Project Report

Text Summarization

Team: Hari Kiran Reddy Mudipalli and Dharani Doppalapudi

Problem Statement:

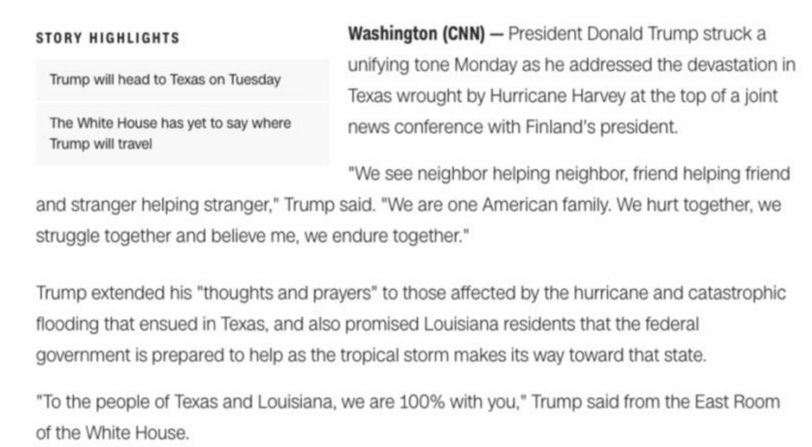
Highlights are a good low-level overview of a news article, but they are not a concrete synopsis. We hope to build a summary generating system with this project that can convey the most relevant information in a few sentences. This project can help two sorts of users: those who want to digest a particle news item as soon as possible, and those who want to follow news via digital media but are concerned about gazing at a screen for an extended period. This program benefits users who have the means and ability to utilize the internet to read news items. This initiative might serve as a supplement to a digital news publication.

Users will navigate through summaries of digital news items on the User Interface. If the user wishes to learn more about the story after reading the summary, they can click on the article to continue reading. We do not see any concerns with this app's trust or privacy at this time because the user selects whether to read the complete narrative based on the summary supplied.

Data Collection:

The data is downloaded from the DeepMind Q&A Dataset at this link (https://cs.nyu.edu/~kcho/DMQA/). This dataset contains both story and story highlights which were already scrapped from the website which has both the main story and a story highlight as shown below.

Structure of CNN story:



STORY HIGHLIGHTS

Trump will head to Texas on Tuesday

The White House has yet to say where Trump will travel

Washington (CNN) — President Donald Trump struck a unifying tone Monday as he addressed the devastation in Texas wrought by Hurricane Harvey at the top of a joint news conference with Finland's president.

"We see neighbor helping neighbor, friend helping friend and stranger helping stranger," Trump said. "We are one American family. We hurt together, we struggle together and believe me, we endure together."

Trump extended his "thoughts and prayers" to those affected by the hurricane and catastrophic flooding that ensued in Texas, and also promised Louisiana residents that the federal government is prepared to help as the tropical storm makes its way toward that state.

"To the people of Texas and Louisiana, we are 100% with you," Trump said from the East Room of the White House.

Data Management:

How did you clean and pre-process the data?

Text data is preprocessed involving the following steps:

- Removing punctuation
- Removing stop words (words that do not add meaning at word level analysis)
- Lemmatization (converts different forms of a single word into one form)

What recording of the variables was done after the data was collected?

- The data essential contained 2 parts, the story (text of a news article) and its highlights.
- After loading the data, a pandas data frame is created with 2 columns for the story and highlights.
- For further analysis, all the story rows are preprocessed using the steps mentioned above and a column of the word count of each story is added to the data frame along with the cleaned stories.

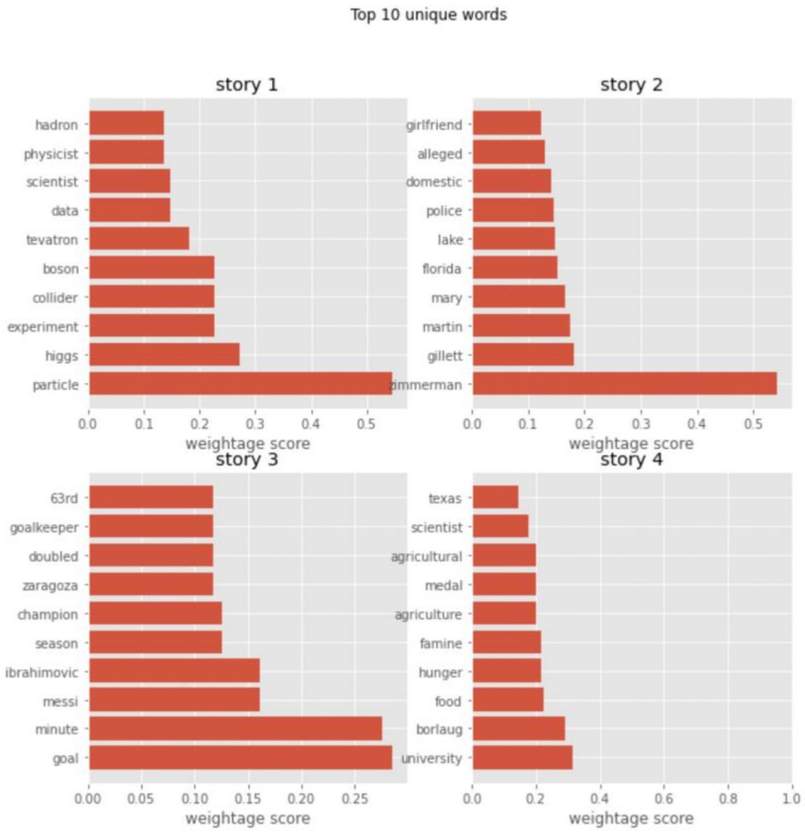
Was any of the data manipulated manually after it was collected?

