828,
'accustomed': 48,
'clerks': 278,
'snarl': 1159,
'if': 630,
'say': 1083,
'anything': 92,
'asked': 109,
'straw': 1206,
'startled': 1190,
'when': 1361,
'smiling': 1155,
'young': 1405,
'found': 510,
'him': 598,
'popped': 951,
'straight': 1205,
'his': 600,
'drink': 391,
'most': 815,
'experience': 443,
'hamburger': 568,
'familiar': 456,
'bag': 128,
'marked': 763,
'golden': 546,
'arches': 99,
'packed': 904,
'wrappers': 1395,
'bearing': 144,
'cyrillic': 338,
'letters': 717,
'approximating': 98,
'gamburger': 526,
'tried': 1298,
'them': 1254,
'handed': 572,
'picked': 935,
'apart': 93,
'examine': 434,
'contents': 307,
'finally': 477,
'squashed': 1182,
'her': 595,
'beeg': 148,
'mak': 752,
'fit': 484,
'lips': 725,
'around': 105,
'tasted': 1236,
'great': 552,
'old': 876,
'boy': 180,
'lot': 741,
'different': 364,
'stolovaya': 1199,
'he': 582,
'continued': 309,
'referring': 1016,
'run': 1067,
'down': 383,
'dirty': 367,
'cafeterias': 218,
'slop': 1152,
'rice': 1045,
'fat': 462,
'boiled': 174,
'sausage': 1081,
'under': 1307,
'sign': 1135,
'accented': 44,
'hammer': 570,
'sickle': 1132,
'flag': 486,
'hundreds': 624,
'long': 736,
'awaited': 123,
'grand': 550,
'am': 80,
'pushkin': 995,
'square': 1181,
'reaching': 1007,
'out': 893,
'excitedly': 438,

```
'flags': 487,
'pins': 936,
'army': 104,
'fulfilled': 521,
'penchant': 924,
'souvenirs': 1174,
'western': 1358,
'logos': 734,
'publicity': 990,
'conscious': 304,
'managers': 757,
'had': 564,
'staff': 1184,
'shout': 1130,
'good': 547,
'morning': 813,
'america': 84,
'english': 416,
'tv': 1303,
'network': 845,
'chairman': 253,
'george': 535,
'cohon': 285,
'man': 754,
'behind': 152,
'deal': 351,
'many': 759,
'people': 927,
'buying': 215,
'multiple': 827,
'orders': 889,
'15': 10,
'just': 681,
'five': 485,
'operation': 886,
'limited': 721,
'purchases': 993,
'customer': 333,
'hopes': 609,
'preventing': 975,
'burger': 209,
'scalping': 1085,
'own': 902,
'bakery': 129,
'dairy': 339,
'meat': 778,
'processing': 981,
'even': 430,
'potato': 959,
'storage': 1201,
'yard': 1399,
'provide': 984,
'guaranteed': 556,
'supplies': 1222,
'country': 322,
'25': 19,
'percent': 929,
'harvest': 576,
'rots': 1060,
'en': 414,
'route': 1061,
'consumer': 306,
'associate': 112,
'wound': 1393,
'importing': 635,
'wooden': 1384,
'crates': 324,
'finland': 480,
'storing': 1202,
'potatoes': 960,
'because': 145,
'went': 1355,
'build': 201,
'there': 1256,
'no': 853,
'wood': 1383,
'nails': 831,
'you': 1404,
'need': 839,
'permit': 931,
'hurricane': 626,
'gilbert': 539,
'packing': 905,
'110': 4,
'mph': 823,
'winds': 1373,
'torrential': 1285,
'rain': 1000,
'moved': 819,
'over': 899,
'capital': 231,
'today': 1276,
'after': 56,
'skirting': 1146,
'puerto': 991,
'rico': 1046,
'haiti': 565,
'dominican': 376,
'republic': 1032,
'immediate': 632,
```

```
'reports': 1031,
'casualties': 237,
'telephone': 1240,
'communications': 291,
'affected': 55,
'right': 1048,
'now': 865,
'actually': 51,
'moving': 822,
'jamaica': 670,
'bob': 172,
'sheets': 1125,
'director': 366,
'national': 835,
'center': 247,
'miami': 786,
'we': 1352,
've': 1326,
'already': 77,
'eastern': 401,
'tip': 1274,
'looks': 740,
'like': 720,
'eye': 448,
'going': 545,
'move': 818,
'lengthwise': 713,
'island': 664,
're': 1005,
'bear': 143,
'full': 522,
'brunt': 196,
'powerful': 962,
'forecasters': 506,
'expected': 441,
'lash': 702,
'throughout': 1270,
'track': 1290,
'strike': 1211,
'cayman': 244,
'islands': 665,
'small': 1153,
'british': 189,
'dependency': 357,
'northwest': 862,
'meanwhile': 775,
'havana': 580,
'radio': 999,
'reported': 1028,
'evacuated': 429,
'guantanamo': 555,
'province': 988,
'cuba': 328,
'southeastern': 1171,
'coast': 282,
'strong': 1212,
'fanning': 457,
'brushing': 198,
'bound': 179,
'flights': 491,
'canceled': 229,
'international': 656,
'airport': 67,
'while': 1365,
'main': 749,
'three': 1268,
'arrived': 106,
'frightened': 519,
'travelers': 1294,
'running': 1068,
'lobby': 731,
'our': 892,
'hotel': 613,
'chickens': 264,
'heads': 586,
'cut': 336,
'off': 872,
'vacationer': 1323,
'returning': 1044,
'home': 607,
'california': 219,
'through': 1269,
'warnings': 1345,
'posted': 958,
'discontinued': 370,
'interests': 654,
'caribbean': 233,
'should': 1129,
'continue': 308,
'monitor': 807,
'progress': 983,
'dangerous': 344,
'adding': 52,
'little': 729,
'change': 254,
'strength': 1209,
'next': 847,
'several': 1118,
'moves': 821,
'westward': 1359,
```

