Technische handleiding

IPFIT6

Yassir Laaouissi | INF3A | 18-10-2020

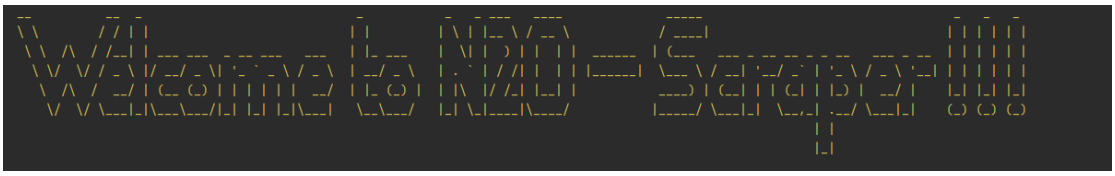
Inhoud

Inleiding

Dit document omvat de beschrijving van alle functionaliteiten die de tool ontwikkeld in dit project bevat. De tool die in dit project is gerealiseerd is gemaakt ten behoeve van het automatiseren van een gedeelte van het OSINT-onderzoek dat gedaan zal worden. In dit document zijn de hoofdstukken ingedeeld op de functies die zijn gemaakt in het python project van de tool.

Main()

Main() is de functie die als eerste wordt aangeroepen. Deze functie kent meerdere verantwoordelijkheden. De eerste is het tonen van de welkomstbanner (afbeelding 1). Deze wordt gerealiseerd middels een http-get request naar een website (afbeelding 2).



Afbeelding 1: Welkomstbanner.

```
## Welcome message
r = requests.get(f'http://artii.herokuapp.com/make?text={"Welcome  to  N2O - Scraper ! ! !"}')
print('\33[33m' + r.text + '\33[0m' + "\n\n")
```

Afbeelding 2: Code voor de welkomstbanner.

Nadat de welkomstbanner is getoond op het scherm is het tijd voor een keuzemenu. Het keuzemenu moet duidelijk maken welke platformen gewenst worden gescraped. Daarnaast plaatst het een check om te kijken of deze platformen wel beschikbaar zijn binnen deze tool (afbeelding 3).

```
##keuze voor welke platformen gescraped moeten worden en gelijk checken of die platforms wel gescraped kunnen worden.
WhatToScrape = input("What platforms do you want scraped? (Options and input structure: twitter, marktplaats, instagram, google maps): ")
Platforms = ["twitter", "marktplaats", "instagram", "google maps"]
x = WhatToScrape.split(", ")
for SelectedPlatform in x:
    if not SelectedPlatform in Platforms:
        print('\33[31m' + "\nOne or more platforms you have given up are not available for scraping in this tool. Please try again with the available platforms." + '\33[0m')
        exit(-1)
```

Afbeelding 3: Code voor het keuzemenu.

Tot slot loopt de main() door de gekozen platformen en start per gekozen menu de daartoe bestemde functie (afbeelding 4). De functies per platform worden in de volgende hoofdstukken verder behandelt.

```
#De scrapers runnen op basis van SelectedPlatform in x
for SelectedPlatform in x:
    if SelectedPlatform == "twitter":
        print('\33[33m' + "\nStarting twitter scraper" + '\33[0m')
        twitter()
    elif SelectedPlatform == "marktplaats":
        print('\33[33m' + "\nStarting marktplaats scraper" + '\33[0m')
        marktplaats()
    elif SelectedPlatform == "instagram":
        print('\33[33m' + "\nStarting instagram scraper" + '\33[0m')
        instagram()
    elif SelectedPlatform == "google maps":
        print('\33[33m' + "\nStarting google maps scraper" + '\33[0m')
        google_maps()
```

Afbeelding 4: Code voor het loopen door de gekozen platformen.

Instagram()

Dit gedeelte van het script is verantwoordelijk voor het scrapen van Instagram aan de hand van een aantal zoektermen. Deze scraper is gebouwd aan de hand van een bestaand script die te vinden is via: https://github.com/arc298/instagram-scraper

De Instagram functie leest eerst alle zoektermen uit en voegt ze toe aan een list, zodat deze in een latere stap makkelijker meegenomen kunnen worden (afbeelding 5).

```
inputfile = open("Zoektermen/keywords_base - Copy.txt", "r")
wordlist = []
for phrase in inputfile:
    phrase = phrase.replace("\n", "")
    wordlist.append(phrase)
```

Afbeelding 5: Code inlezen zoektermen.