- Data were preprocessed using pre-existing libraries and a few manual preprocessing which we thought were necessary like removing text inside brackets.

Data Analysis:

- At this stage of the project, we are using story highlights to compare our generated summaries.
- We are particularly focusing on word-level analysis.

- Implemented TF-IDF to look for unique words.
- A subset of 100 stories was taken for quick analysis.
- We used cosine similarity to check how similar the generated text is with highlights. A decent score of about 80% (rough average) is observed on word-level summaries. For future work, we would like to improve this system using context-based sentence ranking to rank high for more relevant sentences in the story.

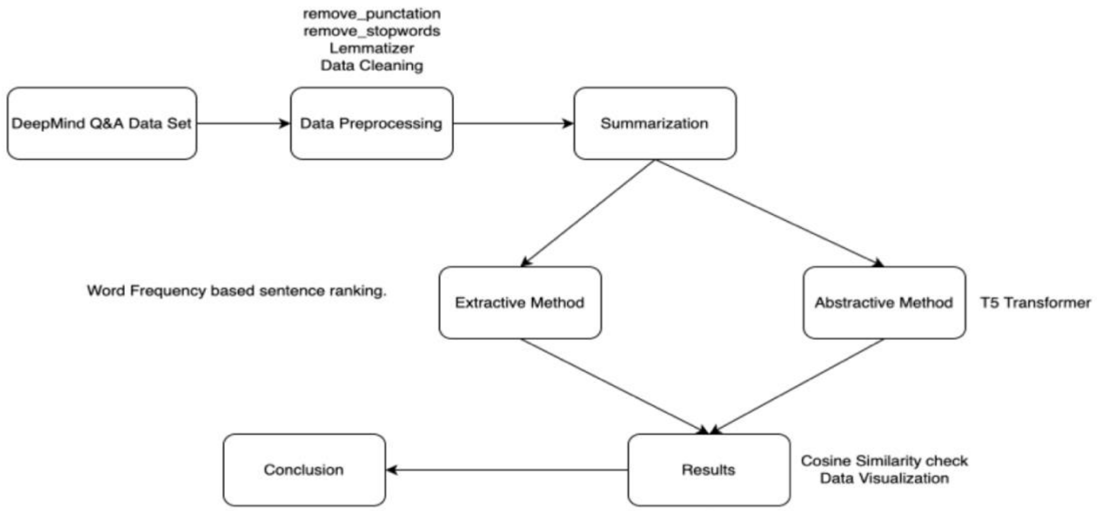


In this visualization, we could see the 10 most unique elements of all four stories. This could help us understand the primary topic concerning the story. I think this feature can extend in categorizing the stories and making the search for news stories easy.

Argument:

- With respect to this project, as the data is text and summaries are generated, the uniqueness of words pertaining to each story might be key in generating summaries.
- This would not be a causal argument since there is no supporting evidence to prove it. It is just intuition in this type of problems

Design:



- The above figure shows the design flow of this project.
- Data from the deep mind is preprocessed first and then it is used for generating summaries using 2 methods, Extractive and Abstractive.
- The results are compared to decide which method is effective for this data.

Intervention:

- Our design flow is expected to produce a proper summary given a large text as input with a target output being a grammatically correct text condensed text of key points.
- But we are least expecting it to give good results on content-based evaluation.
- This is a text-in, text-out design of delivery.
- We are using a content-based evaluation metric, the cosine similarity score for measuring the results.

Stakeholder Analysis:

- News articles have already been digitalized. This project can benefit 2 types of users, one who wants to consume a particle news story as quickly as possible and one who follows news through the digital medium but also worries about staring at the screen for a long time.
- Users with resources and the ability to use the internet to browse through news stories benefit from this application.

Context:

- This project could be a subpart of a digital newspaper. Users will scroll through the User Interface for summaries of digital newspaper stories. User can choose to continue reading by clicking on the article if the user wants to know more about the story after reading the summary.

- At this point, we do not see any trust or privacy issues with this application because it is the user who decides whether to continue reading the full story based on the summary provided.

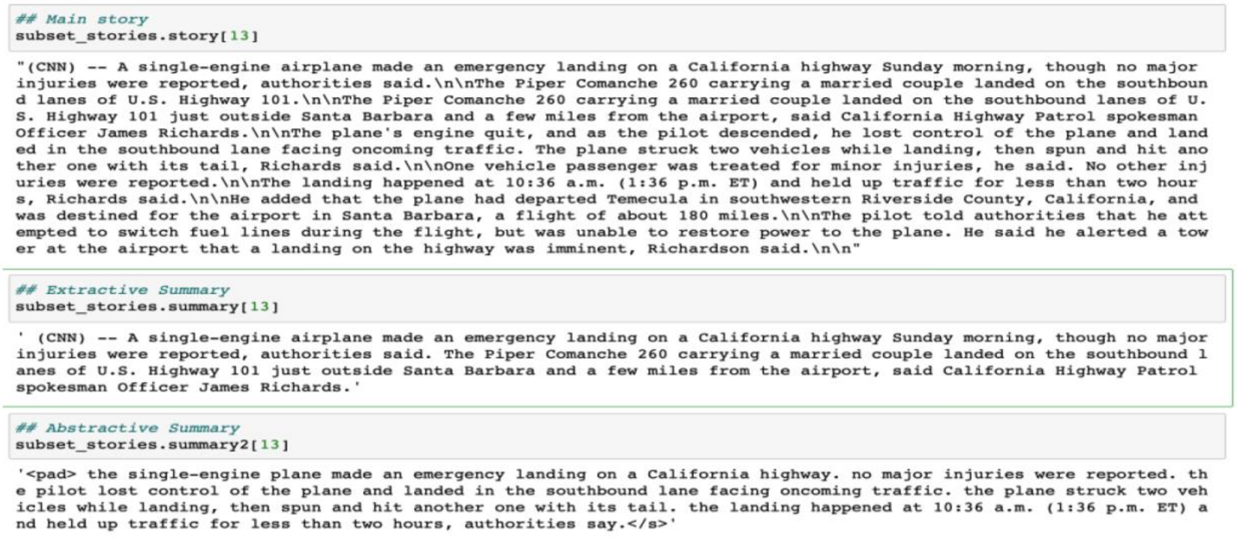
Ethics:

