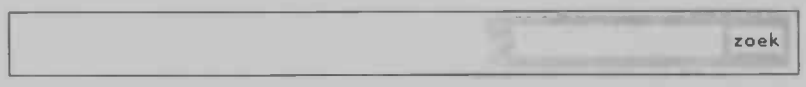
Verbetering van de zoekmachine van het RUG-CMS



Niels Maneschijn

Begeleiders:
Rein Smedinga
Dennis van der Laan

Abstract

Like most other websites, the site of the University of Groningen has an embedded search function. The perceived quality of this search function is not great, but users do not have a clear idea why. Analysis of the search logs reveals that almost half the search actions does not directly lead to satisfying results. In this thesis, we will try to pinpoint the specific weak points in the current search engine.
Also, we will explore different ways to improve search engines, and try to determine the most efficient ones. One of the more interesting possibilities seems to be the use of clustering algorithms to group the results. To explore the feasibility of clustering, some experiments have been done, which lead to not wholly satisfactory results.

There is a computer disease that anybody who works with computers knows about. It's a very serious disease and it interferes completely with the work. The trouble with computers is that you 'play' with them! - Richard P. Feynman

1. Inleiding

De performance van de zoekmachine van het RUG-CMS (het systeem achter de site van de Rijksuniversiteit Groningen, www.rug.nl) kan beter. Mensen vinden niet altijd de informatie die ze nodig hebben. De ordening naar relevantie van resultaten lijkt soms wat willekeurig. Dit levert problemen op wanneer er veel 'hits' zijn. Ook komen de resultaten niet altijd even snel. De indruk bestaat dat hier nog wel wat te verbeteren valt.

Waarom wordt er niet gebruik gemaakt van een standaard-oplossing van bijvoorbeeld Google? De reden hiervoor is dat op het RUG-CMS context-informatie beschikbaar is die niet zichtbaar is voor een conventionele zoekmachine. Dit zou de mogelijkheden beperken om te zoeken op bijvoorbeeld publicatiedatum, auteur en plaats op de site. Daarom wordt er vastgehouden aan een eigen zoeksysteem.

Om dit systeem te kunnen verbeteren is het van belang om te weten wat er precies mankeert aan het huidige systeem. Er zal dus een poging moeten worden gedaan om van een vage omschrijving als 'ik vind niet wat ik zoek' te komen tot een nadere specificatie van de probleempunten.
Wanneer deze problemen nader bekend zijn kan er worden geïnventariseerd wat er voor mogelijkheden voorhanden zijn voor het verbeteren van de ervaren performance. Hierna kan uit deze mogelijkheden een keuze worden gemaakt voor de meest voor de hand liggende opties.
Ten slotte is het van belang dat een theoretisch goede oplossing ook functioneert in de praktijk. Dit zal dan ook getoetst worden.

Dit kan worden vertaald in de volgende concrete onderzoeksvragen:

- **Wat zijn de zwakke punten van de zoekmachine van het RUG-CMS?**
- **Wat zijn er voor mogelijkheden om deze te verbeteren?**
- **Welke van deze oplossingen lijkt het meest veelbelovend?**
- **Hoe functioneert dit in de praktijk?**

In de rest van dit document zal getracht worden antwoorden te geven op deze vragen.

Dit onderzoek is uitgevoerd op het Rekencentrum van de Rijksuniversiteit Groningen (RC), wat verantwoordelijk is voor het beheer en de verdere ontwikkeling van het RUG-CMS. Het RC heeft uitgebreide faciliteiten geboden voor de uitvoering. De auteur is verder dank verschuldigd aan zijn begeleiders, de ontwikkelaars, roommates, meedenkers, en Lonneke van der Plas voor een crash-course Information Retrieval.

Augustus 2006

2. Achtergrond

Het RUG-CMS

Het RUG-CMS is ontwikkeld om een scheiding aan te brengen tussen enerzijds de layout van de website en anderzijds de inhoud. Op deze manier kan voorkomen worden dat er een wildgroei aan verschillende stijlen ontstaat, en wordt het gemakkelijk om deze stijl centraal te beheren. Aan de andere kant wordt het hierdoor makkelijker om de informatie decentraal te beheren. Dit heeft het mogelijk gemaakt dat de inhoud van de website nu wordt beheerd door vele honderden redacteuren.
Het systeem draait op een centrale server, die de pagina's opslaat in een achterliggende Oracle database in een vastgelegd XML-formaat. Wanneer een gebruiker een pagina opvraagt worden de van toepassing zijnde XML-documenten ('XML-objecten') via een reeks XSL-transformaties getransformeerd naar een voor de browser leesbaar HTML-formaat, wat naar de browser van de gebruiker wordt gestuurd. In deze XSL-transformaties ligt het uiterlijk van de site vastgelegd. Het opbouwen van pagina's heet ook wel 'renderen'.
Het renderen is een tijdrovend proces. Daarom worden de 'gerenderde' pagina's en hierbij geproduceerde tussenresultaten bewaard voor later gebruik. Hierdoor kan het opleveren van pagina's drastisch worden versneld. Het nadeel hiervan is dat gebruikers hierdoor soms nog verouderde pagina's te zien krijgen. Dit proces is bekend als 'caching'. Een en ander wordt verwerkt door een in Java geïmplementeerd systeem, in combinatie met een Apache-webserver.
Op het systeem worden verschillende soorten XML-objecten gebruikt. Er zijn objecttypen voor onder andere algemene artikelen, nieuwsberichten, evenementen, FAQ's en formulieren. Elk van deze formaten is precies gedefiniëerd voor het soort informatie wat deze moet kunnen bevatten, en voor ieder soort XML-object is een bijbehorende XSL-transformatie waarin staat hoe deze dient te worden omgezet naar HTML. De objecten kunnen meertalig zijn, en bevatten ook een samenvatting en contextinformatie over onder andere de auteur en de data van aanmaken en bewerken.
De informatie op de site is gerangschikt in verschillende 'portals'. Dit zijn afgebakende stukken website die worden gepresenteerd als een op zichzelf staande website, ook al vallen ze allemaal onder www.rug.nl . Er zijn portals voor de verschillende faculteiten, diverse opleidingen, onderzoeksscholen, diensten en andere organisaties binnen de universiteit. Er zijn ook portals voor bepaalde doelgroepen, zoals studenten, medewerkers en toekomstige studenten. Een bezoeker komt in eerste instantie terecht op de 'corporate' portal. Hiervandaan kan hij verder navigeren naar de verschillende portals. De bezoeker kan natuurlijk ook direct naar de gewenste portal gaan als hij de URL weet.
Er is een hierarchische ordening van portals. Op het hoogste niveau staat de corporate portal. Hieronder staan onder andere de facultaire- en doelgroepenportals. Hieronder kunnen weer portals van bijvoorbeeld opleidingen vallen.
Portals kunnen informatie van andere portals gebruiken zonder deze te kopiëren. Dit gaat door middel van het 'sharing'-mechanisme: dezelfde informatie lijkt op meerdere plaatsen in de boomstructuur te staan. Deze instanties kunnen in één keer worden bijgewerkt, zodat zij allemaal up-to-date blijven.

Zoals eerder gemeld worden de portals decentraal gevuld met informatie. Hiervoor is een organisatie opgezet van portalbeheerders, die het overzicht houden over de inhoud van de portals en het eerste aanspreekpunt hiervoor zijn, en redacteuren, die ieder een voor hen relevant stuk van de portal beheren. Denk bij deze groep aan bijvoorbeeld secretaresses van afdelingen, die de pagina's van hun afdeling up-to-date houden.
Deze organisatie wordt weerspiegeld in een equivalente rechtenstructuur op het systeem. Iedere gebruiker heeft zijn eigen account. Deze accounts zijn gegroepeerd in verschillende groepen gebruikers, bijvoorbeeld per afdeling. De rechten van groepen gebruikers zijn vastgelegd in zogenaamde ACL's (Access Control Lists), waarin per groep wordt aangegeven wat zij mogen doen met objecten die de betreffende ACL toegewezen hebben gekregen. Hierbij gaat het om operaties als lezen, bewerken, maken, en verwijderen.

De werking van zoekmachines

Een zoekmachine, zoals bijvoorbeeld die van het RUG-CMS, werkt door de zoekopdracht van een gebruiker te vertalen naar een zogenaamde 'query' die door de achterliggende database verwerkt kan worden. Hierin staan zowel de zoektermen, als de manier waarop de informatie moet worden geordend, of alle termen aanwezig moeten zijn, het gewenste aantal resultaten, en eventuele andere randcondities. Dit zal doorgaans een SQL-query zijn, of een opdracht in een gespecialiseerd formaat. Vervolgens worden de resultaten die door de database worden opgeleverd gepresenteerd in een formaat dat de gebruiker kan bekijken en waarin hij kan doorklikken naar de resultaten. Doorgaans is dit in de vorm van de titel van een document, samenvatting (of 'snippet'), en link.
De database kan gevuld zijn met de te doorzoeken informatie zelf (zoals op het RUG-CMS), of met informatie die weer verzameld is van andere bronnen en is voorzien van een bronverwijzing (zoals bij zoekmachines als Google). De data is zo geindexeerd dat deze efficiënt doorzocht kan worden.
De termen 'query' en 'zoekopdracht' zullen in de rest van dit stuk door elkaar gebruikt worden in de zin van 'verzameling zoektermen'.

Eisen aan zoekmachines

De ideale zoekmachine voldoet aan een aantal eisen:

- **Alle** relevante zoekresultaten worden gevonden, oftewel het systeem heeft een hoge 'recall'. De **recall** is het aantal relevante resultaten wat gevonden wordt in verhouding tot het totale aantal resultaten wat idealiter gevonden zou moeten worden.
- **Alleen** de relevante zoekresultaten worden gevonden, oftewel de zoekmachine heeft een hoge '**precisie**'. De precisie is de verhouding tussen het aantal relevante gevonden resultaten en het aantal niet ter zake doende gevonden resultaten. De precisie houdt rechtstreeks verband met de 'recall'; als er meer ter zake doende documenten worden gevonden zal de precisie ook toenemen.
- Zoekresultaten worden zo snel mogelijk geretourneerd. Een wachttijd van enkele seconden is acceptabel.

- De zoekresultaten zijn geordend naar relevantie. Hiervoor krijgt ieder zoekresultaat een score. Dit is een getal wat aangeeft hoe relevant een gevonden document zou moeten zijn voor een bepaalde zoekopdracht (zie ook de paragraaf over weging hieronder).
- De zoekresultaten geven de gebruiker een overzicht over het onderwerp, en een idee van hoeveel relevante documenten er zijn.
- De zoekresultaten worden zo beschreven of samengevat dat de gebruiker in één oogopslag kan zien of het betreffende document relevant is.
- De zoekmachine corrigeert fouten van gebruikers.
- De zoekmachine weet wat de gebruiker vraagt om te zetten in wat de gebruiker eigenlijk wil.
- De zoekmachine gebruikt eventueel beschikbare contextinformatie als dit kan helpen om de gewenste resultaten te vinden.
- De gevonden resultaten zijn overzichtelijk.

De bovenstaande eisen zijn in de praktijk niet allemaal haalbaar of zelfs mogelijk strijdig met elkaar. Hier zal dus een balans in gevonden moeten worden.

Problemen met (locale) zoekmachines

Een zoekactie kan op verschillende manieren falen: de zoekmachine vindt geen resultaten, de zoekmachine vindt irrelevante resultaten, of de gebruiker krijgt zoveel resultaten dat hij de relevante pagina's niet meer van de irrelevante kan onderscheiden. Het vinden van irrelevante pagina's is mogelijk zelfs erger dan het vinden van geen pagina's, omdat mensen dan denken dat dat alles is en het dan maar opgeven.
Verder is het mogelijk dat de gebruiker niet precies weet waar hij naar moet zoeken, bijvoorbeeld als deze maar een beperkt begrip heeft van het gebied waarin hij zoekt, of zoekt op een net niet optimaal steekwoord of synoniem. Bovendien blijkt dat bij verreweg de meeste zoekacties slechts één of twee zoektermen wordt opgegeven [Jansen e.a. 99, Jansen e.a. '01], en verder wordt vertrouwd op de zoekmachine voor het zoeken van de meest interessante pagina's. Ook zijn gebruikers niet geneigd om vaak hun zoekquery aan te passen. De overgrote meerderheid probeert het één keer, of doet nog een enkele poging met een aangepaste zoekquery. Overigens is de data van Jansen c.s. (verkregen in 1997, nog voor de lancering van Google) op dit moment wellicht niet meer representatief, omdat de gebruikers meer ervaring zullen hebben gekregen met zoekmachines en nu mogelijk anders zullen zoeken. Daarom zullen ook de zoekacties op het RUG CMS worden geanalyseerd.

De problematiek van locale zoekmachines, dus binnen sites als www.rug.nl, is niet hetzelfde als die van globale zoekmachines zoals Google. Op een locale zoekmachine zullen gebruikers doorgaans zoeken naar meer specifieke informatie. Ze hadden immers al een bepaald idee over de content van de site, en zijn daarom naar deze specifieke site gekomen. Verder zal een locale zoekmachine in minder pagina's zoeken dan een globale. Dit zullen we in het achterhoofd moeten houden bij het bestuderen van dit onderwerp.

Verbeteringsmogelijkheden van zoekmachines

Kleine verbeteringen

Er zijn diverse methoden om de performance van een zoekmachine te verbeteren. Een deel hiervan is redelijk gangbaar en dus standaard voorhanden in bijvoorbeeld Oracle Text, het achterliggende zoeksysteem zoals wordt gebruikt op het RUG-CMS [Oracle '01]. Deze zullen we hier kort behandelen. Van de onderstaande mogelijkheden wordt alleen het negeren van accenten al gebruikt.

Stemming
'Stemming' behelst het afkappen van woorden tot op de grondvorm of woordstam, voor het uitvoeren van de werkelijke zoekactie. Op deze manier wordt voorkomen dat de precieze werkwoordsvorm of meervoudsvorm invloed heeft op het resultaat.
Zoekindices worden gevuld met de grondvorm van woorden. Ook de zoekwoorden worden automatisch afgebroken tot hun grondvorm voor het uitvoeren van de zoekopdracht.
Hiervoor bestaan verschillende algoritmen. Deze algoritmen maken soms fouten in het bepalen van de stam, maar dit hoeft niet erg te zijn, zolang deze fout maar consequent gemaakt wordt. De gezochte termen en de zoekquery zullen dan toch een match opleveren.
Specifiek zijn deze fouten onder te verdelen in 'overstemming' en 'understemming'. Bij overstemming worden woorden te aggressief afgekapt, waardoor woorden met een verschillende betekenis toch op dezelfde grondvorm uitkomen. Hierdoor stijgt de 'recall', maar daalt de precisie. Bij understemming gebeurt het omgekeerde.

Volgens [Tzoukerman '03] blijft het nut van stemming beperkt, en is hier slechts een winst van enkele procenten te behalen op precisie en recall. Dit is waarschijnlijk doordat de gezochte term meestal in meerdere vormen in een document te vinden is.

Alternatieve spellingen
Er zijn veel woorden die op meerdere manieren gespeld kunnen worden. Dit kan bijvoorbeeld het verschil zijn tussen Brits en Amerikaans-Engels ('color' vs. 'colour'), of in het Nederlands het verschil tussen alternatieve spelling, voorkeursspelling en diverse oude spellingen ('quantum' vs. 'kwantum', 'panne(n)koek' etc.). Voor een zoekalgoritme zullen woorden met een andere spelling ook gezien worden als andere termen.
Een oplossing hiervoor is het gebruiken van woordenlijsten om aan te geven wat alternatieve spellingen zijn. Deze kunnen dan automatisch worden toegevoegd aan de zoekquery. 'Kwantum' zal dan kunnen worden vervangen door 'kwantum OR quantum'.
Oracle Text voorziet in kant-en-klare synoniemenlijsten voor Nederlands en Engels.
In plaats van het automatisch toevoegen van alternatieve spellingen aan de zoekquery kan er ook voor worden gekozen om deze alternatieven aan de gebruiker aan te bieden ter overweging. Vergelijk dit met de bekende 'Did you mean:"-suggesties van Google.

Negeren accenten
Het zoeken naar woorden met accenten op letters (é, ë, ê, è) kan moeilijk zijn als de gebruiker niet precies weet hoe het accent moet staan, of wanneer in de relevante documenten gemakshalve deze accenten zijn weggelaten. Dit probleem kan worden ondervangen door bij het indexeren en zoeken deze accenten eenvoudigweg weg te laten.

