**Python PageRank relevantie van sites**

WISB256 - Programmeren in de Wiskunde

Vrince Vriend (3894460), Roeland Krijgsman (3832171) en Halat Naby (3829030)

***Samenvatting***

In deze opdracht word er een met behulp van python 3 een programma geconstrueerd dat het belang van een webpagina bepaald en dit kan ordenen.

Kernwoorden: Ordening, Belang, PageRank waarde, webpagina’s.

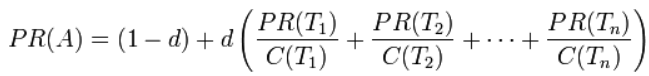
***Inleiding***

PageRank is een methode die de belang van webpagina's ordent, waardoor het zoeken van data op het internet nauwkeuriger en doeltreffender kan gebeuren. De grondleggers van PageRank zijn de oprichters van Google Inc. Het eerste woord in PageRank refereert naar de Google-medeoprichter Larry Page en niet naar het woord pagina. PageRank is een continu proces dat dus up to date is. Het belang van een webpagina wordt bepaald door hoe vaak men verwijst naar een pagina en dit kan door PageRank aangeduid worden in een wiskundige PageRank-formule, zie hieronder. Er wordt een PageRank aangegeven bij een pagina die aangeeft hoe groot de kans eigenlijk is dat iemand op deze pagina terecht komt. PageRank geeft geen (???????) aan relatie met de zoekwoorden die men gebruikt of de werkelijke inhoud van de sites. De PageRank van een pagina zal hoger worden wanneer er verwezen wordt naar andere links met hoge relevantie, maar dit ligt ook aan hoe groot de PageRank is van de pagina met de links erop. De links kunnen gezien worden als stemmen. Hoe meer stemmen, hoe belangrijker de gegeven site is. Hoe belangrijk een stem is, zal ook invloed hebben op de PageRank.

Tegenwoordig is PageRank niet meer zo relevant. Dit komt omdat een site met een hogere PageRank niet hoger in de zoekresultaten te vinden zal zijn als men dit met een site met een lagere PageRank. Dit komt omdat er veel meer factoren binnen het algoritme aanwezig zijn, waarbij een zoekmachine zoals Google rekening mee houdt om webpagina's volledig te kunnen rangschikken op relevantie. De andere factoren houden ook rekening met de zoekwoorden en de inhoud van de sites. Hoewel het eigenlijk geen zin heeft om een hogere PageRank te behalen om hoger in de zoekresultaten voor te komen, geeft PageRank alsnog een zicht op de mate van belang van een pagina aan. Hiermee is het belangrijk dat je weet dat je PageRank in je achterhoofd moet houden, maar dat dit geen grote factor is die de relevantie van een site bepaalt in vergelijking met anderen. In dit verslag zal een programma ontwikkeld worden om het belang van een site te bepalen en hiervoor een PageRank te geven [1,3].

***Hoe bereken je een PageRank?***

Om een PageRank te bepalen voor een site te bepalen zal je ook de links die de pagina omvat mee moeten berekenen en de links buiten de site (bijvoorbeeld als men verwijst naar deze pagina). De wiskundige formule die helpt met de bepalen van de PageRank is als volgt:



PR(A) = De PageRank van een pagina x

C(x) = het totaal aantal uitgaande verwijzende links van pagina x

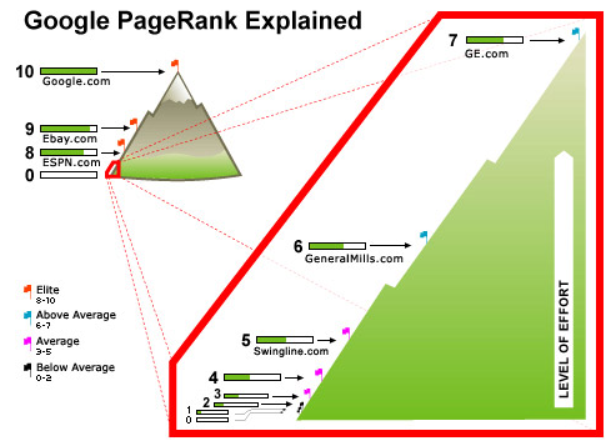
T1,T2,…. Tn = de pagina’s die verwijzen naar pagina A

D = de dempingsfactor (tussen 0-1)

Uit deze formule kan de PR(A) de betekenis hebben dat het de kans aangeeft dat een wikkekeurig persoon op pagina A komt door op random verschillende links te drukken. Een andere betekenis kan bijvoorbeeld bepaling van de populariteit zijn van pagina A.

Een lage waarde dempingsfactor geeft aan dat de persoon die de pagina bezoekt snel zijn interesse verloren heeft en eindigt met het willekeurig aanklikken van de links. Een hoge waarde (1 bijvoorbeeld) geeft dus het omgekeerde aan. De persoon op de pagina heeft meer interesse en zal langer willekeurige links aanklikken. De algemene waarde van d ligt vaak bij 0,85.