- One of the purposes of this project could be to save users time. By looking at the summary, the user gets a brief idea of the topic.
- Though modern NLP (Natural Language Processing) algorithms are performing better in understanding the context of a given text, they are far from understanding the human intentions behind indirect speeches. This could result in an inappropriate summary.
- For such reasons, it is advisable to take the computer-generated summaries with a grain of salt. The intention behind generating these summaries is to get a shorter form of original stories. At the current level of the project, the sentences in the summary are selected from the original text.

Results & observations:

We used story highlights as a reference to compare with the generated summaries. We observed that Abstractive summaries showed **~83%** average similarity score whereas the Extractive method was able to achieve **~75%** average similarity score.

The figure below shows the comparison between the actual story and summaries generated using Extractive and Abstractive methods respectively. It can be observed that the sentences generated by the Extractive method are a collection from the original text/story whereas, in the Abstractive summary, the text is newly generated.



```
## Main story
subset_stories.story[13]
```

```
"(CNN) -- A single-engine airplane made an emergency landing on a California highway Sunday morning, though no major
injuries were reported, authorities said.\n\nThe Piper Comanche 260 carrying a married couple landed on the southboun
d lanes of U.S. Highway 101.\n\nThe Piper Comanche 260 carrying a married couple landed on the southbound lanes of U.
S. Highway 101 just outside Santa Barbara and a few miles from the airport, said California Highway Patrol spokesman
Officer James Richards.\n\nThe plane's engine quit, and as the pilot descended, he lost control of the plane and land
ed in the southbound lane facing oncoming traffic. The plane struck two vehicles while landing, then spun and hit ano
ther one with its tail, Richards said.\n\nOne vehicle passenger was treated for minor injuries, he said. No other inj
uries were reported.\n\nThe landing happened at 10:36 a.m. (1:36 p.m. ET) and held up traffic for less than two hour
s, Richards said.\n\nHe added that the plane had departed Temecula in southwestern Riverside County, California, and
was destined for the airport in Santa Barbara, a flight of about 180 miles.\n\nThe pilot told authorities that he att
empted to switch fuel lines during the flight, but was unable to restore power to the plane. He said he alerted a tow
er at the airport that a landing on the highway was imminent, Richardson said.\n\n"
```

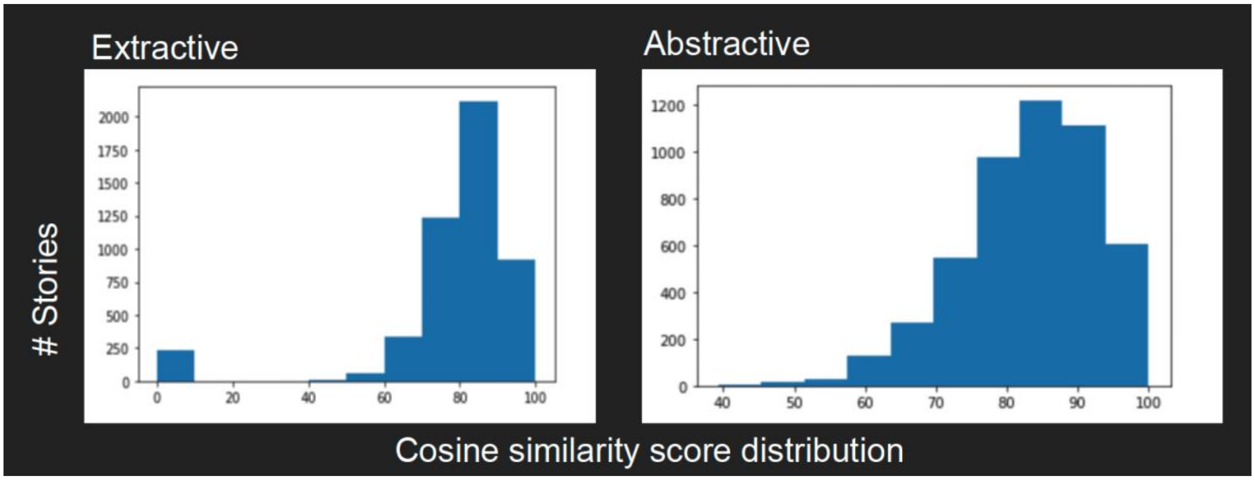
```
## Extractive Summary
subset_stories.summary[13]
```

```
' (CNN) -- A single-engine airplane made an emergency landing on a California highway Sunday morning, though no major
injuries were reported, authorities said. The Piper Comanche 260 carrying a married couple landed on the southbound l
anes of U.S. Highway 101 just outside Santa Barbara and a few miles from the airport, said California Highway Patrol
spokesman Officer James Richards.'
```

```
## Abstractive Summary
subset_stories.summary2[13]
```

```
'<pad> the single-engine plane made an emergency landing on a California highway. no major injuries were reported. th
e pilot lost control of the plane and landed in the southbound lane facing oncoming traffic. the plane struck two veh
icles while landing, then spun and hit another one with its tail. the landing happened at 10:36 a.m. (1:36 p.m. ET) a
nd held up traffic for less than two hours, authorities say.</s>'
```

The following figures shows the distribution of similarity scores among all the stories.

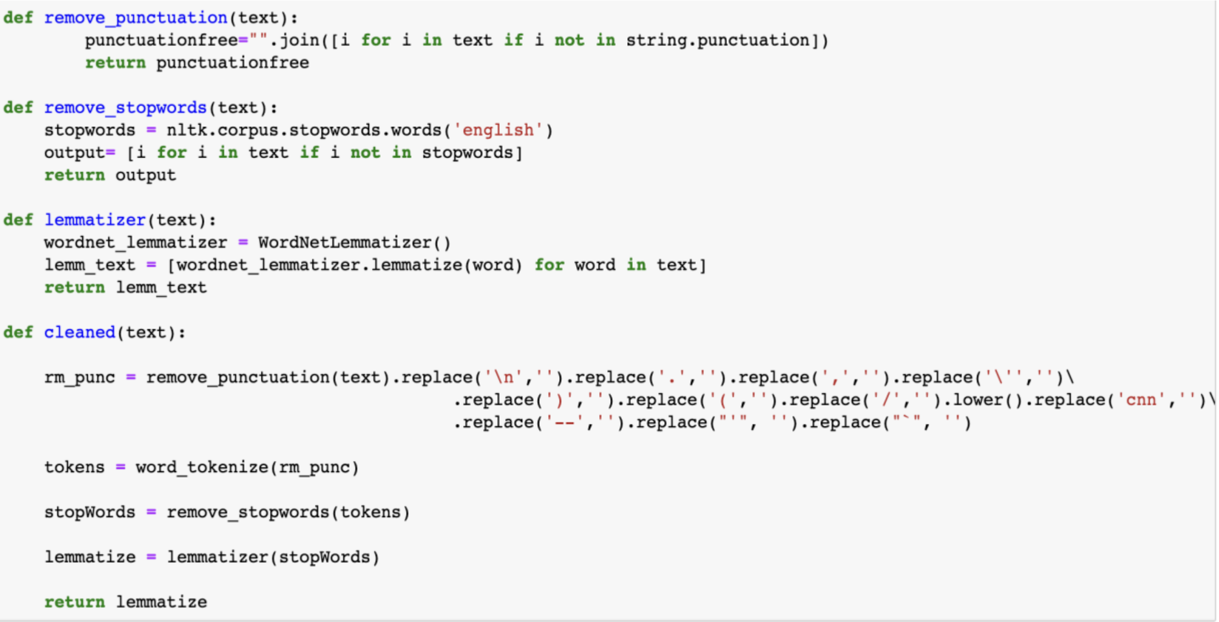


Cosine similarity score distribution

Appendix

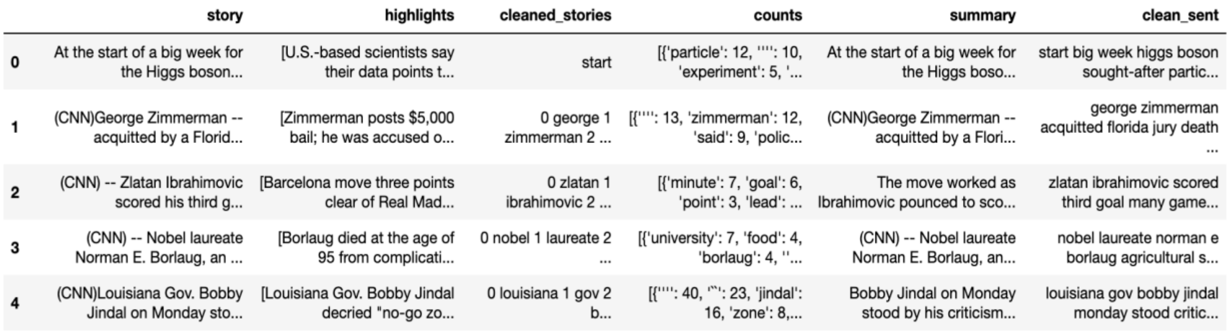
Code for Data Preprocessing and Analysis:

We performed data cleansing and replaced it with necessary annotations. Our code snippet for cleansing data is as follows.