Fuzzy tikfouten
Het is mogelijk om te anticiperen op kleine tikfouten van gebruikers (één of twee letters verkeerd of omgedraaid) door bij het uitvoeren van een zoekquery ook te zoeken naar woorden die sterk lijken op de termen in de zoekopdracht. Deze extra resultaten vervuilen mogelijk de zoekresultaten wanneer er géén tikfout is gemaakt, maar door deze minder zwaar te wegen dan de precies passende resultaten kunnen deze onderaan worden weergegeven, waardoor de gebruiker er weinig last van heeft.

Naburige woorden
Het is mogelijk om een hogere score aan een zoekresultaat te geven wanneer de zoektermen in het gevonden document dicht bij elkaar staan. Dit heet bij Oracle de NEAR-operator. De achterliggende gedachte is dat wanneer woorden dicht bij elkaar staan ze waarschijnlijk in het zelfde zinsverband gebruikt worden, en dus iets met elkaar te maken hebben.
Dit is uiteraard alleen relevant als er wordt gezocht op meer dan één zoekterm.

Al deze verbetermogelijkheden zullen stuk voor stuk slechts een beperkte invloed hebben op de kwaliteit van de zoekresultaten, maar ze zijn relatief makkelijk in te zetten en zullen gebruikers een wat vriendelijker omgeving bieden die toleranter is voor niet zorgvuldig geformuleerde queries. Het verdient aanbeveling om gebruik te maken van deze mogelijkheden.

Query expansion

Query expansion is het automatisch uitbreiden van zoekqueries met termen die met die term te maken hebben. Dit kan gebruikers helpen die niet precies weten op welke termen ze moeten zoeken. De extra termen komen uit een woordenboek of thesaurus. Dit is een lijst van begrippen die thematisch geordend zijn, aan de hand van de relaties 'gelijk aan' (synoniem), 'onderdeel van' (een entiteit), 'is een' (abstracte klasse), en 'is een instantie van'. Deze relaties zijn ook bekend als hyper- en hyponiemen.
Hieruit worden de woorden die hier qua betekenis dicht bij staan uitgehaald, en vervolgens toegevoegd in de zoekquery. Dit zorgt ervoor dat ook pagina's die aan het onderwerp gerelateerd zijn voorkomen in de zoekresultaten.
Ook deze aanpak zorgt wel voor een mogelijke vervuiling van de zoekresultaten, dus een verlaging van de precisie. De recall kan echter wel toenemen. Bovendien daalt de zoekperformance van de achterliggende database naarmate de zoekterm verder geëxpandeerd wordt; er wordt dan immers gezocht op meer termen.

Deze thesaurus kan bestaan uit een met de hand opgebouwde structuur. Voorbeelden hiervan zijn WordNet (voor de Engelse taal) en EuroWordNet (met onder andere het Nederlands). Ook Oracle levert een thesaurus mee.
Het is echter ook mogelijk om de hierin opgeslagen kennis automatisch te destilleren uit een verzameling documenten, bijvoorbeeld de te indexeren documenten [Curran '02]. Dit laatste heeft dan als voordeel dat de thesaurus altijd ter zake doende informatie bevat, en niet meer dan dat (wat zorgt voor een niet nodeloos slechte performance), het nadeel is dat dan mogelijk niet alle relaties gevonden worden.
De verbanden worden bepaald aan de hand van het samen voorkomen van woorden in documenten en zinnen, en bij geavanceerder methoden ook aan de hand van welke functies deze woorden in de zinnen innemen [Bouma '04]. Hiertoe wordt de tekst geparsed en vervolgens onderworpen aan statistische analyses. Dit blijkt overigens beter te werken met data die in een duidelijk vaststaand formaat is opgesteld, zoals in een encyclopedie. De teksten op de pagina's van het RUG-CMS zijn dus wellicht minder geschikt hiervoor; de schrijfstijlen zullen hier meer uiteenlopen [IJzereef '04].
Extractie van relaties is zeer rekenintensief; extractie van 30.000 termen kostte in 2002 ongeveer een week [Curran '02]. Het is echter mogelijk om door middel van schattingen en geoptimaliseerde wegingsfuncties dit drastisch te versnellen, ook bij het verwerken van zeer grote hoeveelheden data.

Volgens [Tzoukerman '03] blijft het nut van query expansion op basis van een thesaurus beperkt. Het is dus de vraag in hoeverre het nut heeft om dit in te zetten op het RUG-CMS. Het zelf extraheren van een thesaurus lijkt in ieder geval te weinig resultaat te bieden voor de grote inspanning in rekenkracht. Bovendien zal deze dan altijd achterlopen op de content van het systeem.

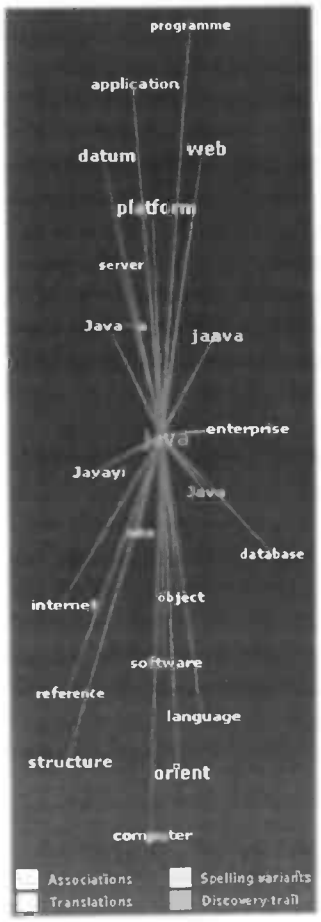
De geparste data zou overigens later gebruikt kunnen worden voor een Question Answering systeem [van der Plas '05]. Dit zijn meer geavanceerde zoeksystemen die antwoorden kunnen geven op vragen zoals 'in welk jaar landde wie op de maan?'. Dit is echter niet waar wij ons op richten.

Grafische representatie en interactief zoeken

Grafisch presenteren van zoekresultaten valt onder te verdelen in het visualiseren van documenteigenschappen en het visualiseren van de verbanden tussen documenten. Dit wordt doorgaans gecombineerd met een interactieve component: de gebruiker wordt niet in één keer voorzien van een verzameling zoekresultaten, maar navigeert door een opeenvolgende reeks van keuzes naar zijn doel toe.
Onder documenteigenschappen vallen hier het aangeven van welke zoektermen relevant zijn voor het gevonden document, en het voldoen aan bepaalde criteria zoals grootte, auteur en publicatiedatum. Binnen het RUG-CMS kan gedacht worden aan het laten zien van het objecttype.
De verbanden tussen documenten kunnen op verschillende manieren worden getoond. Hieronder zullen we enkele manieren behandelen.

Documentnetwerken

Hierbij wordt de verzameling van gevonden documenten getoond als een netwerk met knopen (de documenten). De verbindingen geven aan dat er een verband is tussen de verschillende documenten, en kunnen door kleur, dikte of lengte aangeven hoe sterk het verband is.



Aquabrowser

'Spring embeddings'

In dit model worden de gevonden documenten gerepresenteerd door deeltjes die met veren aan elkaar verbonden zijn. De aantrekkende (of afstotende) kracht wordt gemodelleerd door de overeenkomst tussen de documenten, die in het model de veerconstante bepaalt.[Swan & Allan].

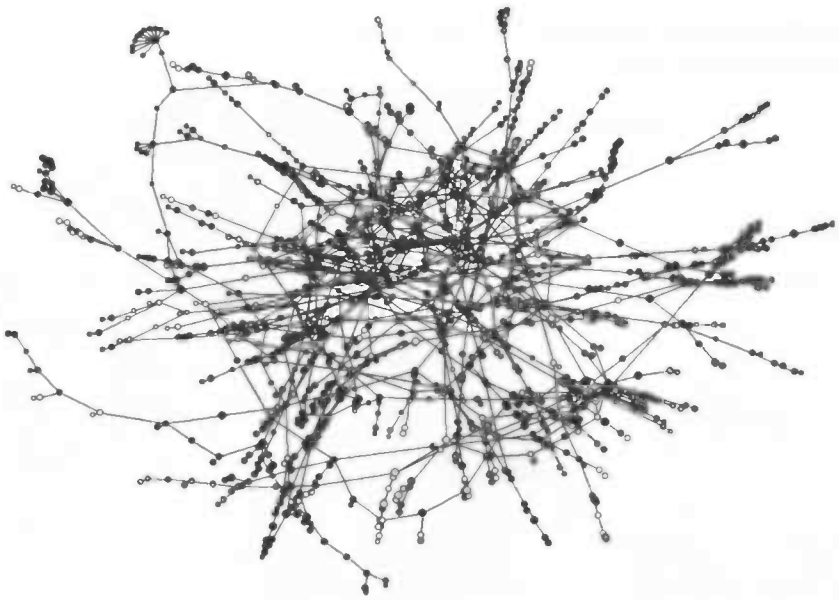


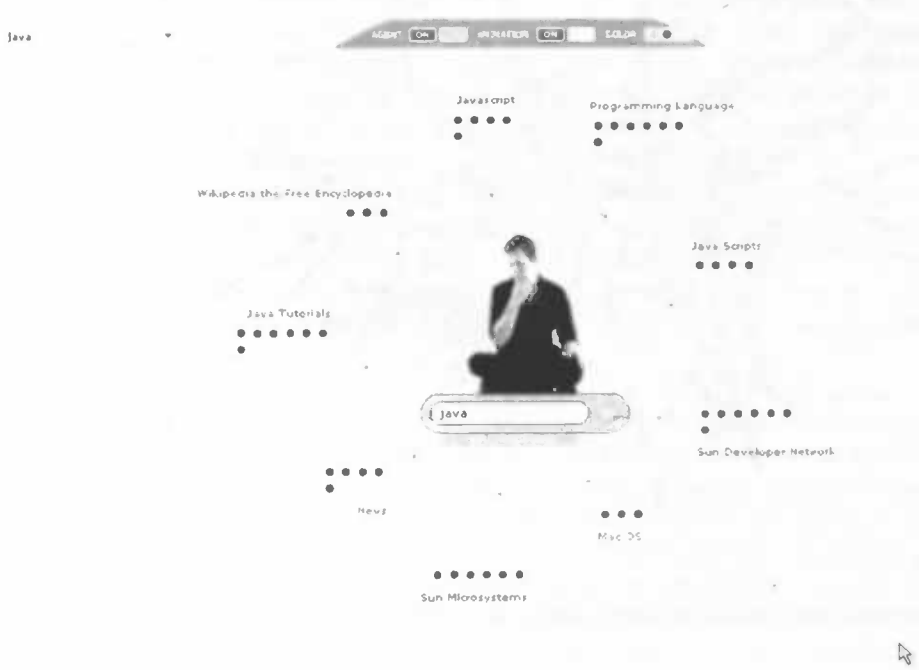
Diagram van familieverbanden in spring embedding [White '98].

Clustering

Clustering is het groeperen van documenten op basis van overeenkomstige woorden of zinsdelen. Hier wordt later nader op ingegaan.

Mind maps i.c.m. clustering

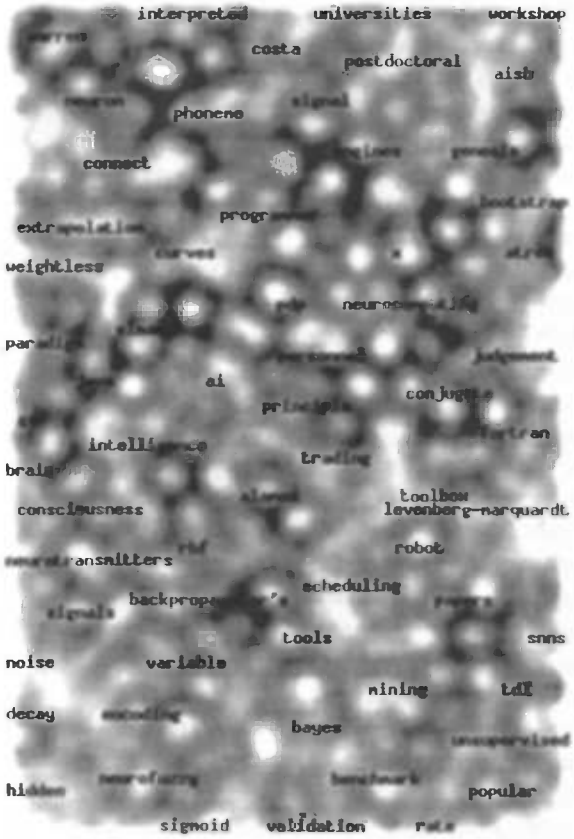
Hierbij wordt de resultaatset eerst geclusterd, en daarna getoond als een 'mind map'. Een voorbeeld hiervan is mex-search.com.



Grafische weergave van clusters op mex-search.com

Self-Organizing Maps

Dit is een methode om door middel van neurale netwerken een tweedimensionale kaart te maken van documenten zodat de meest op elkaar lijkende documenten zo dicht mogelijk bij elkaar liggen. Op deze manier zijn zeer grote hoeveelheden data (in de orde van een miljoen documenten) grafisch te rangschikken. Zie ook het [WEBSOM] project. [Lin, Kohonen]



WEBSOM kaart van 12000 artikelen uit de nieuwsgroep comp.ai.neural-nets. Er kan worden ingezoomd tot op individueel documentniveau door op de kaart te klikken.

Van de bovengenoemde methoden zijn documentnetwerken en spring embeddings meer geschikt voor het navigeren op kleine schaal tussen aan elkaar gerelateerde documenten, en clustering en mind maps beter voor het geven van overzichten over grotere tot zeer grote hoeveelheden documenten. Deze laatste methoden geven de mogelijkheid voor een totaaloverzicht over alle documenten, terwijl de eerste de verbanden tussen individuele documenten kunnen aangeven. Ook zijn de achterliggende technieken geoptimaliseerd voor de betreffende hoeveelheden documenten.

Geavanceerde wegingsmethoden

Met 'weging' wordt bedoeld: een systeem om zoekresultaten te rangschikken naar relevantie. Idealiter zullen de meest interessante zoekresultaten bovenaan de lijst met zoekresultaten komen te staan. Om dit te realiseren zal de onderliggende database van een zoekmachine een bepaalde score meegeven aan een zoekresultaat, normaal gesproken geschaald van 0 (voor een volstrekt niet relevant resultaat) tot 1 (een resultaat wat aan alle zoekcriteria voldoet).
Een score wordt normaliter bepaald aan de hand van het voorkomen van de gezochte zoektermen in het betreffende resultaat. Hierbij wordt niet alleen gekeken naar hoeveel zoektermen voorkomen in het betreffende document, maar wordt ook voor iedere zoekterm bepaald hoe vaak deze voorkomt in verhouding tot de lengte van een document. Een document van honderdvijftig pagina's waarin drie keer een bepaalde zoekterm voorkomt zal immers meestal minder interessant zijn als een document van één pagina met het zelfde aantal voorkomens.
Tevens kan de relevantie van een zoekterm worden bepaald aan de hand van hoe vaak deze voorkomt in de totale documentencollectie. Hoe zeldzamer een woord, hoe beter dit woord te gebruiken is als zoekterm. Documenten waarin deze zoekterm toch voorkomt zullen dus zeer waarschijnlijk relevant zijn, en krijgen daarom een hogere score. Deze laatste techniek is bekend onder de term TF*IDF (Term Frequency * Inverse Document Frequency). [Tzoukerman '03]

Een deel van het succes van Google is te verklaren door het gebruik van het zogenaamde 'Page Rank' algoritme [Page '99]. Hierbij wordt gebruik gemaakt van de graafstructuur van het Web, die gevormd wordt door de hyperlinks waarmee pagina's naar elkaar verwijzen. De achterliggende gedachte is dat naar pagina's die relevant zijn veel verwezen wordt, en weinig naar irrelevante pagina's. Op basis hiervan krijgen pagina's een zogenaamde 'Page Rank'. Deze Page Rank wordt op een iteratieve manier ook weer meegewogen in de pagina's waar deze naar linken, zodat relevante pagina's de score van andere relevante pagina's verhogen. De zoekresultaten kunnen dan uiteindelijk geordend worden naar de Page Rank. Op deze manier zullen de meest relevante zoekresultaten bovenaan te vinden zijn.
Deze methode vertrouwt op het aanwezig zijn van zeer veel links tussen pagina's. Dit zal binnen een (ten opzichte van het hele internet) relatief kleine site als die van de RUG, die bovendien hiërarchisch is opgebouwd, minder het geval zijn. Ook lijkt het waarschijnlijk dat er gezocht wordt naar relatief specifieke informatie. Naar pagina's die een dergelijke kennisniche vullen zal weinig worden gelinkt (in ieder geval vanuit de eigen site), ondanks dat de informatie zeer relevant kan zijn. We gaan er hierbij even vanuit dat het niet haalbaar is om links van buiten het RUG-CMS naar het RUG-CMS te gaan inventariseren.
Daarom lijkt een dergelijke aanpak weinig toe te kunnen voegen aan de ervaren kwaliteit van de zoekmachine van het RUG-CMS.

