Retrievaltechniek; technologieën voor het terugvinden van tekstuele informatie

Eric Sieverts

Inhoud:

[Preprint voor bijdrage aan Handboek Informatiewetenschappen, VakMediaNet, Alphen aan de Rijn, 12-12-2012]

1. Inleiding information retrieval

Geautomatiseerde zoeksystemen bestaan al meer dan 50 jaar. Toch denken veel mensen bij "zoeken" nu vooral aan webzoekmachines. En ook denken ze vooral aan digitale informatie, dat wil zeggen aan informatie waarvan de volledige inhoud digitaal is. Dat laatste is echter nog niet heel lang zo - en zeker geen 50 jaar - en ook nu nog altijd niet voor alle soorten informatie. Denk maar aan multimediale informatie (beeld, geluid), waarvoor metadata de enige tekstuele informatie is waarop je kunt zoeken. Die beperking tot metadata is niet veel anders dan in de beginjaren van retrievalsystemen, toen naar tekstdocumenten ook alleen gezocht kon worden via bijbehorende metadata. Als je over een onderwerp iets wilde weten, was je er dus vrijwel op aangewezen te zoeken in toegekende inhoudelijke ontsluiting.

De techniek die werd toegepast om snel te kunnen zoeken in grote verzamelingen informatie, ook al waren het alleen metadata, ligt in aangepaste vorm nog altijd ten grondslag aan veel van de huidige zoeksystemen. Die *inverted file* methode zullen we hier dus als eerste bespreken (par. 2.1). Nu een steeds groter deel van de inhoud van documenten, artikelen, rapporten en boeken op zijn minst gedeeltelijk digitaal beschikbaar is, kan die tekst met zoekmachine- of retrievalsoftware ook in zijn geheel doorzoekbaar gemaakt worden. Daarbij worden alle woorden uit de tekst in een zoekindex opgenomen. Dat is dus een zeer vergaande vorm van zogenaamd ontleend indexeren. Om ook daarbij nog goede zoekresultaten te verkrijgen, wordt de oude inverted file techniek vaak aangevuld met andere technieken, die zich niet meer beperken tot puur Booleaanse zoekmethoden. De basisprincipes van twee daarvan, de vector-methode en de probabilistische techniek, zullen we - in wat vereenvoudigde vorm - ook behandelen (par. 2.2, 2.3).

Bij veel mensen bestaat de verwachting dat met deze methoden van *free-text retrieval* in principe alles is terug te vinden. Men zou dit het Google-effect kunnen noemen, omdat door het gebruik van webzoekmachines iedereen eraan gewend is geraakt op die manier voldoende relevante informatie te kunnen vinden. De vraag is dus gerechtvaardigd of de zo geboden zoekfunctionaliteit het mogelijk maakt in elke situatie de gewenste informatie met voldoende vangst en precisie terug te vinden. Er bestaat namelijk een levensgroot verschil tussen het web, waar je in vele honderden miljarden webpagina's en documenten altijd wel iets relevants kunt vinden, en een intranet of database met niet meer dan enkele tienduizenden tot maximaal enkele miljoenen documenten. Bovendien kan men zich in die situatie vaak niet veroorloven dat een bepaald relevant document met een zoekactie wordt gemist. In de praktijk bleken eenvoudige methoden van *free-text retrieval* op deze punten nog wel wat problemen op te leveren (par. 3). Sommige van die problemen kunnen intussen voor een deel worden opgelost, door allerlei taaltechnologische technieken toe te passen. Op die technieken en wat ze betekenen voor vangst en precisie van zoekacties, gaan we hier ook in (par. 4).

In het voorgaande ging het steeds over tekst, woorden en zoekwoorden. In dat verband wil ik benadrukken dat deze bijdrage zich inderdaad beperktn tot tekst-retrieval; het met woorden zoeken in en naar "talige" informatie. Ook waar beeld, geluid of objecten zijn beschreven met metadata, is nog sprake van tekst. Moderne multimediale zoekmethode die gebruik maken van het beeld of het geluid zelf, komen hier niet aan de orde.

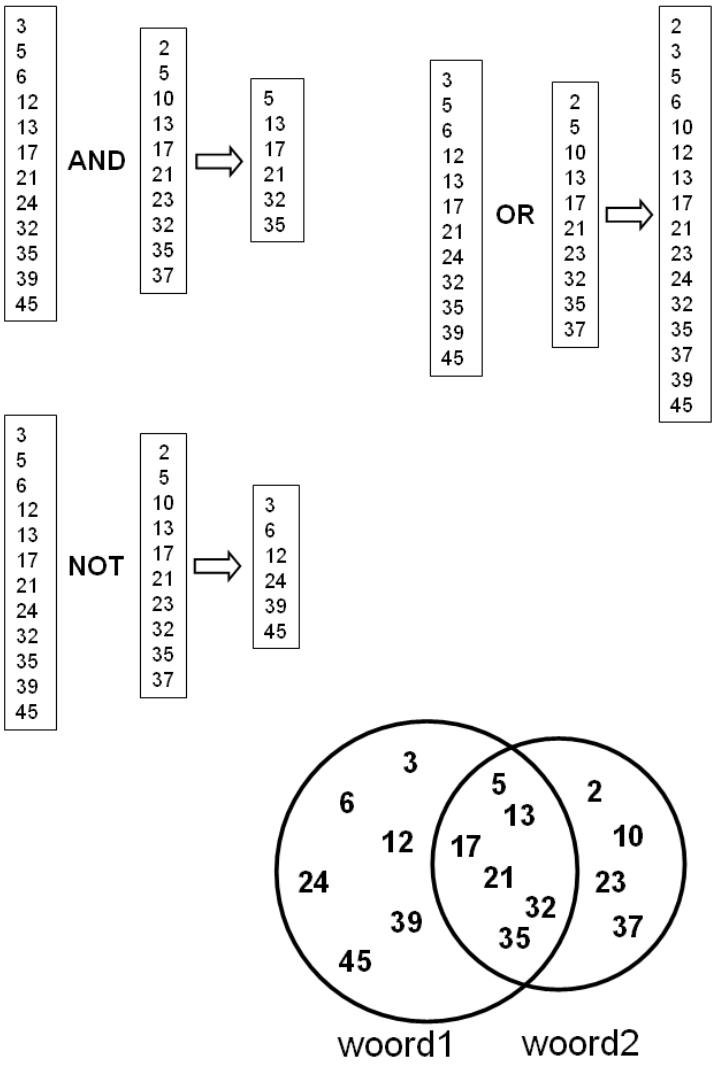
2. Retrievaltechnieken

2.1 De basis: inverted file methode

Om in grote hoeveelheden informatie te zoeken, is het niet handig een computer sequentieel alle tekst te laten doorlezen, om te vinden waar een ingetikt zoekwoord allemaal voorkomt; hoe snel computers tegenwoordig ook zijn en hoeveel intern geheugen ze ook ter beschikking hebben. Dat geldt voor zowel redelijk gestructureerde (bibliografische) databases, als voor collecties full-text documenten of webpagina's. Daarom wordt al vanaf de beginjaren van de grote retrievalsystemen de techniek van de *inverted file* toegepast. Dat houdt in dat automatisch een alfabetische lijst gegenereerd wordt van alle woorden die in de te doorzoeken informatie voorkomen - vandaar dat ook wel van "index" wordt gesproken. Bij elk woord wordt bovendien geregistreerd waar het voorkomt: in welk databaserecord, in welk tekstdocument of in welke webpagina, afhankelijk van de soort informatie die doorzoekbaar gemaakt wordt. De benaming *inverted file* (of geïnverteerd bestand) is gekozen, omdat de informatie hierin "andersom" georganiseerd is dan in de gewone bestanden. Daarin vind je in elk databaserecord of document welke woorden daarin voorkomen. In het geïnverteerde bestand is dat omgekeerd: bij elk woord vind je gegevens in welke records of documenten het voorkomt. Als een gebruiker een zoekterm intikt hoeft het zoeksysteem dat woord dus alleen in die alfabetische lijst op te zoeken en vindt daar meteen alle verwijzingen naar de documenten waarin het woord voorkomt. Zo kan ook meteen het zoekresultaat getoond worden, zowel het aantal gevonden resultaten als de eerste titels. Zelfs in bestanden met tientallen miljoenen documenten werkt dat nog altijd.

Bij webzoekmachines zijn inverted files intussen zo groot geworden dat die gedistribueerd over duizenden servers worden opgeslagen en bevraagd. Daarbij is een antwoord op een zoekvraag aanvankelijk niet gebaseerd op de hele inverted file, maar alleen op het deel met de belangrijkste informatie. Het daarbij vermelde totale aantal resultaten is dan ook een ruwe schatting, die aanzienlijk kan afwijken van hetgeen uiteindelijk echt gevonden kan worden.

Hoe kun je nu zoeken op combinaties van zoektermen, of meer algemeen met Booleaanse operatoren? Door een geïnverteerde bestand wordt dat ook tamelijk eenvoudig. Elke zoekterm levert een rijtje verwijzingen op. In het werkgeheugen van de zoekcomputer kunnen vervolgens bewerkingen uitgevoerd worden met die rijtjes verwijzingen. Bij een AND-operator tussen twee woorden, filtert de computer die verwijzingen eruit die in alle twee de bijbehorende rijtjes voorkomen. Die documenten bevatten immers beide woorden. Bij een OR-operator worden alle verwijzingen daaruit tot één lijst samengevoegd (uiteraard ontdubbeld); bij een NOT-operator worden de verwijzingen naar documenten met het uit te sluiten woord verwijderd. In afbeelding 1 is dit geïllustreerd. Dergelijke bewerkingen kunnen ook nog snel worden uitgevoerd als die rijtjes miljoenen verwijzingen bevatten.



[Afbeelding 1: Uitvoeren van AND, OR en NOT operatoren op basis van een inverted file, aanvullend gellustreerd met een Venn-diagram]

Een specifiekere manier om zoektermen in een AND-relatie te combineren is het zogenaamde nabijheids-zoeken. Daarbij kun je specificeren dat twee te combineren woorden in te vinden teksten dichtbij elkaar moeten voorkomen, bijvoorbeeld maximaal vijf woorden bij elkaar vandaan. Vooral voor zoeken in full-text documenten kan dat een nuttige relevantieverhogende inperking zijn. Via een kleine aanpassing aan de inverted file kan dit gerealiseerd worden. Verwijzingen bij elk indexwoord moeten nu niet alleen gegevens bevatten in welke records/documenten het woord voorkomt, maar ook wat de positie van dat woord is binnen elk van die records/documenten. Als het zoeksysteem in de inverted file ziet dat in een gevonden document het ene woord op positie 8 staat en het andere op positie 23, kan berekend worden dan ze 23-8=15 woorden uit elkaar staan en dus niet voldoen aan de eis "binnen 5 woorden afstand". Als die positieïnformatie laat zien dat ze repectievelijk het 12de en het 16de woord zijn, wordt wel aan de nabijheidseis voldaan. Uiteraard wordt het geïnverteerde bestand hierdoor wat ingewikkelder. Het bevat nu ook de positie-informatie. Wanneer een woord in hetzelfde document vijf keer voorkomt, moeten bovendien alle vijf die posities geregistreerd worden. Ook webzoekmachines bieden soms - en dan vaak ongedocumenteerd - mogelijkheden tot nabijheids-zoeken. Anderzijds speelt de onderlinge afstand van zoekwoorden in de gevonden webpagina's daar al een rol bij de relevantieordening van de zoekresultaten

(par. 5). Bij die relevantieberekeningen zal positie-informatie intern dus ook op een bepaalde slimme manier gebruikt worden.

Veel zoeksystemen bieden de mogelijkheid om zoektermen te trunkeren. Ook dat trunkatie-zoeken vindt meestal plaats via de inverted file. Een rechtse trunkatie:

zoek comput*

gaat betrekkelijk gemakkelijk, omdat alle woorden die met *comput* beginnen in een alfabetische index automatisch al bij (achter) elkaar staan (zie afbeelding 2). Alle daarbij aanwezige verwijzingen kunnen dus makkelijk worden verzameld.

alfabetische index
….
compunction
compunctious
computable
computation
computational
compute
computed
computer
computeraided
computerised
computerization
computerized
computerized
computers
comrade
....