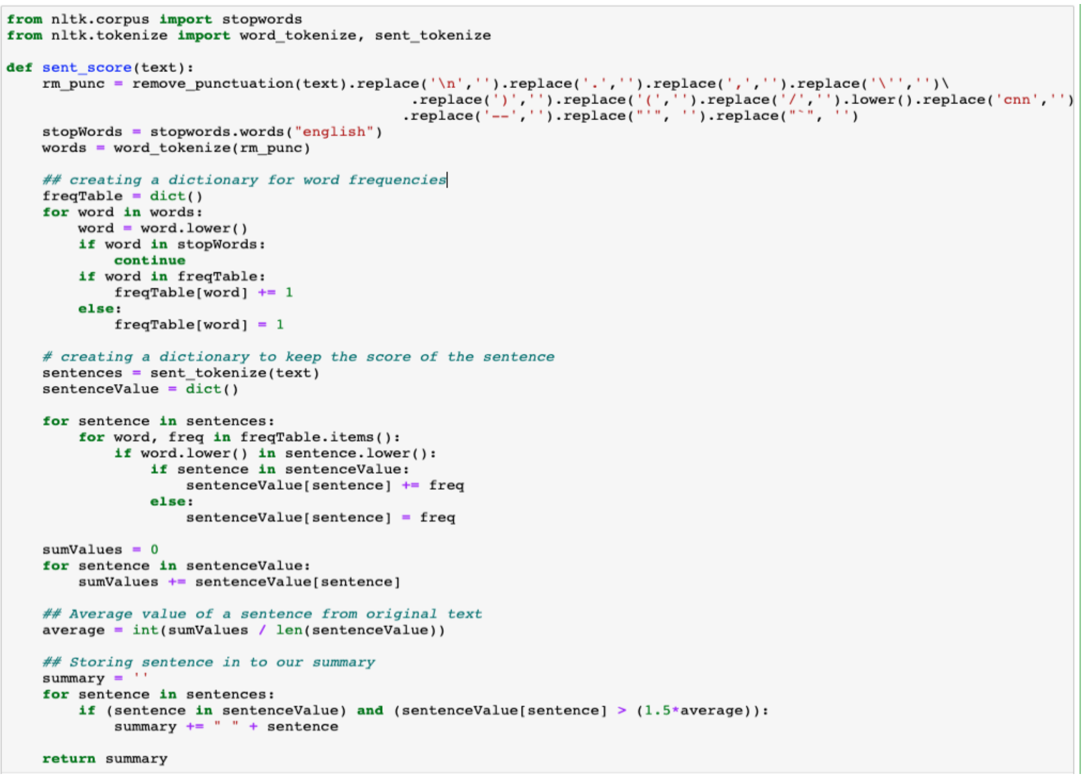
```
def remove_punctuation(text):
        punctuationfree="".join([i for i in text if i not in string.punctuation])
        return punctuationfree

def remove_stopwords(text):
    stopwords = nltk.corpus.stopwords.words('english')
    output= [i for i in text if i not in stopwords]
    return output

def lemmatizer(text):
    wordnet_lemmatizer = WordNetLemmatizer()
    lemm_text = [wordnet_lemmatizer.lemmatize(word) for word in text]
    return lemm_text

def cleaned(text):

    rm_punc = remove_punctuation(text).replace('\n','').replace('.','').replace(',','').replace('\'','')\
                                        .replace(')','').replace('(','').replace('/','').lower().replace('cnn','')\
                                        .replace('--','').replace("'", '').replace("`", '')

    tokens = word_tokenize(rm_punc)

    stopWords = remove_stopwords(tokens)

    lemmatize = lemmatizer(stopWords)

    return lemmatize
```

The following pandas data frame is created with necessary variables after preprocessing the data. The summary column is generated after our analysis.



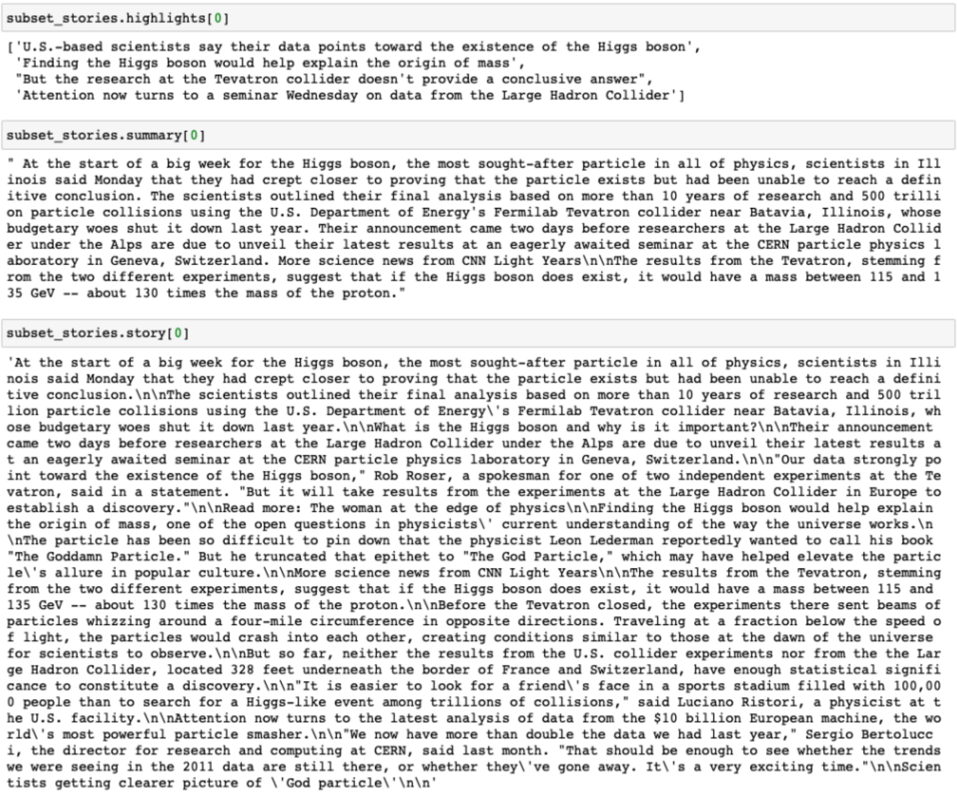
	story	highlights	cleaned_stories	counts	summary	clean_sent
0	At the start of a big week for the Higgs boson...	[U.S.-based scientists say their data points t...	start	[{'particle': 12, '''': 10, 'experiment': 5, '...	At the start of a big week for the Higgs boso...	start big week higgs boson sought-after partic...
1	(CNN)George Zimmerman -- acquitted by a Florid...	[Zimmerman posts $5,000 bail; he was accused o...	0 george 1 zimmerman 2 ...	[{'''': 13, 'zimmerman': 12, 'said': 9, 'polic...	(CNN)George Zimmerman -- acquitted by a Flori...	george zimmerman acquitted florida jury death ...
2	(CNN) -- Zlatan Ibrahimovic scored his third g...	[Barcelona move three points clear of Real Mad...	0 zlatan 1 ibrahimovic 2 ...	[{'minute': 7, 'goal': 6, 'point': 3, 'lead': ...	The move worked as Ibrahimovic pounced to sco...	zlatan ibrahimovic scored third goal many game...
3	(CNN) -- Nobel laureate Norman E. Borlaug, an ...	[Borlaug died at the age of 95 from complicati...	0 nobel 1 laureate 2 ...	[{'university': 7, 'food': 4, 'borlaug': 4, ''...	(CNN) -- Nobel laureate Norman E. Borlaug, an...	nobel laureate norman e borlaug agricultural s...
4	(CNN)Louisiana Gov. Bobby Jindal on Monday sto...	[Louisiana Gov. Bobby Jindal decried "no-go zo...	0 louisiana 1 gov 2 b...	[{'''': 40, '`': 23, 'jindal': 16, 'zone': 8,...	Bobby Jindal on Monday stood by his criticism...	louisiana gov bobby jindal monday stood critic...