Overigens is het mogelijk om pagina's met een hoge PageRank te gebruiken als startpunt bij bepaalde clusteringalgoritmen. Hier zullen we niet verder op ingaan.

Contextinformatie gebruiken

Het RUG-CMS beschikt over contextuele informatie die gebruikelijke zoekmachines niet tot hun beschikking hebben, of waar deze weinig aan hebben. Hierbij gaat het om informatie als: welke gebruiker is ingelogd, komt de gebruiker van binnen het RUG-domein, en op welke pagina was de gebruiker toen hij besloot te gaan zoeken? Ook is er extra informatie beschikbaar in de vorm van de directorystructuur van de portals. Uit deze context kan mogelijk worden afgeleid wat de gebruiker precies zoekt.

Wanneer wordt gezocht vanaf een IP-adres wat niet bij de RUG hoort, kan worden besloten om zoekresultaten aan te bieden die gericht zijn op bezoekers van buiten. Te denken valt aan informatie voor scholieren die wellicht aan de RUG willen studeren of mensen die een baan aan de RUG overwegen. In dit geval zouden zoekresultaten van de /studiekiezers of /prospectiveStaff zwaarder gewogen kunnen worden, zodat deze hoger verschijnen tussen de andere resultaten. Andersom zullen mensen die al binnen het RUG-domein zitten uiteraard minder interesse tonen in dit soort informatie, die dan ook minder zwaar gewogen kan worden.

Ook de pagina vanwaar een zoekopdracht wordt uitgevoerd kan informatie geven over de wensen van de gebruiker. Wanneer een bezoeker zoekt vanaf de eerder genoemde /studiekiezers portal zal het hier waarschijnlijk gaan om iemand die nog niet aan de RUG studeert, en derhalve weinig interesse zal hebben in interne stukken als examenreglementen, arbeidsvoorwaarden en tentamenroosters. Dit soort zoekresultaten zou bij voorkeur minder zwaar gewogen moeten worden.
Dit kan door de afstand in de boomstructuur van de website als scoringscriterium op te nemen. Oracle Text heeft hier geen kant-en-klare voorziening voor, maar het lijkt eenvoudig om na het ophalen van de zoekresultaten de scores van de resultaten aan te passen aan de hand van de overeenstemming tussen het pad van het zoekresultaat en het vertrekpad van de zoekactie. Op deze manier zullen zoekresultaten binnen dezelfde portal zwaarder gewogen worden, met een voorkeur voor resultaten die dichtbij in de boom staan.
Hierbovenop zou dan nog een oplossing gezocht kunnen worden die vergelijkbare portals groepeert. Denk hierbij aan categorieën als faculteit, alfa/beta/gamma, of soort organisatie (faculteit, onderzoeksschool, ondersteunende dienst of wervende portal). Wellicht zou dit kunnen worden gecombineerd met het gebruik van doelgroepentags, wat het systeem al ondersteunt. Hierbij zou de scoring kunnen worden gewogen aan de hand van het al dan niet overeenkomen van de doelgroep van een gevonden document en de doelgroep van de pagina waarvandaan gezocht wordt. Deze aanpak heeft als voordeel dat zelfs op documentniveau kan worden aangegeven welk document voor wie interessant is.

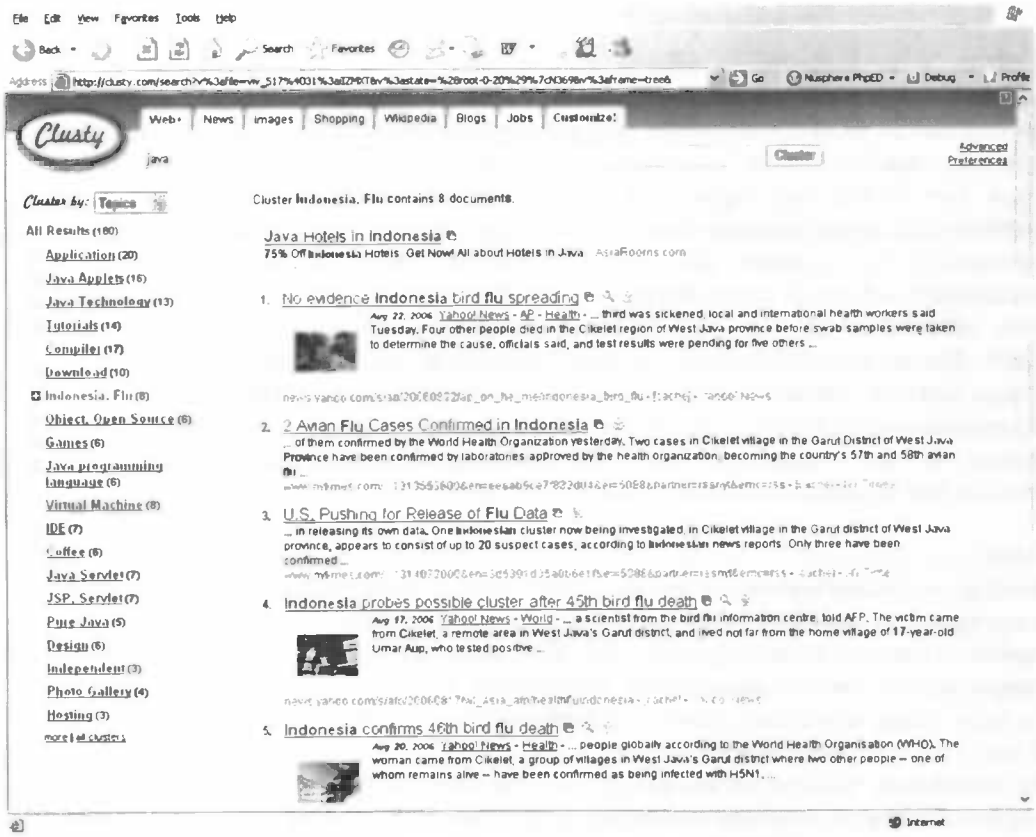
Op een vergelijkbare manier kan ook gebruik worden gemaakt van of een gebruiker is ingelogd, en als wie. Wanneer een gebruiker is ingelogd gaat het kennelijk om een werknemer van de RUG. Het ligt dus voor de hand om deze net zo te behandelen als iemand die werkt vanaf een intern IP-adres (wat doorgaans ook al het geval zal zijn). Ook hier zou deze informatie gebruikt kunnen worden om de informatie van verschillende

portals zwaarder te laten wegen op basis van een vooraf vastgestelde onderverdeling, of de eerder genoemde doelgroepentags.
Behalve of een gebruiker ingelogd is, is ook bekend als wie deze is ingelogd. Binnen het RUG-CMS is bekend welke login (c.q. personeels-, of studentnummer) hoort bij welke afdeling. Dit kan worden gekoppeld aan verschillende interessegebieden (faculteiten, diensten, studies). Aan de hand hiervan kunnen verschillende portals weer een eigen wegingsfactor meekrijgen. Ook dit kan gecombineerd worden met een weging op doelgroepentags.

Clustering van resultaten

Uit onderzoek van de auteurs en van anderen [Mondosoft white paper, Jansen e.a. '99], blijkt dat gebruikers zelden doorklikken naar de tweede pagina met zoekresultaten. Volgens Jansen klikt 58% niet door naar een volgende pagina, volgens Mondosoft, en ons eigen onderzoek, klikt zelfs slechts 1 op de 20 gebruikers daadwerkelijk door op een zoekresultaat op een volgende pagina. Dit betekent gezien het gemiddelde aantal resultaten (172) dat de grote meerderheid van de zoekresultaten helemaal niet bekeken wordt. Het is dus van belang om de belangrijkste zoekresultaten op het eerste scherm weer te geven.
Een manier hiervoor is het groeperen van de zoekresultaten in zogenaamde 'clusters'. Deze clusters worden tijdens het verwerken van de zoekopdracht bepaald en gevuld. Het gaat hier dus om resultaten die voortvloeien uit een zoekactie; deze techniek is geen op zichzelf staande zoekmethode.
De categorieën worden bepaald door termen of zinsneden die de gevonden documenten gezamenlijk hebben. Wanneer deze term of deze zinsnede veel voorkomt in een deel van de gevonden documenten, en juist niet in de rest van de resultaten, is dit wellicht een goede term om een groep te definiëren. Bij het clusteren wordt gestreefd naar een beperkt aantal zo disjunct mogelijke resultaatgroepen. Dit vergroot de kans dat er een bepaalde groep is die duidelijk interessanter is voor de gebruiker, en een titel heeft die de gevonden resultaten duidelijk omschrijft.

Een goed voorbeeld van een toepassing van deze technologie is te vinden op www.clusty.com . Clusty produceert subjectief gezien goed bruikbare clusterlabels. Helaas geeft de producent Vivismo geen informatie over welke algoritmen precies worden gebruikt.



Geclusterde zoekresultaten op clusty.com

Een bijkomend voordeel van deze techniek is dat een gebruiker zich in één oogopslag een beeld kan vormen van een bepaald onderwerp, doordat hij de belangrijkste steekwoorden gepresenteerd krijgt. Op deze manier krijgt hij een overzicht over het onderwerp.

Clustering van zoekresultaten wordt normaliter gedaan op uittreksels ('snippets') van webpagina's, omdat het te lang duurt om de volledige tekst van alle gevonden pagina's op te halen.

Zamir stelt de volgende eisen aan clustering: [Zamir '99]

- Relevantie: clusters moeten relevant zijn ten opzichte van de zoekopdracht
- Bruikbare omschrijvingen: de titel van de cluster moet in een oogopslag duidelijk maken wat de inhoud van deze cluster is.
- Overlappende clusters: documenten kunnen in meerdere categorieën vallen, en moeten daar dan ook in getoond worden
- Uittreksels: het algoritme moet goed presteren bij gebruik op 'snippets'.
- Snelheid: de gebruiker moet geen vertraging merken in vergelijking met een conventionele zoekmachine.
- Incrementeel verwerken: om snel resultaten weer te kunnen geven moet het algoritme al beginnen met werken terwijl nog niet alle gegevens binnen zijn.

Methoden:
Er zijn een aantal gangbare methoden om gegevens te clusteren. Hieronder zullen we enkele behandelen.
Het gaat hier steeds om algoritmen die werken op een beperkte selectie van zoekresultaten die wordt gemaakt door een achterliggende database ('online clustering'), in tegenstelling tot systemen die van te voren de totale hoeveelheid informatie geautomatiseerd indelen in categorieëen ('offline clustering'). Hoewel het van te voren clusteren efficiënt is en een snelheidsvoordeel oplevert zal dit slechtere resultaten opleveren. De optimale clustering van een deelverzameling zal namelijk bijna altijd anders zijn dan een deelverzameling van de optimale clustering van de hele verzameling (in dit geval een van te voren gemaakte clustering). Het meest extreme voorbeeld hiervan is wanneer de deelverzameling slechts één voorgeproduceerde cluster bevat. Deze verzameling zou bij achteraf clusteren verdeeld zijn in meerdere clusters.

K-Means
K-Means is een relatief oud algoritme [Zamir 99]. Hierbij wordt uitgegaan van een van te voren opgegeven aantal clusters (k), en vervolgens worden de documenten verdeeld over dit aantal clusters, ervanuitgaande dat de clusters 'bolvormig' zijn in een vectorrepresentatie van de documenten. Dit gebeurt in een meerdimensionale ruimte waarin ieder uniek woord een dimensie representeert, en waarin lengtes staan voor het aantal keren dat een woord voorkomt.
Bij de initialisatie worden de documenten willekeurig verveeld over de k clusters. Vervolgens worden er steeds twee stappen uitgevoerd: eerst worden de zwaartepunten van de geproduceerde clusters bepaald, en vervolgens worden alle documenten toegewezen aan de cluster met het dichstbijzijnde zwaartepunt. Deze twee stappen worden uitgevoerd tot het resultaat convergeert, oftewel de clusters constant blijven.
Een nadeel van deze aanpak is dat deze zich niet leent voor incrementeel gebruik. Een ander nadeel is dat niet goed te voorspellen is hoe lang het duurt voordat de clusters convergeren. Ook zullen clusters niet overlappen, wat bij gebruik in een clusterende zoekmachine ervoor kan zorgen dat gebruikers een resultaat missen doordat ze in de verkeerde cluster zoeken.

Suffix Tree Clustering (STC)
STC is een algoritme wat gebruik maakt van Suffix Trees. Dit is een datastructuur waarin strings erg efficiënt doorzocht kunnen worden. Bovendien kan deze in lineaire tijd worden opgebouwd. [Zamir '99]
STC werkt in drie stappen: in de eerste stap worden de documenten geparsed en ontdaan van leestekens, nummers en tags. Hiervan blijft alleen de afbakening van de zinnen bewaard. Verder wordt er een stemming-algoritme gebruikt om meervoudsvormen te verwijderen.
In de volgende stap worden alle documenten verdeeld over (overlappende) sets met overeenkomende zinsdelen. Dit gebeurt door de corresponderende zinsdelen op te slaan in een suffix tree. Hieruit kunnen dan weer groepen documenten met overeenkomstige termen worden gedestilleerd. Deze groepen worden gerangschikt door middel van

TF*IDF, waarna de hoogst scorende groepen worden geselecteerd voor verdere verwerking.
In de derde stap worden deze sets samengevoegd tot clusters, op basis van de overlap tussen documenten die in deze sets zitten. Zo worden op elkaar lijkende clusters vermeden.
Het hele STC-algoritme draait in lineaire tijd en produceert overlappende clusters. Verder kan het met enige aanpassingen incrementeel werken.

HAOG-STC
HAOG (Hierarchical Arrangement of Overlapping Groups)-STC is een variant op STC waarbij in plaats van losse clusters een hiërarchische boomstructuur van clusters wordt gegenereerd. Op deze manier kan de gebruiker door een boom van clusters navigeren naar zijn einddoel. Het nadeel hiervan is een vergroot geheugengebruik.

Lingo
Lingo [Osinski '04-2] is een algoritme wat gericht is op zo accuraat mogelijke clusterbeschrijvingen. Om dit te bereiken wordt afgeweken van het gebruikelijk stramien van eerst clusteren, dan labels genereren: eerst wordt een lijst samengesteld van mogelijke clusterbeschrijvingen. Dit gebeurt aan de hand van de frequentie van het voorkomen van zinsdelen. De filosofie hierachter is dat het zinloos is om een cluster te hebben als hier toch geen geschikt clusterlabel bij te vinden is.

Voordat de verdere verwerking begint wordt de data opgeschoond. Dit betekent in de eerste plaats dat alle HTML tags en vreemde karakters worden verwijderd, behalve de scheidingstekens tussen zinnen (deze worden later gebruikt bij het bepalen van de grenzen van zinsdelen). Hierna wordt bepaald welke taal het document heeft, waarna een bij deze taal horende stopwoordenlijst gebruikt kan worden om overbodige informatie te verwijderen.
Deze opgeschoonde data wordt verdeeld in veel voorkomende zinsdelen (woordcombinaties). In deze fase kunnen ook synoniemen worden teruggevoerd tot de bijbehorende abstracties.
Uit deze veel voorkomende zinsdelen worden clusterlabels gegenereerd aan de hand van hoe representatief en onderscheidend deze zijn voor een deel van de documenten, door middel van het eerder genoemde TF*IDF en SVD (Singular Vector Decomposition). Dit gebeurt weer in een vector- of matrixmodel van de zinsdelen.
Pas hierna worden de documenten bij de gegenereerde clusterlabels gezocht, en voorzien van een score.

