Feature extraction from URLs

May 19, 2021

0.0.1 HTML and Javascript Based Features

- Links in Meta, Script and Link tags
- Submitting Information to Email
- Abnormal URL
- Website Forwarding
- status Bar Customization
- Disabling Right Click
- Using Pop-up Window
- IFrame Redirection

Eeach of these features are explained below

Links in Meta, Script and Link tags Here, check whether the links in Meta, scrypt and link tags are linked to the same domain

[ ]:
```
import pandas as pd
import csv
import os
from tldextract import extract
import urllib.request
from bs4 import BeautifulSoup

output = os.path.join('output.csv')
urlinput = pd.read_csv('links.csv')

f = csv.writer(open(output, "w+", newline="\n", encoding="utf-8"))
f.writerow(["URL", "links_in_tags"])

for i in range(0,len(urlinput)):
    url = urlinput.loc[i,'url']

    try:
        subDomain, domain, suffix = extract(url)
        websiteDomain = domain

        opener = urllib.request.urlopen(url).read()
        soup = BeautifulSoup(opener, 'lxml')
```

```
        metas = soup.find_all('meta',href=True)
        links = soup.find_all('link', href=True)
        scripts = soup.find_all('script', src=True)

        no_of_meta = len(metas)
        no_of_link = len(links)
        no_of_script = len(scripts)
        total = no_of_meta + no_of_link + no_of_script

        linked_to_same = 0
        avg =0

        for meta in metas:
            subDomain, domain, suffix = extract(meta['href'])
            meta_domain = domain

            if(websiteDomain == meta_domain or meta_domain == ""):
                linked_to_same = linked_to_same+1

        for link in links:
            subDomain, domain, suffix = extract(link['href'])
            link_domain = domain
            if(websiteDomain == link_domain or link_domain == ""):
                linked_to_same = linked_to_same + 1

        for script in scripts:
            subDomain, domain, suffix = extract(script['src'])
            script_domain = domain
            if(websiteDomain == script_domain or script_domain == ""):
                linked_to_same = linked_to_same + 1

        outside_domain = total - linked_to_same

        if(total!=0):
            avg = round((outside_domain/total)*100,2)

        f.writerow([url,avg])
        print(str(i)+" - Response : "+str(avg)+" - "+url)

    except:
        f.writerow([url,"N/A"])
        print(str(i)+" - Response : N/A - "+url)
        # raised
```

Submitting Information to Email Web forms may allow user to submit his personal information to an email. Here, the availability of 'mail()' function or 'mailto()' function is recorded.

[ ]:
```
import re
import requests
import pandas as pd
import csv
import os

output = os.path.join('output.csv')
urlinput = pd.read_csv('links.csv')

f = csv.writer(open(output, "w+", newline="\n", encoding="utf-8"))
f.writerow(["URL", "submit_to_email"])

for i in range(0,len(urlinput)):
    url = urlinput.loc[i,'url']

    # Stores the response of the given URL
    try:
        response = requests.get(url)
    except:
        response = ""
        # raise

    if response != "" :
        if re.findall(r"[mail\(\)|mailto:?]", response.text):
            result = 1
        else:
            result = -1
    else:
        result = "N/A"

    f.writerow([url,result])
    print(str(i)+" - Response : "+str(result)+" - "+url)
```

Abnormal URL The host name availability in whois database is checked in the given URL

[ ]:
```
import re
import pandas as pd
import csv
import os
import whois

output = os.path.join('output.csv')
urlinput = pd.read_csv('links.csv')

f = csv.writer(open(output, "w+", newline="\n", encoding="utf-8"))
f.writerow(["URL", "abnormal"])
```

```
for i in range(0,len(urlinput)):
    url = urlinput.loc[i,'url']

    try:
        response = whois.whois(url)
    except Exception as e:
        response = ""
        # raise

    if response != "" :
        if response.domain_name :
            result = 1
        else:
            result = -1
    else:
        result = "N/A"

    f.writerow([url,result])
    print(str(i)+" - Response : "+str(result)+" - "+url)
```

Website Forwarding Number of redirections of the website is counted here.