Tot slot wordt een command line commando uitgevoerd met daarin de zoektermen en een variabele genaamd “timeofnow” die worden gebruikt voor het aanmaken van de juiste opslaglocatie (afbeelding 6).

In het onderstaande commando wordt er ingelogd op een instagram account middels -u en -p. Verder wordt gezocht op alle posts die een hashtag bevatten die overeenkomt met de zoektermen. Daarnaast worden locaties, comments en metadata van het profiel van de “poster” meegenomen in de Output. En tot slot zit er ook nog een maximum van laatste 30 posts op als downloadcap.

```
timeofnow = datetime.now().strftime(f"%Y-%m-%d--%H-%M")
for word in wordlist:
    os.system(f'instagram-scraper -u="studenthsleiden" -p="Rolstoel31" --tag "{word}" --include-location --maximum 30 --comments --profile-metadata -d Output/Instagram/{timeofnow}/{word}')
```

Afbeelding 6: Code voor het uitvoeren van de instagram scrape.

Marktplaats()

Deze functie is verantwoordelijk voor het scrapen van alle marktplaats advertenties die worden gevonden op basis van de zoektermen die opgegeven zijn.

Het eerste wat deze scraper doet is het inlezen en omvormen van de zoektermen naar een gangbaar formaat. Daarnaast zet het deze gangbare zoektermen om in een list (afbeelding 7).

```
inputfile = open("Zoektermen/keywords_base - Copy.txt", "r")
wordlist = []

for phrase in inputfile:
    phrase = phrase.replace("\n", "")
    wordlist.append(phrase)
```

Afbeelding 7: Code die de zoektermen inleest en omzet tot list.

Daarna loopt de functie door alle zoektermen heen om te kijken of er amsterdam in voor komt. En vormt aan de hand daarvan een custom query URL (afbeelding 8 en 9). Als er geen amsterdam in de zoekterm voorkomt dan worden variabelen zoals postcode en radius toegevoegd aan de query, zodat er alleen advertenties van 20 km om het centrum van Amsterdam heen worden verzameld. Zit er wel amsterdam in de zoekterm dan zijn deze variabelen niet nodig en wordt er alleen gezocht op de zoekterm.

```
for word in wordlist:
    if "amsterdam" in word:
        #print("amsterdam")
        query = word
        url = 'https://www.marktplaats.nl/q/' + query + '/'
        print(str(url))
```

```
else:
    #print("not amsterdam")
    query = word
    postalcode = '1011ab'
    distance = '20000'

    url = 'https://www.marktplaats.nl/q/' + query + '/'
    url += '#distanceMeters:' + distance
    url += '|postcode:' + postalcode
    print(url)
```

Afbeelding 8 en 9: De check voor amsterdam in zoekterm en custom URL

Daarna voer de code voor zowel zoektermen met Amsterdam als zonder een get request uit op de query URL, en zoekt naar het HTML-element waarin de advertenties staan opgeslagen (afbeelding 10).

```
url = 'https://www.marktplaats.nl/q/' + query + '/'
print(str(url))
source = requests.get(url)
#print(source)
marktplaats = BeautifulSoup(source.text, 'lxml')
# body = marktplaats.find('div', class_='mp-Page-element--main')
body = marktplaats.find('body')
search_result = is_defined_item(body.find('ul', class_='mp-Listings--list-view'))
# print(search_result)
```