Volgens onderzoek door de ontwerpers van het algoritme (onder een beperkt aantal gebruikers) wordt een ruime meerderheid van de gevonden clusters en clusterlabels bruikbaar gevonden [Osinski '04-3]. Ook blijkt het overgrote deel van de resultaten overeen te stemmen met de cluster waarin ze geplaatst zijn. Verder claimen de ontwerpers op basis van een merge-then-cluster test (zie verderop) dat de geproduceerde clusterstructuur kernachtiger en meer divers is dan degene geproduceerd door STC.
Variëren van default parameters verandert weinig aan de resultaten.

Doordat de clusterlabels niet uit de documenten zelf worden gegenereerd, maar uit een beperkte subset van veel voorkomende zinsdelen, kan Lingo sneller clusteren dan 'conventionele' clustermethoden.
Er wordt nog gewerkt aan het verbeteren van het Lingo-algoritme. Op dit moment is er behalve een open-source versie ook een commerciële versie van de derde generatie beschikbaar. Deze versie is voorzien van extra opties als hiërarchisch clusteren, het filteren of promoten van labels, het ondersteunen van synoniemenlijsten, en is geoptimaliseerd voor een betere performance. [Carrot-search]
Lingo kan incrementeel documenten clusteren.

Andere algoritmen
Er zijn nog vele andere algoritmen voor het clusteren van documenten, welke hier verder niet behandeld worden. Enige voorbeelden zijn Buckshot, Fractionation, group-average hierarchical clustering (GAVG) en het single-pass algoritme.

3. Probleemanalyse

Problemen RUG-zoekmachine

Om meer inzicht te krijgen in de problemen die gebruikers ondervinden bij het gebruik van de RUG zoekmachine zijn gedurende ruim twee weken de zoekacties van gebruikers gelogd, inclusief contextuele informatie zoals het gezochte objecttype, doorzochte portal, al dan niet ingelogd zijn, en vertrekpunt van de zoekactie. Ook is geregistreerd hoeveel hits deze zoekopdrachten hebben opgeleverd, op welke zoekresultaten is doorgeklikt, en op welke positie deze stonden. Dit heeft een totaal van 22387 zoekacties en 16711 doorkliks opgeleverd, gemeten van 13-7 tot 28-7-2006, welke zijn opgeslagen en verder verwerkt in een kleine database. De resultaten zijn mogelijk iets gekleurd door de vakantieperiode en daarmee verschuivende aard van de bezoekers, maar het lijkt niet aannemelijk dat dit een grote invloed heeft.
Een substantieel deel van deze zoekacties (ruim 600) is dubbel uitgevoerd binnen dezelfde sessie. Vermoed werd dat dit veroorzaakt werd doordat gebruikers twee keer achter elkaar op 'zoeken' klikken. Het blijkt echter dat tussen twee identieke zoekacties doorgaans een halve tot enkele minuten zit, en nooit enkele seconden. Het gaat hier dus om gebruikers die een zoekactie uitvoeren, iets anders proberen, en het daarna weer proberen met de vorige zoekopdracht. Doordat doorkliks worden bijgehouden op basis van sessie-ID en zoekstring zullen in genoemde gevallen de doorkliks dubbel meegerekend worden. Om de statistieken hiermee niet te vervuilen zijn de dubbele gevallen verwijderd. Er blijven dan 21771 zoekacties over.

Algemene observaties

In eerste instantie mislukt bijna de helft van de zoekacties, hetzij doordat er geen enkel resultaat wordt gevonden, hetzij doordat er niet wordt doorgeklikt op de gevonden resultaten en de resultaten dus kennelijk niet interessant worden bevonden. In totaal faalden 10454 van de onderzochte 21771 zoekacties bij de eerste poging (48%) . In een deel van deze gevallen werd de oorspronkelijke zoekopdracht aangepast. Hier wordt hieronder verder op ingegaan.

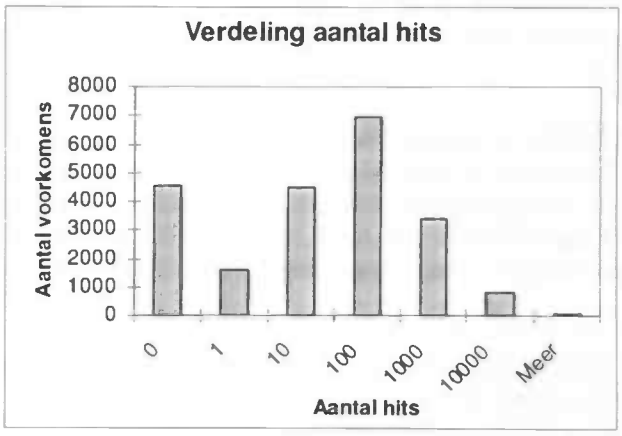
Er wordt veel gezocht naar personen: ongeveer een kwart van de zoekacties lijkt gericht op het zoeken van personen, waarschijnlijk met het doel contactgegevens te vinden. Dit is de grootste groep die te onderscheiden is tussen de zoekacties. Het lijkt dus aanbevelenswaardig om de persoonlijke pagina's prominenter aan te bieden. Dit zou bijvoorbeeld kunnen door gevonden persoonlijke pagina's in een apart kader op de zoekpagina's te presenteren boven of naast de overige resultaten, of door deze pagina's zwaarder te laten wegen en aldus hoger in de zoekresultaten te laten verschijnen. Op dit moment komen de persoonlijke pagina's namelijk midden tussen de overige resultaten terecht.

Er zijn enkele bezoekers (52 zoekacties) welke zoeken op een bepaalde reeks van woorden door deze tussen dubbele aanhalingstekens te zetten, zoals zij gewend zijn van bijvoorbeeld Google. Het huidige zoeksysteem negeert deze aanhalingstekens. Gebruikers zullen desondanks wel de gewenste resultaten vinden, maar zullen niet in staat zijn om te filteren op een exacte woordvolgorde. Deze functionaliteit lijkt niet node gemist te worden, gezien het zeer beperkte aantal pogingen wat ondernomen wordt in deze richting (in de orde van fracties van promilles van het totaal).

De administratie van de doorkliks is niet geheel foutloos, de oorzaak hiervan is onbekend. De fouten zijn niet te reproduceren. Mogelijk worden in bepaalde condities doorkliks aan de verkeerde zoekactie toegeschreven. Van de bestudeerde collectie zijn er 830 gevallen waarbij een aangeklikt document een hoger rangnummer heeft dan zou kunnen volgens het aantal hits. Hiervan zijn er 580 blanco hits. Bij handmatige controle blijkt dat het aantal hits in deze gevallen wel klopt.
Met deze afwijking is bij het verwerken van de resultaten verder geen rekening gehouden. Het is dus mogelijk dat de getallen over het aantal en de verdeling van de doorkliks een enigszins vertekend beeld geven.

Gevonden resultaten

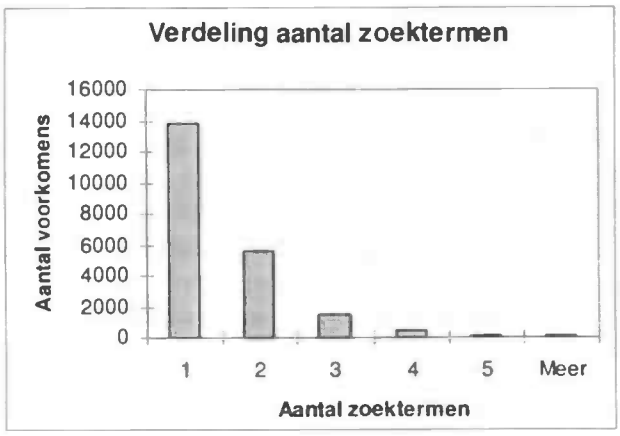
De beschouwde zoekacties leverden een gemiddeld aantal van 172 hits op. In het volgende histogram is inderdaad te zien dat relatief veel zoekacties tussen de 100 en 1000 hits opleveren.



Bij een doorsnee zoekactie zal een gebruiker dus in de orde van tien pagina's met zoekresultaten aangeboden krijgen (met tien resultaten per pagina).

Aantal zoektermen

Ook het aantal gebruikte zoektermen is bepaald en uitgezet in een histogram. Hierbij zijn de door 'comment-spam'-bots uitgevoerde zoekacties buiten beschouwing gelaten. Hier wordt nog nader op ingegaan.

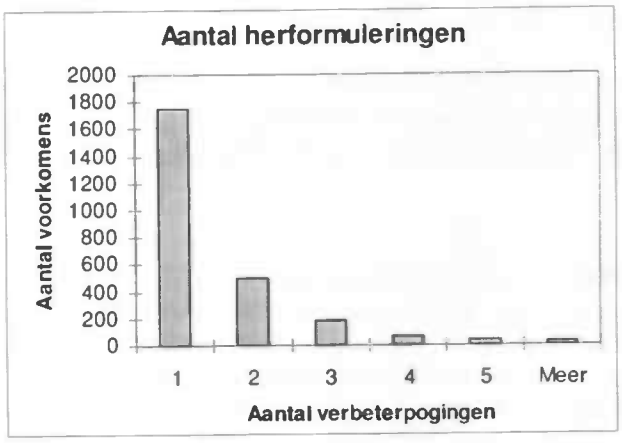


Hier is te zien dat de constatering van Jansen c.s. dat de de grote meerderheid van de gebruikers slechts één of twee zoektermen gebruikt nog steeds van toepassing is, in ieder geval voor de gebruikers van het RUG-CMS. [Jansen e.a. 99, Jansen e.a. '01]

Er zijn gebruikers die zoeken op meer termen (met een uitschieter naar 22, voor een artikeltitel), maar dit is zeldzaam: er zijn slechts 81 zoekacties met meer dan vijf termen geregistreerd. Het gemiddelde aantal hits van deze laatste groep bedroeg 8. Dit gemiddelde wordt omhoog getrokken door een paar uitschieters, ruim de helft (48) van deze groep vindt geen enkel resultaat. Te specifieke queries hebben dus een ondergemiddeld resultaat, te oordelen naar het aantal gevonden pagina's.

Herformuleringen

Jansen c.s. stellen dat de meerderheid van de bezoekers hoogstens één maal, soms twee maal hun zoekopdracht aanpast. De meerderheid van de bezoekers herformuleert helemaal niet. Dit blijkt in overeenstemming te zijn met de bezoekers van het RUG-CMS, getuige het volgende histogram.



Om te voorkomen dat de herformuleringen van herformuleringen dubbel geteld werden is hierbij uitgegaan van de eerste zoekpogingen binnen een sessie, waarbij het gaat om 13824 gevallen. Hiervan werden in totaal 4029 opdrachten gevolgd door één of meerdere nieuwe zoekpogingen (gemiddeld 1,57).

Doodlopende queries

Van de onderzochte zoekopdrachten leverden 4560 queries (**21%**) geen enkel resultaat op. In **56%** (2568) van deze gevallen werd een nieuwe poging gedaan met een aangepaste query, wat leidde tot 2038 queries met resultaten. Deze queries leverden uiteindelijk 988 doorkliks op. 3572 van deze queries (**16%**) leidden dus uiteindelijk tot geen enkele doorklik.

Er is vanuitgegaan dat wanneer binnen dezelfde sessie binnen twintig minuten een nieuwe zoekopdracht gegeven wordt, dit een herformulering van de voorgaande is. Nadere inspectie leert dat dit inderdaad bijna altijd het geval is. Wanneer de tijdsspanne ruimer genomen wordt loopt het genoemde getal nog op, maar dit zijn deels ongerelateerde zoekacties.

In een deel van de gevallen waren er duidelijke oorzaken aan te wijzen voor het mislukken van de zoekopdracht. Om enig kwantitatief inzicht te krijgen in deze oorzaken is van de eerste honderd gevallen bepaald wat de meest waarschijnlijke oorzaken waren van het mislukken van de zoekopdracht (een of meerdere oorzaken per query). De hier gevonden verhoudingen zijn niet bijzonder nauwkeurig, maar lijken op het oog representatief voor de rest van de data.
Bij nader inzien was het beter geweest om honderd willekeurige doodlopende queries te bestuderen in plaats van de eerste honderd, omdat zo series bij elkaar horende pogingen opduiken. Dit is echter erg tijdrovend en is derhalve niet uitgevoerd.

Informatie niet aanwezig
Slechts een bescheiden deel van de zoekacties mislukt doordat er echt geen informatie te vinden is bij de betreffende zoektermen. In de nader beschouwde gevallen gaat het om **7%** van de zoekacties..

Te specifieke queries
Het zoeksysteem is in de huidige vorm niet tolerant tegenover te specifieke zoekacties: verschillende zoektermen worden verbonden met de AND-operator. Met andere woorden, **alle** zoektermen moeten voorkomen op de gezochte pagina. Dit leidt ertoe dat uitgebreide zoekopdrachten kunnen falen op één ongunstig gekozen zoekterm. In **9%** van de nader onderzochte gevallen faalde de zoekactie door het gebruik van een te uitgebreide query. Het lijkt het overwegen waard om in plaats van de AND-operator te kiezen voor een OR-operator, zodat dit niet meteen leidt tot een mislukte zoekactie. Dit zal leiden tot een lagere precisie, maar een hogere recall. Bovendien valt er voor het verlies aan precisie te compenseren met een verstandige weging.

Voornamen
Bij het zoeken naar personen wordt regelmatig gezocht naar namen inclusief voornaam (of soms aan elkaar gespelde voorletters). Omdat op veel persoonlijke en andere relevante pagina's de voornaam van de betreffende persoon niet genoemd wordt, en zoals hierboven beschreven de AND-zoekoperator wordt gebruikt, zullen deze pagina's niet gevonden worden. Of erger, de gebruiker kan stranden op een irrelevante pagina (denk aan 'foto's van de vakgroep' of iets dergelijks), en vervolgens het zoeken opgeven.
10% van de mislukte zoekacties kan worden toegeschreven aan het zoeken op voornamen. Nadere beschouwing van een deel van de onsuccesvolle zoekopdrachten leert dat minder dan de helft van de bezoekers die zoeken inclusief voornaam op het idee komt om het nog eens te proberen zonder voornaam of met alleen een voorletter!

Verkeerde portal
De huidige zoekmachine laat gebruikers standaard zoeken binnen de portal waar zij op dit moment zijn (met uitzondering van de corporate portal). Dit leidt ertoe dat zij geen informatie zullen vinden op andere portals.
Ook bij het zoeken naar personen zullen persoonsobjecten niet gevonden worden als er in een specifieke portal gezocht wordt (tenzij het /staff is).
In **38%** van de nader bekeken gevallen is het waarschijnlijk dat het falen van de zoekactie werd veroorzaakt doordat werd gezocht in een verkeerde portal! Het lijkt dan ook verstandig om gebruikers niet 'by default' slechts te laten zoeken binnen de huidige portal. Een betere keuze zou zijn om de gebruikers deze keuze later te laten maken, om indien nodig de hoeveelheid resultaten te beperken. Anders zullen zij eerder concluderen dat de gezochte informatie er niet is.

Verkeerde objecttype
Op dit moment wordt standaard gezocht binnen webpagina's, dat wil zeggen in speciaal voor het RUG-CMS opgestelde objecttypen als (onder andere) algemene artikelen, evenementen en nieuwsberichten. Er is echter meer content voorhanden op het CMS, meestal in de vorm van PDF- en Word-documenten. De gebruiker zal echter geen weet

hebben van het bestaan van eventuele relevante documenten van de laatste soort, wanneer hij hier niet expliciet naar zoekt.
Van de nader onderzochte zoekacties leidde **15%** tot 0 hits, terwijl er waarschijnlijk wel relevante informatie aanwezig was in PDF- of Word-formaat. Een zeer klein deel van de gebruikers (**1%**) zocht op een specifiek objecttype terwijl dit duidelijk niet de bedoeling was.

Spambots
Het blijkt dat de RUG-site regelmatig bezocht wordt door zogenaamde "comment-spam" bots. Deze proberen berichten achter te laten op gastenboeken om bepaalde produkten onder de aandacht te brengen. Deze bots proberen ieder formulier wat zij passeren in te vullen met een reclameboodschap. Van het totale aantal onderzochte zoekacties zijn 235 veroorzaakt door deze spambots. Een groot deel adverteert ringtones.

http://www.la-ringtones.com/mp3/ ringtones site. Download ringtones FREE, Best free samsung ringtones, Cingular ringtones more. from website .
Hello, Great site! http://pet.cyberfreehost.com/
I just wanted say thank you..your site is very nice well thought out... http://cheap-cigarette.be/cheap_marlboro_cigarette/

Behalve het vervuilen van de zoekstatistieken hebben deze spambots een verwaarloosbare invloed op de zoekperformance. Hier zal dan ook niet verder op worden ingegaan.

Tikfouten
Een groot deel van de zoekacties zonder resultaat worden veroorzaakt door simpele tikfouten. Van de onderzochte gevallen zonder hits was **33%** verkeerd gespeld.

Creshe
Kresche
Kreshe
Cresche
[gebruiker geeft op]

Doorgaans gaat het hier om enkele weggelaten of omgedraaide letters, en niet om totaal verhaspelde woorden. Er is dus veel winst te behalen door het inschakelen van de al aanwezige 'Fuzzy search'-mogelijkheden.
Een minder ingrijpende mogelijkheid is om de gebruiker bij een geconstateerde waarschijnlijke spelfout een suggestie aan te bieden voor verbetering, zoals de hiervoor aangehaalde 'Did you mean...?'.

Woordvorm
Hieraan gerelateerd komt het met enige regelmaat voor dat een gebruiker een afwijkende woordvorm gebruikt. Wanneer er gezocht wordt op een meervoudsvorm zal de zoekmachine geen pagina's retourneren waar alleen de enkelvoudsvorm in staat. Van de nader bestudeerde queries liep **3%** hier aanwijsbaar op spaak. Dit beperkte voorkomen is in overeenstemming met de literatuur. Het nut van het gebruik van een stemmer is dus beperkt.

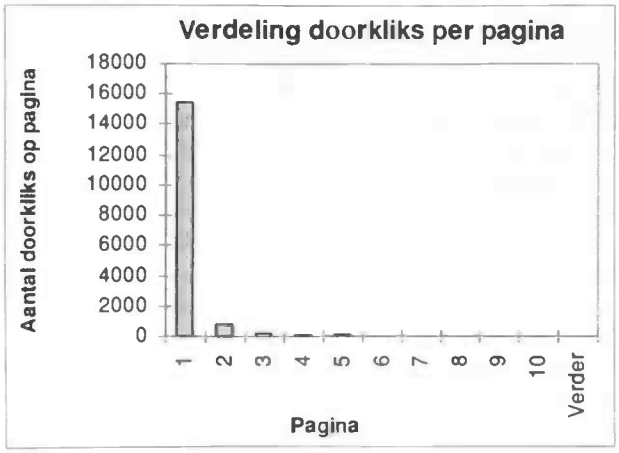
Verder komt het hier vaak (**8%** van de gevallen) voor dat gebruikers woorden aan elkaar spellen (`buluitreiking`, `stagebeoordeling`, `bulaanvraag`, `avv-vakken`). Wanneer deze woorden los gespeld zouden worden zouden er veel meer resultaten gevonden kunnen worden. Hier liggen dus mogelijkheden voor een systeem wat aan de hand van een woordenlijst samengestelde woorden opbreekt in de samenstellende woorden, om daar mee verder te zoeken.
Dit laatste is een probleem wat redelijk specifiek is voor het Nederlandse taalgebied, omdat het gebruik van samenstellingen hier veel vaker voorkomt. Een standaardoplossing is dus waarschijnlijk niet voorhanden.

Queries zonder doorkliks

Behalve de zoekacties zonder resultaten is het ook interessant te weten welke zoekacties wel resultaten opleverden, maar de gebruiker niet konden verleiden tot het klikken op (minstens) één van de zoekresultaten. Dit kan betekenen dat de gebruiker geen relevant lijkend zoekresultaat te zien kreeg, een nieuwe zoekquery geformuleerd heeft, of simpelweg zijn interesse verloor. Van de 17210 queries die minstens één resultaat opleverden werd er op **34%** (5894) niet doorgeklikt. Van deze laatste gevallen werden 1857 (**32%**) gevolgd door één of meerdere (gemiddeld 2,77) geherformuleerde queries, waarbij in 1720 gevallen hits werden gevonden. 891 van deze herformuleringen leverden alsnog doorkliks op (in totaal 1420 doorkliks).
In totaal **29%** van de gevallen (5003) werd een zoekactie met resultaten dus niet gevolgd door enige doorklik, ook niet na een herformulering.

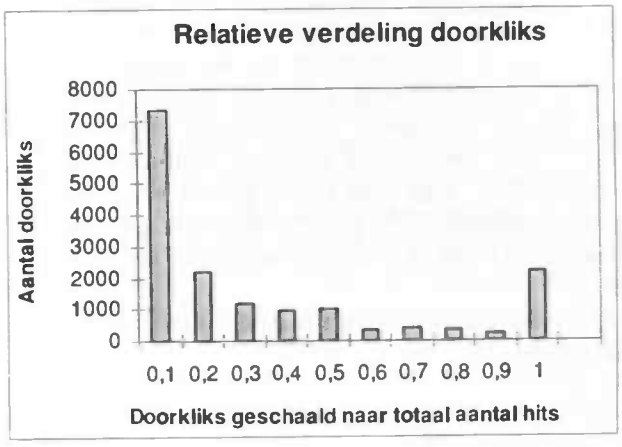
Verdeling doorkliks

Volgens Jansen c.s. wordt er zeer weinig doorgeklikt op zoekresultaten die niet op de eerste pagina met resultaten te vinden zijn. De metingen op het RUG-CMS ondersteunen deze stelling, getuige het onderstaande histogram. Bij het bekijken hiervan dient in het achterhoofd gehouden te worden dat de gemiddelde zoekactie hier 172 resultaten retourneert, wat overeenkomt met 18 pagina's.



Dit kan twee dingen betekenen: óf het RUG-CMS levert een buitengewone prestatie in het naar relevantie rangschikken van resultaten, zodat de gezochte gegevens altijd op pagina 1 staan, óf de gebruikers willen niet doorklikken naar volgende pagina's. De waarheid zal ergens in het midden liggen.

Uiteraard wordt dit beeld vertekend doordat bij het vinden van weinig resultaten deze altijd op de eerste pagina('s) zullen komen te staan. Daarom is er ook gekeken naar de relatieve verdeling geschaald naar het totaal aantal gevonden hits. Ook hier blijkt dat er vooral op de eerste links geklikt wordt. De piek bij 1 wordt veroorzaakt door een afrondingseffect bij een laag totaal aantal hits.



Conclusies

De belangrijkste oorzaken van problemen zijn het zoeken in de verkeerde portal, en het maken van tikfouten. Andere noemenswaardige oorzaken zijn het zoeken op het verkeerde objecttype, het zoeken op voornamen, het gebruiken van een te specifieke zoekterm, en het gebruik van een verkeerde woordvorm. Deze problemen zijn technisch goed te ondervangen door middel van respectievelijk standaard zoeken op de hele site, het inschakelen van de 'fuzzy search', het aanvullen van de persoonlijke pagina's met voornamen, het zoeken door middel van de OR-operator, en het opknippen van zoektermen. Dit laatste vereist nog wel nader onderzoek.

Keuze verbeteringsmogelijkheden

Er lijkt veel te verwachten van de genoemde relatief eenvoudige ingrepen als het afvangen van tikfouten, het zoeken met de OR-operator, het beter presenteren van gevonden personen en relevante documenten, en het uitbreiden van het standaard zoekgebied. Het is zonder meer aan te raden om deze zaken te implementeren.

Van de geavanceerdere mogelijkheden om de performance van de zoekmachine te verbeteren lijkt clustering van zoekresultaten het meest veel belovend. Aangezien de zoekresultaten worden gegroepeerd en er dus een orde van grootte meer zoekresultaten op een pagina past, biedt dit een oplossing voor het probleem dat gebruikers meestal niet verder doorklikken dan de eerste pagina met resultaten. Bovendien is het een goede methode om gebruikers te 'verleiden' tot het de facto ingeven van extra zoektermen om het resultaat nader te specificeren, in plaats van de meestal enkele zoekterm die normaliter gebruikt wordt. Zij hoeven deze term niet expliciet in te geven, klikken om een cluster te openen volstaat. Ook biedt het de gebruikers de mogelijkheid om een overzicht te krijgen over de onderdelen van een bepaald onderwerp, en op die manier met een betrekkelijk vaag idee van het gezochte onderwerp toch constructief te zoeken. Het feit dat gebruikers meestal op één of twee termen zoeken doet immers vermoeden dat dit het geval is.

Een positieve uitwerking op het zoekresultaat kan ook worden verwacht van Query expansion met behulp van een thesaurus. Maar aangezien mensen niet snel gebruik zullen maken van geavanceerde functionaliteit [Jansen e.a. 99] heeft deze uitbreiding alleen zin als deze op een niet te ontkomen manier wordt gepresenteerd, of wanneer dit automatisch, dus niet interactief, wordt toegepast. Aangezien dit laatste afbreuk doet aan een essentieel aspect van query expansion en de resultaten potentieel vervuilt, zal deze mogelijkheid in eerste instantie niet verder worden onderzocht.

Clustering op het RUG-CMS

Volgens Zamir en Etzioni [Zamir '98] zijn de resultaten van 'whole document clustering' merkbaar beter dan de clustering van 'snippets'. De precisie neemt toe met 10 à 20%. Het lijkt dus een goed idee om te kijken of dit haalbaar is binnen het RUG-CMS, en dit is ook onderzocht.
Bij globale zoekmachines is deze aanpak niet haalbaar, omdat het teveel tijd kost om de originele documenten op te halen, en omdat een zoekopdracht zeer veel hits oplevert. Ook al slaat Google kopieën van alle pagina's op, dan nog zullen al deze documenten ter clustering naar een centrale processor moeten worden verstuurd. Dit kost (te)veel tijd, omdat de data verdeeld over vele machines is opgeslagen.
In de situatie van het RUG CMS is de hoeveelheid documenten minder groot, waardoor een gemiddelde zoekopdracht een beheersbaar aantal resultaten zal opleveren (in de orde van 100, in plaats van in 100.000). Verder zijn alle documenten opgeslagen op een centrale database. Deze moet een document kunnen retourneren in de orde van 1 ms. Dit

betekent dat het ophalen van de benodigde informatie binnen een seconde zou moeten kunnen lukken, wat acceptabel is bij een zoekmachine.
Deze omstandigheden betekenen ook dat er eventueel een niet-incrementeel clustering algoritme gebruikt zou kunnen worden, en algoritmes die slecht overweg kunnen met snippets.
Wanneer de fulltext zoekresultaten toch niet snel genoeg blijken te kunnen worden geretourneerd, kan er voor gekozen worden om te werken met minder zoekresultaten en de minder relevante niet verder te verwerken. Dit levert mogelijk een minder compleet beeld op, maar heeft aan de andere kant wellicht minder last van vervuiling van de clusters door de minder relevante hits.

Evaluatie van clustering-algoritmen

Hoe vergelijk je de ervaren performance van een clustering algoritme? Dit is erg moeilijk objectief te meten, en meestal wordt er dan ook uitgegaan van een subjectieve indruk, ook bij ontwikkelaars van clustering algoritmen zoals Weiss.
Problemen bij het beoordelen van clustering-resultaten zijn [Osinski '04]

- Het begrip van clusters bij mensen is erg subjectief. Het is erg moeilijk om automatisch te bepalen hoe goed mensen zullen begrijpen wat een bepaalde cluster precies omvat, en verschillende mensen zullen verschillende ideeën hebben over de gewenste optimale clustering.
- Er zijn veel mogelijke oplossingen die allemaal even goed zijn, doordat er verschillende uitgangspunten zijn om een documentcollectie onder te verdelen. Hierbij is vaak niet aan te geven welk uitgangspunt beter is.
- De overeenstemming van de clusteromschrijvingen met de inhoud van de geproduceerde clusters is nauwelijks te formaliseren. Ook hier kan dus slecht een objectieve score worden bepaald.
- Door bovengenoemde redenen is er geen 'Gold Standard' om (al dan niet automatisch) de performance van een clusteralgoritme mee te vergelijken.

Ieder clusteralgoritme is heeft zijn eigen zwakke plekken voor bepaalde soorten invoersets. Er zijn enkele duidelijke probleemgevallen:

- Invoerdata met ongelijke clustergrootte. Sommige algoritmen (o.a. K-Means) gaan er vanuit dat de invoer groepen bevat van een vergelijkbare grootte, maar dit zal in de praktijk slechts zelden het geval zijn. Dit kan er toe leiden dat grote clusters doormidden gesneden worden en kleine clusters worden samengevoegd.
- Uitbijters in de invoerset. Enkele documenten die erg afwijken van de rest van de invoerset zullen door sommige algoritmen een eigen cluster toegewezen krijgen, terwijl de meer gemiddelde documenten samengevoegd worden. Dit terwijl er best een duidelijke onderverdeling in deze laatste groep kan zitten, die veel belangrijker is.
- Niet-'bolvormige' invoerclusters. Wat hiermee bedoeld wordt is dat er documenten zullen zijn die het onderwerp steeds op een iets andere wijze belichten, maar die toch een duidelijk centraal onderwerp zullen hebben. Documenten die iets verder afwijken van het centrale onderwerp kunnen op

zichzelf gezien dichter bij een ander onderwerp liggen, maar toch beter bij de de eerste groep passen. Ieder algoritme wat uitgaat van een bepaalde aanname over de invoer zal dit verband niet zien.

- Kettingen van documenten. Dit is min of meer het omgekeerde van het voorgaande geval: het is denkbaar dat er series van documenten zijn die op zich goed bij elkaar passen, maar over het geheel genomen beter in aparte groepen zouden kunnen worden ingedeeld.

Osinski [Osinski '04] stelt drie beoordelingsmethoden voor (zie ook [Zamir 99]):

- Gebruikmaken van de uit het Information Retrieval (IR) vakgebied bekende metrieken: 'precision' en 'recall'. Hierbij wordt er vanuitgegaan dat de resultaten in één cluster corresponderen met de resultaten van een bepaalde query. De overeenkomst hiertussen is eenvoudig te berekenen. Het probleem met deze aanpak is dat niet precies bekend is wat deze query dan zou moeten zijn; dit is erg subjectief. Er is dus ook geen testset hiervoor.
- Evaluatie door gebruikers. Dit kan gebeuren door gebruikers te ondervragen, of door de gebruikers te observeren terwijl zij zoeken door middel van de search logs. Dit laatste vereist echter een draaiend systeem. Bovendien zal ook hier het resultaat erg subjectief zijn.
- 'Merge-then-cluster' aanpak. Bij deze aanpak worden verschillende documentsets waarvan het onderwerp bekend is samengevoegd, waarna een clusteralgoritme wordt gebruikt om de oorspronkelijke documentsets te achterhalen. Op deze manier wordt een objectieve maatstaf bereikt voor het meten van de kwaliteit van clustering. Een ander belangrijk voordeel is dat de dit proces volledig geautomatiseerd kan worden, zodat de resultaten reproduceerbaar zijn, en invloed van verschillende parameters objectief becijferd kan worden. Deze aanpak wordt ook gebruikt in [Osinski '04-3]. Hierbij werd gebruik gemaakt van documenten uit het Open Directory Project, welke door mensen geclassificeerd zijn in een fijnvertakte structuur.

Van deze methoden lijkt de 'merge-then-cluster' aanpak het best bruikbaar. Deze zal dan ook gebruikt worden in dit onderzoek. Hierbij kan gebruik gemaakt worden van de verdeling van documenten in gespecialiseerde portals: ook de RUG-website heeft een boomstructuur. Hierdoor zijn documentsets samen te stellen met overeenkomstige onderwerpen.

Mogelijke criteria voor beoordeling van clusteralgoritmen zijn

- Kwaliteit van de gegenereerde clusters
- Onderscheidend vermogen tussen verschillende clusters
- 'Outlier detection': de tolerantie voor vervuilende data
- Snelheid
- Kwaliteit van de clusterlabels

De snelheid van het cluster-algoritme kan in de praktijk bepaald worden met een eenvoudige benchmark-applicatie.

De kwaliteit van de geproduceerde clusters en het onderscheidend vermogen is minder eenvoudig te bepalen. Wij zullen dit doen aan de hand van 'topic coverage', 'snippet coverage' en 'cluster contamination'. [Osinski '04-3].

- **'Snippet coverage'** is het percentage documenten wat toegewezen wordt aan minstens één cluster. Dit is dus een maat voor hoeveel documenten niet geclusterd kunnen worden, of in een 'overige' cluster worden samengevoegd.