Code for generating summary:



```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize

def sent_score(text):
    rm_punc = remove_punctuation(text).replace('\n','').replace('.','').replace(',','').replace('\'','')\
                                         .replace(')','').replace('(','').replace('/','').lower().replace('cnn','')
                                        .replace('--','').replace("'", '').replace("`", '')
    stopWords = stopwords.words("english")
    words = word_tokenize(rm_punc)

    ## creating a dictionary for word frequencies
    freqTable = dict()
    for word in words:
        word = word.lower()
        if word in stopWords:
            continue
        if word in freqTable:
            freqTable[word] += 1
        else:
            freqTable[word] = 1

    # creating a dictionary to keep the score of the sentence
    sentences = sent_tokenize(text)
    sentenceValue = dict()

    for sentence in sentences:
        for word, freq in freqTable.items():
            if word.lower() in sentence.lower():
                if sentence in sentenceValue:
                    sentenceValue[sentence] += freq
                else:
                    sentenceValue[sentence] = freq

    sumValues = 0
    for sentence in sentenceValue:
        sumValues += sentenceValue[sentence]

    ## Average value of a sentence from original text
    average = int(sumValues / len(sentenceValue))

    ## Storing sentence in to our summary
    summary = ''
    for sentence in sentences:
        if (sentence in sentenceValue) and (sentenceValue[sentence] > (1.5*average)):
            summary += " " + sentence

    return summary
```

As can be seen below, the first cell shows the story highlight, 2nd cell displays generated summary of the story and 3rd cell displays the original story.



```
subset_stories.highlights[0]
```

```
['U.S.-based scientists say their data points toward the existence of the Higgs boson',
 'Finding the Higgs boson would help explain the origin of mass',
 "But the research at the Tevatron collider doesn't provide a conclusive answer",
 'Attention now turns to a seminar Wednesday on data from the Large Hadron Collider']
```

```
subset_stories.summary[0]
```