[ ]:
```
import re
import requests
import pandas as pd
import csv
import os

output = os.path.join('output.csv')
urlinput = pd.read_csv('links.csv')

f = csv.writer(open(output, "w+", newline="\n", encoding="utf-8"))
f.writerow(["URL", "redirect"])

for i in range(0,len(urlinput)):
    url = urlinput.loc[i,'url']

    # Stores the response of the given URL
    try:
        response = requests.get(url)
    except:
        response = ""
        # raise

    if response != "" :
        result = len(response.history)
    else:
```

```
        result = "N/A"

    f.writerow([url,result])
    print(str(i)+" - Response : "+str(result)+" - "+url)
```

Status Bar Customization There can be javascripts to show fake url in the status bar. So that, particularly the "onMouseOver" event is serached in the source code.

[ ]:
```
import re
import requests
import pandas as pd
import csv
import os

output = os.path.join('output.csv')
urlinput = pd.read_csv('links.csv')

f = csv.writer(open(output, "w+", newline="\n", encoding="utf-8"))
f.writerow(["URL", "status_bar"])

for i in range(0,len(urlinput)):
    url = urlinput.loc[i,'url']

    # Stores the response of the given URL
    try:
        response = requests.get(url)
    except:
        response = ""
        # raise

    if response != "" :
        if re.findall("<script>.+onmouseover.+</script>", response.text):
            result = 1
        else:
            result = -1
    else:
        result = -1

    f.writerow([url,result])
    print(str(i)+" - Response : "+str(result)+" - "+url)
```

Disabling Right Click There can be JavaScript to disable the right-click function, so that users cannot view and save the webpage source code. For this feature, we search for event "event.button==2" in the webpage source code and check if the right click is disabled.

[ ]:
```
import re
import requests
```

```
import pandas as pd
import csv
import os

output = os.path.join('output.csv')
urlinput = pd.read_csv('links.csv')

f = csv.writer(open(output, "w+", newline="\n", encoding="utf-8"))
f.writerow(["URL", "disble_write_click"])

for i in range(0,len(urlinput)):
    url = urlinput.loc[i,'url']

    # Stores the response of the given URL
    try:
        response = requests.get(url)
    except:
        response = ""
        # raise

    if response != "" :
        if re.findall(r"event.button ?== ?2", response.text):
            result = 1
        else:
            result = -1
    else:
        result = -1

    f.writerow([url,result])
    print(str(i)+" - Response : "+str(result)+" - "+url)
```

Using Pop-up Window Here, the availability of text fields in the popup window is searched

[ ]:
```
import re
import requests
import pandas as pd
import csv
import os

output = os.path.join('output.csv')
urlinput = pd.read_csv('links.csv')

f = csv.writer(open(output, "w+", newline="\n", encoding="utf-8"))
f.writerow(["URL", "popup"])

for i in range(0,len(urlinput)):
    url = urlinput.loc[i,'url']
```

```
    # Stores the response of the given URL
    try:
        response = requests.get(url)
    except:
        response = ""
        # raise

    if response != "" :
        if re.findall(r"alert\(", response.text):
            result = 1
        else:
            result = -1
    else:
        result = -1

    f.writerow([url,result])
    print(str(i)+" - Response : "+str(result)+" - "+url)
```

IFrame Redirection IFrame can be used by making it invisible without frame borders. Here we check whether the source code includes frameBorder attribute for hiding Iframe border.

[ ]:
```
import re
import requests
import pandas as pd
import csv
import os

output = os.path.join('output.csv')
urlinput = pd.read_csv('links.csv')

f = csv.writer(open(output, "w+", newline="\n", encoding="utf-8"))
f.writerow(["URL", "iframe"])

for i in range(0,len(urlinput)):
    url = urlinput.loc[i,'url']

    # Stores the response of the given URL
    try:
        response = requests.get(url)
    except:
        response = ""
        # raise

    if response != "" :
        if re.findall(r"[<iframe>|<frameBorder>]", response.text):
            result = 1
```

```
        else:
            result = -1
    else:
        result = -1

    f.writerow([url,result])
    print(str(i)+" - Response : "+str(result)+" - "+url)
```

[ ]: