Text Summarization using NLP

- Akshay Sharma

Introduction

As we know, the internet contains large volumes of data, both descriptive as well as quantitative. While the quantitative data can be analysed in several ways, it becomes cumbersome to read through the large paragraphs in descriptive data. So, the most efficient way to get access to these important parts of the data is to summarize the data in a way that it contains non-redundant and useful information only.

Text summarization is a subdomain of Natural Language Processing (NLP) that deals with extracting summaries from huge chunks of texts. There are two main techniques used for text summarization - NLP-based techniques and deep learning based techniques. In this project, we have used a simple NLP-based technique for text summarization.

In [4]:
```
from newspaper import Article
import re
import bs4 as bs
import urllib.request
import numpy as np
import nltk
```

Steps Involved in Text Summarization

1. We first begin by importing the neccessary libraries for the project. We then use the **urlopen** function form the **urllib.request** utility to scrape the data. Next, we need to call **read** function on the object returned by urlopen function in order to read the data. To parse the data, we use **BeautifulSoup** object and pass it the scraped data object. In Wikipedia articles, all the text is enclosed inside the ¡p¿ tags. To retrieve the text we

In [6]:
```
#Using the urlopen function from the urllib.request library to scrap the data
scraped_data = urllib.request.urlopen('https://en.wikipedia.org/wiki/FIFA_World_Cup')

#Reading the data
article = scraped_data.read()

#To parse the data, we use beautiful soup library
parsed_article = bs.BeautifulSoup(article, 'lxml')
paragraphs = parsed_article.find_all('p')

article_text = ""

for p in paragraphs:
    article_text += p.text
```

need to call **find all** function on the object returned by BeautifulSoup. This function returns all the paragraphs in the article in the form of a list.

2. **Pre - processing** - This step involves removing references from the article. It also involves removing the square brackets an replacing the resulting multiple spaces by a single space. Apart from that, we also remove the special characters which are not needed in the summary. Now, we have two objects **article text** which contains the

```
In [8]:
#Removing square brackets and extra spaces
article_text = re.sub(r'\[[0-9]*\]', ' ', article_text)
article_text = re.sub(r'\s+', ' ', article_text)
```

```
In [9]:
#Removing special characters and digits
processed_text = re.sub('[^a-zA-Z0-9]', ' ', article_text)
processed_text = re.sub(r'\s+',' ', processed_text)
```

original article and **processed-text**, which contains the processed article. We will use the latter to create the weighted frequency histograms which will then be replaced with the words in the original text.

3. **Converting text into sentences** - Now that we have pre-processed the data, we tokenize the article into sentences. The original text is used to tokenize the article into sentences since it contains full-stops.

```
In [10]:
sentences = nltk.sent_tokenize(article_text)
```

```
In [11]:
len(sentences)
#The number of sentences in the text
```

```
Out[11]: 242
```

4. **Find the weighted frequency of occurrence** - To find the frequency of occurrence of each word, we use the **processed text** variable. This is because this does not contain special characters, punctuations etc.

   - First we store all the stopwords in the English language from nltk library into a variable named **stopwords**.
   - We then loop through all the sentences and corresponding words to check if they are stop words. If not, we proceed to check whether the words exist in the **word frequencies** dictionary or not.
   - If the word is encountered for the first time, the frequency of that word is initialized to one.
   - If the word is already present, the frequency of that word is updated.
   - Finally, to find the weighted frequency, we divide the frequency of that word with the maximum frequency of any word in the text.

In [12]:
```
from nltk.corpus import stopwords
```

In [13]:
```
stopwords = stopwords.words('english')

word_frequencies = {}
for word in nltk.word_tokenize(processed_text):
    if word not in stopwords:
        if word not in word_frequencies.keys():
            word_frequencies[word] = 1
        else:
            word_frequencies[word] += 1
```

In [14]:
```
max_freq = max(word_frequencies.values())

for word in word_frequencies.keys():
    word_frequencies[word] = (word_frequencies[word]/max_freq)
```

5. **Calculating Sentence Scores** - We calculate the scores for each sentence by adding the weighted frequencies of all the words that occur in that particular sentence.