[Afbeelding 2: Rechtse trunkatie via de alfabetische index]

Bij een linkse trunkatie:

zoek *therapie

ligt dat veel moeilijker, want hiervoor moet de computer de hele index sequentieel van a t/m z doorkijken op zoek naar alle woorden die eindigen op *therapie,* omdat ze met een willekeurige letter kunnen beginnen (zie afbeelding 3). Dat kunnen tientallen miljoenen woorden en vele gigabytes zijn.

alfabetische index
....
arbeidstherapie
....
aromatherapie
....
chemotherapie
....
fysiotherapie
....
gentherapie
....
psychotherapie
....
radiotherapie
....
shocktherapie
....
zonnetherapie
....

[Afbeelding 3: Linkse trunkatie in een alfabetische index]

Dit probleem, samen met het feit dat in het Engels, met zijn weinige samengestelde woorden, niet zo veel behoefte bestaat aan linkse trunkatie, maken dat dit in de meeste zoeksystemen niet mogelijk is. Toch is er wel een betrekkelijk eenvoudige praktische oplossing voor. In een "achterstevoren" of *retrograde* index gaat dat zoeken wel weer makkelijk. Woorden die op *-therapie* eindigen, zijn in een retrograde index namelijk degenen die beginnen met *eipareht* en dus weer alfabetisch bij elkaar staan (zie afbeelding 4). De zoekactie *zoek *therapie* wordt dus de rechtse trunkatie *zoek eipareht**.

retrograde index
....
eiparcs
eiparehtamora
eiparehtennoz
eiparehtkcohs
eiparehtneg
eiparehtohcysp
eiparehtoidar
eiparehtoisyf
eiparehtomehc
eiparehtsdiebra
eiplahtne
eipocselet
....

[Afbeelding 4: Linkse trunkatie wordt rechtse trunkatie in een retrograde index]

2.2 Het vectormodel

Het wordt wel als een nadeel gezien dat puur Booleaanse combinaties op basis van een klassieke inverted file een erg zwart-wit beeld opleveren. Of een document voldoet exact aan de gevraagde combinatie van termen en wordt gevonden, of het voldoet niet (helemaal) aan die combinatie en komt niet in het zoekresultaat terecht. Enige nuance zit daar niet in. Voor relevantieordening is dus eigenlijk geen plaats. Een methode die wel relevantieordening mogelijk maakt en daarbij ook resultaten kan opleveren die maar deels aan de zoekvraag voldoen, is al in 1975 door Salton bedacht: het vectormodel (Salton 1975). In feite is dat een variant op de inverted file methode. Ook hier vormt een index van alle in een systeem aanwezige termen het uitgangspunt. Elk document wordt vervolgens in een vectorindex gerepresenteerd door een reeks 1-en en 0-en die aangeven welke indextermen wel (1) en welke niet (0) in dat document voorkomen. Als eenvoudig voorbeeld nemen we hier een zeer beperkt informatiesysteem waarin de volgende tien (zeer korte en soms wellicht wat onzinnige) documenten aanwezig zijn:

1: Voor de **toegankelijkheid** van deze **informatie gebruiken** we een **thesaurus**.
2: Voor **browsen gebruiken** we een **classificatie**.
3: **Zoeksystemen** worden **beter** als we een **thesaurus gebruiken** voor **toegankelijkheid**
4: In **zoeksystemen** wordt **vaak postcoordinatie toegepast**
5: In een **classificatie** worden **aspecten** van een **document precoordinatief gekoppeld**
6: Als een **thesaurus** wordt **toegepast** zijn **aspecten** van de **inhoud** van een **document** niet **precoordinatief gekoppeld**
7: **Zoeksystemen** die voor **toegankelijkheid** een **thesaurus gebruiken** worden **vaak toegepast**
8: **Postcoordinatie** van **zoeksystemen** maakt dat je niet kunt **browsen**
9: De **inhoud** van **informatie** in een **document** is **gekoppeld** aan de **postcoordinatie** van een **thesaurus**
10: Hoe **beter** de **toegankelijkheid** van **informatie**, hoe **beter** we de **inhoud gebruiken**

Bij het indexeren worden stopwoorden zoals lidwoorden, voorzetsels, hulpwerkwoorden en dergelijke overgeslagen. De index van dit kleine zoeksysteempje bevat dan de zestien termen in de linker kolom in onderstaande tabel (afbeelding 5). De kolommen daarnaast, genummerd 1 t/m 10, representeren de tien documenten in het systeem. In elke kolom in deze tabel is te zien welke woorden in een bepaald document voorkomen (in document 2 bijvoorbeeld de woorden 'browsen', 'classificatie' en 'gebruiken'). In elke rij zien we in welke documenten een bepaalde term voorkomt (bijvoorbeeld de term 'informatie' in de documenten 1, 9 en 10). Dat is in feite de klassieke "inverted file".

document index	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**zoek-vraag**
aspecten	0	0	0	0	1	1	0	0	0	0	0
beter	0	0	1	0	0	0	0	0	0	1	1
browsen	0	1	0	0	0	0	0	1	0	0	0
classificatie	0	1	0	0	1	0	0	0	0	0	0
document	0	0	0	0	1	1	0	0	1	0	0
gebruiken	1	1	1	0	0	0	1	0	0	1	0
gekoppeld	0	0	0	0	1	1	0	0	1	0	0
informatie	1	0	0	0	0	0	0	0	1	1	0
inhoud	0	0	0	0	0	1	0	0	1	1	0
postcoordinatie	0	0	0	1	0	0	0	1	1	0	0
precoordinatief	0	0	0	0	1	1	0	0	0	0	0
thesaurus	1	0	1	0	0	1	1	0	1	0	1
toegankelijkheid	1	0	1	0	0	0	1	0	0	1	1
toegepast	0	0	0	1	0	1	1	0	0	0	0
vaak	0	0	0	1	0	0	1	0	0	0	0
zoeksystemen	0	0	1	1	0	0	1	1	0	0	0

[Afbeelding 5: Voorbeeld van index en documentvectoren in een vereenvoudigd vectormodel]

De kolommen met 0-en en 1-en in deze tabel kun je wiskundig als vectoren beschouwen, dat wil zeggen als "pijlen" die in een bepaalde richting wijzen. In dit geval zijn dat vectoren in een 16-dimensionale ruimte. Hoewel we ons visueel geen hogere dan een 3-dimensionale ruimte kunnen voorstellen, kan wiskundig ook worden gerekend met vectoren in zulke hoger dimensionale ruimtes. En dat kan ook als er geen 16 maar 16.000 indextermen zijn, dus ook in een 16.000- (of nog veel hoger) dimensionale ruimte.

In een zoeksysteem dat de vectormethode hanteert, worden zoekvragen ook als vectoren voorgesteld, met daarin een 1 voor elke term die we vragen (en impliciet 0-en voor alle termen die we niet vragen). In de rechterkolom van de tabel is de zoekvraag 'beter toegankelijkheid thesaurus' zo als vector gerepresenteerd. Het zoeksysteem voert dan een wiskundige vectorvermenigvuldiging uit tussen de vraagvector en elk van de documentvectoren. Voor elk document levert dat een getal op, dat een maat is voor de overeenkomst tussen de zoekvraag en dat document. Die vermenigvuldiging levert in dit geval het grootste getal op voor document 3 waarin inderdaad alle drie zoekwoorden voorkomen. Dat is dus waarschijnlijk het meest relevante antwoord op onze vraag. De documenten 1, 7 en 10 waarin twee van de drie zoekwoorden voorkomen scoren wat lager, maar kunnen best ook nog enigszins relevant zijn als antwoord op de zoekvraag. Ze komen echter wel op een lagere positie in de resultatenlijst terecht. De documenten met één zoekwoord scoren nog lager, die met geen enkel zoekwoord leveren 0 op.

In de praktijk past men hierop nog allerlei verfijningen toe:

- Voor het samenstellen van de index worden woorden tot hun woordstam gereduceerd, zodat enkel- en meervoud en andere varianten geen afzonderlijke indextermen opleveren (en niet tot onnodige extra "dimensies" leiden).
- Aan de wijze van voorkomen van de indextermen in de documenten worden gewichten gekoppeld, zodat woorden die in een titel of trefwoordveld staan of die herhaald voorkomen, in de vectoren een hogere waarde krijgen. De componenten van de vectoren kunnen dan dus heel andere waarden krijgen dan alleen 0 of 1.
- Op soortgelijke wijze kunnen ook woorden in een zoekvraag een gewicht krijgen op basis van hun (vermoedelijke) belang voor die vraag. Dit is een verfijning die in principe gebaseerd is op het probabilistische model dat in de volgende paragraaf aan de orde komt.
- Voor het vermenigvuldigen worden de lengtes van de vectoren genormeerd, waardoor lange documenten (met heel veel woorden) relatief lager zullen scoren.

De uitkomsten van de vectorvermenigvuldigingen geven zo een veel genuanceerder beeld welke documenten vermoedelijk het beste antwoord op de zoekvraag geven. Wiskundig gezien zijn die uitkomsten dan ook een maat voor de hoek tussen de vraagvector en de documentvectoren; hoe groter de uitkomst, hoe kleiner die hoek, dus hoe meer de twee vectoren in dezelfde richting wijzen en hoe groter de overeenkomst met de vraag.

Ditzelfde model kun je ook toepassen om documenten te vergelijken. Hoe kleiner de hoek tussen twee documentvectoren, zoals afgeleid uit hun vermenigvuldiging, hoe meer die documenten op elkaar lijken. Zo lijken de documenten 3 en 7 enigszins op elkaar, omdat ze 4 woorden gemeen hebben. Datzelfde geldt voor de documenten 5 en 6 en voor 6 en 9. De *more-like-this* zoekfunctie die sommige zoeksystemen bieden, kan hierop berusten. Bovendien kan een systeem zo clusters van op elkaar lijkende documenten herkennen (par. 4.6).

De toepassing van wegingsfactoren voor de woorden in zoekvragen en in de gevonden documenten, zijn eigenlijk al een vorm van probabilistisch zoeken, waarop we in de volgende paragraaf verder ingaan.

2.3 Het probabilistische model

Uitgangspunt van probabilistische zoektechnieken is dat daarin geprobeerd wordt te berekenen wat de *kans* is dat een bepaald document relevant is voor een gestelde zoekvraag. Dergelijke kansen kunnen in een getal, bijvoorbeeld een percentage, worden uitgedrukt. Dat betekent dat deze methode ook sterk gericht is op de relevantieordening van zoekresultaten. Hoe groter de kans, hoe hoger een document in de resultatenlijst komt. Dergelijke kansen worden uitgerekend uitgaande van gegevens zoals:

- de inhoud van de gestelde zoekvraag
- de inhoud van het document
- eerdere reacties van de gebruiker
- et cetera.

De statistiek die daarbij wordt toegepast is de zogenaamde Bayesiaanse waarschijnlijkheidsrekening, genoemd naar de 19de eeuwse Engelse dominee (en wiskundige) Thomas Bayes die deze tak van de statistiek heeft ontwikkeld. Hoe meer gegevens en vooronderstellingen bekend (en getoetst) zijn, hoe groter de betrouwbaarheid waarmee de waarschijnlijkheid van de relevantie van een document voorspeld kan worden. Voor het uitrekenen van die kansen gebruikt men onder meer het gewicht dat woorden in de zoekvraag toegekend krijgen, het gewicht van woorden in de

documenten, de relatie tussen de positie van zoekwoorden in documenten en de kans op relevantie van die documenten en het gewicht (of belang) van individuele documenten.