'associated': 113,
'press': 974,
'headquarters': 585,
'san': 1076,
'juan': 679,
'unable': 1305,
'phone': 934,
'calls': 222,
'kingston': 691,
'high': 596,
'heavy': 592,
'preceding': 965,
'storm': 1203,
'drenched': 390,
'overnight': 901,
'toppling': 1281,
'trees': 1297,
'causing': 242,
'local': 732,
'flooding': 494,
'littering': 728,
'streets': 1208,
'branches': 182,
'jamaicans': 671,
'boarding': 171,
'windows': 1372,
'preparation': 969,
'some': 1164,
'companies': 294,
'broadcast': 191,
'appeals': 94,
'technicians': 1239,
'electricians': 409,
'report': 1027,
'weather': 1353,
'bureau': 208,
'predicted': 966,
'140': 9,
'miles': 791,
'southeast': 1170,
'would': 1392,
'pass': 915,
'south': 1169,
'hit': 602,
'southern': 1172,
'parish': 909,
'clarendon': 274,
'flash': 489,
'flood': 492,
'issued': 666,
'parishes': 910,
'portland': 956,
'northeast': 859,
'st': 1183,
'mary': 766,
'north': 858,
'tourist': 1287,
'region': 1018,
'montego': 808,
'bay': 140,
'west': 1357,
'ocho': 870,
'rios': 1049,
'east': 400,
'far': 458,
'impact': 634,
'zone': 1410,
'separated': 1109,
'mountains': 816,
'receive': 1009,
'urged': 1317,
'residents': 1037,
'higher': 597,
'risk': 1051,
'areas': 102,
'along': 76,
'seek': 1102,
'ground': 554,
'certainly': 251,
'larger': 700,
'systems': 1227,
'seen': 1104,
'hal': 566,
'gerrish': 536,
'forecaster': 505,
'maximum': 768,
'sustained': 1224,
'near': 837,
'tropical': 1299,
'force': 504,
'extending': 447,
'250': 20,
'100': 2,
'prime': 979,
'minister': 799,
'edward': 404,
'seaga': 1093,
'alerted': 72,
'government': 549,
'agencies': 59,

```
'saying': 1084,
'sunday': 1221,
'night': 850,
'appears': 96,
'be': 141,
'real': 1008,
'threat': 1267,
'everyone': 431,
'follow': 500,
'instructions': 652,
'precautions': 964,
'office': 873,
'disaster': 369,
'preparedness': 972,
'order': 888,
'minimize': 798,
'danger': 343,
'been': 149,
'gaining': 525,
'passed': 916,
'ocean': 869,
'dumped': 395,
'inches': 637,
'share': 1121,
'hispaniola': 601,
'know': 696,
'within': 1376,
'72': 35,
'whether': 1363,
'major': 751,
'martin': 765,
'nelson': 844,
'another': 89,
'meteorologist': 783,
'17': 12,
'hurricanes': 627,
'take': 1229,
'northward': 861,
'turn': 1301,
'central': 248,
'prensa': 968,
'latina': 706,
'news': 846,
'agency': 60,
'state': 1191,
'alert': 71,
'declared': 355,
'midday': 788,
'cuban': 329,
'provinces': 989,
'holguin': 606,
'santiago': 1078,
'de': 349,
'granma': 551,
'received': 1010,
'mexico': 785,
'civil': 270,
'defense': 356,
'broadcasting': 192,
'bulletins': 205,
'television': 1242,
'recommending': 1013,
'emergency': 411,
'measures': 777,
'providing': 987,
'information': 642,
'stiff': 1197,
'downed': 384,
'power': 961,
'lines': 723,
'caused': 241,
'barahona': 136,
'peninsula': 925,
'then': 1255,
'less': 715,
'neighboring': 841,
'ripped': 1050,
'roofs': 1055,
'houses': 618,
'flooded': 493,
'coastal': 283,
'southwestern': 1173,
'saturday': 1080,
'eugenio': 426,
'cabral': 217,
'parts': 914,
'santo': 1079,
'domingo': 375,
'outages': 894,
'swept': 1226,
'toward': 1288,
'heavily': 591,
'populated': 953,
'prepare': 971,
'rains': 1001,
'seas': 1096,
'approaching': 97,
'gusting': 559,
'92': 40,
'alarm': 69,
```

```
'shortly': 1128,
'closely': 279,
'movement': 820,
'estimated': 425,
'live': 730,
'70': 33,
'125': 7,
'formed': 509,
'strengthened': 1210,
'position': 957,
'latitude': 707,
'longitude': 737,
'67': 32,
'ponce': 950,
'200': 17,
'broad': 190,
'area': 101,
'cloudiness': 281,
'rotating': 1059,
'watch': 1348,
'virgin': 1333,
'least': 710,
'12': 6,
'feet': 468,
'gusts': 560,
'subsided': 1215,
'during': 396,
'florence': 498,
'downgraded': 385,
'remnants': 1023,
'pushed': 994,
'inland': 647,
'gulf': 558,
'returned': 1043,
'find': 478,
'damage': 340,
'80': 38,
'sixth': 1143,
'named': 832,
'1988': 15,
'atlantic': 115,
'season': 1097,
'second': 1100,
'debby': 353,
'reached': 1006,
'minimal': 797,
'briefly': 186,
'hitting': 603,
'mexican': 784,
'last': 704,
'month': 809,
'explosion': 446,
'flattened': 490,
'military': 793,
'barracks': 137,
'tore': 1282,
'nearby': 838,
'homes': 608,
'killing': 689,
'11': 3,
'injuring': 645,
'22': 18,
'police': 948,
'ira': 660,
'claimed': 273,
'responsibility': 1040,
'blast': 163,
'more': 812,
'rescue': 1034,
'workers': 1386,
'frantically': 513,
'dug': 394,
'rubble': 1063,
'story': 1204,
'building': 202,
'collapsed': 287,
'royal': 1062,
'marines': 762,
'school': 1088,
'music': 829,
'stunned': 1214,
'neighbors': 842,
'gathered': 530,
'damaged': 341,
'destroyed': 362,
'chief': 265,
'inspector': 650,
'alan': 68,
'butterfield': 213,
'kent': 685,
'provided': 985,
'casualty': 238,
'figures': 474,
'coordinated': 314,
'effort': 406,
'person': 932,
'missing': 802,
'accounted': 47,
'injured': 643,
'seriously': 1111,
```

```
'hurt': 628,
'terrific': 1246,
'crash': 323,
'reminded': 1022,
'blitz': 164,
'ceiling': 246,
'started': 1188,
'fall': 455,
'pensioner': 926,
'joan': 674,
'betteridge': 157,
'secretary': 1101,
'tom': 1278,
'king': 690,
'inspecting': 649,
'wreckage': 1396,
'not': 863,
'yet': 1403,
'absolutely': 43,
'confirmed': 302,
'bomb': 175,
'evidence': 433,
'quite': 998,
'clearly': 277,
'atrocity': 116,
'installations': 651,
'frequent': 515,
'bombing': 176,
'target': 1233,
'irish': 662,
'republican': 1033,
'campaign': 226,
'rid': 1047,
'northern': 860,
'ireland': 661,
'rule': 1066,
'town': 1289,
'london': 735,
'worst': 1390,
'attack': 117,
'mainland': 750,
'seven': 1117,
'occurred': 868,
'26': 21,
'lounge': 743,
'thebarracks': 1252,
'bands': 132,
'stopped': 1200,
'playing': 945,
'parade': 908,
'ministry': 800,
'spokesman': 1178,
'speaking': 1177,
'anonymously': 88,
'keeping': 683,
'custom': 332,
'dozens': 387,
'four': 511,
'witnesses': 1378,
'hearing': 589,
'away': 124,
'how': 620,
'servicemen': 1114,
'civilians': 271,
'included': 638,
'however': 621,
'told': 1277,
'reporters': 1030,
'directed': 365,
'against': 57,
'unarmed': 1306,
'bandsmen': 133,
'firefighters': 482,
'used': 1319,
'lifting': 719,
'equipment': 421,
'thermal': 1257,
'cameras': 225,
'search': 1095,
'debris': 354,
'fire': 481,
'brigade': 187,
'kevin': 687,
'simmons': 1139,
'ten': 1244,
'doctors': 373,
'giving': 542,
'treatment': 1296,
'scene': 1087,
'ambulances': 83,
'hospitals': 612,
'ambulance': 82,
'statement': 1192,
'telephoned': 1241,
'dublin': 393,
'visited': 1335,
'response': 1039,
'margaret': 760,
'thatcher': 1250,
'visit': 1334,
```

```
 'nine': 852,
 'days': 348,
 'mrs': 824,
 'message': 782,
 'war': 1342,
 'still': 1198,
 'want': 1339,
 'peace': 923,
 'leave': 711,
 'signed': 1136,
 'neill': 843,
 'nom': 854,
 'guerre': 557,
 'usually': 1322,
 'uses': 1320,
 'claim': 272,
 'actions': 50,
 'charles': 257,
 'haughey': 579,
 'condemning': 300,
 'calling': 221,
 'outrage': 897,
 'attempt': 119,
 'february': 465,
 '60': 31,
 'soldiers': 1163,
 'shropshire': 1131,
 'england': 415,
 'exploded': 445,
 'soldier': 1162,
 'killed': 688,
 'wounded': 1394,
 'august': 120,
 'july': 680,
 '1982': 14,
 'eight': 407,
 'died': 363,
 'bombings': 177,
 'household': 617,
 'cavalry': 243,
 'bandstand': 134,
 'regent': 1017,
 'park': 911,
 'band': 131,
 'total': 1286,
 '51': 27,
 'training': 1292,
 'recruits': 1015,
 'play': 943,
 'men': 781,
 'between': 158,
 'based': 138,
 'musical': 830,
 'roof': 1054,
 'janet': 672,
 'minnock': 801,
 'house': 616,
 'torn': 1283,
 'shattered': 1123,
 'has': 577,
 'blown': 170,
 'bits': 160,
 'shaken': 1119,
 'door': 378,
 'neighbor': 840,
 'heather': 590,
 'hackett': 563,
 'standing': 1187,
 'kitchen': 692,
 ...}
```