In [15]:
```
sentence_scores = {}

for sent in sentences:
    for word in nltk.sent_tokenize(sent.lower()):
        for word in word_frequencies.keys():
            if len(sent.split(' ')) < 40:
                if sent not in sentence_scores.keys():
                    sentence_scores[sent] = word_frequencies[word]
                else:
                    sentence_scores[sent] += word_frequencies[word]
```

- We first create an empty **sentence scores** dictionary. The keys of the dictionary will be the sentence themselves and the values will be the corresponding scores of the sentences.
- We loop through each sentence in the **sentence list** and tokenize the sentence into words.
- We then check if the word exists in the **word frequencies** dictionary. This check is performed since we created the sentence list from the **article text** object, on the other hand, the word frequencies were calculated using the **processed text** object, which doesn't contain any stop words, numbers etc.
- We do not want extremely long sentences in the summary. So, we can specify the maximum length of the sentences that we will consider for our summary. Here, I have taken it to be 40 words .
- We check if the sentence exists in the sentence scores dictionary or not. If it doesn't, we add it to the sentence scores dictionary as a key and assign it to the weighted frequency dof the first word in the sentence.
- If it already exists, we add the weighted frequency of the word to the existing value.

6. Finally, we get the summary as follows -

In [19]:

```
import heapq
summary_sent = heapq.nlargest(25, sentence_scores, key = sentence_scores.get)
#returns the top n sentences with the highest scores

summary = ' '.join(summary_sent)
print(summary)
```

Example

The url of the text we need to summarize is given below along with the summary we got after performing the above processes.

URL - FIFA World Cup article from Wikipedia

Summary -

```
In November 2007, FIFA announced that all members of World Cup-winning squads between 1930 and 1974 were to be retroactively awarded winners' medal
s. The reigning champions are France, who won their second title at the 2018 tournament in Russia. The format involves a qualification phase, which
takes place over the preceding three years, to determine which teams qualify for the tournament phase. In the tournament phase, 32 teams compete for
the title at venues within the host nation(s) over about a month. The host nation(s) automatically qualify. As of the 2018 FIFA World Cup, twenty-on
e final tournaments have been held and a total of 79 national teams have competed. The trophy has been won by eight national teams. Brazil have won
five times, and they are the only team to have played in every tournament. The World Cup is the most prestigious association football tournament in
the world, as well as the most widely viewed and followed single sporting event in the world. Seventeen countries have hosted the World Cup. The wor
ld's first international football match was a challenge match played in Glasgow in 1872 between Scotland and England. The first international tourna
ment for nations, the inaugural British Home Championship, took place in 1884. After FIFA was founded in 1904, it tried to arrange an international
football tournament between nations outside the Olympic framework in Switzerland in 1906. These were very early days for international football, and
the official history of FIFA describes the competition as having been a failure. At the 1908 Summer Olympics in London, football became an official
competition. Planned by The Football Association (FA), England's football governing body, the event was for amateur players only and was regarded su
spiciously as a show rather than a competition. Great Britain (represented by the England national amateur football team) won the gold medals. They
repeated the feat at the 1912 Summer Olympics in Stockholm. With the Olympic event continuing to be contested only between amateur teams, Sir Thomas
Lipton organised the Sir Thomas Lipton Trophy tournament in Turin in 1909. The Lipton tournament was a championship between individual clubs (not na
tional teams) from different nations, each one of which represented an entire nation. Lipton invited West Auckland, an amateur side from County Durh
am, to represent England instead. West Auckland won the tournament and returned in 1911 to successfully defend their title. and the Heart of Midloth
ian F.C., which Sunderland won. In 1914, FIFA agreed to recognise the Olympic tournament as a "world football championship for amateurs", and took r
esponsibility for managing the event. This paved the way for the world's first intercontinental football competition for nations, at the 1920 Summer
Olympics, contested by Egypt and 13 European teams, and won by Belgium.
```