Termen in de zoekvraag kunnen initieel gewicht krijgen op basis van hun zeldzaamheid, waarbij de aanname is dat het in een document voorkomen van een heel algemeen zoekwoord veel minder doorslaggevend is voor de relevantie van dat document dan de aanwezigheid van een tamelijk zeldzaam woord. Voor het uitrekenen van dat gewicht wordt meestal de volgende wiskundige formule gebruikt:

$$W_{termA} = {}^{2}\log (N_{doc}/N_{termA}) + 1$$

Daarin is W_{termA} het gewicht dat aan een woord (termA) wordt toegekend, N_{doc} het totaal aantal documenten in het systeem en N_{termA} het aantal documenten waarin dat woord (termA) voorkomt. Bedenk overigens wel dat zo'n formule voor de wegingsfactor geen "natuurwet" is. Met die logaritme wordt alleen gezorgd dat berekende gewichten voor zeldzame woorden niet al te groot en voor algemene woorden niet al te klein worden. Voordat deze formule algemeen geaccepteerd werd, moest experimenteel geverifieerd worden dat hij goed bleek te werken.

Als praktijkvoorbeeld van de initiële gewichten van zoekwoorden laten we hier de waarden zien die een probabilistische zoekmachine uitrekende voor de zoekvraag *"asymptotic hopf bifurcation theory"* (een wiskundig onderwerp). Hierbij zijn die gewichten zo genormeerd dat aanwezigheid van alle vier zoekwoorden uitkomt op een totaal van 100% (zie afbeelding 6).

term	frequentie	gewicht
hopf	1763	33 %
bifurcation	4786	29 %
asymptotic	20811	23 %
theory	189984	14 %

[Afbeelding 6: De relatie tussen termfrequentie en termgewicht voor de zoekwoorden in een voorbeeldzoekvraag]

in document aanwezige termen	"relevantie-kans" van document	
asymptotic hopf bifurcation theory	100 %	
asymptotic hopf bifurcation	86 %	
hopf bifurcation theory	77 %	←
asymptotic hopf theory	71 %	
asymptotic bifurcation theory	67 %	
hopf bifurcation	62 %	
asymptotic hopf	56 %	
asymptotic bifurcation	52 %	
hopf theory	47 %	
bifurcation theory	43 %	
asymptotic theory	37 %	
hopf	33 %	
bifurcation	29 %	
asymptotic	23 %	
theory	14 %	

[Afbeelding 7: De initiële relevantiekans van documenten afhankelijk van welke van de gevraagde zoekwoorden daarin voorkomen]

Afhankelijk van welk van die vier woorden in een document aanwezig is, kan dan de relevantiekans van dat document berekend worden. Dat levert de tabel in afbeelding 7. Deze initiële gewichten kunnen eventueel aangepast worden op basis van het zoekgedrag van een gebruiker, zodat de relevantieberekeningen in feite gepersonaliseerd worden. Als een gebruiker vaak op documenten blijkt te klikken met alleen de drie termen waar in afbeelding 7 een pijltje achter staat, dan krijgen die voor latere zoekacties wat hoger gewicht en de niet aanwezige term wat lager gewicht. Kennelijk is voor deze gebruiker het algemene woord "theory" toch wel belangrijk en het veel zeldzamer woord "asymptotic" niet zo erg. Zo worden de gewichten bijgesteld op basis van impliciete gebruikersterugkoppeling; een zelflerend proces. Systemen met expliciete terugkoppeling, waarbij de gebruiker zelf gewichten kan bijstellen, kom je zelden meer tegen, omdat dat teveel ingewikkelde acties van gebruikers vraagt. Toch gaat de methode van Google, waarbij de gebruiker het belang van een bepaalde term kan benadrukken door die in de zoekvraag enkele keren te herhalen, wel een eindje in die richting.

In een gevonden document aanwezige zoekwoorden kunnen gewicht krijgen op grond van hun positie in het document. Een systeem kan bijvoorbeeld zo zijn ingesteld dat de kans op relevantie 30% hoger is wanneer een zoekwoord in de titel van het document voorkomt, dan wanneer het ergens in de gewone tekst staat, en dat die kans 10% hoger is als het woord al in de eerste zin van de tekst staat. Zelfs bepaalde typografische kenmerken, zoals een woord in hoofdletters of in afwijkende kleur of lettertype, die voor bepaald materiaal standaard worden toegepast, kunnen in principe in rekening worden gebracht. De frequentie van voorkomen van een zoekwoord binnen een document speelt vaak eveneens een rol. Het gaat dan echter om de relatieve woordfrequentie. Die bepaalt wat de inhoudelijk belangrijkste en meest karakteristieke woorden in een document zijn. Een daarvoor veel gebruikte techniek is de *TF*IDF* methode *(spreek uit tie-ef-ai-die-ef)*. Daarbij wordt van elk woord uit een document geteld hoe vaak dat woord in het document voorkomt (de term-frequentie *TF*) en dat getal wordt vermenigvuldigd met de *inverse* document frequentie *IDF* (wat een ingewikkelde manier is om te zeggen dat het wordt gedeeld door het aantal documenten waarin dat woord voorkomt). Woorden zijn dus alleen belangrijk als ze in een document vaker voorkomen, maar tegelijk in weinig andere documenten aanwezig zijn. Heel algemene woorden als lidwoorden en voorzetsels met een hoge *TF* scoren dus toch heel laag, doordat door een heel hoge document frequentie gedeeld moet worden (bijv. TF=50; DF=100.000; TF*IDF=0,0005). Een woord dat maar één keer in een document voorkomt kan toch belangrijk zijn, als er maar twee documenten zijn waarin het voorkomt (TF=1; IDF=0,5; TF*IDF=0,5).

Ook worden wel formules opgesteld voor de kans dat een document relevant is voor een complexe zoekvraag, als functie van de onderlinge afstand en de volgorde waarin de zoektermen in dat document voorkomen. Terugkomend op het eerder gebruikte voorbeeld, zou de zin
"according to asymptotic hopf bifurcation theory, we have proved that ..."
dan waarschijnlijk relevanter zijn dan
"in his theory, hopf predicted that this bifurcation behavior was not asymptotic at all".

Ook individuele documenten krijgen vaak gewicht, waarbij men relevantie kan laten samenhangen met kwaliteit en (soms) met "populariteit". Zo kan de kans dat een bepaald document relevant is - nog afgezien van de aanwezige zoekwoorden, samenhangen met

- de mate van gebruik van het document
- de geregistreerde waardering door andere gebruikers
- het aantal hyperlinks naar het document (zoals bij webzoekmachines het geval is).

Door ook de mate van gelijkenis met door gebruiker eerder als relevant aangemerkte documenten te laten meespelen wordt nog een extra vorm van personalisatie geïntroduceerd.

Zoals eerder al aangegeven, zullen al dit soort factoren gekwantificeerd en/of in wiskundige formules moeten worden uitgedrukt, wil het zoeksysteem er aan kunnen rekenen. Daarin zitten een heleboel aannames die eerst nog in de praktijk geverifieerd moeten worden. Zulke grootschalige praktijktests en vergelijkingen worden bijvoorbeeld uitgevoerd bij de jaarlijkse retrieval-competitie TREC (Text REtrieval Conference), waarbij een gecontroleerde vergelijking van vangst en precisie van verschillende systemen plaats vindt. Een heleboel van de daar uitgeteste methodes zitten intussen verwerkt in zowel commerciële enterprise-search systemen, als in de relevantie-ordening van de grote zoekmachines. Allerlei taaltechnologische verfijningen zoals die in paragraaf 4 aan de orde komen, spelen overigens ook een belangrijke rol bij de geleidelijke verbetering van de resultaten van zoekmachines.

3. Problemen van free-text search (vangst en precisie)

Met de eerste generatie *free-text* retrievalsystemen werden alleen documenten gevonden die exact de combinatie van zoekwoorden bevatten, zoals die in de zoekvraag waren ingetikt. Dat leverde een aantal intussen bekende problemen op met betrekking tot vangst (*recall*) en precisie van zoekresultaten (Sieverts 2011). Daarbij is vangst gebruikelijk gedefinieerd als dat deel van de aanwezige relevante documenten, dat met de zoekactie ook daadwerkelijk gevonden is; precisie als dat deel van de gevonden documenten dat ook werkelijk relevant blijkt te zijn.

Los van echte fouten die zoekers kunnen maken, zien we vaak de volgende oorzaken voor het missen van relevante informatie, dus voor slechte vangst:

- in relevante documenten komen andere woordvormen (enkel-/meervoud, vervoegingen, verbuigingen) voor, dan het exacte woord uit de zoekvraag,
- relevante documenten zijn in een andere taal dan de woorden waarmee wordt gezocht,
- in relevante documenten komen woorden in andere spelling voor,
- in relevante documenten zijn synoniemen van de zoekwoorden gebruikt,
- in relevante documenten komen specifiekere woorden voor dan het meer algemene begrip waarmee is gezocht,
- van relevante documenten is zo weinig tekst digitaal beschikbaar en dus doorzoekbaar, dat lang niet alle daarin behandelde onderwerpen en invalshoeken in die digitale tekst gerepresenteerd zijn (zoals in veel bibliotheekcatalogi).

Evenzo kennen we de volgende oorzaken voor het vinden van niet-relevante informatie (ook wel "ruis" genoemd), dus voor slechte precisie:

- een gebruikt zoekwoord komt in documenten (ook) in andere betekenis voor dan die welke door de zoeker bedoeld was,
- een gebruikt zoekwoord komt in andere, niet bedoelde context voor,
- de in een Booleaanse AND-combinatie gekoppelde zoekwoorden komen wel samen in een gevonden document voor, maar blijken daarin niet de inhoudelijke relatie te hebben die met de zoekvraag bedoeld werd (foute of ontbrekende verbanden),
- in lange teksten komen veel woorden voor die weinig representatief zijn voor datgene waar het document echt over gaat, maar waarop het wel wordt gevonden.

Veel van die problemen spelen veel minder als je bij het zoeken gebruik kunt maken van gecontroleerd vocabulaire. Als voornaamste doel heeft dat immers om

- terminologie eenduidig te maken en te standaardiseren, zodat geen problemen meer optreden met woordvormen, spellingsverschillen, taalverschillen, synoniemen of homoniemen,
- zoekingangen bij benadering gelijk gewicht te geven, zodat documenten alleen nog op daarvoor relevante termen worden gevonden en bij diepte-indexering, waarbij ook termen voor deelonderwerpen van een publicatie worden toegekend, in principe ook op alle relevante termen,
- relaties te leggen tussen begrippen, zodat zoekresultaten makkelijk (soms automatisch) kunnen worden uitgebreid met publicaties over specifiekere of deelonderwerpen (zoals bij generiek zoeken met behulp van een thesaurus) of over verwante onderwerpen,
- (soms) precoördinatief syntactische relaties te leggen tussen begrippen die in een document aan de orde komen (zoals de subheadings uit de MeSH-thesaurus die in de PubMed database worden gebruikt).

Gezien het gebrek aan flexibiliteit van gecontroleerde ontsluiting en vooral ook gezien de kosten die verbonden zijn aan het handmatig ontsluiten van informatie is het goed te kijken welke automatische technieken intussen in free-text retrievalsoftware worden toegepast, die op zijn minst een gedeeltelijke oplossing voor de genoemde problemen bieden. In de volgende paragrafen komen die aan de orde.

4. Verbeteren van vangst en precisie door taaltechnologie

4.1 Woordstammen

Al lang bestaat in veel zoeksystemen de mogelijkheid van trunkeren of maskeren: het zoeken naar alle woorden die beginnen met een bepaald door de gebruiker ingetikt woorddeel ('computer*' levert ook 'computeronderwijs') of waar op een bepaalde plaats in het woord een variabele letter mag voorkomen ('publi?atie' levert zowel 'publicatie' als 'publikatie').

Trunkeren biedt een tamelijk ruwe manier om op variaties van woorduitgangen te zoeken. Bovendien moeten gebruikers zelf bedenken hoe dat te doen. Daarom wordt in zoeksystemen steeds vaker *word stemming* toegepast. Ook Google is daar een voorbeeld van. Door bij het indexeren van tekst de daarin voorkomende woorden te reduceren tot hun morfologische woordstam (bijvoorbeeld: computer → comput) en dat bij zoekwoorden ook te doen, maakt het niet meer uit welke woordvormen toevallig in de te doorzoeken teksten voorkomen (enkelvoud, meervoud, verbuiging, vervoeging, enzovoort - computers, computing, computation, computed) en evenmin op welke woordvorm iemand zoekt. Als zodanig heeft het een positieve invloed op de vangst van zoekacties. De meest toegepaste technieken voor *word-stemming* zijn gebaseerd op regels die in principe voor (bijna) alle woorden in een taal zouden moeten gelden.

Probleem met deze methode is dat er altijd uitzonderingen kunnen voorkomen, waarbij de algemene regels voor *stemming* niet tot correcte resultaten leiden (bijvoorbeeld: communism - community - communication). Om *word-stemming* effectief te laten zijn, moet gezorgd worden dat noch *under-stemming*, noch *over-stemming* optreedt (Braschler 2004). Moet van het woord 'hypothetical' alleen de uitgang '..al', wellicht beter '..ical' of misschien zelfs '..etical' worden afgehakt om de juiste woordstam (hypothetic / hypothet / hypoth) over te houden? En is dat bij het woord 'chemical' net zo? *Over-stemming*, het

verwijderen van te lange woorduitgangen, zal over het algemeen nogal nadelig zijn voor de precisie van zoekresultaten; *under-stemming* juist weer voor de vangst. *Stemming* geeft meer verbetering van de vangst, naarmate de hoeveelheid digitaal doorzoekbare tekst van de documenten kleiner is. In lange documenten zullen de verschillende vormen van belangrijke woorden toch vaak al allemaal aanwezig zijn.

Een ander probleem is dat deze methode sterk taalafhankelijk is en niet voor alle talen even eenvoudig mogelijk. Zo zal bij (regelmatige) werkwoorden in het Engels alleen aan het eind van het woord iets variëren (walk - walks - walked - walking), maar in het Nederlands ook aan het begin en zelfs vaak in aanwezige klinkers (lopen - loopt - liep - gelopen). Gezien de moeilijkheden die het werken met vaste regels soms oplevert, wordt ook wel eens met woordenlijsten gewerkt waarin expliciet is vastgelegd welke afzonderlijke woorden tot dezelfde woordstam gereduceerd moeten worden.

4.2 Fuzzy zoeken

Een ander soort woordvarianten, met name die in spelling, kan automatisch worden meegenomen door technieken van *fuzzy* zoeken. Er bestaan verschillende technieken om de computer te laten zoeken naar woorden die wat betreft spelling of uitspraak sterk lijken op de woorden in de zoekvraag. Deze techniek kan compenseren voor spelfouten in zowel de documenten als in de zoekvragen, alsmede voor spellingsvariaties en in beperkte mate voor morfologische varianten. Ook voor OCR-fouten in gescande teksten geldt dit. Hierbij hoeft de gebruiker dus niet meer zelf letters te maskeren. Soms kan bij het zoeken de mate van *fuzziness* worden ingesteld – ruwweg het aantal letters waarin gevonden en gevraagde woorden maximaal mogen verschillen. Bij deze techniek gaat de verbetering van de vangst overigens vaak ten koste van de precisie. Eigenlijk alleen bij vrij lange zoekwoorden zal de precisie van zo verkregen zoekresultaten voldoende goed blijven. Korte woorden waarvan één letter verschilt, hebben meestal al een heel andere betekenis (bijvoorbeeld: boek → boer – bonk – koek).

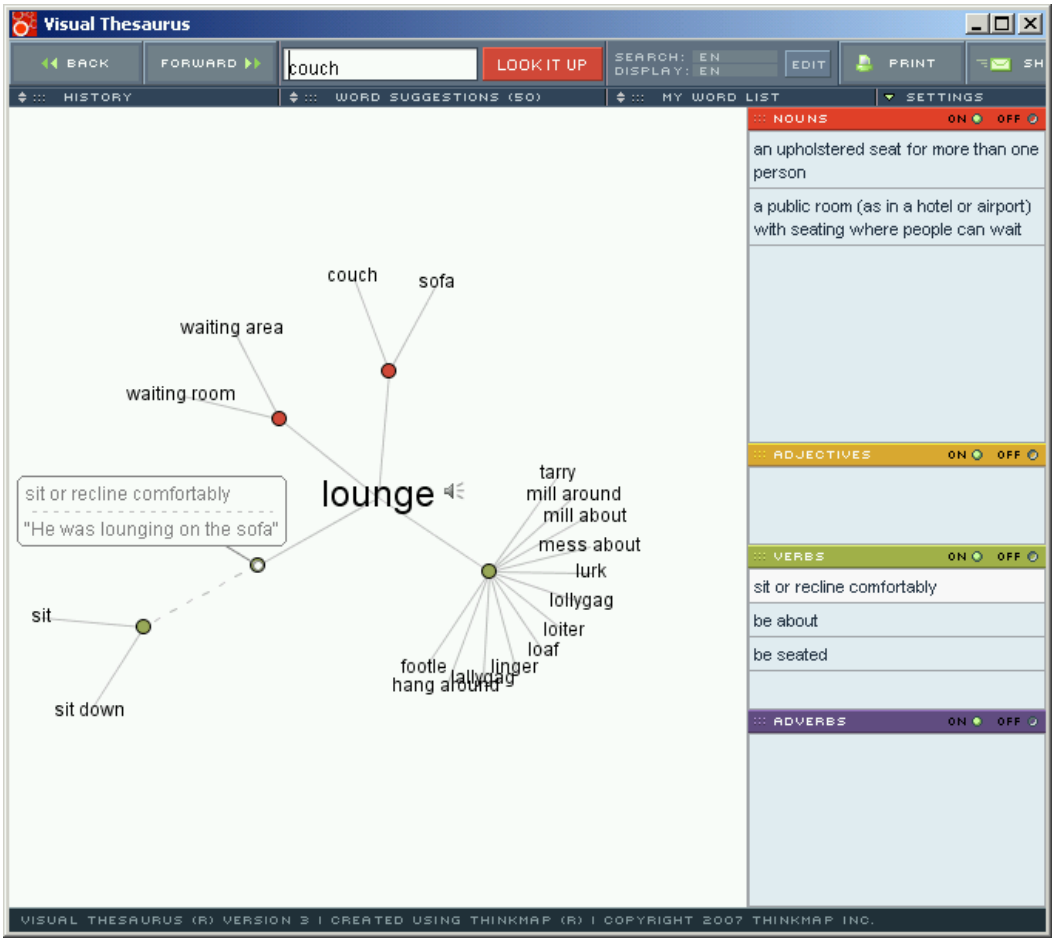
4.3 Splitsen van samengestelde woorden

In het Engels komen haast geen samengestelde woorden voor. In talen als Nederlands, Duits en Deens echter wel (Braschler 2004). Daardoor zul je een document over bijvoorbeeld een 'hockeytoernooi' niet automatisch vinden op de zoekterm 'toernooi'. Gewone trunkatie, *word-stemming* en *fuzzy* zoeken zijn hiervoor geen remedie. Om ook in dit soort situaties de vangst te verbeteren zullen dergelijke samengestelde woorden bij het indexeren automatisch in hun afzonderlijke bestanddelen opgesplitst moeten worden, zogenaamde *decompounding*. Daarbij ligt het vaak heel subtiel voor welke woorden het zinnig is om ze wel of juist niet te splitsen. Zo zullen de woorden 'voorjaar', 'voorhoede' of 'najaar' zeker niet gesplitst moeten worden, maar voor betere retrieval is dat bij 'vooraankondiging', 'jubileumjaar' of 'naheffing' weer wel nuttig. Ook het punt waar het systeem een woord moet splitsen is niet altijd makkelijk eenduidig te bepalen. In kranten gebruikte automatische afbreek-routines geven hier soms hilarische voorbeelden van. Denk bijvoorbeeld aan kerst-omaatjes of park-eerst-rook.

4.4 Disambiguering

De technieken in de vorige paragraaf waren vooral gebaseerd op tamelijk mechanische manipulatie met letters en onderdelen binnen woorden, waarbij de betekenis van woorden nog vrijwel geen rol speelde. Door kennis over de betekenis van woorden toe te voegen kan zoeken ook verder worden verbeterd. Dat kan onder meer door gebruik te maken van een semantisch netwerk. Dat kun je beschouwen als een digitale versie van een woordenboek, waarin niet alleen (verschillende) betekenissen van woorden zijn

vastgelegd, maar ook allerlei betekenisrelaties tussen de woorden. Door het leggen van die relaties vormen de woorden een netwerk, waarin elk woord met elk ander woord verbonden kan zijn, ook al is dat misschien via de tussenstappen van relaties met een heleboel tussenliggende woorden. Hoeveel tussenstappen daarvoor nodig zijn voor een willekeurig tweetal woorden, noemt men wel de semantische afstand van die twee woorden. Zo zal de semantisch afstand tussen de woorden 'voetbal' en 'stadion' waarschijnlijk kleiner zijn dan die tussen 'voetbal' en 'viool'. Afbeelding 8 toont een visualisatie van een stukje van zo'n semantisch netwerk voor de Engelse taal.



[Afbeelding 8: Visualisatie van de semantische omgeving van een woord in het Engelse Wordnet (gemaakt met VisualThesaurus software)]

Bij het vastleggen van de betekenisrelaties worden bovendien de verschillende betekenisen van een woord onderscheiden. Door de in de omgeving van een woord voorkomende tekst in een document te vergelijken met de omgeving van de verschillende betekenissen van dat woord in het semantisch netwerk, kan nu worden afgeleid in welke betekenis dat woord in de tekst voorkomt. Als in de buurt van het woord 'bank' woorden voorkomen als geld, giro, beurs of pinnen, kan een computer in principe afleiden dat het op die plaats in de tekst zeer waarschijnlijk om een financiële instelling zal gaan, omdat deze woorden in het semantisch netwerk op enigerlei wijze aan die betekenis van bank gerelateerd zijn. Als er daarentegen sprake is van kussens, zitten, tv-kijken, tafeltje of schemerlamp, dan zal het waarschijnlijk om een zitmeubel gaan. Als een computerprogramma op deze wijze de betekenis van multi-interpretabele woorden kan afleiden, kan dus specifiek op een gewenste betekenis worden gezocht, hetgeen goed is voor de precisie.

Voorwaarde voor toepassing van deze techniek is uiteraard de beschikbaarheid van een semantisch netwerk van de taal (of talen) waarin de te doorzoeken documenten gesteld zijn. Voor veel talen bestaan intussen dergelijke geavanceerde digitale woordenboeken. Zelfs bestaan die meertalig, zoals het meertalige semantische netwerk dat op de Van Dale vertaalwoordenboeken is gebaseerd. Ze beperken zich echter meestal tot de gewone dagelijkse taal. Gespecialiseerd vakmatig of wetenschappelijk vocabulaire maakt daar vrijwel nooit standaard al deel van uit. Als dat van belang is, zal men een gespecialiseerde leverancier in de arm moeten nemen om dat toe te voegen.

Ook zonder dat zulke inhoudelijke kennis al a priori in een semantisch netwerk is vastgelegd, bestaan er mogelijkheden om betekenissen te onderscheiden. Die berusten dan op statistische analysemethoden van het samen voorkomen van woorden in gevonden documenten. Clustering van zoekresultaten is daar een voorbeeld van. Daarop gaan we in paragraaf 4.6 nog afzonderlijk in. Een ander voorbeeld zijn statistisch gegenereerde extra zoektermen, waarmee vragen verfijnd en daarmee tot een specifieke betekenis ingeperkt kunnen worden. Daar komen we in paragraaf 4.7 verder op terug. In de afbeeldingen die daar ter illustratie worden getoond, zien we overigens dat niet alle uit statistiek afgeleide onderscheiden ook inhoudelijk zinvol zijn.

4.5 Vraaguitbreiding

Ter verbetering van de vangst kan een zoekactie automatisch uitgebreid worden door ook verwante (of zelfs synonieme) begrippen uit het semantisch netwerk in de zoekvraag mee te nemen. Dat zijn dan de woorden die binnen een bepaalde semantische afstand van de oorspronkelijke zoekterm vallen. In principe kan een zoeksysteem deze termen ook eerst ter keuze aan de zoeker voorleggen.