- **'Topic coverage'** is een maatstaf voor de mate waarin een algoritme erin slaagt om alle origineel aanwezige onderwerpen te laten terugkomen in de geproduceerde clustering. Dit wordt gemeten in het percentage originele onderwerpen wat terugkomt als een cluster (niet te verwarren met het aantal documenten). Een uitgebreidere versie van dit criterium houdt ook rekening met de positie in de lijst met gegenereerde clusters waarop alle originele onderwerpen zijn gedekt, dus hoeveel clusters er minimaal nodig waren geweest om alle onderwerpen te dekken.

- **'Cluster contamination'** is een maat voor de 'puurheid' van een gegenereerde cluster [Weiss '04]. Een cluster is 'puur' wanneer er slechts documenten met één bepaald onderwerp in zijn ingedeeld. De vervuiling ('contamination') is in dat geval 0. Deze grootheid is zo gedefiniëerd dat in het slechtste geval, dus wanneer een cluster bestaat uit een verzameling documenten met een gelijkmatige verdeling van onderwerpen, de vervuiling gesteld wordt op 1.

De bovengenoemde metrieken zijn niet bruikbaar om in een productie-omgeving te gebruiken voor het goed- of afkeuren van gegenereerde clusters. In deze situatie is er immers niet sprake van duidelijk afgebakende originele clusters om deze statistieken uit te berekenen.

De kwaliteit van clusterlabels is niet machinaal te beoordelen. Hiervoor zal dus een subjectief oordeel geveld moeten worden.

4. Implementatie

Carrot^2

Om de bruikbaarheid van clustering in de praktijk te testen zijn enige experimenten gedaan.
Hiervoor is gebruik gemaakt van het Carrot^2 open-source clustering framework. Dit framework is opgezet voor researchdoeleinden. Het maakt gebruik van Java en XML, en is dus een logische keuze om te gebruiken samen met het RUG-CMS, wat ook gebruikmaakt van deze technologie.
Carrot^2 heeft een duale architectuur; de verschillende componenten kunnen met elkaar communiceren door middel van locale procedurecalls, of door middel van remote components die de data in XML-formaat naar elkaar toesturen. Deze laatste mogelijkheid zal in een volgende release verdwijnen omdat hier geen gebruik meer van wordt gemaakt.

Er zijn verschillende soorten componenten: input-, filter-, output-, en controller-componenten. De te clusteren gegevens worden verzameld door een inputcomponent. Deze geeft de verzamelde gegevens door aan een pipeline van één of meerdere filtercomponenten, die zorgdragen voor de eigenlijke verwerking en clustering van de data. Tenslotte wordt de data verzameld door een outputcomponent, die zorgdraagt voor de uitvoer of presentatie van de resultaten. De verwerking van de gegevens wordt gestart en aangestuurd door een controllercomponent..
Er is een vastliggend formaat voor iedere tussenstap in dit proces, zodat de componenten elkaars resultaten kunnen lezen.

Voor de testen is er gebruik gemaakt van de demo-applicatie die met het Carrot^2 framework wordt meegeleverd. Deze applicatie vraagt de gebruiker om een zoekquery, die vervolgens wordt doorgegeven aan een bekende zoekmachine zoals Yahoo of Google. De zoekresultaten die door de zoekmachine worden geretourneerd worden hierna verwerkt volgens een door de gebruiker te kiezen proces (een 'recept' met een serie filtercomponenten). In dit proces is vastgelegd met welk algoritme de zoekresultaten geclusterd worden.
Er kunnen eenvoudig nieuwe componenten en processen worden toegevoegd door middel van XML-configuratiebestanden en het toevoegen van de benodigde classes.

Voor onze experimenten is een nieuwe invoercomponent geschreven welke gegevens kan inlezen van het RUG-CMS. Deze is gebaseerd op een bestaande component voor het inlezen van XML-data. De zoekquery wordt via een HTTP GET-request naar het RUG-CMS gestuurd, die hierop de resultaten in XML-formaat retourneert. Deze resultaten bevatten behalve de titel, het pad en de samenvatting ook de volledige inhoud van de gevonden documenten. De RUG-invoercomponent selecteert hieruit door middel van een XSL-transformatie de benodigde data. Deze XSL-transformatie kan gemakkelijk worden aangepast om andere velden te gebruiken voor experimenten.
Deze nieuwe invoercomponent kan ook gebruikt worden voor het inlezen van een file met van te voren verzamelde informatie, zodat een standaard testset gebruikt kan worden.

De demo-applicatie geeft niet alle details weer over de geproduceerde clusters, en berekent ook geen statistieken over de kwaliteit van de clusters. Om hier toch zicht op te krijgen is een extra component geschreven die in de processing pipeline kan worden ingevoegd voor de uitvoercomponent, en als het ware de informatie onderschept. Op basis hiervan worden de 'snippet coverage', 'simple topic coverage', 'extended topic coverage', minimale, maximale en gemiddelde cluster-vervuiling, en de standaarddeviatie hiervan bepaald, door middel van hiervoor aanwezige standaardfuncties.
Ook wordt een overzicht getoond van de correspondentie tussen gegenereerde clusters en de originele groepen, voor gebruik bij een gemengde testset.

Omdat Carrot[2] geen ingebouwde taaldetectie heeft voor de Nederlandse taal is een filtercomponent geschreven die aan een ingevoerd document een Nederlands taallabel meegeeft (analoog aan de bestaande engelse versie). Dit wordt door sommige algoritmen gebruikt om een bijpassende stopwoordenlijst te vinden.

De bestaande processen zijn aangepast voor gebruik met de RUG-invoercomponent, de alternatieve taalcodering en de statistiekencomponent.
Verder is de aanroep van de Java virtual machine aangepast zodat er meer geheugen beschikbaar is (256 MB). Dit bleek nodig bij het clusteren van grote hoeveelheden data.

5. Resultaten

Merge-then-cluster test

Om te kunnen beoordelen hoe goed de clustering in de praktijk werkt is gebruikgemaakt van de eerder besproken merge-then-cluster aanpak Hiervoor zijn vier groepen van ongeveer honderd willekeurige documenten van vier verschillende portals gehaald. Deze sets zijn samengesteld door te zoeken op 'een' met een limiet van 100 resultaten, waardoor de eerste honderd hits geselecteerd zijn.
Voor het samenstellen van de testset is gekozen voor de portals van de bibliotheek, de portal voor aankomende studenten (studiekiezers), de portal voor het onderzoeksinstituut GBB, en een stuk van de bureauportal, namelijk het deel voor het expertisecentrum ABJZ (Algemeen Bestuurlijke en Juridische Zaken). Deze selectie is zo gemaakt met de bedoeling vier duidelijk te onderscheiden kennisgebieden te hebben, met een beperkte overlap.
De vier groepen zijn vervolgens samengevoegd, en worden tijdens de verdere verwerking op titel alfabetisch gesorteerd om een semi-willekeurige volgorde te verkrijgen. Deze hele verzameling zullen we voortaan aanduiden met 'testset'.
Deze testset is minder duidelijk gedefiniëerd dan degene die gebruikt is door Osinski. Het is dus interessant te zien of deze aanpak ook werkt met losser samenhangende invoerdocumenten.

De testset is verwerkt door vier verschillende algoritmen: STC, HAOG-STC, K-Means en Lingo. Dit met de standaardinstellingen zoals deze gebruikt worden in de demo-applicatie. Zoals uit de literatuur bekend hebben deze parameters maar een beperkte invloed, en dit blijkt ook uit enige experimenten. Hierbij zijn verschillende parameters gevariëerd, maar zonder extreme waarden te kiezen lukte het niet om de gemeten waarden in de orde van tientallen procenten te laten veranderen. Daarom is hier verder geen aandacht aan besteed.

De standaardprocessen zoals gebruikt zijn hebben behalve een in- en uitvoercomponent en de daadwerkelijke clustercomponent hulpcomponenten voor het nummeren van de invoerdocumenten, een onderschepper voor het eventueel bewaren van resultaten op schijf, en de toegevoegde onderschepper voor het genereren van de statistieken. Ook worden in de STC en HAOG-STC processen extra filtercomponenten gebruikt voor het toevoegen van (hardcoded) taalinformatie voor de tokenizer, het tokeniseren van de invoer en het normaliseren van hoofdlettergebruik.
De K-Means en Lingo clusteraars bleken niet compatible met de STC-tokenizer, waardoor deze algoritmen niet gebruik konden maken van de aanwezige stopwoordenlijst. Dit zou een beperkte invloed kunnen hebben op de resultaten. Er is wel getest met de STC-algoritmen zonder (Nederlandse) stopwoordenlijst; bij deze algoritmen blijkt het gebruik van een stopwoordenlijst inderdaad een beperkte gunstige invloed te hebben.

Bij het beoordelen van de resultaten gaan we uit van de door Carrot[2] gegenereerde statistieken voor 'simple topic coverage', 'extended topic coverage', 'snippet coverage' en 'average cluster contamination'.

De resultaten voor de testset zijn als volgt:

Testset snippets	STC	HAOG-STC	K-Means	Lingo
Simple topic coverage	50%	75%	50%	100%
Extended topic coverage	26%	27%	70%	67%
Snippet coverage	39%	48%	68%	72%
Avg. Cluster contamination	0,24	0,37	0,44	0,38

Op het eerste gezicht is hier te zien dat zowel Lingo als K-means goed scoren op deze criteria. Nadere inspectie leert echter dat K-Means voor deze testset slechts twee clusters genereert, waarvan één slechts een zinnige cluster lijkt; de andere is opgebouwd rond het stopwoord 'van'. Dat de gemiddelde clustervervuiling niet hoger uitvalt komt doordat de ene 'zinnige' cluster het gemiddelde omlaag trekt. Een weging op basis van de clustergrootte lijkt in dit geval beter. Het probleem met het stopwoord 'van' zou onderdrukt kunnen worden door een stopwoordenlijst te gebruiken en hierop te filteren. Toch blijft het onwenselijk dat een clusteralgoritme zo gevoelig is voor veel voorkomende termen.
Verder scoort STC goed op de clustervervuiling. Dit wordt echter bereikt door het grootste deel van de documenten en onderwerpen niet te clusteren. Dit is niet wenselijk bij gebruik in een zoekmachine. Dezelfde conclusie gaat in mindere mate op voor het op STC gebaseerde HAOG-STC algoritme.
Op basis van deze testset en deze criteria lijkt Lingo dus het meest geschikte algoritme.

Een heel andere conclusie is dat ook met een testset met vrij los gedefiniëerde onderwerpen er aanzienlijke verschillen zitten in de gemeten prestaties van de verschillende algoritmen, en dat de 'merge-then-cluster' aanpak op basis van documenten van verschillende portals dus een plausibele test lijkt. Toch kan er niet alleen van deze getallen worden uitgegaan, aangezien het niet onmogelijk is dat een voor een gebruiker onbruikbare cluster toch een redelijke score behaalt.

Fulltext versus snippet

Volgens Zamir c.s. [Zamir '98] levert het clusteren van volledige documenten in plaats van samenvattingen een precisiewinst op van 10 à 20%. Het ligt dus voor de hand te onderzoeken of er ook een prestatiewinst te vinden is aan de hand van de hier voorhanden zijnde statistieken.
In plaats van de samenvatting van het document is hier de inhoud van het nederlandstalige deel van de documenten aan de clusteraar aangeboden. Verder is de opzet van het experiment als hierboven.

Dit leverde de volgende resultaten op:

Testset fulltext	STC	HAOG-STC	K-Means	Lingo
Simple topic coverage	75%	75%	-	75%
Extended topic coverage	39%	45%	-	53%
Snippet coverage	73%	71%	-	58%
Avg. Cluster contamination	0,82	0,84	-	0,33

Het K-Means algoritme is na tien minuten onderbroken omdat het op dat moment nog niet klaar was.
Wat hier opvalt is dat de STC-gebaseerde algoritmen hier een veel grotere clustervervuiling opleveren. Kennelijk heeft de extra beschikbare informatie hier een negatieve invloed. Lingo brengt het er hier beter vanaf, het resultaat is hier beter dan bij het clusteren van snippets. Waarschijnlijk kunnen er betere clusterlabels (en dus clusters) gegenereerd worden wanneer er meer informatie beschikbaar is.
Wat betreft extended topic coverage en snippet coverage profiteren de STC-gebaseerde algoritmen juist van de extra beschikbare informatie. De vraag is of dit opweegt tegen de toegenomen vervuiling.

Over het algemeen genomen lijken alle genoemde algoritmen geen baat te hebben bij de extra informatie in de volledige documenten in onze testset. Mogelijk gaan de uiteindelijke prestaties hier zelfs van achteruit.

Gezien de ervaringen in andere onderzoeken is deze conclusie onverwacht. Er is niet precies aan te geven waarom de resultaten van fulltext clusteren in dit onderzoek niet beter zijn. Vermoed wordt dat dit komt door de relatief hoge kwaliteit van de samenvattingen die aanwezig zijn op het RUG-CMS. Deze worden namelijk door mensen ingevoerd bij het plaatsen van een pagina of document. De 'snippets' die bij andere onderzoeken worden gebruikt zijn automatisch gegenereerde samenvattingen, die worden geproduceerd door de achterliggende database op basis van zinnen die in de buurt staan van de gezochte termen. De samenvattingen zullen hierdoor een groter aantal relevante termen bevatten, waardoor de toegevoegde waarde van de rest van het document daalt en het effect van de vervuiling zwaarder telt. Verhoudingsgewijs zal fulltext clusteren dan een minder grote voorsprong hebben of zelfs een negatief effect behalen ten opzichte van de gebruikelijke methode.

Snelheid

Om bruikbaar te zijn in de praktijk moeten sets van zoekresultaten geclusterd kunnen worden binnen een tijdsspanne in de orde van een seconde. Dit is gemeten voor de hiervoor ook gebruikte testset. Deze bevat bijna 400 documenten, wat representatief is voor de gemeten praktijk op het RUG-CMS.
De snelheid van de verschillende clusteralgoritmen is vergeleken door de testset 100 keer te laten clusteren met de benchmarkfunctie van de demo-applicatie. De eerste 25 runs worden niet meegerekend om eventuele invloeden van caching-mechanismen en

dergelijke uit te sluiten. Hiervan is alleen afgeweken bij de fulltext test met HAOG-STC, hier zijn 30 runs gedaan waarvan de eerste tien niet meegerekend zijn. De gemeten tijd is alleen de tijd die benodigd is voor het clusteren zelf, de uitvoer naar het scherm wordt hier niet bijgerekend.
De benchmarks zijn uitgevoerd op een AMD Athlon 1467 CPU met 1 GB geheugen.

Gemiddelde tijd	STC	HAOG-STC	K-Means	Lingo
Snippets (σ)	260 ms (148)	262 ms (144)	13245 ms (361)	1206 ms (210)
Fulltext (σ)	9344 ms (204)	20136 ms (9791)	>360000	7517 ms (565)

Er zit soms een grote spreiding tussen de verschillende runs, dit uit zich in een hoge standaarddeviatie (σ).
Hier is te zien dat het K-Means algoritme te traag is voor gebruik in de praktijk; een wachttijd in de orde van een seconde is acceptabel, maar een orde van 10 seconden niet meer. De fulltext test is hier afgebroken nadat bleek dat er op deze wijze in tien minuten nog geen clustering te produceren viel.
STC en HAOG-STC blijken erg snel te zijn bij het clusteren van snippets. Lingo heeft in dit geval nog geen voordeel van de 'omgekeerde' aanpak van eerst clusterlabels produceren en dan pas clusteren; kennelijk wordt dit pas gunstig op het moment dat er grote hoeveelheden gegevens moeten worden geclusterd, zoals bij het fulltext clusteren.
Op basis van snelheid zijn STC en HAOG-STC dus aan te bevelen voor het clusteren van samenvattingen. Lingo is bruikbaar hiervoor, maar niet de snelste optie.
Wanneer er geclusterd wordt op basis van de hele documenten is Lingo de snelste optie. Ook STC lijkt geen slechte keuze. Om fulltext clustering in de praktijk bruikbaar te maken zal het proces dan nog wel sneller gemaakt moeten worden. Dit kan door het inzetten van snellere hardware, het gebruik van een sneller algoritme (wat inmiddels commerciëel verkrijgbaar is), of het beperken van het aantal documenten wat geclusterd wordt. Dit laatste doet echter tekort aan één van de voordelen van clustering: het overzichtelijk maken van een groot aantal documenten.