In [252…

```
# get cosine similarity matrix by using created dataframe
print(cosine_similarity(News_df.values, News_df.values))
```

```
[[1.         0.54082435 0.42279101 0.56849572 0.50821448 0.48433927
  0.5472568  0.59938462]
 [0.54082435 1.         0.68250002 0.61184088 0.53099415 0.53330571
  0.84194483 0.50997942]
 [0.42279101 0.68250002 1.         0.4670546  0.40854181 0.41944037
  0.65355313 0.3833241 ]
 [0.56849572 0.61184088 0.4670546  1.         0.74806744 0.7313034
  0.61104358 0.54061195]
 [0.50821448 0.53099415 0.40854181 0.74806744 1.         0.55023635
  0.53310933 0.45222212]
 [0.48433927 0.53330571 0.41944037 0.7313034  0.55023635 1.
  0.52849343 0.46656382]
 [0.5472568  0.84194483 0.65355313 0.61104358 0.53310933 0.52849343
  1.         0.51410949]
 [0.59938462 0.50997942 0.3833241  0.54061195 0.45222212 0.46656382
  0.51410949 1.        ]]
```

Additional Work done for the Similarity Checking

--

In [150…
```
new_data1.iloc[0]
```
Out[150…
```
Information    Thousands of queue-hardened Soviets on Wednesd...
Name: 0, dtype: string
```
In [157…
```
new_data1.values[0]
```
Out[157…
```
array(["Thousands of queue-hardened Soviets on Wednesday cheerfully lined up to get a taste of ''gamburgers'', ''chizburgers'' and ''Filay-o-feesh'' sandwiches as McDonald's opened in the land of Lenin for the first time."],
      dtype=object)
```
In [193…
```
new_data1.head(10)
```
Out[193…

	Information
0	Thousands of queue-hardened Soviets on Wednesd...
1	The world's largest version of the landmark Am...
2	The Soviets, bundled in fur coats and hats, se...
3	The crush of customers was so intense the comp...
4	I only waited an hour and I think they served ...
5	And it was only 10 rubles for all this, she sa...
6	Big Macs were priced at 3.75 rubles and double...
7	The official exchange rate is 1.59 dollar per ...
8	Half the day's sales were donated to the Sovie...
9	The restaurant, built by the company in a join...

