Data Pre_Processing_NLP_Removing Urls

By : Venkatesh Mungi venkateshmungi1247@gmail.com

1.Using the re.sub() function to remove URLs from Text in Python

The re.sub() function provides the most straightforward approach to remove URLs from text in Python.

This function is used to substitute a given substring with another substring in any provided string. It uses a regex pattern to find the substring and then replace it with the provided substring.

In [1]:
```
import re
```

In [2]:
```
ven ="This is a text with a URL https://www.venkateshmungi.com/ to remove."
v1 = re.sub('http://\S+|https://\S+', '', ven)
v2 = re.sub('http[s]?://\S+', '', ven)
v3 = re.sub(r"http\S+", "", ven)
print(v1)
print(v2)
print(v3)
```

```
This is a text with a URL  to remove.
This is a text with a URL  to remove.
This is a text with a URL  to remove.
```

In [3]:
```
def remove_URL(sample):
    """Remove URLs from a sample string"""
    return re.sub(r"http\S+", "", sample)

sample = "This is a text with a URL https://www.venkateshmungi.com/ to remove."

remove_URL(sample)
```

Out[3]:
```
'This is a text with a URL  to remove.'
```

2.Using the re.findall() function to remove URLs from Text in Python

The re.findall() function is used to find the total occurrences of a substring in a given string based on a regex pattern. It returns a list of all the occurrences of the substring.

We can use this function to find the URLs in a given string and then remove them using the replace() function. With the replace() function, we will replace the occurrence of the given URL with an empty string.

In [4]:
```
Krishna = "He is One and only Legend in https://www.kurukshetra.com/ and no one can not defeat him"

sri = re.findall('http://\S+|https://\S+', Krishna)

for i in sri:
    result = Krishna.replace(i, '')
    print(result)
```

```
He is One and only Legend in  and no one can not defeat him
```

In [5]:
```
def rmv_url(Krishna):
    """Remove URLs from a sample string"""
    sri = re.findall('http://\S+|https://\S+', Krishna)
    return Krishna.replace(i, '')
Krishna = "He is One and only Legend in https://www.kurukshetra.com/ and no one can not defeat him"
rmv_url(Krishna)
```

Out[5]: 'He is One and only Legend in and no one can not defeat him'

Extract URL from Text

URL extraction is achieved from a text file by using regular expression. The expression fetches the text wherever it matches the pattern. Only the re module is used for this purpose.We can take a input file containig some URLs and process it thorugh the following program to extract the URLs.

The findall()function is used to find all instances matching with the regular expression.

We can take a input file containig some URLs and process it thorugh the following program to extract the URLs. The findall()function is used to find all instances matching with the regular expression.

INPUT Text : Now a days you can learn almost anything by just visiting http://www.google.com (http://www.google.com). But if you are completely new to computers or internet then first you need to leanr those fundamentals. Nextyou can visit a good e-learning site like - https://www.tutorialspoint.com (https://www.tutorialspoint.com) to learn further on a variety of subjects.

In [6]:
```
with open("extractingurls.txt") as file:
    for line in file:
        urls = re.findall('https?://(?:[-\w.]|(?:%[\da-fA-F]{2}))+', line)
        print(urls)
```

```
['http://www.google.com.']
['https://www.tutorialspoint.com']
```

3.Using the re.search() function to remove URLs from Text in Python

We can also use the re.match() and re.search() function to find a substring based on the regex pattern. However, both these functions only return the first occurrence of the substring. So, if a string contains more than one URL, these methods will fail.

Another downside of the re.match() function is that it only searches the first line of the string. So, if we have a string with only one URL, we can use the re.search() function.

In [7]:
```
Krishna = "He is One and only Legend in https://www.kurukshetra.com/ and no one can not defeat him"
arjuna = re.search("http://\S+|https://\S+", Krishna)
bheem = arjuna.group(0)
dharma = Krishna.replace(bheem,"")
print(dharma)
```

```
He is One and only Legend in  and no one can not defeat him
```

In [8]:
```
def rm_url(krishna):
    arjuna = re.search("http://\S+|https://\S+", Krishna)
    bheem = arjuna.group(0)
    return Krishna.replace(bheem,"")
Krishna = "He is One and only Legend in https://www.kurukshetra.com/ and no one can not defeat him"
rm_url(Krishna)
```

Out[8]: 'He is One and only Legend in and no one can not defeat him'

4.Using the urllib.urlparse class to remove URLs from Text in Python

In Python, we can send requests to a given address using modules like urllib, requests, and more. With the urllib.urlparse class, we can parse URLs and break them into components.

The urllib.parse object parses a URL string. We can use the scheme attribute of this object to check whether a string matches the structure of a URL or not.

To remove URLs from text in Python with this method, we will first break the text into a list of strings. This can be achieved using the split() function that can split strings into a list of strings based on some character.

We will then use the scheme attribute to check if each string in the list matches a URL or not. If the match is True, we will ignore that string. Finally, we will combine the remaining elements of the list using the join() function.

In [9]:
```
from urllib.parse import urlparse
Krishna ="He is One and only Legend in https://www.kurukshetra.com/ and no one can not defeat him."
lst = [l for l in Krishna.split() if not urlparse(l).scheme]
s = ' '.join(lst)
s
```

Out[9]:
```
'He is One and only Legend in and no one can not defeat him.'
```

In [10]:
```
def rmve_url(krishna):
    lst = [l for l in Krishna.split() if not urlparse(l).scheme]
    s = ' '.join(lst)
    return s
Krishna ="He is One and only Legend in https://www.kurukshetra.com/ and no one can not defeat him."
rmve_url(Krishna)
```

Out[10]:
```
'He is One and only Legend in and no one can not defeat him.'
```