Kwaliteit clusterlabels

De kwaliteit van clusterlabels valt niet objectief te meten. Om toch een beeld hiervan te krijgen is een subjectief oordeel geveld over de resultaten voor vier 'real-life' zoekopdrachten. Hiervoor zijn op het RUG-CMS de zoekresultaten opgevraagd voor de termen 'fiets', 'propedeuse', 'nano' en 'geschiedenis'. Hierbij zijn slechts de eerste honderd resultaten opgevraagd. Deze zijn vervolgens geclusterd op basis van zowel snippets als de gehele documenten (de gegenereerde clusters zijn te vinden in de appendix). Dit leidde tot de volgende observaties.

De resultaten zijn zeer wisselend. De kwaliteit van de onderwerpen varieert van zeer slecht tot redelijk bruikbaar. In geen enkel geval is er sprake van dat een totaalbeeld over het onderwerp van de oorspronkelijke query ontstaat, zoals gehoopt. De verzameling termen is fragmentarisch.

De kwaliteit van de clusterlabels verbetert niet door het verwerken van de volledige documenten. De resultaten van fulltext clusteren lijken minder bruikbaar dan de resultaten behaald op basis van snippets.

Het enige algoritme wat consistent redelijke resultaten levert is Lingo. K-Means levert ook bruikbare resultaten, maar is te langzaam voor het uitvoeren van fulltext clustering. STC en HAOG-STC geven zeer wisselende resultaten, met uitschieters naar de negatieve kant. Lingo lijkt dus de minst slechte keuze.

Verder duiken er regelmatig stopwoorden en werkwoorden op in de gegenereerde labels, ook bij algoritmen die voorzien zijn van een stopwoordenlijst. Dit verdient nadere aandacht.

Samengevat lijken de geproduceerde clusters goed bruikbaar om een gebruiker suggesties te geven voor verdere navigatie, maar als hoofdnavigatie zal het waarschijnlijk niet voldoen. Er moet ook een ander mechanisme zijn om de zoekresultaten te ontsluiten voor het geval dat de gebruiker geen interessante cluster vindt, bijvoorbeeld de lijst zoals deze nu ook al gebruikt wordt.

Keuze clusteralgoritmen

Op basis van de in dit hoofdstuk onderzochte eigenschappen lijkt Lingo de beste keuze als clusteralgoritme voor een clusterende zoekmachine. Het is niet het snelste algoritme voor gebruik met 'snippets', maar de snelheid is acceptabel. Bovendien blijft de snelheid ook acceptabel wanneer er meer data dan normaal wordt geclusterd. Dit zou in de praktijk voor kunnen komen wanneer mensen het samenvatting-veld in documenten gevuld hebben met een kopie van de inhoud van het hele document. Verder levert Lingo volgens de statistieken de hoogste kwaliteit van clusters. Deze indruk wordt bevestigd door een subjectieve beoordeling.

6. Conclusies en suggesties voor verder onderzoek

In dit hoofdstuk zullen we een antwoord pogen te geven op - en aan de hand van - de in de inleiding gestelde onderzoeksvragen. Ook zullen we enige suggesties doen voor nader onderzoek.

Wat zijn de zwakke punten van de zoekmachine van het RUG-CMS?
Er zijn een paar duidelijke zwakke punten aan te wijzen. Het systeem heeft geen tolerantie voor spelfouten, en is niet gericht op het vinden van personen, alhoewel dit laatste het belangrijkste gebruik van de zoekmachine blijkt te zijn. Verder is er een grote hoeveelheid informatie aanwezig in de vorm van onder andere Word- en PDF-documenten, terwijl deze niet standaard doorzocht worden. Ook de keuze om standaard alleen de huidige portal te doorzoeken leidt tot problemen.
Verder blijkt dat het grootste deel van de gebruikers alleen de eerste pagina met zoekresultaten gebruikt. Gemiddeld gesproken is dit het topje van de ijsberg. Ook is het zo dat gebruikers relatief weinig hun zoekopdracht aanpassen, en meestal één en soms twee zoektermen gebruiken. Dit doet vermoeden dat er een hoop relevante informatie niet gevonden wordt, en dat de informatie beter gepresenteerd kan worden.

Wat zijn er voor mogelijkheden om de zwakke punten te verbeteren?
Welke van deze oplossingen lijkt het meest veelbelovend?
Er is een aantal oplossingen aangereikt die de werking van de zoekmachine zouden kunnen verbeteren. De belangrijkste hiervan zijn het invoeren van een al dan niet automatische spellingssuggestie, het prominenter presenteren van gevonden personen en mogelijk relevante documenten, en de standaard zoekinstellingen aanpassen zodat ook wordt gezocht buiten de huidige portal. De gebruiker zal op zijn minst op de hoogte gebracht moeten worden van extra zoekresultaten die te vinden zullen zijn bij het uitbreiden van het zoekgebied, hetzij door het zoeken op meer portals, hetzij door het zoeken op een ander of algemener objecttype. Dit kan uitgebreid worden door niet meer strikt te eisen dat *alle* zoektermen voorkomen in een zoekresultaat. Dit laatste zal onder andere de problemen met het zoeken op voor- en achternaam kunnen verhelpen.
Ook lijkt het interessant om zoekresultaten te groeperen in clusters, zodat er meer resultaten op een scherm weergegeven kunnen worden. Hiermee wordt in eerste instantie gedoeld op het ordenen van zoekresultaten in automatisch gegenereerde clusters, maar er valt ook te denken aan het standaard aanbieden van bij de zoekopdracht passende personen en documenten, bijvoorbeeld in aparte kaders. Deze suggesties sluiten elkaar niet uit, en kunnen samen ingezet worden.

Hoe functioneert dit in de praktijk?
Er is onderzocht of het gebruik van clustering in de praktijk zou kunnen werken. De resultaten rechtvaardigen niet om de zoekresultaten exclusief via een geclusterd overzicht aan te bieden, maar bieden wel uitzicht op een extra ingang om door resultaten te navigeren. Lingo lijkt het meest bruikbare van de onderzochte algoritmen. Wellicht zullen nieuwere versies van dit algoritme betere resultaten opleveren.

Het clusteren op basis van hele documenten in plaats van samenvattingen blijkt slechtere resultaten op te leveren. Dit is niet in overeenstemming met eerder onderzoek. Mogelijk is de kwaliteit van de samenvattingen die op het RUG-CMS gebruikt worden hoger dan de automatisch gegenereerde 'snippets' welke gebruikt worden in andere onderzoeken, waardoor fulltext zoeken een verhoudingsgewijs slechter resultaat oplevert.
In het verlengde hiervan zou er zelfs geëxperimenteerd kunnen worden met het clusteren op documenttitel. Een zeer bescheiden experiment (zie appendix) laat zien dat dit nette en bruikbare clusterlabels oplevert, vergelijkbaar met het zoeken in snippets.

Verder onderzoek

Weging van resultaten op basis van context
Zoals eerder genoemd in dit rapport zou het mogelijk moeten zijn om zoekresultaten te wegen op basis van de plaats waar zij in de directoryboom staan, de plaats waar de zoekactie plaatsvindt, en andere contextinformatie als IP en loginnaam. Dit lijkt een interessant onderwerp voor verder onderzoek.

Grafische representatie van clusters
Wanneer er bevredigende clusterresultaten bereikt kunnen worden, kan er overwogen worden om de gegenereerde clusters op een grafische manier weer te geven. Op die manier kan de gebruiker een visueel overzicht krijgen van de grootte van de verschillende clusters, en navigeren door een eventuele hiërarchische clusterstructuur.

Samenstellingen automatisch uitsplitsen
Een significant deel van de falende zoekacties wordt veroorzaakt doordat wordt gezocht naar samengestelde woorden ("buluitreiking"). Het zou mogelijk moeten zijn om deze woorden voor het verwerken van de zoekactie automatisch te splitsen in de samenstellende woorden, aan de hand van een gegenereerde of bestaande woordenlijst.

Afromen van resultaten
Mogelijk kunnen de resultaten van de clustering verbeterd worden door slechts de hoogst scorende documenten te gebruiken voor verdere clustering. Op deze manier zullen clusters minder vervuild worden door minder relevante pagina's. Het nadeel is in dit geval dat onvoorzien slecht scorende documenten dus helemaal niet meer gevonden worden. Verder onderzoek zou kunnen uitwijzen of bij de beperkte hoeveelheid betrekkelijk specialistische documenten op het RUG-CMS het voordeel zwaarder weegt dan het nadeel.

7. Referenties

[Bouma '04] Syntactic Contexts for Finding Similar Words (Bouma en van der Plas, 2004)

[Carrot-search] http://www.carrot-search.com/lingo-3g-vs-classic.html

[Curran '02] Curran, James R. and Marc Moens. 2002. Improvements in Automatic Thesaurus Extraction.

[IJzereef '04] Afstudeerscriptie Leonie IJzereef: Automatische extractie van hyponiemrelaties uit grote tekstcorpora

[Jansen e.a. '99] Jansen, B. J., Spink, A., and Saracevic, T. 2000. Real life, real users, and real needs: A study and analysis of user queries on the web. *Information Processing and Management. 36*(2), 207-227.

[Jansen e.a. '01] Searching The Web: The Public and Their Queries (2001) Amanda Spink, Deitmar Wolfram, Bernard Jansen, Tefko Saracevic

[Oracle '01] Oracle Text Application Developer's Guide

[Osinski '04] Dimensionality reduction techniques for search results clustering. Stanisław Osiński.

[Osinski '04-2] Stanislaw Osinski, Jerzy Stefanowski, Dawid Weiss Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition

[Osinski '04-3] Stanislaw Osinski, Dawid Weiss: Conceptual Clustering Using Lingo Algorithm: Evaluation on Open Directory Project Data

[Osinski '05] Stanislaw Osinski, Dawid Weiss, "A Concept-Driven Algorithm for Clustering Search Results," IEEE Intelligent Systems, vol. 20, no. 3, pp. 48-54, May/Jun, 2005.

[Page '99] Page, Lawrence; Brin, Sergey; Motwani, Rajeev; Winograd, Terry. The PageRank Citation Ranking: Bringing Order to the Web

[Tzoukerman '03] Tzoukerman, Klavans & Strzalkowski - The Oxford Handbook of Computational Linguistics, hoofdstuk over Information Retrievel

[Van der Plas '05] Lonneke vd Plas - Automatic Knowledge Acquisition for Question Answering

[Weiss '04] Dawid Weiss - Cluster Contamination Measure

[WEBSOM] http://websom.hut.fi/websom/

[White '98]
http://eclectic.ss.uci.edu/~drwhite/LocalPopulations/WarrenCoTennessee.html

[Zamir '98] Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In Proc. ACM SIGIR'98, 1998

[Zamir, 99] Oren E. Zamir. Clustering Web Documents: A Phrase-Based Method for Grouping Search Engine Results. Doctoral Dissertation, University of Washington, 1999.

A. Appendix

Voorbeelden van doodlopende zoekacties

Hieronder de eerste honderd doodlopende zoekacties met de waarschijnlijkste redenen voor het mislukken.

emptySearchesTab

query	portal	objecttype	typo	spam	persoon	vooraam	objecttypebad	portalbad	tespecifiek	meervoud	samenstelling	geeninfo
	/umcg	w3object	Yes	No	Yes	Yes	No	No	No	No	No	No
{Ormel} and {Johan}	/gmw	w3object	No	No	Yes	Yes	No	No	No	No	No	No
{Johan} and {Ormel}	/gmw	w3object	No	No	Yes	Yes	No	No	No	No	No	No
{mirates}	/farmacie	w3object	No	No	No	No	No	No	No	No	No	Yes
{*81401}	/staff	w3object	No	No	Yes	No	No	No	No	No	No	Yes
{erwin} and {wilbers}	/fwn	w3object	No	No	Yes	Yes	No	No	No	No	No	No
{stubbe}	/	w3object	No	No	Yes	No	Yes	No	No	No	No	No
{out} and {of} and {office} and {reply}	/fwn	w3object	No	No	No	No	No	Yes	No	No	No	No
{glebeek}	/	w3object	Yes	No	Yes	No	No	No	No	No	No	No
{abul} and {azia}	/economie	w3object	No	No	Yes	Yes	No	No	No	No	No	No
{tanveer} and {shehhzad} and {choudhry}	/economie	w3object	No	No	Yes	No	No	No	Yes	No	No	No
{J.} and {Neef}	/gbb	w3object	No	No	Yes	No	No	Yes	No	No	No	No
{adreswijzinging}	/	w3object	Yes	No	No	No	No	No	No	No	No	No
{groeneveld}	/ufb	w3object	No	No	Yes	No	No	Yes	No	No	No	No
{pleinen}	/frw	w3object	No	No	No	No	Yes	Yes	No	No	No	Yes
{bull}	/studenten	w3object	Yes	No	No	No	No	No	No	No	No	No
{bullultreiking}	/studenten	w3object	Yes	No	No	No	No	No	No	No	No	No
{bul+data}	/studenten	w3object	Yes	No	No	No	No	Yes	No	No	No	No
{bulluitrijking}	/studenten	w3object	Yes	No	No	No	No	No	No	No	No	No
{telencentrum}	/talencentrum	w3object	Yes	No	No	No	No	No	No	No	No	No
{talencentrum}	/talencentrum	w3object	No	No	No	No	No	Yes	No	No	No	No
{big} and {guns}	/bibliotheek	w3object	No	No	No	No	No	Yes	No	Yes	No	No
{hyva}	/bibliotheek	w3object	No	No	No	No	Yes	No	No	No	No	No
{oerd}	/bureau	w3object	No	No	No	No	Yes	No	No	No	No	No
{H:\} and {drive}	/biologie	w3object	No	No	No	No	No	Yes	Yes	No	No	No
{iwinet} and {wiskunde} and {b}	/	w3object	No	No	No	No	No	No	No	No	No	No
{biofisica}	/studenten	w3object	Yes	No	No	No	No	No	No	No	No	No
{marleen} and {schippers}	/bcn	w3object	No	No	Yes	Yes	No	Yes	No	No	No	No
{Prof.} and {Oscar} and {Kuipers}	/umcg	w3object	No	No	Yes	No	No	Yes	Yes	No	No	No
{Prof.} and {Oscar} and {Kuipers} and {(Groningen)}	/umcg	w3object	No	No	Yes	No	No	Yes	Yes	No	No	No
{icho34}	/scheikunde	w3object	No	No	No	No	Yes	Yes	No	No	No	No
{abmulder-bakker@let.rug.nl}	/	w3object	Yes	No	Yes	No	No	No	No	No	No	No
{stagebeoordeling}	/let	w3object	No	No	No	No	Yes	Yes	No	No	Yes	No
{stagebeoordelingsformulier}	/let	w3object	No	No	No	No	Yes	Yes	No	No	Yes	No
{bachelor} and {bul}	/psy/onderwijs/bacheloropleidingen	w3object	No	No	No	No	No	Yes	No	No	No	No
{bachelorbul}	/psy	w3object	No	No	No	No	No	No	No	No	Yes	No
{introductiedag}	/	w3organization	No	No	No	No	Yes	No	No	No	No	No
{faculteitskamp}	/bdk	w3object	No	No	No	No	No	No	No	Yes	Yes	No
{tiution} and {fee}	/	w3object	Yes	No	No	No	No	No	No	No	No	No