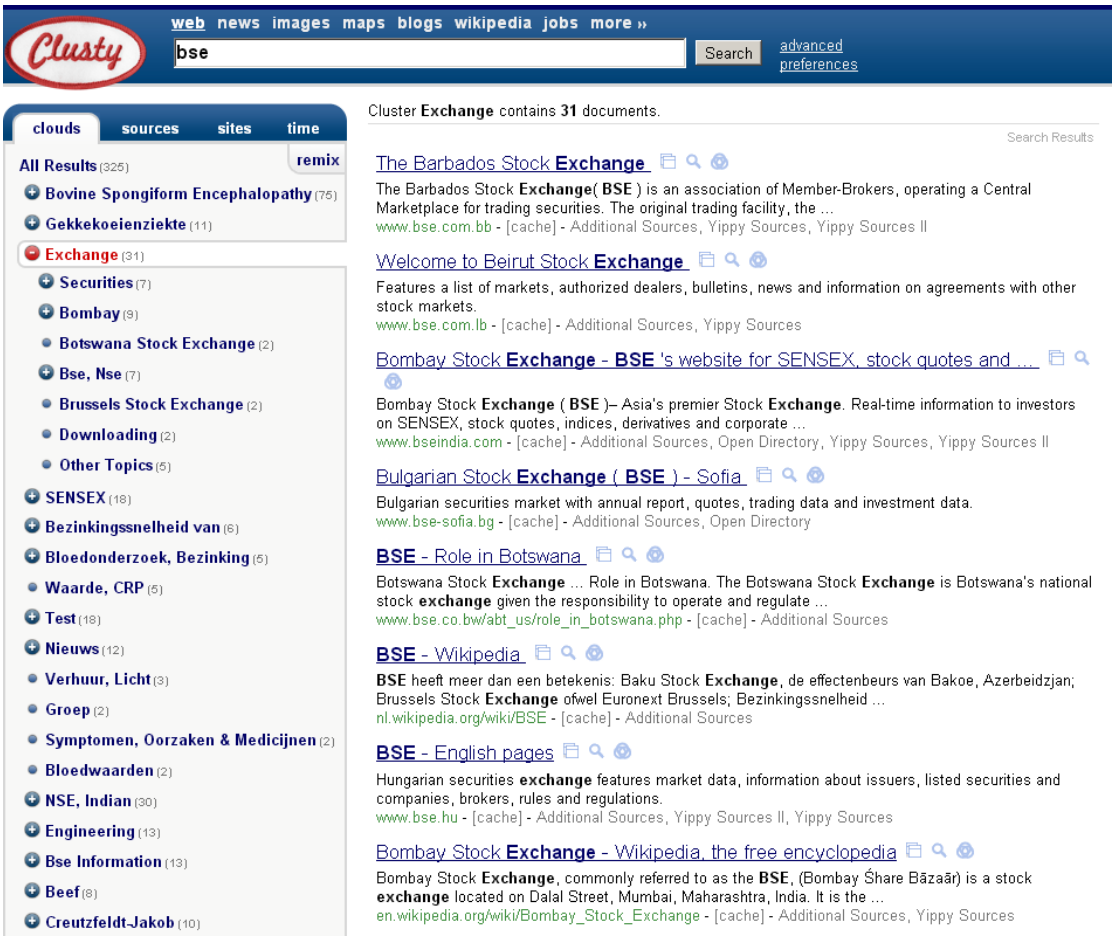
Van nature zijn gebruikers van zoeksystemen zich er nauwelijks van bewust dat bij het zoeken op een algemeen begrip niet altijd automatisch resultaten voor specifiekere begrippen worden gevonden, zodat je bij zoeken op bijvoorbeeld 'Europa' vaak veel informatie over 'Frankrijk' mist. Veel gebruikers realiseren zich evenmin dat zoeken op specifieke begrippen vaak meer resultaten oplevert dan zoeken op meer algemene begrippen, zeker in collecties met veel gespecialiseerd materiaal, zoals wetenschappelijke artikelen en hoofdstukken uit monografieën. Ook daarvoor kan vraaguitbreiding dus heel nuttig zijn. Wanneer een hiërarchische thesaurus voor een bepaald vakgebied beschikbaar is, kan daarvan gebruik worden gemaakt om de zoekvraag via zogenaamd generiek zoeken - liefst automatisch - te expanderen. In de PubMed database op internet gebeurt dit inderdaad ongevraagd, waarbij zelfs een koppeling met relevante thesaurustermen wordt gelegd als de gebruiker een ander zoekwoord heeft ingetikt.

Onderzoek heeft ook in de praktijk bevestigd dat zo'n automatische vraaguitbreiding met specifiekere termen uit een thesaurus tot betere vangst leidt, zonder verslechtering van de precisie (Greenberg 2001a,b). Dat bleek ook te gelden voor het toevoegen van synoniemen. Vraaguitbreiding met ruimere en verwante begrippen (BT en RT) ging - zoals ook wel te verwachten viel - wel ten koste van de precisie. Voor die termen kan de gebruiker dus beter zelf de keuze worden gelaten welke daarvan hij toch aan de zoekvraag zou willen toevoegen. De gebruiker moet die dus als keuzelijstje van mogelijke aanvullende termen gepresenteerd krijgen.

4.6. Clusteren van resultaten

Een manier om de precisie van een al verkregen zoekresultaat te verbeteren is automatische opdeling van dat resultaat in clusters. Die clusters kunnen gebaseerd zijn op tevoren gedefinieerde categorieën of tot stand komen op basis van statistische analyse van de tekst van de gevonden documenten. De gebruiker kan vervolgens die cluster(s) kiezen die het best aansluit(en) bij zijn/haar informatiebehoefte. Wie bijvoorbeeld op BSE zoekt, kan dan de keuze krijgen tussen clusters als 'mad cow disease' (bovine spongiform encephalopathy), 'stock exchange' (Bombay / Beirut / Boston Stock Exchange) en 'cancer detection' (breast self examination), waarvan de omschrijvingen allemaal tot BSE worden afgekort.

Voor het werken met tevoren (door mensen) gedefinieerde categorieën dienen kennisregels beschikbaar te zijn, op basis waarvan de computer de documenten kan indelen. Die kennisregels kunnen handmatig zijn opgesteld of uit voorbeelddocumenten zijn afgeleid, op vergelijkbare wijze zoals dat ook gebeurt bij automatisch categoriseren of metadateren van documenten (Becker 2010, van Aalten 2011). Bij het full-text indexeren van de documenten, worden die dan ook meteen volgens deze regels gecategoriseerd. Het resultaat van een zoekactie kan vervolgens gemakkelijk worden uitgesplitst op basis van de aan de gevonden documenten toegekende categorieën. De benaming van die categorieën is tevoren al door hun bedenker vastgelegd.



[Afbeelding 9: Een in clusters opgedeeld zoekresultaat van de metazoekmachine Clusty (http://clusty.com)]

Omdat het opstellen van kennisregels voor categorisering erg arbeidsintensief is, zeker voor brede onderwerpsdomeinen, wordt tegenwoordig veel meer gebruik gemaakt van automatische processen. Daartoe kan bijvoorbeeld gebruik gemaakt worden van vectorrepresentaties van de documenten in het zoekresultaat, zoals beschreven in paragraaf 2.2. Groepen documenten waarvan de vectoren ongeveer in dezelfde richting wijzen, vormen dan een cluster. Het resultaat hiervan is erg vergelijkbaar met dat van een methode waarbij een statistische analyse wordt uitgevoerd van de tekst van de gevonden documenten. Het samen voorkomen van groepen karakteristieke woorden is dan een maat voor de gelijkenis tussen documenten. Dit kan alleen worden toegepast op niet te grote zoekresultaten, dus bijvoorbeeld niet op 300.000 resultaten uit een webzoekmachine, maar wel op de enkele honderden resultaten die een metazoekmachine uit achterliggende echte zoekmachines heeft opgehaald, zoals onder meer gebeurt bij Clusty (zie afbeelding 9) en bij Polymeta. De benaming van dergelijke clusters wordt automatisch vastgesteld op grond van een karakteristiek woord of zinsnede uit zo'n cluster, hetgeen niet altijd tot even zinnige omschrijvingen leidt.

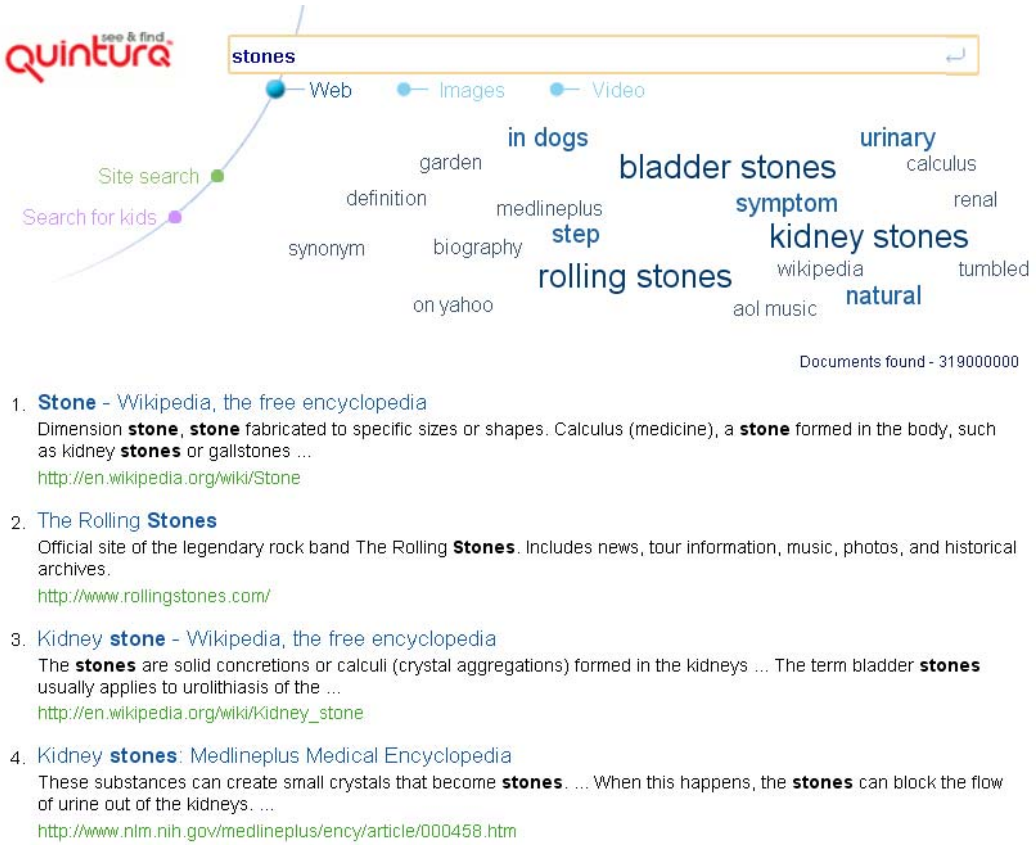
Een andere wijze van uitsplitsing die je in steeds meer zoeksystemen tegenkomt, is gebaseerd op gestandaardiseerde metadata die toch al in het doorzochte materiaal aanwezig zijn. Bij deze zogenaamde *parametric* of *faceted search* kan een zoekresultaat worden uitgesplitst op basis van verschillende facetten, bijvoorbeeld de aard van gevonden documenten (webpagina, pdf, word-document, boek, krantenartikel), de taal van de gevonden publicaties, de namen van daarin frequent voorkomende auteurs, de titels van de tijdschriften waaruit gevonden artikelen afkomstig zijn, globale onderwerps-categorieën en dergelijke. Die uitsplitsing vermeldt vaak meteen al hoeveel na inperking overblijft van het oorspronkelijke resultaat. Het clusteren van een zoekresultaat in tevoren bedachte categorieën, waarmee we deze paragraaf begonnen, is in feite ook een toepassing van faceted search, alleen dan op basis van automatisch toegekende metadata voor die categorieën. In het voorbeeld van de Aquabrowser in afbeelding 10 is een opdeling in diverse facetten zichtbaar.