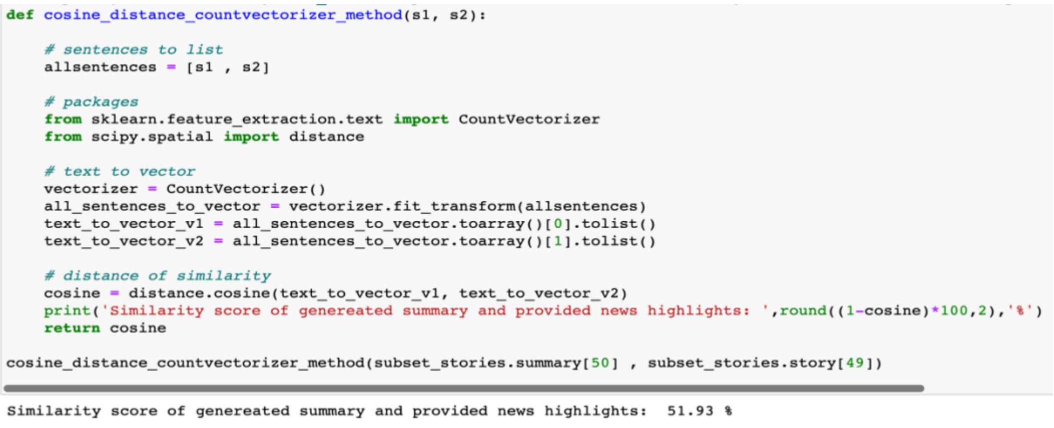
```
" At the start of a big week for the Higgs boson, the most sought-after particle in all of physics, scientists in Ill
inois said Monday that they had crept closer to proving that the particle exists but had been unable to reach a defin
itive conclusion. The scientists outlined their final analysis based on more than 10 years of research and 500 trilli
on particle collisions using the U.S. Department of Energy's Fermilab Tevatron collider near Batavia, Illinois, whose
budgetary woes shut it down last year. Their announcement came two days before researchers at the Large Hadron Collid
er under the Alps are due to unveil their latest results at an eagerly awaited seminar at the CERN particle physics l
aboratory in Geneva, Switzerland. More science news from CNN Light Years\n\nThe results from the Tevatron, stemming f
rom the two different experiments, suggest that if the Higgs boson does exist, it would have a mass between 115 and 1
35 GeV -- about 130 times the mass of the proton."
```

```
subset_stories.story[0]
```

```
'At the start of a big week for the Higgs boson, the most sought-after particle in all of physics, scientists in Illi
nois said Monday that they had crept closer to proving that the particle exists but had been unable to reach a defini
tive conclusion.\n\nThe scientists outlined their final analysis based on more than 10 years of research and 500 tril
lion particle collisions using the U.S. Department of Energy\'s Fermilab Tevatron collider near Batavia, Illinois, wh
ose budgetary woes shut it down last year.\n\nWhat is the Higgs boson and why is it important?\n\nTheir announcement
came two days before researchers at the Large Hadron Collider under the Alps are due to unveil their latest results a
t an eagerly awaited seminar at the CERN particle physics laboratory in Geneva, Switzerland.\n\n"Our data strongly po
int toward the existence of the Higgs boson," Rob Roser, a spokesman for one of two independent experiments at the Te
vatron, said in a statement. "But it will take results from the experiments at the Large Hadron Collider in Europe to
establish a discovery."\n\nRead more: The woman at the edge of physics\n\nFinding the Higgs boson would help explain
the origin of mass, one of the open questions in physicists\' current understanding of the way the universe works.\n
\nThe particle has been so difficult to pin down that the physicist Leon Lederman reportedly wanted to call his book
"The Goddamn Particle." But he truncated that epithet to "The God Particle," which may have helped elevate the partic
le\'s allure in popular culture.\n\nMore science news from CNN Light Years\n\nThe results from the Tevatron, stemming
from the two different experiments, suggest that if the Higgs boson does exist, it would have a mass between 115 and
135 GeV -- about 130 times the mass of the proton.\n\nBefore the Tevatron closed, the experiments there sent beams of
particles whizzing around a four-mile circumference in opposite directions. Traveling at a fraction below the speed o
f light, the particles would crash into each other, creating conditions similar to those at the dawn of the universe
for scientists to observe.\n\nBut so far, neither the results from the U.S. collider experiments nor from the the Lar
ge Hadron Collider, located 328 feet underneath the border of France and Switzerland, have enough statistical signifi
cance to constitute a discovery.\n\n"It is easier to look for a friend\'s face in a sports stadium filled with 100,00
0 people than to search for a Higgs-like event among trillions of collisions," said Luciano Ristori, a physicist at t
he U.S. facility.\n\nAttention now turns to the latest analysis of data from the $10 billion European machine, the wo
rld\'s most powerful particle smasher.\n\n"We now have more than double the data we had last year," Sergio Bertolucc
i, the director for research and computing at CERN, said last month. "That should be enough to see whether the trends
we were seeing in the 2011 data are still there, or whether they\'ve gone away. It\'s a very exciting time."\n\nScien
tists getting clearer picture of \'God particle\'\n\n'
```

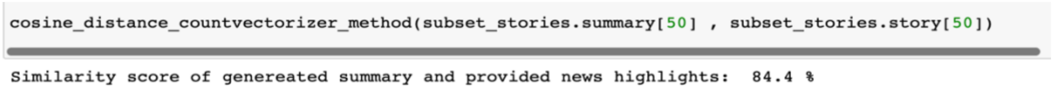
Code for similarity check:

Below is our code for checking the similarity. For example, the similarity between story number 49 and the summary of story number 50 is only 52% whereas the similarity score between story number 50 and its summary is 84%.



```
def cosine_distance_countvectorizer_method(s1, s2):

    # sentences to list
    allsentences = [s1 , s2]

    # packages
    from sklearn.feature_extraction.text import CountVectorizer
    from scipy.spatial import distance

    # text to vector
    vectorizer = CountVectorizer()
    all_sentences_to_vector = vectorizer.fit_transform(allsentences)
    text_to_vector_v1 = all_sentences_to_vector.toarray()[0].tolist()
    text_to_vector_v2 = all_sentences_to_vector.toarray()[1].tolist()

    # distance of similarity
    cosine = distance.cosine(text_to_vector_v1, text_to_vector_v2)
    print('Similarity score of genereated summary and provided news highlights: ',round((1-cosine)*100,2),'%')
    return cosine

cosine_distance_countvectorizer_method(subset_stories.summary[50] , subset_stories.story[49])
```

```
Similarity score of genereated summary and provided news highlights:  51.93 %
```