In [194…
```
r1=tfidf_doc1.fit_transform(new_data1.iloc[0],Head.iloc[0])
r1.sum()
```
Out[194…
```
5.059644256269405
```
In [197…
```
for i in range(len(new_data1)):
    r1=tfidf_doc1.fit_transform(new_data1.iloc[i],Head.iloc[0])
    print(r1)
    print(r1.sum())

# print(r1_tot)

#     r1=tfidf_doc1.fit_transform(new_data1.iloc[i],Head.iloc[0])
#     df2 = pd.DataFrame({'info':new_data1['Information'].values[i],
#                        'Cosine Similarity':Head.iloc[0]})
```
```
  (0, 25)	0.15811388300841897
  (0, 6)	0.15811388300841897
  (0, 7)	0.15811388300841897
  (0, 13)	0.15811388300841897
  (0, 12)	0.15811388300841897
  (0, 23)	0.31622776601683794
  (0, 11)	0.15811388300841897
  (0, 18)	0.15811388300841897
  (0, 15)	0.15811388300841897
  (0, 1)	0.15811388300841897
  (0, 20)	0.15811388300841897
  (0, 4)	0.15811388300841897
  (0, 5)	0.15811388300841897
  (0, 0)	0.15811388300841897
  (0, 3)	0.15811388300841897
  (0, 8)	0.15811388300841897
  (0, 22)	0.15811388300841897
  (0, 9)	0.15811388300841897
  (0, 26)	0.15811388300841897
  (0, 27)	0.15811388300841897
  (0, 14)	0.15811388300841897
  (0, 2)	0.15811388300841897
  (0, 28)	0.15811388300841897
  (0, 17)	0.15811388300841897
  (0, 21)	0.15811388300841897
  (0, 10)	0.15811388300841897
  (0, 19)	0.15811388300841897
  (0, 16)	0.4743416490252569
  (0, 24)	0.15811388300841897
5.059644256269405
  (0, 22)	0.16666666666666666
  (0, 16)	0.16666666666666666
  (0, 27)	0.16666666666666666
  (0, 13)	0.16666666666666666
  (0, 10)	0.16666666666666666
  (0, 20)	0.16666666666666666
  (0, 7)	0.16666666666666666
  (0, 18)	0.16666666666666666
  (0, 4)	0.16666666666666666
  (0, 21)	0.16666666666666666
  (0, 5)	0.16666666666666666
  (0, 1)	0.16666666666666666
```

```
  (0, 17)	0.16666666666666666
  (0, 14)	0.16666666666666666
  (0, 0)	0.16666666666666666
  (0, 2)	0.16666666666666666
  (0, 24)	0.16666666666666666
  (0, 19)	0.16666666666666666
  (0, 6)	0.16666666666666666
  (0, 9)	0.16666666666666666
  (0, 8)	0.16666666666666666
  (0, 3)	0.16666666666666666
  (0, 11)	0.16666666666666666
  (0, 15)	0.16666666666666666
  (0, 25)	0.16666666666666666
  (0, 12)	0.16666666666666666
  (0, 26)	0.16666666666666666
  (0, 23)	0.5
4.999999999999999
  (0, 23)	0.15811388300841897
  (0, 19)	0.15811388300841897
  (0, 2)	0.15811388300841897
  (0, 15)	0.15811388300841897
  (0, 0)	0.15811388300841897
  (0, 13)	0.15811388300841897
  (0, 8)	0.15811388300841897
  (0, 16)	0.15811388300841897
  (0, 17)	0.15811388300841897
  (0, 1)	0.15811388300841897
  (0, 14)	0.15811388300841897
  (0, 7)	0.15811388300841897
  (0, 4)	0.15811388300841897
  (0, 24)	0.15811388300841897
  (0, 12)	0.15811388300841897
  (0, 22)	0.15811388300841897
  (0, 18)	0.15811388300841897
  (0, 10)	0.15811388300841897
  (0, 3)	0.15811388300841897
  (0, 6)	0.15811388300841897
  (0, 9)	0.15811388300841897
  (0, 11)	0.15811388300841897
  (0, 5)	0.15811388300841897
  (0, 20)	0.15811388300841897
  (0, 21)	0.6324555320336759
4.427188724235731
  (0, 9)	0.22360679774997896
  (0, 12)	0.22360679774997896
  (0, 5)	0.22360679774997896
  (0, 3)	0.22360679774997896
  (0, 14)	0.22360679774997896
  (0, 6)	0.22360679774997896
  (0, 15)	0.22360679774997896
  (0, 8)	0.22360679774997896
  (0, 11)	0.22360679774997896
  (0, 0)	0.22360679774997896
  (0, 4)	0.22360679774997896
  (0, 10)	0.22360679774997896
  (0, 16)	0.22360679774997896
  (0, 2)	0.22360679774997896
  (0, 7)	0.22360679774997896
  (0, 1)	0.22360679774997896
  (0, 13)	0.4472135954999579
4.024922359499623
  (0, 11)	0.20412414523193154
  (0, 1)	0.20412414523193154
  (0, 4)	0.20412414523193154
  (0, 20)	0.20412414523193154
  (0, 18)	0.20412414523193154
  (0, 19)	0.20412414523193154
  (0, 0)	0.20412414523193154
  (0, 9)	0.20412414523193154
  (0, 6)	0.20412414523193154
  (0, 12)	0.20412414523193154
  (0, 8)	0.20412414523193154
  (0, 5)	0.20412414523193154
  (0, 16)	0.20412414523193154
  (0, 13)	0.20412414523193154
  (0, 14)	0.20412414523193154
  (0, 15)	0.20412414523193154
  (0, 3)	0.20412414523193154
  (0, 7)	0.20412414523193154
  (0, 2)	0.4082482904638631
  (0, 17)	0.20412414523193154
  (0, 10)	0.20412414523193154
4.4907311951024935
  (0, 17)	0.1889822365046136
  (0, 16)	0.1889822365046136
  (0, 5)	0.1889822365046136
  (0, 3)	0.1889822365046136
  (0, 7)	0.1889822365046136
  (0, 14)	0.3779644730092272
  (0, 4)	0.1889822365046136
  (0, 13)	0.1889822365046136
  (0, 11)	0.1889822365046136
  (0, 12)	0.1889822365046136
  (0, 15)	0.1889822365046136
  (0, 1)	0.1889822365046136
  (0, 6)	0.3779644730092272
  (0, 10)	0.1889822365046136
  (0, 0)	0.1889822365046136
  (0, 9)	0.1889822365046136
  (0, 18)	0.1889822365046136
```

```
  (0, 8)	0.3779644730092272
  (0, 2)	0.1889822365046136
4.157609203101499
  (0, 21)	0.15811388300841897
  (0, 32)	0.15811388300841897
  (0, 25)	0.15811388300841897
  (0, 11)	0.15811388300841897
  (0, 29)	0.15811388300841897
  (0, 22)	0.15811388300841897
  (0, 20)	0.15811388300841897
  (0, 17)	0.15811388300841897
  (0, 28)	0.15811388300841897
  (0, 7)	0.15811388300841897
  (0, 15)	0.15811388300841897
  (0, 6)	0.15811388300841897
  (0, 24)	0.15811388300841897
  (0, 4)	0.15811388300841897
  (0, 30)	0.15811388300841897
  (0, 16)	0.15811388300841897
  (0, 26)	0.15811388300841897
  (0, 14)	0.15811388300841897
  (0, 27)	0.15811388300841897
  (0, 10)	0.15811388300841897
  (0, 18)	0.15811388300841897
  (0, 12)	0.15811388300841897
  (0, 31)	0.15811388300841897
  (0, 1)	0.15811388300841897
  (0, 8)	0.15811388300841897
  (0, 9)	0.15811388300841897
  (0, 2)	0.15811388300841897
  (0, 23)	0.31622776601683794
  (0, 0)	0.15811388300841897
  (0, 3)	0.31622776601683794
  (0, 19)	0.15811388300841897
  (0, 33)	0.15811388300841897
  (0, 13)	0.15811388300841897
  (0, 5)	0.15811388300841897
5.69209978830308
  (0, 15)	0.1643989873053573
  (0, 3)	0.1643989873053573
  (0, 17)	0.1643989873053573
  (0, 25)	0.1643989873053573
  (0, 8)	0.1643989873053573
  (0, 24)	0.1643989873053573
  (0, 2)	0.1643989873053573
  (0, 10)	0.1643989873053573
  (0, 7)	0.1643989873053573
  (0, 0)	0.1643989873053573
  (0, 12)	0.1643989873053573
  (0, 21)	0.1643989873053573
  (0, 5)	0.1643989873053573
  (0, 6)	0.1643989873053573
  (0, 23)	0.1643989873053573
  (0, 13)	0.1643989873053573
  (0, 4)	0.1643989873053573
  (0, 20)	0.1643989873053573
  (0, 18)	0.1643989873053573
  (0, 9)	0.1643989873053573
  (0, 1)	0.1643989873053573
  (0, 14)	0.3287979746107146
  (0, 19)	0.1643989873053573
  (0, 11)	0.1643989873053573
  (0, 16)	0.1643989873053573
  (0, 22)	0.4931969619160719
4.767570631855362
  (0, 25)	0.15617376188860607
  (0, 8)	0.15617376188860607
  (0, 19)	0.15617376188860607
  (0, 2)	0.15617376188860607
  (0, 12)	0.15617376188860607
  (0, 14)	0.15617376188860607
  (0, 16)	0.15617376188860607
  (0, 26)	0.15617376188860607