emptySearchesTab

query	portal	objecttype	typo	spam	persoon	voormaam	objecttypebad	portalbad	tespecifiek	meervoud	samenstelling	geeninfo
{creshe}	/umcg	w3object	Yes	No	No	No	No	No	No	No	No	No
{kresche}	/umcg	w3object	Yes	No	No	No	No	No	No	No	No	No
{kreshe}	/umcg	w3object	Yes	No	No	No	No	No	No	No	No	No
{cresche}	/umcg	w3object	Yes	No	No	No	No	No	No	No	No	No
{studentenverenging}	/	w3object	Yes	No	No	No	No	No	No	No	No	No
{ubblad}	/medewerkers	w3object	No	No	No	No	No	Yes	No	No	Yes	No
{ub} and {blad}	/medewerkers	w3object	No	No	No	No	No	Yes	No	No	No	No
{horsel}	/	document	Yes	No	No	No	No	No	No	No	No	No
{hunneman}	/medewerkers	w3object	No	No	Yes	No	No	No	No	No	No	No
{Temporal} and {stability} and {of} and {rhythmic} and {tapping} and {"on"} and {and} and {"off} and {the} and {beat":} and {a} and {developmental} and {study.}	/bibliotheek	w3object	No	No	No	No	No	No	Yes	No	No	No
{a} and {personal} and {member} and {mister} and {bicord} and {bolla} and {bong}	/	w3object	No	Yes	No	No	No	No	Yes	No	No	No
{graduation} and {dates}	/studenten	w3object	No	No	No	No	No	Yes	No	Yes	No	No
{"}	/economie	w3object	No	No	No	No	No	No	No	No	No	Yes
{bullaanvraag}	/rechten	w3object	Yes	No	No	No	No	No	No	No	No	No
{refwrence} and {manager}	/bibliotheek	w3object	Yes	No	No	No	No	No	No	No	No	No
{refrence} and {manager}	/bibliotheek	w3object	Yes	No	No	No	No	No	No	No	No	No
{studentenblogs}	/	w3object	No	No	No	No	No	No	No	No	No	Yes
{contact} and {mvr} and {flobbe}	/let	w3object	No	No	Yes	No	No	Yes	Yes	No	No	No
{mvr} and {flobbe}	/let	w3object	No	No	Yes	No	No	Yes	Yes	No	No	No
{huivesting}	/prospectivestudents	w3object	Yes	No	No	No	No	No	No	No	No	No
{huisvesting}	/prospectivestudents	w3object	No	No	No	No	No	Yes	No	No	No	No
{batchelor} and {geneeskunde}	/	w3object	Yes	No	No	No	No	No	No	No	No	No
{cims}	/bdk	w3object	No	No	No	No	No	No	No	No	No	Yes
{Elsevier} and {SD} and {Backfile} and {Complete}	/umcg	w3object	No	No	No	No	No	No	Yes	No	No	No
{fokke} and {boersma}	/ufb	w3object	No	No	Yes	Yes	No	Yes	No	No	No	No
{persoongsgegevens}	/	w3object	Yes	No	No	No	No	No	No	No	No	No
{E.h.} and {Vd} and {voort}	/	w3object	No	No	Yes	Yes	No	No	No	No	No	No
{trientje} and {eleveld}	/umcg	w3object	No	No	Yes	Yes	No	No	No	No	No	No
{life} and {science} and {en} and {technologhy}	/	w3object	Yes	No	No	No	No	No	No	No	No	No
{chandler}	/bibliotheek	w3object	No	No	No	No	Yes	Yes	No	No	No	No
{RUG-FN2}	/medewerkers	w3object	No	No	No	No	No	Yes	No	No	No	No
{anglo} and {amerikan} and {school}	/economie	w3object	Yes	No	No	No	No	No	No	No	No	No
{anglo} and {american} and {school}	/economie	w3object	No	No	No	No	Yes	Yes	No	No	No	No
{anglo} and {amerikaanse} and {school}	/economie	w3object	No	No	No	No	Yes	Yes	No	No	No	No
{summer} and {dresses} and {summer} and {dresses} and {summer} and {dresses} and {http://www.beep.com/members101/kilondi/} and {summer} and {dresses}	/ai	w3object	No	Yes	No	No	No	No	No	No	No	No
{munneke}	/bdk	w3object	No	No	Yes	No	No	Yes	No	No	No	No
{I} and {am} and {impressed}	/frw	w3object	No	Yes	No	No	No	No	No	No	No	No

emptySearchesTab

query	portal	objecttype	typo	spam	persoon	voornaam	objecttypebad	portalbad	tespecifiek	meervoud	samenstelling	geenInfo
and {with} and {this} and {page...setup} and {really} and {nice.} and {http://cheap-cigarette.be/cheap_cigarette_online/} and {Thanks!}												
{talencentrum}	/talencentrum	w3object	No	No	No	No	No	Yes	No	No	No	No
{bulaanvraag}	/let	w3object	No	No	No	No	Yes	Yes	No	No	Yes	No
{http://www.ringtones-dir.com/get/} and {ringtones} and {site} and {free.} and {Free} and {nokia} and {ringtones} and {here,} and {Download} and {ringtones} and {FREE,} and {Best} and {free} and {samsung} and {ringtones.} and {From} and {website} and {.}	/bcn	w3object	No	Yes	No	No	No	No	No	No	No	No
{} and {http://www.ringtones-dir.com/download/} and {download} and {ringtones.} and {nokia} and {ringtones:} and {Free} and {nokia} and {ringtones} and {here,} and {Download} and {ringtones} and {FREE,} and {Best} and {free} and {samsung} and {ringtones.} and {[url]http://www.ringtones-dir.com/free/[/url]} and {[link=http://www.ringtones-dir.com]ring} and {tones[/link]} and {From} and {site} and {.}	/bcn	w3object	No	Yes	No	No	No	No	No	No	No	No
{http://www.ringtones-dir.com/get/} and {ringtones} and {site} and {free.} and {[URL=http://www.ringtones-dir.com]ringtones} and {download[/URL]:} and {Free} and {nokia} and {ringtones} and {here,} and {Download} and {ringtones} and {FREE,} and {Best} and {free} and {samsung} and {ringtones.} and {[url=http://www.ringtones-dir.com]samsung} and {ringtones[/url]} and {From} and {website} and {.}	/bcn	w3object	No	Yes	No	No	No	No	No	No	No	No
{beemer}	/economics	w3object	No	No	No	No	Yes	Yes	No	No	No	No
{tijhoff}	/	w3personalinfo	Yes	No	Yes	No	Yes	No	No	No	No	No
{oratie} and {fiscaliteit}	/bdk	w3object	No	No	No	No	No	No	No	No	No	Yes
{haeike} and {harms}	/	w3personalinfo	Yes	No	Yes	Yes	No	No	No	No	No	No
{technische} and {berdijiswetenschappen}	/	w3object	Yes	No	No	No	No	No	No	No	No	No
{plaza} and {kluwe}	/rechten	w3object	Yes	No	No	No	No	No	No	No	No	No
{codes} and {ibg}	/studiekiezers	w3object	No	No	No	No	Yes	Yes	No	No	No	No
{opcic}	/scheikunde	w3object	No	No	No	No	Yes	Yes	No	No	No	No
{avv} and {vakken}	/psy	w3object	No	No	No	No	No	Yes	No	No	No	No
{avv-vakken}	/psy	w3object	No	No	No	No	No	Yes	No	No	Yes	No
{avv}	/psy	w3object	No	No	No	No	No	Yes	No	No	No	No
{I} and {am} and {impressed} and {with} and {this} and {page...setup} and {really} and {nice.} and {http://cheap-cigarette.be/cheap_cigarette_online/} and {Thanks!}	/sterrenkunde	w3object	No	Yes	No	No	No	No	No	No	No	No

emptySearchesTab

query	portal	objecttype	typo	spam	persoon	voormaam	objecttypebad	portalbad	tespecifiek	meervoud	samenstelling	geeninfo
{talencentrum}	/talencentrum	w3object	No	No	No	No	No	Yes	No	No	No	No
{chiropractor}	/bibliotheek	w3object	No	No	No	No	No	Yes	No	No	No	No
{talencentrum}	/talencentrum	w3object	No	No	No	No	No	Yes	No	No	No	No
{teskt} and {oratie}	/umcg	w3object	Yes	No	No	No	No	No	No	No	No	No
{crodil}	/	w3object	Yes	No	No	No	No	No	No	No	No	No
{krodil}	/	w3object	Yes	No	No	No	No	No	No	No	No	No
{crodil}	/	w3object	Yes	No	No	No	No	No	No	No	No	No

Voorbeelden van real-life clusteringen

Resultaten voor de vier zoektermen 'fiets', 'propedeuse', 'nano' en 'geschiedenis', voor snippets en volledige teksten, en voor de algoritmen STC, HAOG-STC, K-Means en Lingo.

"Fiets"

Snippet

STC

Meewerken dan zou de Totale Nederlandse Koolstofdioxide-uitstoot 54 Procent (14)
Huisregels (8)
Vindt u de Huisregels (4)

HAOG-STC

ProcentNederlandseNederlandse Huishoudens Meewerken dan zou de Totale NederlandseMeewerken dan zou de Totale NederlandseHuishoudens Meewerken dan zou de Totale Nederlandse (14)
Huisregels (8)
+ Vindt u de Huisregels (4)

K-Means

Van (58)
Warffum (12)
Van (33)
Stallen Van (7)
Routebeschrijving Naar (16)
Other (12)

Lingo

December Warffum (12)
November Harkstede (12)
Huisregels Van Het (11)
En (14)
Hoe Spaarlampen Het Energieverbruik Doen Toenemen (10)
Veilig Naar School September (6)
Augustus Zuidlaren (6)
Routebeschrijving Naar Studium Generale Groningen (4)

Fulltext

STC

Zie; Dient; Ongeveer (57)
Parkeergarage Ossenmarkt; Dichtsbijzijnde Parkeergarage; 15 Richting (12)
Week; Illustreren de Aanwezigheid van de RUG Discovery in Harkstede; Groepen Stond Natuurkunde van de Fiets Centraal (39)
Basisscholen in Warffum zijn Enthousiast Gemaakt voor de Natuurkunde; Leidde tot Jaloerse Blikken bij het Hogeland College; Kortom in 2006

Komen we Terug (27)
Natuurkunde van de Fiets; Groepen (35)
Fiets (83)
Weet; Totale (28)

HAOG-STC

Fiets (87)
+ Natuurkunde van de FietsNatuurkundeGroepen (87)
++ KomenGemaaktCollegeTerug2006 Komen we Terug (27)
++ CentraleKijkRUGWeekAanwezig (69)
+++ OpenGaansLaat1 (57) ++++ Lijn 6Lijn 6Lijn 6Lijn 6Lijn 6 (12)
+ WeetTotale (69)
++ CentraleKijkRUGWeekAanwezig (69)
+++ OpenGaansLaat1 (57)
++++ Lijn 6Lijn 6Lijn 6Lijn 6Lijn 6 (12)

Lingo

En Te (21)
December Warffum (12)
November Harkstede (12)
Veilig Naar School (6)
(Other) (49)

"Propedeuse"

Snippet

STC

Propedeuse (9)
Jaar (5)

HAOG

Jaar (5)
Propedeuse (9)

K-Means

Van (41)
Inleiding (2)
Propedeuse 2002 2003 (2)
Propedeuse Vestiging Friesland (3)
Propedeuse (2)
Van (2)
Other (45)

Lingo

Bachelor Informatica 1e Jaar (9)
Paragraaf Propedeuse (10)
Bacheloropleiding Sterrenkunde (4)
Studeren in Friesland (3)
Voor Natuurkunde Scheikunde En Informatica (6)
Hoe is de Studie Opgebouwd (4)

Algemene Studie-informatie (2)
Studieopbouw (2)
Bacheloropleidingen Cohort (2)
Deeltijdstudies (2)
Propedeutisch Examen (2)
TBW (2)
Overgangsregeling Bedrijfskunde (2)
Deblokkaderegeling (2)
Het Opleiding En Examenregeling Biologie (3)
Tutoraat (2)
Aan de Rijksuniversiteit Groningen (3)
(Other) (46)

Fulltext

STC

Propedeuse; Studiejaar; Studenten (85)
B; A (23)
VWO-diploma; Colloquium Doctum (14)
Informatie (28)
Contacten (26)
Gaan (23)
Natuurkunde (22)
60 EC (12)
2 Jaar (12)
Betreffende (21)
Studerende (20)
Studielast (20) Programma s (11)
Technische (19)
Gevolgd (19)

HAOG
(out of memory)

LINGO

Het Uitvoeren Van Een (40)
Eeuw Ec (17)
Oriëntatie Informatica Info (9)
Jaar Je (12)
Natuurkunde Sterrenkunde En Technische Natuurkunde (8)
Jaar Filosofie (9)
Economie En (5)
Uw Propedeuse (3)
Van de Juridische (8)
Studeren in Friesland (3)
Propedeuse Bachelor Bedrijfskunde (5)
P.j. Lont Opleidingscoördinator TBW TBK (3)
(Other) (19)

"Nano"

Snippet

STC

Groningse; Gepresenteerd die zijn Verdedigd aan de Faculteit Der Wijsbegeerte (5)

HAOG

Groningse; Gepresenteerd die zijn Verdedigd aan de Faculteit Der Wijsbegeerte (5)

K-Means

Proces (4)
Wereld Veranderen (1)
Jaar Van Uitgave (4)
Electronic Devices (6)
Other (85)

Lingo

Physics of Electronic Devices (5)
Nanoscience Research Master (3)
Faculteit Der (4)
(Other) (88)

Fulltext

STC

2; 1; Second (11)

HAOG

12AndAndAnd (11)

Lingo

Van Het (10)
(Other) (90)

"Geschiedenis"

Snippet

STC

Literatuurbronnen op het Vakgebied Algemene Geschiedenis (3)
Geschiedenis (11)

Geschiedenis (12)
+Literatuurbronnen op het Vakgebied Algemene GeschiedenisVakgebied Algemene GeschiedenisAlgemene Geschiedenis (3)

K-Means

Geschiedenis Van (25)
Onze Vaderlandse Geschiedenis (2)

Van (7)
Finoegrische (7)
Hoogleraren Faculteit Der (4)
Geschiedenis (4)
Educatieve Master (5)
Van Het Bibliografisch (4)
Onderwijs (6)
Het (13)
Oude Geschiedenis (5)
Van (9)
Plaatsingsschema Bibliografisch Centrum (4)
Other (25)

Lingo

Geschiedenis Van (20)
Alfabetische Lijst (5)
Van de Collecties (7)
Volledige Lijst Van Publicaties Van (6)
Oude Geschiedenis (5)
Indeling Geschiedenis Bibliografisch Centrum (4)
Plaatsingsschema Zaal (4)
Master Geschiedenis in Het Kort (4)
Onderwijs En Examenregeling Bachelor Godgeleerdheid (4)
Ernst Kossmann Instituut (4)
Curriculum Vitae (2)
Tweede Jaar (3)
Opinie (3)
Faculteit Der (3)
Geschiedenis Aan (4)
Onderzoek (2)
Stef Van Den Hof (2)
(Other) (36)

Fulltext

STC

1; Staat; Modern (90)
Geschiedenis; Nederlandse; Politieke (90)
Europese; Landen; Europa (50)
Nederlandse Politieke Partijen (18)
20e Eeuw; Kunst; Basis (30)
Economische en Sociale Geschiedenis (14)
Ten; Ter; Plaats (35)
September 2002; Vrije Ruimte; 2002 of Eerder met hun Studie zijn Begonnen zal het Programma Afwijken (13)
taal- en Letterkunde; Romaanse (28)
Jaren; Onderzoek (47)
Economische; Sociale Geschiedenis (31)
Historisch (49)
Volg; Inhoud; Scriptie (27)
Der (47)
Cultuur (44)

HAOG
out of memory

Lingo

Je Aan (16)
Algemeen Thematisch (10)
Info Geschiedenis (7)
Fins ii Hongaars ii Taalvaardigheid (5)
Kossmann Instituut (6)
Hoogleraar Oude Geschiedenis En (4)
Geschiedenis Van de Farmacie (2)
Nederland Den Haag (5)
Medieval History (4)
Theorie En Geschiedenis (3)
Berlijn Wereldstad (2)
Geschiedenis Annotatie (2)
Van de Wiskunde Links (2)
(Other) (36)

Voorbeeld van clustering op titels

De volgende clustering is gegenereerd op basis van de teksten in de titels van de documenten, met het algoritme Lingo en de resultaten voor de zoekterm 'fiets'.

Huisregels (10)
Hoe Spaarlampen Het Energieverbruik Doen Toenemen (10)
November Harkstede (8)
Routebeschrijving Naar (10)
December Warffum (7)
Sociologie in Groningen (4)
Parkeren Fietsstalling Drs-gebouw (4)
Introductie Kunstmatige Intelligentie (3)
Nieuwsbrief (3)
Samenvatting (3)
Fietsprivéregeling RUG (3)
Bijlage (2)
Enquête SPRDe Flash-manier (2)
Uw Bijdrage Aan Kyoto (2)
Toptien Energiebesparingtips (2)
Kindertaal (2)
Top Energie-besparingstips (2)
Veelgestelde Vragen Lustrumtourlied (2)
Hoe Ons Te Bereiken (2)
Huis En Gedragsregels Algemeen (2)
Einstein Fietstocht Op Juni Aanstaande (2)
Doen Column Van Twee Co-assistenten Deel (2)
Er Gaat Niets Boven Groningen Behalve Borkum (2)
(Other) (60)