Afbeelding 10: Get request en inzoomen op mp-listings

Daarna wordt per advertentie in de mp-listings de link naar de advertentie, de beschrijving, de prijs, de naam van de verkoper, de locatie van de verkoper en de link van het profiel van de verkoper gescraped (zie afbeelding 11). Deze informatie wordt weggeschreven in een object genaamd myobj (zie afbeelding 12).

```
listOfArticles = []
try:
    for article in search_result:
        try:
            # advertentieinformatie
            title = is_defined_item(article.find('h3', class_="mp-Listing-title")).text
            # print(title)

            href1 = article.a['href']
            href = "https://www.marktplaats.nl" + href1

            summary_ = is_defined_item(article.find('p', class_='mp-text-paragraph')).text
            # print(summary_)

            price = is_defined_item(article.find('span', class_='mp-text-price-label')).text
            price = price.replace("\xc2\xa0", " ")
            # print(price)

            # seller informatie
            seller_name = is_defined_item(article.find('span', class_='mp-Listing-seller-name')).text
            # print(seller_name)

            seller_location = is_defined_item(article.find('span', class_='mp-Listing-location')).text
            # print(seller_location)

            seller_link = is_defined_item(article.find('a', class_='mp-TextLink'))['href']
            if "/u/" in seller_link:
                def_seller_link = "https://www.marktplaats.nl" + seller_link
            else:
                def_seller_link = seller_link
```

Afbeelding 11: Verschillende elementen van een advertentie worden gextraheerd.

```
# ff al die advertentieinfo wegschrijven naar een object
myObj = Item()
myObj.title = title.encode("utf-8").strip()
myObj.url = href.encode("utf-8").strip()
myObj.price = price.encode("utf-8").strip()
myObj.summary = summary_.encode("utf-8").strip()
myObj.seller_name = seller_name.encode("utf-8").strip()
myObj.seller_location = seller_location.encode("utf-8").strip()
myObj.seller_link = def_seller_link.encode("utf-8").strip()
listOfArticles.append(myObj)
```

Afbeelding 12: MyObj()

Tot slot wordt het object in een CSV en XLSX-bestand opgeslagen (zie afbeelding 13 en 14). Maar daarvoor wordt eerst de juiste mappenstructuur aangemaakt.

```
timeofnow = datetime.now().strftime(f"%Y-%m-%d--%H-%M")
smallertimeofnow = datetime.now().strftime(f"%Y-%m-%d")
if not os.path.exists(f"Output/Marktplaats/{smallertimeofnow}"):
    os.mkdir(f"Output/Marktplaats/{smallertimeofnow}")
csvfilename = f"Output/Marktplaats/{smallertimeofnow}/{word}_MP_{timeofnow}.csv"
xlsxfilename = f"Output/Marktplaats/{smallertimeofnow}/{word}_MP_{timeofnow}.xlsx"
writetocsv(listOfArticles, csvfilename)
convert_csv_to_xsl(csvfilename, xlsxfilename)
```

Afbeelding 13: Opslag van MyObj in CSV en XLSX

```
def writetocsv(items, csvfilename):
    csv_file = open(csvfilename, 'w')
    csv_writer = csv.writer(csv_file)
    csv_writer.writerow(['Title', 'URL', 'Price', 'Summary', 'Seller name', 'Seller location',
                         'Seller link'])
    for item in items:
        csv_writer.writerow([item.title, item.url, item.price, item.summary,
                             item.seller_name, item.seller_location, item.seller_link])
    csv_file.close()


def convert_csv_to_xsl(csvfilename, xlsxfilename):
    wb = Workbook()
    ws = wb.active

    with open(csvfilename, 'r') as f:
        for row in csv.reader(f):
            ws.append(row)
    wb.save(xlsxfilename)
```

Afbeelding 14: writetocsv() en convert_csv_to_xls()

Twitter()

Deze functie staat ervoor garant dat twitter wordt gescraped. Deze functie heeft een gehele tijd goed gefunctioneerd, echter is er vermoedelijk iets aan de kant van twitter verandert waardoor deze functie niet meer naar toebehoren werkt. Echter zal dit gedeelte van de scraper toch beschreven worden, gezien dit wellicht handig is als deze toch werkend gemaakt wordt. Iets wat in de tijd van dit project niet meer zal gebeuren.