  (0, 21)	0.15617376188860607
  (0, 18)	0.15617376188860607
  (0, 10)	0.15617376188860607
  (0, 6)	0.15617376188860607
  (0, 15)	0.15617376188860607
  (0, 1)	0.15617376188860607
  (0, 0)	0.31234752377721214
  (0, 3)	0.15617376188860607
  (0, 13)	0.15617376188860607
  (0, 17)	0.15617376188860607
  (0, 28)	0.15617376188860607
  (0, 9)	0.15617376188860607
  (0, 4)	0.31234752377721214
  (0, 22)	0.15617376188860607
  (0, 24)	0.31234752377721214
  (0, 7)	0.15617376188860607
  (0, 27)	0.15617376188860607
  (0, 20)	0.15617376188860607
  (0, 5)	0.15617376188860607
  (0, 23)	0.31234752377721214
  (0, 11)	0.15617376188860607
5.153734142324
  (0, 16)	0.15249857033260467
  (0, 12)	0.15249857033260467
  (0, 22)	0.15249857033260467
  (0, 9)	0.15249857033260467
  (0, 14)	0.15249857033260467
```

```
  (0, 1)        0.15249857033260467
  (0, 20)       0.15249857033260467
  (0, 4)        0.15249857033260467
  (0, 2)        0.15249857033260467
  (0, 24)       0.15249857033260467
  (0, 0)        0.15249857033260467
  (0, 3)        0.15249857033260467
  (0, 18)       0.15249857033260467
  (0, 13)       0.15249857033260467
  (0, 15)       0.30499714066520933
  (0, 7)        0.15249857033260467
  (0, 23)       0.15249857033260467
  (0, 21)       0.15249857033260467
  (0, 11)       0.15249857033260467
  (0, 10)       0.15249857033260467
  (0, 8)        0.15249857033260467
  (0, 6)        0.15249857033260467
  (0, 5)        0.15249857033260467
  (0, 17)       0.15249857033260467
  (0, 19)       0.6099942813304187
4.422458539645536
  (0, 10)       0.13483997249264842
  (0, 0)        0.13483997249264842
  (0, 4)        0.13483997249264842
  (0, 1)        0.13483997249264842
  (0, 12)       0.13483997249264842
  (0, 14)       0.13483997249264842
  (0, 13)       0.13483997249264842
  (0, 11)       0.13483997249264842
  (0, 22)       0.13483997249264842
  (0, 24)       0.13483997249264842
  (0, 19)       0.13483997249264842
  (0, 9)        0.13483997249264842
  (0, 2)        0.13483997249264842
  (0, 20)       0.26967994498529685
  (0, 15)       0.26967994498529685
  (0, 5)        0.13483997249264842
  (0, 3)        0.13483997249264842
  (0, 8)        0.40451991747794525
  (0, 25)       0.13483997249264842
  (0, 21)       0.13483997249264842
  (0, 7)        0.13483997249264842
  (0, 18)       0.13483997249264842
  (0, 6)        0.13483997249264842
  (0, 16)       0.13483997249264842
  (0, 17)       0.13483997249264842
  (0, 23)       0.5393598899705937
4.449719092257398
  (0, 20)       0.14907119849998599
  (0, 37)       0.14907119849998599
  (0, 28)       0.14907119849998599
  (0, 36)       0.14907119849998599
  (0, 0)        0.14907119849998599
  (0, 34)       0.14907119849998599
  (0, 14)       0.14907119849998599
  (0, 21)       0.14907119849998599
  (0, 31)       0.14907119849998599
  (0, 29)       0.14907119849998599
  (0, 38)       0.14907119849998599
  (0, 16)       0.14907119849998599
  (0, 25)       0.14907119849998599
  (0, 7)        0.14907119849998599
  (0, 22)       0.14907119849998599
  (0, 33)       0.14907119849998599
  (0, 6)        0.14907119849998599
  (0, 35)       0.14907119849998599
  (0, 18)       0.14907119849998599
  (0, 23)       0.14907119849998599
  (0, 5)        0.14907119849998599
  (0, 8)        0.14907119849998599
  (0, 10)       0.14907119849998599
  (0, 26)       0.14907119849998599
  (0, 27)       0.14907119849998599
  (0, 15)       0.14907119849998599
  (0, 12)       0.14907119849998599
  (0, 2)        0.29814239699997197
  (0, 11)       0.14907119849998599
  (0, 3)        0.29814239699997197
  (0, 9)        0.14907119849998599
  (0, 1)        0.14907119849998599
  (0, 32)       0.14907119849998599
  (0, 4)        0.14907119849998599
  (0, 24)       0.14907119849998599
  (0, 19)       0.14907119849998599
  (0, 13)       0.14907119849998599
  (0, 17)       0.14907119849998599
  (0, 30)       0.14907119849998599
6.111919138499429
  (0, 17)       0.19611613513818404
  (0, 7)        0.19611613513818404
  (0, 15)       0.19611613513818404
  (0, 13)       0.19611613513818404
  (0, 19)       0.19611613513818404
  (0, 22)       0.19611613513818404
  (0, 2)        0.19611613513818404
  (0, 12)       0.19611613513818404
  (0, 11)       0.19611613513818404
  (0, 10)       0.19611613513818404
  (0, 4)        0.19611613513818404
  (0, 3)        0.19611613513818404
```

```
  (0, 20)	0.19611613513818404
  (0, 6)	0.19611613513818404
  (0, 5)	0.19611613513818404
  (0, 18)	0.19611613513818404
  (0, 9)	0.19611613513818404
  (0, 21)	0.19611613513818404
  (0, 1)	0.3922322702763681
  (0, 16)	0.19611613513818404
  (0, 8)	0.19611613513818404
  (0, 14)	0.19611613513818404
  (0, 0)	0.19611613513818404
4.706787243316419
  (0, 7)	0.16222142113076254
  (0, 11)	0.16222142113076254
  (0, 13)	0.16222142113076254
  (0, 23)	0.16222142113076254
  (0, 14)	0.16222142113076254
  (0, 17)	0.16222142113076254
  (0, 10)	0.16222142113076254
  (0, 9)	0.16222142113076254
  (0, 30)	0.16222142113076254
  (0, 21)	0.16222142113076254
  (0, 31)	0.16222142113076254
  (0, 19)	0.16222142113076254
  (0, 28)	0.16222142113076254
  (0, 22)	0.16222142113076254
  (0, 27)	0.16222142113076254
  (0, 2)	0.3244428422615251
  (0, 24)	0.16222142113076254
  (0, 8)	0.16222142113076254
  (0, 4)	0.16222142113076254
  (0, 1)	0.16222142113076254
  (0, 5)	0.16222142113076254
  (0, 3)	0.16222142113076254
  (0, 18)	0.16222142113076254
  (0, 25)	0.16222142113076254
  (0, 12)	0.16222142113076254
  (0, 20)	0.16222142113076254
  (0, 29)	0.16222142113076254
  (0, 6)	0.16222142113076254
  (0, 26)	0.16222142113076254
  (0, 0)	0.16222142113076254
  (0, 15)	0.16222142113076254
  (0, 16)	0.3244428422615251
5.515528318445927
  (0, 11)	0.15811388300841897
  (0, 0)	0.15811388300841897
  (0, 16)	0.15811388300841897
  (0, 6)	0.15811388300841897
  (0, 3)	0.15811388300841897
  (0, 27)	0.15811388300841897
  (0, 19)	0.15811388300841897
  (0, 4)	0.15811388300841897
  (0, 1)	0.15811388300841897
  (0, 12)	0.15811388300841897
  (0, 17)	0.15811388300841897
  (0, 2)	0.15811388300841897
  (0, 8)	0.15811388300841897
  (0, 22)	0.31622776601683794
  (0, 14)	0.31622776601683794
  (0, 21)	0.15811388300841897
  (0, 25)	0.31622776601683794
  (0, 20)	0.15811388300841897
  (0, 13)	0.15811388300841897
  (0, 26)	0.31622776601683794
  (0, 7)	0.15811388300841897
  (0, 9)	0.15811388300841897
  (0, 23)	0.15811388300841897
  (0, 24)	0.15811388300841897
  (0, 15)	0.15811388300841897
  (0, 5)	0.15811388300841897
  (0, 18)	0.15811388300841897
  (0, 10)	0.15811388300841897
5.059644256269404
  (0, 9)	0.16666666666666666
  (0, 1)	0.16666666666666666
  (0, 10)	0.16666666666666666
  (0, 6)	0.16666666666666666
  (0, 11)	0.16666666666666666
  (0, 2)	0.16666666666666666
  (0, 8)	0.3333333333333333