4.7 Genereren van termen voor vraagverfijning

Een statistische analyse van de woorden in een zoekresultaat kan ook aanleiding geven tot een lijst karakteristieke losse woorden of zinsneden die daarin vaker voorkomen, zonder dat de gevonden documenten op basis daarvan al worden geclusterd. Voor het vaststellen welke woorden karakteristiek (of specifiek) genoeg zijn, worden dezelfde TF*IDF technieken toegepast als we in paragraaf 2.3 al tegenkwamen. Die techniek voorkomt dat woorden in de keuzelijstjes terechtkomen die wel veel in het zoekresultaat voorkomen, maar zo algemeen zijn dat ze in bijna elk document veel voorkomen.

De zoeker kan dan woorden of zinsneden uit het getoonde lijstje aanklikken om ze (in een AND-relatie) aan de oorspronkelijke zoekvraag toe te voegen en zo het zoekresultaat verder in te perken. De bij de Openbare Bibliotheken in Nederland en Vlaanderen gebruikte Aquabrowser, visualiseert de zo gegenereerde woorden in een soort woordenwolk. Daarin worden overigens ook woorden met enige (fuzzy) gelijkenis met een gevraagd zoekwoord en woorden uit eventuele semantische hulpsystemen getoond (zie afbeelding 10). Deze statistische methode is ook toepasbaar op zeer grote zoekresultaten, door de analyse alleen te laten plaatsvinden op een behapbaar deel van het zoekresultaat onder de aanname dat die wel representatief zal zijn voor het hele zoekresultaat. De Quintura software die dergellijke analyses maakt op basis van de zoekresultaten van een bestaande zoekmachine, is hier een voorbeeld van (afbeelding 11).

[Afbeelding 10: Zoekresultaat in een Aquabrowser. Links een woordenwolk met zowel statistisch gegenereerde woorden, als verwante woorden uit een thesaurus of een semantisch netwerk, alsmede fuzzy gelijkende woorden. Rechts is het zoekresultaat opgedeeld op basis van formele facetten.]



[Afbeelding 11: Woordenwolk van statistisch gegenereerde woorden en begrippen uit een eerder zoekresultaat in de Quintura webzoekmachine. (Voor dit voorbeeld werden nog resultaten uit Yahoo! gebruikt; op het moment van schrijven maakt Quintura gebruik van een Russische zoekmachine)]

Om behalve tot losse woorden ook tot zinsnedes (of andere combinaties van woorden) te komen is niet zo eenvoudig. Hoe een computerprogramma kan bepalen wat zinnige zinsneden zijn om in de statistische analyse te betrekken, is een belangrijk veld van onderzoek. Niet elk twee- of drietal woorden uit een zin komt daar namelijk voor in aanmerking. Naast vooral in het Engels veelvuldig voorkomende *noun phrases* (opeenvolgingen van twee zelfstandig naamwoorden, zoals 'information retrieval'), kunnen dat ook complexere stukken zin zijn, met voorzetsels, bijvoeglijke naamwoorden en dergelijk (zoals 'grote rode auto' of 'opwarming van de aarde'), die als geheel een concept of begrip kunnen representeren. Om deze zogenaamde *lexical phrases* te kunnen onderscheiden van willekeurige reeksen opvolgende woorden, is tamelijk geavanceerde taalkundige analyse van de teksten nodig.

5. Terugkoppeling door gebruikers

Behalve de in paragraaf 4.7 besproken suggesties voor inperkingen waaruit een gebruiker kan kiezen, zijn er ook andere manieren waarop terugkoppeling van het oordeel van de gebruiker tot verbetering van de precisie van zoekresultaten kan leiden. Een eerste methode daarvoor is de *more-like-this* functie. Daarbij wordt op basis van een al gevonden en door de gebruiker als zeer relevant beoordeeld document verder gezocht naar documenten die daar sterk op lijken. Die gelijkenis zal veelal worden bepaald op basis van de woorden die in het voorbeelddocument voorkomen. De vectormethode is een hiervoor vaak gebruikte techniek, zoals in paragraaf 2.2 al werd beschreven. Er zijn echter ook wel andere technieken om op de meest karakteristieke woorden uit het voorbeelddocument verder te zoeken. In sommige gevallen wordt de gelijkenis ook op heel andere wijze bepaald. Google gebruikt daarvoor bijvoorbeeld linkpatronen tussen webpagina's. Hoewel je zou verwachten dat het resultaat van deze zoekmethoden altijd tot goede precisie leidt, blijkt dat in de praktijk toch nog wel eens tegen te vallen.

Een andere wijze van terugkoppeling is die waarbij de gebruiker actief kan aanvinken welke van de zoekresultaten hij als relevant beoordeelt. Zoeksystemen kunnen echter ook zo geprogrammeerd worden dat ze zelfstandig deduceren welke documenten de gebruiker kennelijk relevant vindt. Ze registreren dan welke resultaten de gebruiker metterdaad opvraagt en hoe lang hij daarnaar kijkt. Op basis daarvan zullen de meest karakteristieke woorden uit die documenten een hoger gewicht krijgen ten behoeve van toekomstige zoekacties. Documenten die deze woorden bevatten, zullen in de relevantieordening dan hoger scoren dan tevoren. Bij de introductie van de probabilistische zoekmethode (par. 2.3) werd al uitgelegd hoe zoiets tot een zelflerend systeem kan leiden, dat geleidelijk - in de praktijk heel subtiel - steeds iets betere precisie moet opleveren, specifiek toegesneden op de persoonlijke belangstelling en voorkeuren van de gebruiker en daarmee ook mee-evoluerend.

6. Relevantieordening van resultaten

De relevantie-ordening die steeds meer zoeksystemen en zoekmachines toepassen - zeker op het web - is bedoeld als remedie tegen lage precisie. Die techniek is er immers op gericht om de precisie van het eerst getoonde deel van de resultaten te verbeteren. De beste antwoorden moeten bij de eerste tien - of liever nog de eerste drie - resultaten zitten. Een aantal factoren dat daarbij meespeelt kwam al aan de orde bij de probabilistische zoektechnieken (par. 2.3) en de methoden voor zoekverbeteringen. Hier sommen we de belangrijkste factoren nog eens wat systematischer op.:

1. Waar een gevraagde term in de gevonden documenten staat is van belang. Als die op een belangrijke plek in een document staat (in de titel, in koppen, in opsommingen, in de eerste paar regels) zal dit document hoger scoren.
2. Als een gevraagde term in een gevonden document herhaald voorkomt, zal dit document hoger scoren.
3. Als gevraagde woorden in een gevonden document dicht bij elkaar staan, zal dit document hoger scoren.
4. Dat laatste effect zal nog sterker gelden als de gevraagde woorden in een gevonden document ook in dezelfde volgorde voorkomen als in de zoekvraag.
5. Bij de voor deze ordening uitgevoerde berekeningen krijgen zeldzame termen zwaarder gewicht dan algemene, omdat zeldzame begrippen meer onderscheidend zijn en belangrijker, want specifieker, delen van de zoekvraag representeren (zie 2.3).
6. Als (in een webomgeving) veel hyperlinks van andere sites naar een gevonden document verwijzen, zal dit document hoger scoren (denk aan Google's "pagerank").
7. Als een gevonden document of website veel wordt gebruikt of bezocht, zal dit document hoger scoren.
8. Een document zal hoger scoren wanneer het (sterk) lijkt op documenten die de gebruiker al eerder raadpleegde.

Hierbij proberen de factoren 1 en 2 onderscheid te maken tussen woorden die in een document inhoudelijk centraal staan en die welke min of meer toevallig voorkomen. De factoren 3 en 4 zorgen dat die documenten eerder worden getoond, waarin de zoekwoorden vermoedelijk in de gewenste (syntactische) relatie tot elkaar staan. De factoren 6 en 7 zijn meer een poging om ook de factor kwaliteit (soms wellicht verward met populariteit) te laten meespelen. Factor 8 probeert het resultaat aan te passen aan de ingeschatte persoonlijke voorkeur van de gebruiker.

Webzoekmachines houden meestal ook nog rekening met tientallen andere verfijnde factoren bij het berekenen van relevantiescores van gevonden documenten. Veel van die factoren zijn echter vooral gericht op het neutraliseren van ongewenste technieken van *search engine optimalisatie* (SEO), waarmee websitebouwers op kunstmatige wijze proberen pagina's hoog te laten scoren in de relevantieberekeningen.

7. Semantisch zoeken

Semantisch zoeken is het nieuwe paradigma waarop alle zoekmachineontwikkelingen zich lijken te richten - zeker voor het web. Dat "semantisch" houdt in dat zoek- en analyse-techieken worden toegepast waarbij de computer iets weet over de betekenis van de woorden in zowel zoekvragen als gevonden documenten. In het voorafgaande waren we al wat klassiekere semantische aspecten tegengekomen, toen het gebruik van semantische netwerken en thesauri aan de orde kwam. Echt semantische zoeksystemen horen echter nog een stapje verder te gaan met dat begrijpen. Of dat ook werkelijk het geval is bij alle systemen die van het etiket "semantisch" worden voorzien, is nog maar de vraag. In de praktijk blijkt semantisch zoeken namelijk een soort containerbegrip waar men allerlei doelstellingen en technieken onder laat vallen (Grimes 2010, John 2012, Starr 2012). Doelen en methoden die vaak genoemd worden in overzichten van hetgeen semantisch zoeken inhoudt, vallen globaal in de volgende drie categorieën:

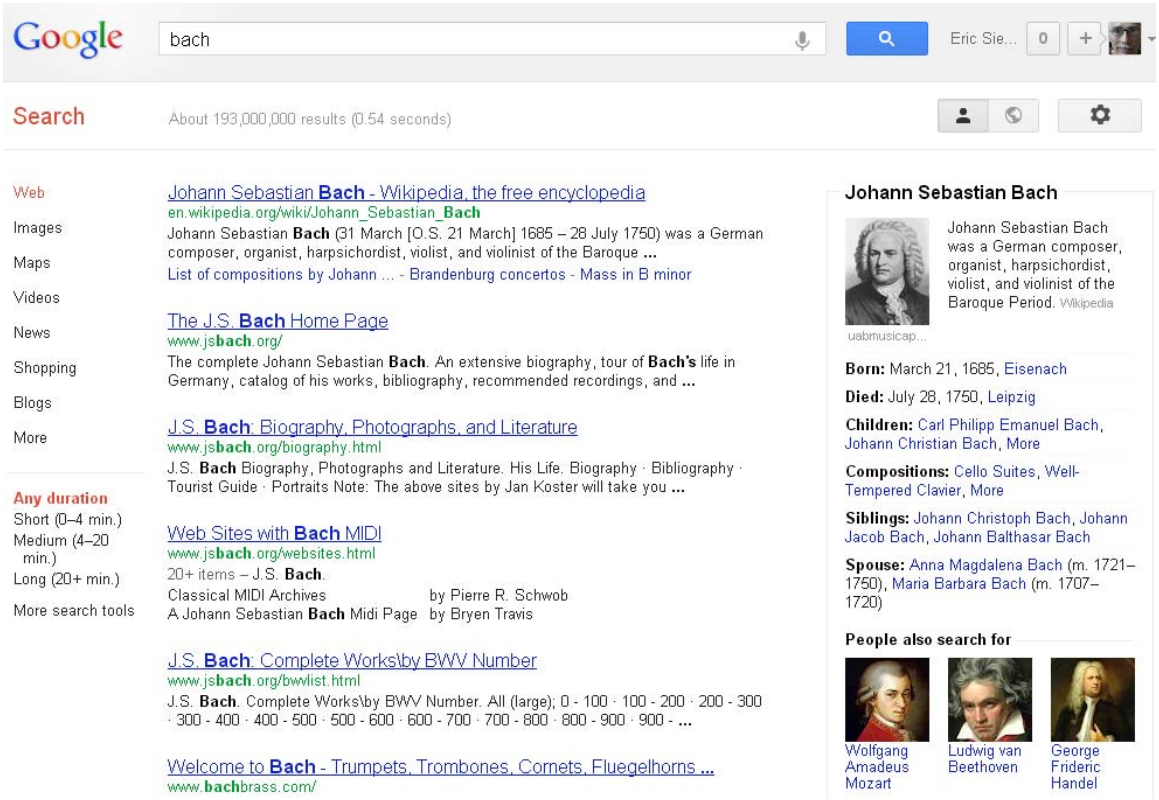
- inschatten van doel of context van zoekvragen,
- vooraf (bij indexeren) of achteraf (na een zoekactie) analyseren van tekst om betekenis van woorden af te leiden,
- automatisch aanpassen van zoekvragen op basis van de betekenis van de oorspronkelijk gebruikte zoekwoorden.

