Ethische aspecten

Webscraping

Onze webscraper werkt door zich voor te doen als een gewone internet gebruiker. De webscraper voet een request uit naar de website en leest alle inhoud. Hierin gaat de scraper opzoek naar nieuwe webpagina’s op hetzelfde domein en bezoekt deze een voor een. De tekstuele inhoud van iedere pagina slaan we dan op in een database. Hieronder bespreken een aantal voor- en nadelen van deze methode alsook enkele bedenkingen en best practices.

Een “gewone” gebruiker

**De scraper gedraagt zich als een gewone gebruiker en doet zijn ding dus enkel met wat een gewone gebruiker ook kan zien. Hij gaat niet opzoek naar verstopte webpagina’s of probeert geen data te verzamelen uit achterliggende systemen waar de gebruiker geen toegang tot zou mogen hebben. Onze webscraper bezoekt enkel pagina’s waarnaar expliciet verwezen wordt door de webpagina die hij bezoekt. Webscraper kunnen ook zodanig geprogrammeerd worden dat ze opzoek gaan naar verborgen subpagina’s zoals ‘/admin’ of ‘/login’. Deze pagina’s kunnen dan weer op hun beurt gebruikt worden om informatie te bekomen waar de gebruiker geen recht toe heeft.**

**In tegenstelling tot onze scraper zich te laten gedragen als een gewone gebruiker is gebruik maken een API. Een API, Application Programming Interface, is een interface die een website beschikbaar kan stellen om op een programmatische wijze informatie beschikbaar te stellen. Zo stelt Twitter bijvoorbeeld een API beschikbaar om alle tweets op te vragen met een bepaalde zoekterm in. Hierdoor maakt Twitter het onnodig om een agressieve webscraper te gebruiken om de website te bezoeken en een voor een alle tweets op te vragen en te controleren. Moesten te veel mensen dit tegelijkertijd doen zou een website overbelast kunnen geraken en onbereikbaar worden voor gewone gebruikers.**

**Als zo’n API beschikbaar is is dit de beste manier om aan je gezochte data te komen. Voor ons probleem konden we geen gebruik maken van zo’n API dus hebben we moeten opteren voor een agressievere webscraper.**

Denial of service

In het vorig puntje vermelde we al dat indien het aantal requests naar een bepaalde website te groot in een te korte tijd dit ervoor kan zorgen dat de website onbereikbaar wordt. Indien dit opzettelijk gedaan wordt spreken we over een denial-of-service attack. Zoals eerder vermeld gedraagt onze webscraper zich als een gewone gebruiker en wordt iedere pagina slecht eenmaal opgevraagd. Het aantal requests zou dus zeker geen probleem mogen zijn om correct verwerkt te worden. De scraper voert deze requests echter wel op een zeer korte tijd uit en het is dus wel mogelijk dat deze geflagged wordt. Ook kan het zijn dat we tijdens het testen van onze scraper telkens dezelfde site opnieuw bezoeken. Hierdoor kan het aantal requests vlug oplopen en kan de scraper geblokkeerd worden door de website.

Om dergelijke problemen te voorkomen nemen kunnen enkel voorzorgen genomen worden. Allereerst probeer je best de eigenaar van de website te contacteren en bespreek je je intenties. Zo kan er voorzorgen genomen worden zodat de webscraper niet geflagged wordt en niet de toegang tot de website ontzegt wordt. Ook is het aangeraden om de webscraper te gebruiken buiten piekuren. Zo wordt de website niet extra belast wanneer gewone gebruikers deze ook gebruiken.

GDPR

In ons geval gaan we de gescrapte data ook opslaan om verder te gebruiken. Hierbij moeten we opletten dat de data die onze webscraper vindt ook gekopieerd, opgeslagen en gebruikt mag worden door ons. Wij hebben geen zicht op welke data onze webscraper exact verzamelt. We weten enkel dat deze publiekelijk beschikbaar is. Zo zal er zeker contact informatie gevonden worden op de contact pagina van een onderneming. Zolang dit de contact informatie van een bedrijf is zal dit geen probleem zijn. Maar ik vermoed dat voor kleinere ondernemingen het ook zal voorkomen dat adresgegevens en telefoonnummers die op de website vermeld zijn ook voor privé doeleinden gebruikt wordt. Persoonlijke gegevens mogen zonder expliciete goedkeuren niet zomaar opgeslagen worden. Zo moet het bijvoorbeeld te allen tijde mogelijk zijn voor een persoon om zijn data op te vragen, te wijzigen en te verwijderen. Momenteel is het voor ons onmogelijk om na te gaan welke persoonlijke data wij opslaan en zouden wij op dergelijke vragen geen antwoord kunnen geven.

Explainable AI

Vorig jaar werd aan iedere website een score toegewezen aan de hand van hoe vaak een van de zoektermen voorkwam. Hoe meer zoektermen gevonden werden hoe hoger de score en dus hoe hoger de duurzaamheidsrapportage-score. Echter werd er hier geen rekening gehouden met de context waarin de zoektermen in voorkwamen hierdoor kon de eindscore een zeer vertekent beeld geven over de duurzaamheid van een bedrijf. In dit project maken we gebruik van de full text search functionaliteit binnen PostgreSQL. Deze functie houdt rekening met de context waarbinnen woorden voorkomen in een document en zal hieruit een score berekenen. In ons eerste project konden we perfect uitleggen wanneer een bedrijf een score van 100 kreeg, dit bedrijf dus 100 zoektermen vermelde op zijn website. Echter wanneer een website nu een score van 100 krijgt kan ik hierover niets vertellen. Mogelijks kan de ontwikkelaars van deze full text search funtie niet eens vertellen hoe deze score berekent werd. Dit noemen we non exlainable AI. Alle berekeningen en logica gebeurt in een black box die niet expliciet geprogrammeerd werd. Het algoritme heeft zelf bepaald welke factoren belangrijk zijn en welke minder. Hoeveel deze moeten doorwegen bij de berekening van een score en welke helemaal genegeerd kunnen worden. We kunnen dus niet uitleggen aan de klant, stakeholders of in ons geval aan ondernemingen waarom een bedrijf een hoge of lage score toegewezen krijgt. Dit kan zorgen voor wantrouwen in het systeem.