Men kan de PageRank verhogen van A doordat andere pagina’s meer te verwijzen naar de pagina a [1]. De PageRank loopt van 0 tot 10. De sites met PageRank 0 zouden (zonder de andere factoren meegeteld die de zoekresultaat plaats bepalen) helemaal niet in de zoekresultaten moeten voorkomen en een sites met PageRank 10 zoals Google, Facebook en CNN zullen boven aan moeten staan. Hieronder staat een figuur beschreven die de PageRank aangeeft van verschillende sites. Hoe hoger ‘’de berg’’, hoe meer inspanning het zal kosten. Een website hoort bij de ‘’Elite’’ als het een PageRank heeft tussen 8 en 10. [3]



***Figuur 1. Uitleg over PageRank [2]***

***Hoe gaat het in zijn werk?***

De eerste stap is informatie van een document per link (document, pagina, webpage) bepalen:

* Titel van de pagina
* URL
* Hyperlink text waarmee verwezen wordt naar de pagina
* Hyperlink text waarmee verwezen wordt naar andere pagina’s
* PageRank waarde

Hierna zal men informatie opzoeken per link:

* Site-informatie (Titel, URL)
  + Bevat meestal de kern van de zaak, id’s en onderwerp.  
    Meer hebben we eigenlijk niet nodig, dus indexeren we niet de complete inhoud van de site. Dit zou namelijk ook meer schijfruimte en meer tijd kosten, en voegt niet voldoende toe om dit te verantwoorden.
  + Op te slaan in een dictionary (zoekwoord, [URL])
    - Te sorteren op PageRank
* Links naar deze pagina
* Links naar andere pagina’s
  + Ook binnen frames zoeken naar links
  + We negeren alles behalve php/html (dus geen plaatjes, scripts, opmaak, etc.)
    - Links binnen plaatjes negeren vanwege advertenties etc.  
      Hierbij accepteren we dat de gewone applicatie van plaatjes laten verwijzen naar pages binnen de site geen hogere PageRank geeft.
* PageRank waarde
  + We gebruiken de eenvoudige waarde, volgens de formule:  
    PR(A) = (1-d) + d ( i=0nPR(Ti)/C(Ti) )

Wat we hier nog wel aan veranderen, is dat C(Ti) altijd een plus één krijgt,  
    om delen door nul te voorkomen.

<http://infolab.stanford.edu/pub/papers/google.pdf>

d: vaak op 0.85 gezet, uit bovenstaande paper

Google itereert voor het hele internet 100 keer voor een goede schatting van de daadwerkelijke PageRank. Aangezien wij minder dan een honderdste van het internet gaan bekijken, is het voldoende om dit slechts één keer te doen.

Vervolgens wordt er een datamodel gecontrueerd:

* URL Table
  + n \* 1, URL -> docID
* DocID Table
  + n \* 1, docID -> URL
* PageRank Table
  + n \* 1, docID -> PageRank
  + Iedere query opnieuw gevuld
* Reference Table
  + n \* n, docID -> docID
  + Wordt gevuld tijdens WebCrawl
  + Bevat 0/1 voor link van site met docID -> andere site met docID
* Outgoing
  + docID -> aantal
* Incoming
  + docID -> aantal
* Webpages
  + Webpage
* Words
  + word -> docID

GUI

* Zoekplek
  + Simple parsing
    - Woord moet de goede letters bevatten, in volgorde (duplicates achter elkaar worden verwijderd)
  + Kan meerdere woorden zijn
* Lijst met informatie van sites
  + Geeft url + titel
  + Gesorteerd op PageRank

Webcrawler

* Zoekt binnen één domein
  + Eén WebCrawl inputfile voor het gekozen domein
    - Dus: altijd een 100% verbonden domein
* Slaat op op schijf
  + Wordt gebruikt voor PageRank

***Techniek***

We gebruiken vooral dictionaries en lists.

***Taakverdeling***

|  |  |
| --- | --- |
| Taken: | Uitgevoerd door: |
| Het verslag | Halat Naby |
| Algoritme | Vrince Vriend en Roeland Krijgsman |

***Referenties***

[1] Webworkshop. What is PageRank?. Geraadpleegd op: 30-06-2015.

URL:http://www.webworkshop.net/PageRank.html.

[2] Een korte uitleg van de PageRank-techniek

URL: http://www.PageRank-check.nl/over\_google\_PageRank.php. Geraadpleegd op: 30-06-2015.

[3] WebSonic. Google - Wat is PageRank. URL: [http://www.websonic.nl/tutorials/google/website\_googlePageRank.php](http://www.websonic.nl/tutorials/google/website_googlepagerank.php). Geraadpleegd op: 30-06-2015.