  (0, 15)	0.16666666666666666
  (0, 5)	0.16666666666666666
  (0, 22)	0.16666666666666666
  (0, 23)	0.16666666666666666
  (0, 3)	0.16666666666666666
  (0, 16)	0.16666666666666666
  (0, 4)	0.16666666666666666
  (0, 20)	0.3333333333333333
  (0, 0)	0.16666666666666666
  (0, 14)	0.16666666666666666
  (0, 17)	0.16666666666666666
  (0, 13)	0.16666666666666666
  (0, 7)	0.16666666666666666
  (0, 12)	0.3333333333333333
  (0, 18)	0.16666666666666666
  (0, 21)	0.16666666666666666
  (0, 19)	0.3333333333333333
4.666666666666666
  (0, 5)	0.35355339059327373
```

```
  (0, 1)	0.35355339059327373
  (0, 4)	0.35355339059327373
  (0, 7)	0.35355339059327373
  (0, 0)	0.35355339059327373
  (0, 2)	0.35355339059327373
  (0, 6)	0.35355339059327373
  (0, 3)	0.35355339059327373
2.82842712474619
  (0, 19)	0.19245008972987526
  (0, 1)	0.19245008972987526
  (0, 15)	0.19245008972987526
  (0, 9)	0.19245008972987526
  (0, 0)	0.19245008972987526
  (0, 17)	0.19245008972987526
  (0, 20)	0.19245008972987526
  (0, 23)	0.19245008972987526
  (0, 3)	0.19245008972987526
  (0, 7)	0.19245008972987526
  (0, 8)	0.19245008972987526
  (0, 18)	0.19245008972987526
  (0, 2)	0.19245008972987526
  (0, 4)	0.19245008972987526
  (0, 14)	0.19245008972987526
  (0, 24)	0.19245008972987526
  (0, 25)	0.19245008972987526
  (0, 16)	0.19245008972987526
  (0, 21)	0.19245008972987526
  (0, 26)	0.19245008972987526
  (0, 5)	0.19245008972987526
  (0, 11)	0.19245008972987526
  (0, 22)	0.19245008972987526
  (0, 10)	0.19245008972987526
  (0, 6)	0.19245008972987526
  (0, 13)	0.19245008972987526
  (0, 12)	0.19245008972987526
5.19615242270663
  (0, 22)	0.10425720702853739
  (0, 41)	0.10425720702853739
  (0, 42)	0.10425720702853739
  (0, 35)	0.10425720702853739
  (0, 29)	0.10425720702853739
  (0, 15)	0.10425720702853739
  (0, 5)	0.10425720702853739
  (0, 10)	0.10425720702853739
  (0, 18)	0.10425720702853739
  (0, 6)	0.10425720702853739
  (0, 30)	0.10425720702853739
  (0, 13)	0.10425720702853739
  (0, 24)	0.10425720702853739
  (0, 11)	0.10425720702853739
  (0, 28)	0.10425720702853739
  (0, 32)	0.10425720702853739
  (0, 37)	0.10425720702853739
  (0, 31)	0.10425720702853739
  (0, 26)	0.10425720702853739
  (0, 2)	0.10425720702853739
  (0, 0)	0.10425720702853739
  (0, 7)	0.10425720702853739
  (0, 27)	0.10425720702853739
  (0, 17)	0.10425720702853739
  (0, 8)	0.10425720702853739
  (0, 23)	0.10425720702853739
  (0, 14)	0.3127716210856122
  (0, 40)	0.10425720702853739
  (0, 21)	0.10425720702853739
  (0, 20)	0.10425720702853739
  (0, 12)	0.10425720702853739
  (0, 33)	0.10425720702853739
  (0, 3)	0.20851441405707477
  (0, 19)	0.10425720702853739
  (0, 36)	0.20851441405707477
  (0, 9)	0.10425720702853739
  (0, 1)	0.10425720702853739
  (0, 4)	0.10425720702853739
  (0, 16)	0.10425720702853739
  (0, 25)	0.10425720702853739
  (0, 34)	0.10425720702853739
  (0, 38)	0.6255432421712244
  (0, 39)	0.10425720702853739
5.421374765483943
  (0, 12)	0.22941573387056174
  (0, 18)	0.22941573387056174
  (0, 1)	0.22941573387056174
  (0, 2)	0.22941573387056174
  (0, 6)	0.22941573387056174
  (0, 14)	0.22941573387056174
  (0, 3)	0.22941573387056174
  (0, 5)	0.22941573387056174
  (0, 9)	0.22941573387056174
  (0, 0)	0.22941573387056174
  (0, 11)	0.22941573387056174
  (0, 7)	0.22941573387056174
  (0, 15)	0.22941573387056174
  (0, 16)	0.22941573387056174
  (0, 17)	0.22941573387056174
  (0, 8)	0.22941573387056174
  (0, 10)	0.22941573387056174
  (0, 4)	0.22941573387056174
  (0, 13)	0.22941573387056174
4.358898943540672
```

```
  (0, 21)	0.13608276348795434
  (0, 13)	0.13608276348795434
  (0, 11)	0.13608276348795434
  (0, 10)	0.13608276348795434
  (0, 15)	0.13608276348795434
  (0, 14)	0.13608276348795434
  (0, 2)	0.13608276348795434
  (0, 28)	0.13608276348795434
  (0, 0)	0.2721655269759087
  (0, 1)	0.13608276348795434
  (0, 26)	0.13608276348795434
  (0, 24)	0.13608276348795434
  (0, 3)	0.13608276348795434
  (0, 22)	0.13608276348795434
  (0, 19)	0.13608276348795434
  (0, 5)	0.13608276348795434
  (0, 29)	0.13608276348795434
  (0, 23)	0.2721655269759087
  (0, 17)	0.13608276348795434
  (0, 25)	0.13608276348795434
  (0, 9)	0.13608276348795434
  (0, 4)	0.13608276348795434
  (0, 16)	0.13608276348795434
  (0, 27)	0.5443310539518174
  (0, 8)	0.13608276348795434
  (0, 12)	0.13608276348795434
  (0, 7)	0.13608276348795434
  (0, 6)	0.13608276348795434
  (0, 20)	0.2721655269759087
  (0, 18)	0.13608276348795434
4.898979485566357
  (0, 13)	0.25
  (0, 2)	0.25
  (0, 10)	0.25
  (0, 8)	0.25
  (0, 4)	0.25
  (0, 5)	0.25
  (0, 3)	0.25
  (0, 9)	0.25
  (0, 7)	0.25
  (0, 1)	0.25
  (0, 0)	0.25
  (0, 15)	0.25
  (0, 11)	0.25
  (0, 6)	0.25
  (0, 12)	0.25
  (0, 14)	0.25
4.0
  (0, 4)	0.1414213562373095
  (0, 25)	0.1414213562373095
  (0, 7)	0.1414213562373095
  (0, 24)	0.1414213562373095
  (0, 11)	0.1414213562373095
  (0, 28)	0.282842712474619
  (0, 17)	0.1414213562373095
  (0, 19)	0.1414213562373095
  (0, 0)	0.1414213562373095
  (0, 30)	0.1414213562373095
  (0, 31)	0.1414213562373095
  (0, 5)	0.1414213562373095
  (0, 12)	0.1414213562373095
  (0, 27)	0.1414213562373095
  (0, 10)	0.1414213562373095
  (0, 23)	0.1414213562373095
  (0, 29)	0.4242640687119285
  (0, 32)	0.1414213562373095
  (0, 26)	0.1414213562373095
  (0, 21)	0.1414213562373095
  (0, 8)	0.1414213562373095
  (0, 1)	0.1414213562373095
  (0, 20)	0.1414213562373095
  (0, 22)	0.1414213562373095
  (0, 16)	0.1414213562373095
  (0, 6)	0.1414213562373095
  (0, 2)	0.1414213562373095
  (0, 13)	0.1414213562373095
  (0, 9)	0.1414213562373095
  (0, 18)	0.282842712474619
  (0, 14)	0.282842712474619
  (0, 3)	0.1414213562373095
  (0, 15)	0.1414213562373095
5.374011537017762
  (0, 12)	0.16222142113076254
  (0, 0)	0.16222142113076254
  (0, 26)	0.16222142113076254
  (0, 13)	0.3244428422615251
  (0, 23)	0.16222142113076254
  (0, 19)	0.16222142113076254
  (0, 8)	0.16222142113076254
  (0, 3)	0.16222142113076254
  (0, 21)	0.16222142113076254
  (0, 24)	0.16222142113076254
  (0, 20)	0.3244428422615251
  (0, 25)	0.16222142113076254
  (0, 2)	0.16222142113076254
  (0, 15)	0.16222142113076254
  (0, 17)	0.16222142113076254
  (0, 7)	0.16222142113076254
  (0, 6)	0.16222142113076254
  (0, 9)	0.16222142113076254
```

```
  (0, 5)	0.3244428422615251
  (0, 27)	0.16222142113076254
  (0, 10)	0.16222142113076254
  (0, 22)	0.16222142113076254
  (0, 28)	0.16222142113076254
  (0, 4)	0.16222142113076254
  (0, 18)	0.16222142113076254
  (0, 16)	0.16222142113076254
  (0, 1)	0.16222142113076254
  (0, 11)	0.16222142113076254
  (0, 14)	0.16222142113076254
5.191085476184403
  (0, 2)	0.35355339059327373
  (0, 0)	0.35355339059327373
  (0, 6)	0.35355339059327373
  (0, 4)	0.35355339059327373
  (0, 3)	0.35355339059327373
  (0, 7)	0.35355339059327373
  (0, 1)	0.35355339059327373
  (0, 5)	0.35355339059327373
2.82842712474619
```