Deze functie is gemaakt aan de hand van de volgende github repository: https://github.com/twintproject/twint

Het eerste wat deze functie doet is het inlezen van de zoektermen en omvormen tot een gangbaar formaat (afbeelding 15). Daarna checkt het of er Amsterdam in de zoekterm zit. Zit er Amsterdam in dan wordt alleen op zoekterm en tijd gefilterd. Zit er geen Amsterdam in de zoekterm dan wordt er gekeken naar tweets binnen een straal van 25km vanaf het centrum van Amsterdam (afbeelding 16 en 17). De tweets worden vervolgens weggeschreven naar een list, die dan weer opgeslagen wordt in een JSON-bestand (zie afbeelding 18).

```
def twitter():

    total_tweets = []
    inputfile = open("Zoektermen/keywords_base - Copy.txt", "r")
    for word in inputfile:
        word = word.replace("\n", "")
        print('\33[33m' + "Het woord is nu: " + word + '\33[0m')
```

Afbeelding 15: zoektermen inladen en omvormen

```
if "amsterdam" in word:
    c = twint.Config()
    c.Search = str(word)
    c.Profile_full = True

    # use this for initialscrape
    c.Since = "2020-01-01 00:00:01"
    c.Until = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    c.Store_object = True
    c.Limit = 1
    twint.run.Search(c)
    tlist = c.search_tweet_list
    total_tweets.append(tlist)
```

Afbeelding 16: Scrapen met Amsterdam

```
else:
    c = twint.Config()
    c.Search = str(word)
    c.Geo = "52.378909,4.900244,25km"

    # use this for initialscrape
    c.Since = "2020-01-01 00:00:01"
    c.Until = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    c.Store_object = True
    c.Limit = 1
    c.Profile_full = True
    twint.run.Search(c)
    tlist = c.search_tweet_list
    total_tweets.append(tlist)
```

Afbeelding 17: Scrapen zonder Amsterdam

```
#print(total_tweets)
filename_json = f"Output/Twitter/Export_" + datetime.now().strftime("%Y-%m-%d--%H-%M") + ".json"
file = open(filename_json, 'w')
json.dump(total_tweets, file)
```

Afbeelding 18: Opslaan tweets in JSON-formaat.

Google_Maps()

De laatste functie van deze scraper is Google_Maps(). Zoals de naam suggereert is deze functie bedoelt voor het scrapen van Google Maps. Deze functie is gebasseerd op de github repository: https://github.com/lyyka/google-maps-businesses-scraper

Net zoals de andere functies leest deze functie alle zoektermen in en vormt ze om tot een gangbaar formaat. Daarna schrijft hij de zoekterm weg naar een bestand genaamd settings, en op basis daarvan wordt de scraper functie uitgevoerd met de meegegeven settings (afbeelding 19).

```
def google_maps():
    inputfile = open("Zoektermen/keywords_base - Copy.txt", "r")
    #inputfile = open("Zoektermen/yeet.txt", "r")
    for word in inputfile:
        #print(word)
        word = word.replace("\n", "")
        settings.SETTINGS['BASE_QUERY'].append(str(word))
    scraper.scrape()
```

Afbeelding 19: Google maps functie

Dockerfile en dockercompose

Docker heeft een Dockerfile nodig zodat de software weet wat er geinstalleerd moet worden in de mini virtuele machine die docker aanmaakt. Daarom is de dockerfile gevuld met de volgende gegevens:

```
FROM debian:latest
MAINTAINER Yassir Laaouissi
WORKDIR /app

COPY . /app

RUN apt-get update
RUN apt-get upgrade -y
RUN apt-get install git openjdk-11-jdk chromium chromium-driver -y
RUN apt install -y gcc g++ python3 python3-pip python3-dev python3-wheel ffmpeg libxslt
RUN pip3 install -r requirements.txt
RUN pip3 install --upgrade git+https://github.com/yunusemrecatalcam/twint.git@twitter_legacy2

CMD ["python3", "2hrs.py"]
```

Tot slot is er een docker compose file om ervoor te zorgen dat de docker VM de output van de scraper weg schrijft naar de host machine. De docker compose file ziet er als volgt uit:

```
version: "3.3"
services:
  web:
    build: .
    container_name: ipfit6_scraper_yassir
    volumes:
    - ./Output:/app/Output
```