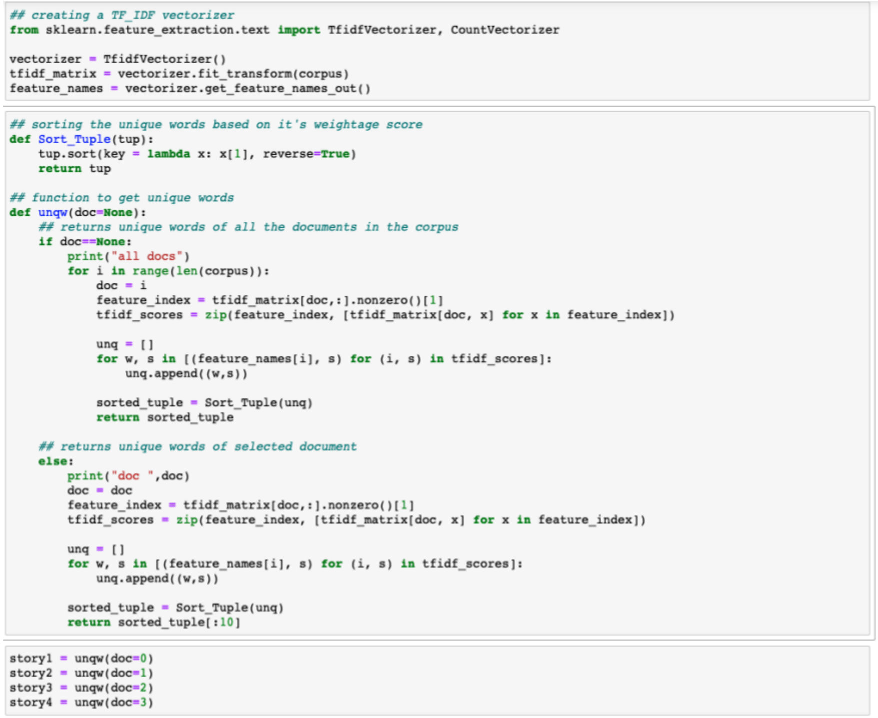
```
cosine_distance_countvectorizer_method(subset_stories.summary[50] , subset_stories.story[50])
```

```
Similarity score of genereated summary and provided news highlights:  84.4 %
```

TF-IDF Code:

Before the similarity check, a TF-IDF (Term Frequency – Inverse Document Frequency) vectorizer is trained. The IDF part gives us the unique words used in each story. Using these words, we checked if the same words were used in the story highlights.

For our analysis purpose, we made a few visualizations of the unique words in the first 4 stories. The following code is used to extract the top 10 unique words.

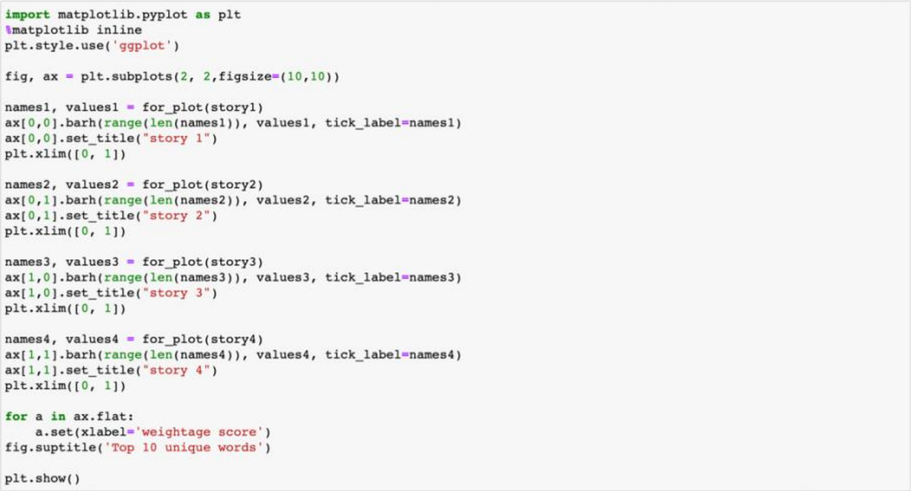


```
## creating a TF_IDF vectorizer
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer

vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(corpus)
feature_names = vectorizer.get_feature_names_out()
```

```
## sorting the unique words based on it's weightage score
def Sort_Tuple(tup):
    tup.sort(key = lambda x: x[1], reverse=True)
    return tup

## function to get unique words
def unqw(doc=None):
    ## returns unique words of all the documents in the corpus
    if doc==None:
        print("all docs")
        for i in range(len(corpus)):
            doc = i
            feature_index = tfidf_matrix[doc,:].nonzero()[1]
            tfidf_scores = zip(feature_index, [tfidf_matrix[doc, x] for x in feature_index])

            unq = []
            for w, s in [(feature_names[i], s) for (i, s) in tfidf_scores]:
                unq.append((w,s))

            sorted_tuple = Sort_Tuple(unq)
            return sorted_tuple

    ## returns unique words of selected document
    else:
        print("doc ",doc)
        doc = doc
        feature_index = tfidf_matrix[doc,:].nonzero()[1]
        tfidf_scores = zip(feature_index, [tfidf_matrix[doc, x] for x in feature_index])

        unq = []
        for w, s in [(feature_names[i], s) for (i, s) in tfidf_scores]:
            unq.append((w,s))

        sorted_tuple = Sort_Tuple(unq)
        return sorted_tuple[:10]
```

```
story1 = unqw(doc=0)
story2 = unqw(doc=1)
story3 = unqw(doc=2)
story4 = unqw(doc=3)
```

The following code is used for visualizing the unique words and their corresponding importance. Just, for example, the first 4 stories are visualized here.



```
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('ggplot')

fig, ax = plt.subplots(2, 2,figsize=(10,10))

names1, values1 = for_plot(story1)
ax[0,0].barh(range(len(names1)), values1, tick_label=names1)
ax[0,0].set_title("story 1")
plt.xlim([0, 1])

names2, values2 = for_plot(story2)
ax[0,1].barh(range(len(names2)), values2, tick_label=names2)
ax[0,1].set_title("story 2")
plt.xlim([0, 1])

names3, values3 = for_plot(story3)
ax[1,0].barh(range(len(names3)), values3, tick_label=names3)
ax[1,0].set_title("story 3")
plt.xlim([0, 1])

names4, values4 = for_plot(story4)
ax[1,1].barh(range(len(names4)), values4, tick_label=names4)
ax[1,1].set_title("story 4")
plt.xlim([0, 1])

for a in ax.flat:
    a.set(xlabel='weightage score')
fig.suptitle('Top 10 unique words')

plt.show()
```