In [200…

```
# for i in range(len(new_data1)):
#     r2=tfidf_doc1.fit_transform(new_data1.iloc[i],Head.iloc[2])
#     print(r2)
#     print(r2.sum())
```

In [50]:

```
News_head1 = ["Hurricane Gilbert Heads Toward Dominican Coast"]
News_head2 = ["IRA terrorist attack"]
News_head3= ["McDonald's Opens First Restaurant in China"]

# Create the pandas DataFrame with column name is provided explicitly
Headlines1 = pd.DataFrame(News_head1, columns=['Headline'])
Headlines2 = pd.DataFrame(News_head2, columns=['Headline'])
Headlines3 = pd.DataFrame(News_head3, columns=['Headline'])
```

Text file1 headline Revealing

In [51]:

```
Head1_title1=Headlines1.append([Headlines1]*24,ignore_index=True)
Head1_title2=Headlines2.append([Headlines2]*24,ignore_index=True)
Head1_title3=Headlines3.append([Headlines3]*24,ignore_index=True)
```

In [52]:

```
# Merge default pandas DataFrame without any key column
Head1_title1 = pd.concat([Head1_title1,new_data1], join = 'outer', axis = 1)
Head1_title2 = pd.concat([Head1_title2,new_data1], join = 'outer', axis = 1)
Head1_title3 = pd.concat([Head1_title3,new_data1], join = 'outer', axis = 1)
```

In [53]:

```
Head1_title3.head(5)
```

Out[53]:

	Headline	Information
0	McDonald's Opens First Restaurant in China	Thousands queue-hardened Soviets Wednesday cheerfully lined get taste `` gamburgers '' , `` chizburgers '' `` Filay-o-feesh '' sandwich McDonald 's opened land Lenin first time .
1	McDonald's Opens First Restaurant in China	The world 's large version landmark American fast-food chain rang 30,000 meal 27 cash register , breaking opening-day record McDonald 's worldwide , official said .
2	McDonald's Opens First Restaurant in China	The Soviets , bundled fur coat hat , seemed unfazed , lining dawn outside 700 seat restaurant , first 20 planned across Soviet Union .
3	McDonald's Opens First Restaurant in China	The crush customer intense company stayed open midnight , two hour late planned .
4	McDonald's Opens First Restaurant in China	I waited hour I think served thousand , said happy middle-aged woman work aluminum plant .

In [54]:

```
frames = [Head1_title1 , Head1_title2 , Head1_title3]
Doc1 = pd.concat(frames)
```

In [55]:

```
Doc1.shape
Doc1
```

Out[55]:

	Headline	Information
0	Hurricane Gilbert Heads Toward Dominican Coast	Thousands queue-hardened Soviets Wednesday cheerfully lined get taste `` gamburgers '' , `` chizburgers '' `` Filay-o-feesh '' sandwich McDonald 's opened land Lenin first time .
1	Hurricane Gilbert Heads Toward Dominican Coast	The world 's large version landmark American fast-food chain rang 30,000 meal 27 cash register , breaking opening-day record McDonald 's worldwide , official said .
2	Hurricane Gilbert Heads Toward Dominican Coast	The Soviets , bundled fur coat hat , seemed unfazed , lining dawn outside 700 seat restaurant , first 20 planned across Soviet Union .
3	Hurricane Gilbert Heads Toward Dominican Coast	The crush customer intense company stayed open midnight , two hour late planned .

	Headline	Information
4	Hurricane Gilbert Heads Toward Dominican Coast	I waited hour I think served thousand , said happy middle-aged woman work aluminum plant .
...	...	...
20	McDonald's Opens First Restaurant in China	McDonald 's Canada Chairman George Cohon , man behind deal , said many people buying multiple order restaurant served 15,000 20,000 people first five hour operation .
21	McDonald's Opens First Restaurant in China	The restaurant limited purchase 10 Big Macs per customer hope preventing burger scalping .
22	McDonald's Opens First Restaurant in China	McDonald 's built factory , including bakery , dairy , meat processing plant even potato storage yard , provide guaranteed supply country 25 percent harvest rot en route consumer .
23	McDonald's Opens First Restaurant in China	One McDonald 's associate said company wound importing wooden crate Finland storing potato went build crate , found wood , nail .
24	McDonald's Opens First Restaurant in China	They found need permit buy nail .