Over elk daarvan hier iets meer details.

7.1 Doel en context van zoekvragen

Sommige zoekmachines - vooral op het web - proberen een inschatting te maken van het meest waarschijnlijke doel waarvoor een zoekvraag wordt gesteld. Daartoe kan een zoekmachine diverse soorten indicaties gebruiken, zoals gegevens over de locatie van de gebruiker, eerder zoekgedrag van die gebruiker en de aard en formulering van de zoekvraag. De te gebruiken locatieïnformatie kan zich beperken tot de landenversie van de zoekmachine die wordt gebruikt. Wie op *google.nl* zoekt krijgt bij veel vragen andere antwoorden dan wie *google.be* gebruikt of standaard naar *google.com* gaat. Zoek maar eens op "apple" in deze drie versies. Nu voor het stellen van zoekvragen steeds vaker mobiele apparatuur wordt gebruikt, kan het ook om veel gedetailleerder informatie gaan, omdat de locatie van de gebruiker dan veelal heel precies bekend is. Een zoekvraag als "pizza" zal dan geïnterpreteerd kunnen worden als de wens er eentje te gaan eten en niet als een zoekactie naar een recept of naar de geschiedenis van de pizza, zodat informatie over restaurants in de onmiddellijke nabijheid van de zoeker getoond zal worden.

Ook eerder zoekgedrag van de gebruiker en eerder geraadpleegde websites kunnen een indicatie geven van de waarschijnlijke context waarbinnen een vraag gesteld wordt. Dit lijkt erg op de eerder (paragraaf 2.3 en 6) besproken personalisatie van zoekresultaten.



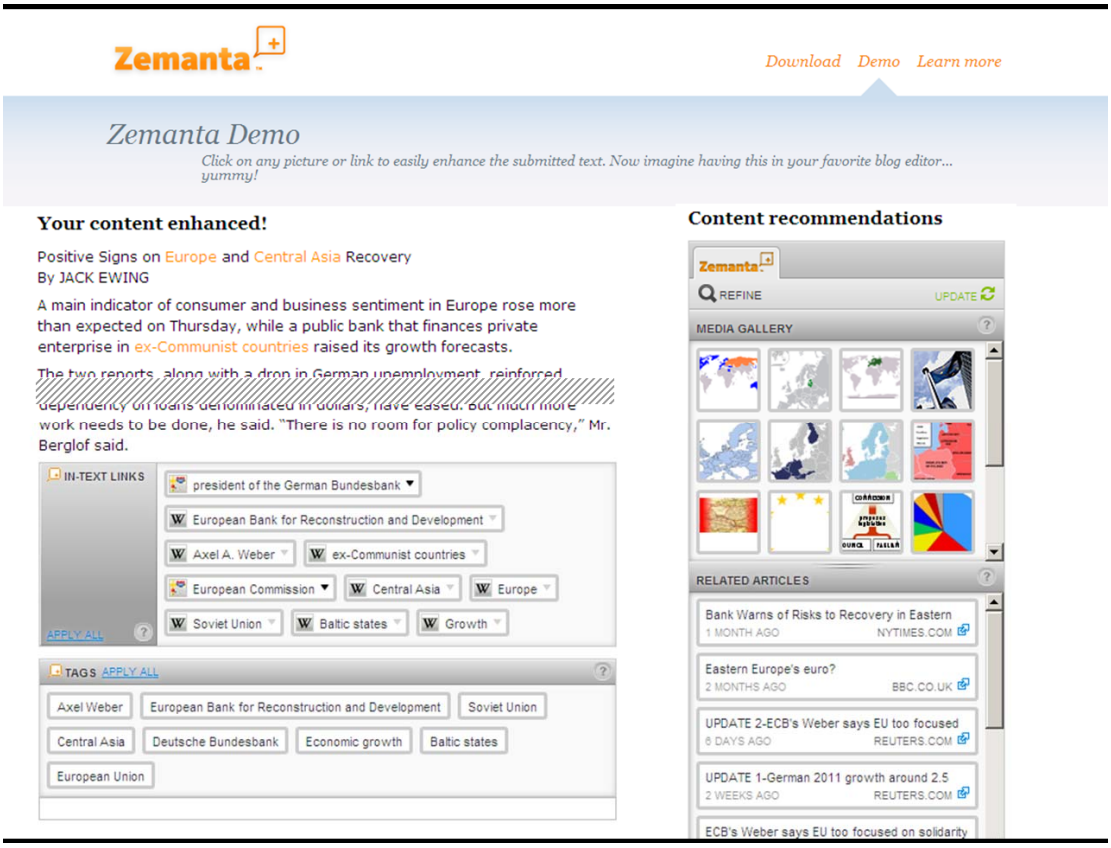
[Afbeelding 12: Voorbeeld van automatisch getoonde feitelijke informatie uit Google's "Knowledge Graph" bij een daarvoor in aanmerking komende zoekvraag]

Bij wat uitgebreider geformuleerde zoekvragen kan door middel van natuurlijke taaltechnieken getracht worden de aard van de zoekvraag te achterhalen. Bij een vraag als "geboortejaar Beethoven" zal het heel waarschijnlijk zijn dat het de zoeker alleen te doen is om een jaartal, en dat geen uitgebreide lijst zoekresultaten gewenst is. Als antwoord

kan dan dat feitelijke gegeven getoond worden, dat is opgevraagd uit een aan het systeem gekoppelde reference collectie. De door Google in 2012 geïntroduceerde Knowledge Graph is een voorbeeld van een systeem waarbij dit gebeurt (afbeelding 12). Zelfs bij kortere vragen naar alleen een persoonsnaam, stad, land of bedrijf, zullen vaak meteen al feitelijke gegevens getoond worden, overigens nog wel naast een lijst gewone zoekresultaten. Bij dubbelzinnig woorden of namen worden soms zelfs gegevens met betrekking tot de verschillende betekenissen getoond. De zoekmachine WolframAlpha, die zichzelf een "computational knowledge engine" noemt, komt zelfs uitsluitend met feitelijke gegevens op basis van inhoudelijke interpretatie van de zoekvraag. Zo evolueren zoekmachines dus in feite tot antwoordmachines. Overigens is dat niet eens heel nieuw, want de in 1996 geïntroduceerde zoekmachine AskJeeves, probeerde ook al feitelijke antwoorden te geven op dit soort zoekvragen.

7.2 Analyse van de informatie

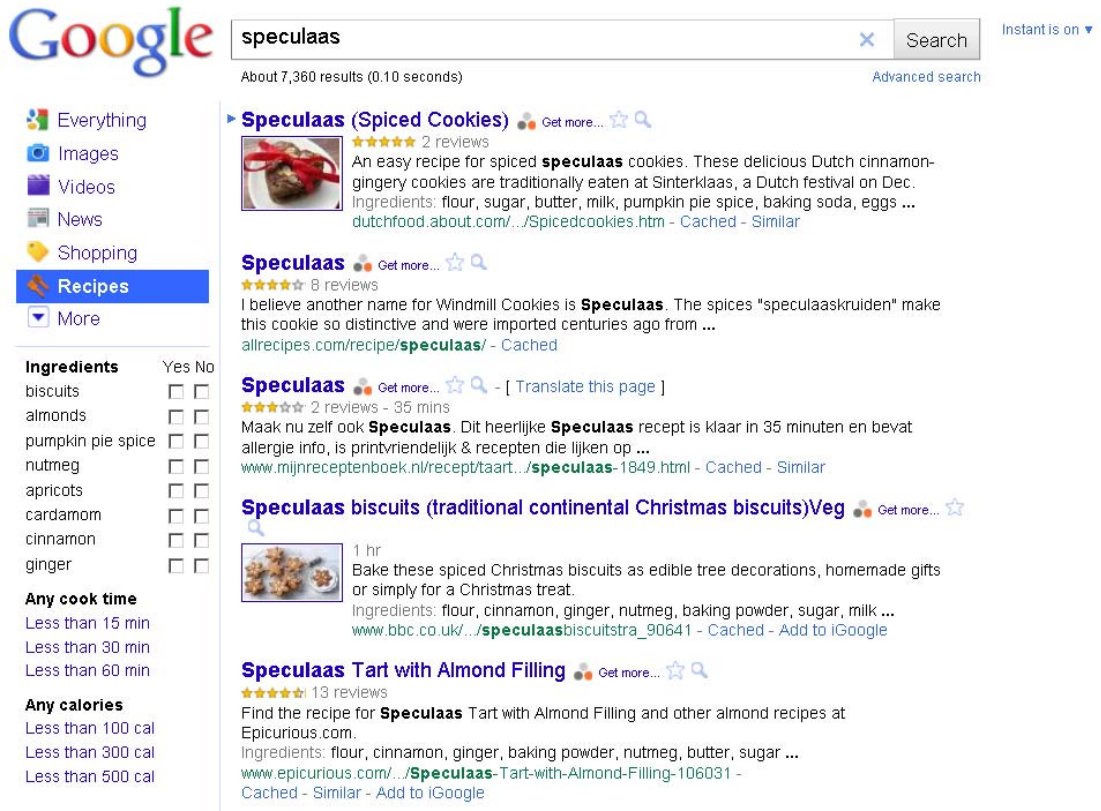
In tekstdocumenten kunnen concepten, woorden of namen herkend worden als behorend tot een bepaalde categorie. Daarbij kun je denken aan plaatsnamen, namen van personen, producten of bedrijven, gebeurtenissen en dergelijke. Dat herkennen kan vooraf gebeuren, bij het indexeren van de teksten, zodat gerichte zoek- en filter-mogelijkheden kunnen worden aangeboden. Het kan ook achteraf met de resultaten van een zoekopdracht, zodat de gebruiker een duidelijker beeld geboden kan worden van hetgeen gevonden is of een resultaat makkelijk uitgesplitst kan worden. De herkenning van dergelijke specifieke elementen in de tekst (ook wel entiteiten genoemd) kan automatisch plaats vinden of op basis van vooraf al in de tekst aangebrachte coderingen, een soort interne (of "embedded") metadata.



[Afbeelding 13: Voorbeeld van de analyse door Zemanta van een artikel uit de NewYork Times]

Voorbeelden van systemen die de inhoud van beschikbare teksten kunnen scannen op het daarin voorkomen van dergelijke entiteiten zijn OpenCalais van Thomson Reuters en de webdienst Zemanta. Deze diensten koppelen daar overigens zelf geen zoekfunctionaliteit aan. Wel kunnen ze voor verdere verwerking semantische codering (in XML) of hyperlinks naar gerelateerde informatie aan de geanalyseerde teksten toevoegen. Zemanta maakt hierbij vooral gebruik van aan de Wikipedia ontleende concepten (zie afbeelding 13). OpenCalais heeft eigen lijsten van te herkennen termen voor geografie, bedrijven, personen, feiten, gebeurtenissen en dergelijke. Beide systemen hebben een demonstratie web-viewer, waar je zelf stukken te analyseren tekst in kunt plakken: http://viewer.opencalais.com/ en http://www.zemanta.com/demo/.

Voor metadata die in de HTML-code van webpagina's verwerkt worden (embedded metadata), zijn op RDF (Resource Description Framework) gebaseerde standaarden ontwikkeld. Op basis van één van die technische standaarden, "microformats", heeft Google voor bepaalde soorten informatie verdere inhoudelijke standaarden vastgelegd, zodat bij zoekresultaten zogenaamde "rich snippets" getoond kunnen worden. Een leuk voorbeeld is een zo gerealiseerde receptenzoekmachine (alleen op google.com). De recepten van aanbieders die zich aan Google's standaard houden, kunnen zo heel gericht doorzocht worden op het al dan niet benodigd zijn van ingrediënten, op bereidingstijd en op aantallen calorieën (afbeelding 14).



[Afbeelding 14: Voorbeeld van de toepassing van interne metadata (microformats) volgens Google's "rich snippets" standaard voor recepten, waardoor gericht gezocht of gefilterd kan worden op ingrediënten en andere receptkarakteristieken (in linker kolom)]

Behalve voor recepten heeft Google ook "rich snippet" standaarden vastgelegd voor zinvolle kenmerken van recencies, personen, producten, organisaties, gebeurtenissen en muziek. Deze worden echter vooral in de presentatie van zoekresultaten gebruikt. Een meer algemene hiërarchie van standaard begrippen en eigenschappen voor gebruik op

het web, die door meer bedrijven wordt ondersteund, is Schema.org. In feite kan hierbij al van een soort ontologie worden gesproken (zie 7.4). Een algemene standaard om metadata in de (X)HTML-code van webpagina's te verwerken is RDFa (RDF in attributes). Daarbij worden in de praktijk allerlei onderwerps- of toepassingsspecifieke ontologieën gebruikt, zoals de GoodRelations ontologie voor E-Commerce of de Music Ontology voor sites met muziek-informatie. Dat aanbieders van informatie op het web deze standaarden ook werkelijk toepassen, is een gevolg van het feit dat men er het belang van inziet dat de aangeboden informatie op deze manier beter gevonden kan worden en in de praktijk vaak ook hoger in resultatenlijsten terecht komt.

Het eerder genoemde facetzoeken (paragraaf 4.6) is een hieraan verwante techniek die ook gebaseerd is op de aanwezigheid van gestandaardiseerde metadata, maar dan in een gestructureerde database-omgeving. Een andere techniek voor analyse achteraf van zoekresultaten kwam in paragraaf 4.6 ook al ter sprake, namelijk de statistische analyse van in een zoekresultaat voorkomende karakteristieke woorden, op basis waarvan dat resultaat geclusterd kan worden. Hierbij is er echter geen sprake van dat de computer iets weet over de betekenis van woorden en teksten, zodat hier in feite niet van een semantische zoektechniek gesproken kan worden.

7.3 Aanpassen van zoekvragen

De in de paragrafen 4.1 - 4.3 genoemde taaltechnieken om woordvarianten in het zoekproces te betrekken, worden ook wel als "semantisch" aangeduid. Hiervoor geldt echter ook dat betekenis van woorden hierbij vrijwel geen rol speelt, zodat het nauwelijks als een vorm van semantisch zoeken beschouwd kan worden. Dat is wel het geval bij het gebruik van synoniemenlijsten, thesauri, semantisch netwerken of ontologieën, omdat de betekenis van woorden en begrippen daar wel centraal staat. Daarop werd in paragraaf 4.4 en 4.5 al ingegaan. Omdat in het kader van het semantisch web het gebruik van ontologieën een belangrijke rol speelt, onder meer bij het herkennen van de betekenis van woorden en het identificeren van inhoudelijk gerelateerde begrippen, gaan we in de volgende paragraaf nog apart op ontologieën in.

7.4 Ontologieën en semantisch web

Niet alleen semantisch zoeken, maar ook het "semantisch web" staat volop in de belangstelling. In dat semantisch web zal veel informatie al bij voorbaat met betekenis gelabeld worden, zoals bij de hiervoor al genoemde technieken van microformats en RDFa. In dat kader en ook meer algemeen bij semantisch zoeken, speelt het begrip ontologie een centrale rol. In principe beogen ontologieën op een geformaliseerde en door computers interpreteerbare manier een gedetailleerde beschrijving te geven van (een stukje van) de werkelijkheid. Een ontologie moet daartoe in een formele computertaal beschreven kunnen worden. OWL, de Web Ontology Language is een voorbeeld van zo'n taal. Formeel wordt een ontologie wel gedefinieerd als "een strikt en uitputtend schema voor een bepaald onderwerpsdomein, meestal in een hiërarchische structuur, die alle relevante grootheden en hun relaties bevat, alsmede de regels waaraan die grootheden en relaties binnen dat domein voldoen" (Becker 2010). Voor de soorten relaties die tussen de begrippen in een ontologie kunnen worden gedefinieerd, bestaat een veel grotere vrijheid dan bij een thesaurus of een semantisch netwerk.

In toepassingen voor het semantisch web moet een computer echt betekenis van (tekst in) documenten en metadata kunnen afleiden en op basis daarvan kunnen redeneren en gevolgtrekkingen maken. Voor het semantisch web en bij semantisch zoeken worden ontologieën op dit moment nog vooral gebruikt om af te leiden welke begrippen

inhoudelijk iets met elkaar te maken hebben en welke woorden min of meer gelijke betekenis hebben, ook als die woorden in verschillende ontologieën voorkomen. Daartoe hoeven niet altijd "echte" ontologieën, volgens de eerdere definitie, beschikbaar te zijn. Ook woordenlijsten, thesauri, taxonomieën, semantische netwerken en concordanties worden in dit kader wel als ontologie beschouwd. Dergelijke systemen die termen inhoudelijk met elkaar in verband brengen, ook tussen verschillende systemen, maken interpretatie van de bedoeling van zoekvragen en uitbreiding of aanpassing van zoekvragen met andere zoekwoorden mogelijk. Dat geldt ook voor het verbreden of specifieker maken van zoekvragen. Ook kan zo aanvullende inhoudelijke informatie uit bepaalde bronnen worden opgevraagd, zoals uit Google's Knowledge Graph of uit een linked-data versie van de Wikipedia, de DBpedia.

7.5 Linked data en semantisch web

Een belangrijke bron voor gegevens ten behoeve van het semantisch web wordt intussen gevormd door zogenaamde "linked data". Dit zijn feitelijke gegevens die op technisch gestandaardiseerde wijze op internet beschikbaar worden gesteld, vrij voor iedereen te gebruiken, zodat ook wel van "linked open data" wordt gesproken. Deze gegevens zijn afgeleid uit allerlei bestaande systemen en databases. Een belangrijke bron hiervoor is de in de vorige paragraaf al genoemde DBpedia, waarvan de beginletters "DB" aangeven dat het om een soort "database"-versie van de Wikipedia gaat. Omdat op zijn minst een deel van de inhoud van elke beschrijving in de Wikipedia, de "infoboxen", gebruik maakt van een vaste structuur, kunnen gegevens daaruit voor een deel automatisch worden omgevormd tot het standaard format voor linked data zodat ze in de DBpedia kunnen worden opgenomen.

Op dit moment gebeurt dat nog vooral met gegevens uit de Engelstalige Wikipedia. In de praktijk blijkt in het Nederlandse deel van de Wikipedia wat minder zorgvuldig met de vaste structuren te zijn omgesprongen. Vanuit het Bibliotheek.nl project van de Nederlandse Openbare Bibliotheken, is daarom een actie gestart om deze gegevens beter te mappen met de vaste DBpedia-structuur (Kuys 2012). Daarmee moeten uiteindelijk de zoekmogelijkheden in het zoekplatform van Bibliotheek.nl verder worden verbeterd. Het met behulp van deze gegevens aan elkaar koppelen van diverse in het verleden al gebruikte ontsluitingssystemen is daar ook een essentieel onderdeel van.

Behalve de DBpedia zijn nog vele honderden andere collecties van gestructureerde gegevens als linked open data beschikbaar (Cyganiak 2011). Voorbeelden daarvan zijn gegevens uit de Internet Movie Database (Linked MDB), de wereldwijde authorisatielijst van auteursnamen VIAF, collectiegegevens van het Amsterdam Museum, et cetera. Specifiek vanuit de bibliotheekwereld worden ook diverse initiatieven ontwikkeld om zoveel mogelijk datasets beschikbaar te stellen (Library 2012). Ook diverse ontologieën, zoals besproken in de vorige paragraaf, worden in de vorm van linked open data beschikbaar gesteld. In de erfgoedsector is Europeana een systeem waarin al veel gebruik wordt gemaakt van de technieken van het semantsich web en linked open data om gegevens uit diverse collecties te kunnen combineren en gebruikers zo veelzijdiger antwoorden op hun zoekvragen te kunnen bieden.

Zoals uit de hier genoemde voorbeelden blijkt, is op dit moment vooral het web nog een proeftuin voor het ontwikkelen en uittesten van semantische zoektechnieken. Vermoedelijk zal het niet lang meer duren voordat ook in interne zoeksystemen voor bedrijven en organisaties (enterprise search) dergelijke technieken algemeen ingang vinden.

Literatuur:

Joyce van Aalten (2011) - Automatisch classificeren - In: Handboek Informatiewetenschap voor bibliotheek en archief. IV F 900. Alphen aan den Rijn: Vakmedianet

Peter Becker, Marjolein van der Linden, Henk Magrijn & Eric Sieverts (2010) - Organiseer je informatie; aan de slag met thesauri, taxonomieën, tags en topics. Leidschendam, Biblion Uitgeverij, ISBN 978-0-5483-954-5

M. Braschler & B. Ripplinger (2004) - How effective is stemming and decompounding for German text retrieval? - Information Retrieval 7, nr. 3, 291-316

Richard Cyganiak (2011) - The Linking Open Data cloud diagram (http://richard.cyganiak.de/2007/10/lod/, geraadpleegd 2/12/2012)

Jane Greenberg (2001a) - Automatic query expansion via lexical-semantic relationships - Journal of the American Society for Information Science and Technology 52, nr 5, 402-415

Jane Greenberg (2001b) - Optimal query expansion (QE) processing methods with semantically encoded structured thesauri terminology - Journal of the American Society for Information Science and Technology 52, nr 6, 487-498

Seth Grimes (2010) - Breakthrough Analysis: Two + Nine Types of Semantic Search - In: InformationWeek Software, 21 january 2010 (http://www.informationweek.com/software/business-intelligence/breakthrough-analysis-two-nine-types-of/222400100, geraadpleegd 4/10/2012)

Tony John (2012) - What is semantic search? - In: Techulator Resources, 14 march 2012 (http://www.techulator.com/resources/5933-What-Semantic-Search.aspx, geraadpleegd 4/10/2012)

Gerard Kuys (2012) - Zoekplatform Openbare bibliotheken: beter content vinden met DBpedia - Presentatie op studiedag FOBID Commissie Ontsluiting, 8 november 2012 (http://www.slideshare.net/gerardkuys1/20121108-fobid, geraadpleegd 2/12/2012)

Library linked data incubator group wiki (2012) (http://www.w3.org/2005/Incubator/lld/wiki/Main_Page, geraadpleegd 2/12/2012)

G. Salton, A. Wong, & C. S. Yang (1975) - A Vector Space Model for Automatic Indexing - Communications of the ACM 18, nr 11, 613–620 (http://www.cs.uiuc.edu/class/fa05/cs511/Spring05/other_papers/p613-salton.pdf, geraadpleegd 2/12/2012)

Eric Sieverts (2011) - De informatie vinden die je zoekt - In: Handboek Informatiewetenschap voor bibliotheek en archief. IV F 660. Alphen aan den Rijn: Vakmedianet (http://www.library.uu.nl/medew/it/eric/zoeken-en-vinden-2011.pdf, geraadpleegd 4/10/2012)

Barbara Starr (2012) - How Search & Social Engines Are Using Semantic Search - In: SearchEngineLand, 26 september 2012 (http://searchengineland.com/semantic-search-what-is-it-how-are-major-search-and-social-engines-use-it-part-1-133160, geraadpleegd 4/10/2012)