75 rows × 2 columns

In [56]:
```
Doc1['Headline'] = Doc1['Headline'].str.replace('\d+', '',regex=True)
```

In [59]:
```
Doc1[['Headline','Information']] = Doc1[['Headline','Information']].astype('string')
```

In [106…
```
Doc1.dtypes
```

Out[106…
```
Headline       string
Information    string
dtype: object
```

In [107…
```
Doc1.head(5)
```

Out[107…

	Headline	Information
0	Hurricane Gilbert Heads Toward Dominican Coast	Thousands queue-hardened Soviets Wednesday cheerfully lined get taste `` gamburgers '' , `` chizburgers '' `` Filay-o-feesh '' sandwich McDonald 's opened land Lenin first time .
1	Hurricane Gilbert Heads Toward Dominican Coast	The world 's large version landmark American fast-food chain rang 30,000 meal 27 cash register , breaking opening-day record McDonald 's worldwide , official said .
2	Hurricane Gilbert Heads Toward Dominican Coast	The Soviets , bundled fur coat hat , seemed unfazed , lining dawn outside 700 seat restaurant , first 20 planned across Soviet Union .
3	Hurricane Gilbert Heads Toward Dominican Coast	The crush customer intense company stayed open midnight , two hour late planned .
4	Hurricane Gilbert Heads Toward Dominican Coast	I waited hour I think served thousand , said happy middle-aged woman work aluminum plant .

In [101…
```
headline_vectorizer = CountVectorizer()
# headline_features   = headline_vectorizer.fit_transform(Doc1['Headline'])
```

In [80]:
```
headline_features.get_shape()
```

Out[80]:
```
(75, 15)
```

Using Bag of Words method

> Bag of Word model not provides the expected level of accuracy for similarity checking and it's being neglected

A **Bag of Words(BoW)** method represents the occurence of words within a **document**. Here, each headline can be considered as a **document** and set of all headlines form a **corpus**.

Using **BoW** approach, each **document** is represented by a **d-dimensional** vector, where **d** is total number of **unique words** in the corpus. The set of such unique words forms the **Vocabulary**.

In [99]:
```
# def bag_of_words_based_model(row_index, num_similar_items):
#     couple_dist = pairwise_distances(headline_features,headline_features[row_index])
#     indices = np.argsort(couple_dist.ravel())[0:num_similar_items]
#     df = pd.DataFrame({'Information':Head1_title1['Information'][indices].values,
#                'Euclidean similarity with the queried article': couple_dist[indices].ravel()})
#     print("="*30,"Queried article details","="*30)
#     print('headline : ',Doc1['Headline'][indices[1]])
#     print("\n","="*25,"Recommended articles : ","="*23)
#     #return df.iloc[1:,1]
#     return df.iloc[1:,]

# # bag_of_words_based_model(20, 10) # Change the row index for any other queried article
```

```
# name=input('News Title For Recommendation :')
# clear_output()
# ind=Doc1[Doc1['Headline']==name].index[0]
# dd=bag_of_words_based_model(ind, 20)
# dd.head(10)  # Change the row index for any other queried article
```

In [39]:

```
text1 = new_data1.to_numpy()
process.extract("Hurricane Gilbert Heads Toward Dominican Coast",  new_data1, scorer=fuzz.ratio)
```

Out[39]:

```
[(0     Thousands of queue-hardened Soviets on Wednesday cheerfully lined up to get a taste of ''gamburgers'', ''chizburgers'' and
''Filay-o-feesh'' sandwiches as McDonald's opened in the land of Lenin for the first time.
  1     The world's largest version of the landmark American fast-food chain rang up 30,000 meals on 27 cash registers, breaking t
he opening-day record for McDonald's worldwide, officials said.
  2     The Soviets, bundled in fur coats and hats, seemed unfazed, lining up before dawn outside the 700 seat restaurant, the fir
st of 20 planned across the Soviet Union.
  3     The crush of customers was so intense the company stayed open until midnight, two hours later than planned.
  4     I only waited an hour and I think they served thousands before me, said a happy middle-aged woman who works at an aluminum
plant.
  5     And it was only 10 rubles for all this, she said. I'm taking it back for the girls at the factory to try.
  6     Big Macs were priced at 3.75 rubles and double cheeseburgers at 3 rubles about two hours' pay for a starting McDonald's st
affer or the average Soviet, but much cheaper than other private restaurants that have sprung up recently.
  7     The official exchange rate is 1.59 dollar per ruble but foreign visitors can buy rubles for 16 cents each, about what the
currency is worth on the black market.
  8     Half the day's sales were donated to the Soviet Children's Fund, which provides medical care and assistance to orphans and
disadvantaged children, Gary Reinblatt, senior vice president of McDonald's Canada, said from Toronto.
  9     The restaurant, built by the company in a joint venture with the city of Moscow that began 14 years ago, brought to 52 the
number of countries where McDonald's operates.
  10    The previous opening-day record for sales was in Budapest, company officials said. Besides its restaurants in the United S
tates, the leading number of McDonald's are in Canada and Japan, the officials said.
  11    Soviets got a first-hand look at such alien concepts as efficiency and fast, friendly service. Normally dour citizens brok
e into grins as they caught the infectious cheerful mood from youthful Soviet staffers hired for their ability to smile and work h
ard.
  12    Accordions played folk songs and women in traditional costumes danced with cartoon characters, including Mickey Mouse and
Baba Yaga, a witch of Russian fairy tales.
  13    One Muscovite, accustomed to clerks who snarl if they say anything at all, asked for a straw and was startled when a smili
ng young Soviet woman found him one and popped it straight into his drink.
  14    For most customers, it was their first experience with a hamburger. Sandwiches were served in the familiar bag marked with
the golden arches, but were packed in wrappers bearing Cyrillic letters, approximating ``gamburger.''
  15    They tried them one-handed.They picked their sandwiches apart to examine the contents. One young woman finally squashed he
r ``Beeg Mak'' to fit her lips around it.
  16    ''It tasted great!'' a 14 years old boy said.
  17    It's a lot different from a stolovaya,'' he continued with a smile, referring to the much cheaper but run down dirty cafet
erias that slop rice and fat or boiled sausage.
  18    Under the sign of the golden arches, accented by the Soviet hammer and sickle flag, hundreds lined up for the long awaited
grand opening at 10 am on Pushkin Square, reaching out excitedly for McDonald's flags and pins as the hamburger chain's army fulfi
lled the Soviet penchant for souvenirs with Western logos.
  19    Publicity conscious managers had the staff shout ''Good morning, America!'' in English and Russian, for an American TV net
work.
  20    McDonald's of Canada Chairman George Cohon, the man behind the deal, said many people were buying multiple orders and the
restaurant served 15,000 to 20,000 people in just the first five hours of operation.
  21    The restaurant limited purchases to 10 Big Macs per customer in hopes of preventing burger scalping.
  22    McDonald's built its own factory, including bakery, dairy, meat processing plant and even potato storage yard, to provide
its own guaranteed supplies in a country where up to 25 percent of the harvest rots en route to the consumer.
  23    One McDonald's associate said the company wound up importing wooden crates from Finland for storing potatoes because when
they went to build crates, they found there was no wood, and no nails.
  24    They found you need a permit to buy nails.
  Name: Information, dtype: string,
  0.0,
  'Information')]
```

In [ ]:

In [ ]: