Preprint van bijdrage aan:
Wegwijzer bibliotheken en documentatiecentra (2020)
Politeia Uitgeverij (Brussel) ISBN 9057180278

Retrievaltechniek; technologieën voor het terugvinden van tekstuele informatie

Eric Sieverts

1. Inleiding information retrieval

Geautomatiseerde zoeksystemen bestaan al meer dan 60 jaar. Toch denken veel mensen bij "zoeken" nu vooral aan webzoekmachines en aan digitale informatie, dat wil zeggen aan informatie waarvan de volledige inhoud digitaal is. Dat laatste is nog niet heel lang zo - zeker geen 60 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. Wel kunnen voor dit soort documenten in toenemende mate automatisch inhoudelijke metadata worden toegekend. Deze 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 dus aangewezen op zoeken in titels en 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 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 filetechniek vaak aangevuld met andere technieken, waardoor het zoekproces zich niet meer beperkt tot puur Booleaanse methoden. De basisprincipes van twee daarvan, de vectormethode en de probabilistische techniek, zullen we - in wat vereenvoudigde vorm - ook behandelen (par. 2.2 en 2.3). Daarnaast wordt zogenaamd semantisch zoeken steeds belangrijker (par. 7). Daarbij spelen technieken uit de kunstmatige intelligentie (AI), zoals *Machine learning*, een belangrijke rol. Ook de introductie van knowledge graphs (par. 4.4) is daarop van invloed geweest (Reinanda, 2020).

Veel mensen denken dat met 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 met een enkele zoekopdracht voldoende relevante informatie te vinden. De vraag is dus gerechtvaardigd of de huidige zoekfunctionaliteit het mogelijk maakt om 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 tientallen 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). Ook in het pre-web-tijdperk kon men zich overigens al de vraag stellen of fulltext search de "panacee" was, die veel mensen erin zagen, of dat het niet meer dan een "placebo" was voor gecontroleerde ontsluiting. Een *eye-opener* in dit verband was een beroemd experiment van Blair & Maron

(Blair, 1985). Advocaten op een groot Amerikaans advocatenkantoor bleken in hun interne fulltext zoeksysteem nog geen 20% van de relevante informatie terug te vinden, terwijl ze zelf inschatten dat dat minstens 75% zou zijn. Het missen van meer dan 80% van de voor een zaak relevante documenten was in de Amerikaanse rechtspraktijk volstrekt onacceptabel.

Sommige van deze vindproblemen kunnen intussen voor een deel worden opgelost, door toepassing van allerlei taaltechnologische technieken. Op die technieken en wat ze betekenen voor recall en precisie van zoekacties, gaan we ook in (par. 4).

In het voorgaande ging het al regelmatig over tekst, woorden en zoekwoorden. Deze bijdrage zal zich dan ook inderdaad beperken tot tekstretrieval; het met woorden zoeken in en naar "talige" informatie. Waar beeld, geluid of objecten zijn beschreven met metadata, is uiteraard ook sprake van tekst. Moderne multimediale zoekmethoden die gebruik maken van het beeld of het geluid zelf, komen hier niet aan de orde.

2. Retrievaltechnieken

2.1 De basis: inverted filemethode

Om in grote hoeveelheden informatie te zoeken, is het niet handig een computer sequentieel alle tekst te laten doorlezen, om te vinden waar een zoekwoord allemaal voorkomt; hoe snel computers tegenwoordig ook zijn en hoeveel intern geheugen ze ook ter beschikking hebben. Dat geldt zowel redelijk gestructureerde (bibliografische) databases, als collecties fulltext 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 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. Ook in bestanden met vele miljoenen documenten werkt dat nog altijd.

Hoe kun je hiermee nu ook zoeken op combinaties van zoektermen, of meer algemeen met Booleaanse operatoren? Ook dat wordt door dat geïnverteerde bestand tamelijk eenvoudig. Elke zoekterm levert een rijtje verwijzingen op. In het werkgeheugen van de zoekcomputer kunnen hiermee vervolgens bewerkingen uitgevoerd worden. 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. Door slimme technieken kunnen dergelijke bewerkingen ook nog snel worden uitgevoerd als die rijtjes miljoenen verwijzingen bevatten.

3 5 6 12 13 17 21 24 32 35 39 45	AND	2 5 10 13 17 21 23 32 35 37	→	5 13 17 21 32 35

3 5 6 12 13 17 21 24 32 35 39 45	OR	2 5 10 13 17 21 23 32 35 37	→	2 3 5 6 10 12 13 17 21 23 24 32 35 37 39 45

3 5 6 12 13 17 21 24 32 35 39 45	NOT	2 5 10 13 17 21 23 32 35 37	→	3 6 12 24 39 45

[Afbeelding 1: Uitvoeren van AND, OR en NOT operatoren op basis van een inverted file]

Een specifiekere manier om zoektermen te combineren is het zogenaamde nabijheidszoeken. Daarbij kun je in de zoekvraag specificeren dat twee woorden dicht bij elkaar moeten voorkomen; dat ze bijvoorbeeld maximaal 5 woorden uit elkaar mogen staan. Vooral voor zoeken in fulltext documenten kan dat een nuttige relevantie-verhogende inperking zijn. Via een aanvulling van de inverted file kan dit gerealiseerd worden. Verwijzingen bij elk indexwoord bevatten daarbij niet alleen gegevens 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, weet het dat ze 23-8=15 woorden uit elkaar staan en ze dus niet voldoen aan de eis "binnen 5 woorden afstand". Als de positie-informatie laat zien dat ze 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 en wanneer een woord vijf keer in een document voorkomt, moeten bovendien alle vijf die posities geregistreerd worden. Soms bieden webzoekmachines - vaak ongedocumenteerd - ook mogelijkheden tot nabijheidszoeken. Daarnaast speelt de onderlinge afstand van zoekwoorden in de gevonden webpagina's ook 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 trunkatiezoeken vindt meestal plaats via de inverted file. Een rechtse trunkatie:

zoek comput*

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

<u>alfabetische index</u>
....
compunction
compunctious
computable
computation
computational
compute
computed
computer
computeraided
computerisation
computerized
computerization
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 tot en met 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 maar weinig samengestelde woorden voorkomen, maakt dat aan linkse trunkatie minder behoefte bestaat. De meeste zoeksystemen bieden die optie dan ook niet. Toch bestaat er wel een 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 die welke beginnen met *eipareht*, zodat ze daarin wel weer alfabetisch achter elkaar staan (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 geen plaats. Een - ook al klassieke - methode die wel relevantieordening mogelijk maakt en daarbij ook resultaten kan opleveren die

maar deels aan de zoekvraag voldoen, is het zogenaamde vectormodel. In feite is dat een variant op de inverted filemethode. 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 enen en nullen 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 postcoördinatie toegepast**
5: In een **classificatie** worden **aspecten** van een **document precoördinatief gekoppeld**
6: Als een **thesaurus** wordt **toegepast** zijn **aspecten** van de **inhoud** van een **document** niet **precoördinatief gekoppeld**
7: **Zoeksystemen** die voor **toegankelijkheid** een **thesaurus gebruiken** worden **vaak toegepast**
8: **Postcoördinatie** van **zoeksystemen** maakt dat je niet kunt **browsen**
9: De **inhoud** van **informatie** in een **document** is **gekoppeld** aan de **postcoördinatie** van een **thesaurus**
10: Hoe **beter** de **toegankelijkheid** van **informatie**, hoe **beter** we de **inhoud gebruiken**

Bij het indexeren in dit voorbeeld zijn stopwoorden zoals lidwoorden, voorzetsels en hulpwerkwoorden overgeslagen. De index van dit kleine zoeksysteempje bevat dan de zestien termen in de linkerkolom in onderstaande tabel (afbeelding 5). De kolommen daarnaast, genummerd 1 tot en met 10, representeren de tien documenten in het systeem. In elke kolom in deze tabel is te zien welke woorden in elk document voorkomen (in document 2 bijvoorbeeld de woorden 'browsen', 'classificatie' en 'gebruiken'). In elke rij zien we in welke documenten een bepaald woord voorkomt (bijvoorbeeld 'informatie' in de documenten 1, 9 en 10). Dit is in feite een andere vorm van 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
postcoördinatie	0	0	0	1	0	0	0	1	1	0	0
precoördinatief	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 nullen en enen 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 hogerdimensionale ruimtes. En dat kan ook als er geen 16 maar 16.000 indextermen zijn, dus ook in 16.000- (of nog veel hoger) dimensionale ruimtes.

In een zoeksysteem dat de vectormethode hanteert, worden zoekvragen ook als vectoren voorgesteld, met daarin een 1 voor elke term die we vragen (en nullen voor alle termen die we niet vragen). Zo'n vraagvector zal dus veel meer nullen dan enen bevatten. 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. In dit geval levert die vermenigvuldiging 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 ook nog wel enigszins relevant zijn als antwoord op de zoekvraag. Ze zullen echter op een lagere positie in de resultatenlijst terechtkomen. 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 genormaliseerd, dat wil zeggen dat ze onder meer tot hun woordstam worden gereduceerd, zodat varianten als enkel- en meervoud geen afzonderlijke indextermen opleveren die de vermenigvuldigingen zouden vertroebelen en tot onnodige extra dimensies zouden 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.
- Ook kan gewicht gekoppeld worden aan de zeldzaamheid van woorden, wat een maat is hoe kenmerkend ze zijn voor de inhoud van een document. De daarvoor vaak gebruikte TF*IDF-techniek komt in de volgende paragraaf aan de orde.
- Op soortgelijke wijze kunnen ook woorden in een zoekvraag een gewicht krijgen op basis van hun (vermoedelijke) belang voor die vraag.
- Voor het vermenigvuldigen worden de lengtes van de vectoren genormeerd, waardoor lange documenten (met heel veel woorden) r niet op basis van hun lengte al hoog 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 een maat voor de hoek tussen de vraagvector en de document-vectoren; hoe groter de uitkomst, hoe kleiner die hoek, dus hoe meer de twee vectoren in dezelfde richting wijzen en hoe groter de overeenkomst tussen vraag en document. De wegingsfactoren voor de woorden in de zoekvragen en in de gevonden documenten, zijn eigenlijk al een vorm van probabilistisch zoeken, waarop we hierna ingaan.

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. 4.8). Documenten waarvan de vectoren ruwweg in dezelfde richting wijzen, vormen dan een cluster.

2.3 Het probabilistische model

Bij probabilistische zoektechnieken wordt geprobeerd 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 berekend uitgaande van gegevens zoals:

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

De statistiek die daarbij wordt toegepast is de zogenaamde Bayesiaanse waarschijnlijkheidsrekening. Deze is 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 betrouwbaarder de waarschijnlijkheid van relevantie van een document voorspeld kan worden.

Hulpmiddelen bij het uitrekenen van die kansen hebben betrekking op het gewicht dat woorden in de zoekvraag toegekend krijgen, op het gewicht dat woorden in de documenten toegekend krijgen, op de relatie die er kan zijn tussen de positie van zoekwoorden in documenten en de kans op relevantie van die documenten, het gewicht (of belang) van individuele documenten, etc.

Zo kunnen termen in de zoekvraag al 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. Anders gezegd: zeldzame woorden zijn informatiever. Voor het uitrekenen van dat gewicht wordt meestal de volgende wiskundige formule gebruikt:

$$W = \log (N/DF) + 1$$

Daarin is W het gewicht dat aan een woord wordt toegekend, N het totaal aantal documenten in het systeem en DF de documentfrequentie, het aantal documenten waarin het betreffende woord voorkomt. Naarmate dat laatste aantal groter wordt, wordt het gewicht voor dat woord dus kleiner. Bedenk overigens wel dat zo'n formule voor de wegingsfactor geen natuurwet is. Met die logaritme wordt vooral gezorgd dat het berekende gewicht niet al te groot of al te klein kan worden. Voordat deze formule algemeen geaccepteerd werd, moest eerst nog 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	34 %
bifurcation	4786	29 %
asymptotic	20811	23 %
theory	189984	14 %

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

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.

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	63 %
asymptotic hopf	57 %
asymptotic bifurcation	52 %
hopf theory	48 %
bifurcation theory	43 %
asymptotic theory	37 %
hopf	34 %
bifurcation	29 %
asymptotic	23 %
theory	14 %

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

Deze initiële gewichten kunnen eventueel nog 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 in de zoekvraag zelf gewichten kan instellen, kom je niet meer tegen, omdat dat te veel 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. Zo kan een systeem 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. Bepaalde typografische kenmerken,

zoals hoofdletters, afwijkende kleur of afwijkend lettertype, kunnen hierbij eveneens in rekening worden gebracht. Ook de frequentie van voorkomen van een zoekwoord binnen een document speelt vaak 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)*, een uitbreiding van de eerder beschreven methode voor gewichtsbepaling van zoekwoorden. Daarbij wordt van elk woord uit een document geteld hoe vaak dat woord in het document voorkomt (de termfrequentie *TF*) en dat getal wordt vermenigvuldigd met de *inverse* documentfrequentie *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 documentfrequentie gedeeld moet worden (bijv. TF=50; DF=100.000; TF*IDF=0,0005). Een woord dat maar één keer in een document voorkomt kan juist wel belangrijk zijn, als er maar twee documenten zijn waarin het voorkomt (TF=1; IDF=0,5; TF*IDF=0,5). In de praktijk wordt voor berekening van het gewicht van een woord meestal een wat ingewikkelder formule gebruikt:

$$W = \log(1+TF) \times \log(N/DF)$$

N is daarbij het totaal aantal documenten. De logaritme dient vooral om de variatie in de uitkomsten van de berekening te "dempen", zodat die binnen een beperkt interval blijft.

Er worden ook 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 ..."
relevanter kunnen 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 de gebruiker eerder als relevant aangemerkte documenten te laten meespelen wordt nog een extra vorm van personalisatie geïntroduceerd. In paragraaf 6 komen we nog nader op relevantieordening terug.

Zoals eerder al aangegeven zullen al dit soort factoren gekwantificeerd en in wiskundige formules uitgedrukt moeten worden, opdat het zoeksysteem ermee kan 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 retrievalcompetitie TREC (Text REtrieval Conference), waarbij een gecontroleerde vergelijking van recall en precisie van verschillende systemen plaatsvindt. Veel van de daar uitgeteste methodes zitten intussen verwerkt in zowel commerciële enterprise-searchsystemen, als in de relevantieordening van webzoekmachines. Ook taaltechnologische verfijningen zoals die in 4 aan de orde komen, spelen overigens een belangrijke rol bij de geleidelijke verbetering van de resultaten van zoekmachines.

3. Problemen van free-text search (recall 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 recall (vangst) en precisie van zoekresultaten (Sieverts, 2011). Daarbij is recall 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 recall:

- in relevante documenten komen andere woordvormen (enkel-/meervoud, vervoegingen, verbuigingen) voor, dan het exacte woord uit de zoekvraag,
- 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,
- relevante documenten zijn in een andere taal dan de zoekwoorden,
- van relevante documenten is zo weinig tekst digitaal beschikbaar, dat lang niet alle daarin behandelde onderwerpen en invalshoeken in de digitale tekst gerepresenteerd en dus vindbaar 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 de zoeker met de zoekvraag bedoelde (foute of ontbrekende verbanden),
- in lange teksten komen veel woorden voor die weinig te maken hebben met het werkelijke onderwerp van het document, maar waarop het wel gevonden wordt.

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 - in principe ook op alle relevante termen,
- hiërarchische en semantische 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 recall 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 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. 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. Deze methode is sterk taalafhankelijk 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).

Probleem met deze methode is dat er altijd uitzonderingen kunnen voorkomen, waarbij de algemene stemmingregels 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 vaak toch al meer verschillende vormen van belangrijke woorden aanwezig zijn.

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 technieken kunnen 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). Zoekmachinesoftware met deze functionaliteit is bijvoorbeeld die van Exalead. Daarvan bestaat als demo ook een webzoekmachine (http://www.exalead.com/search/). Deze biedt zelfs twee soorten fuzzy: "phonetic search" (*soundslike*) en "approximate spelling search" (*spellslike*).

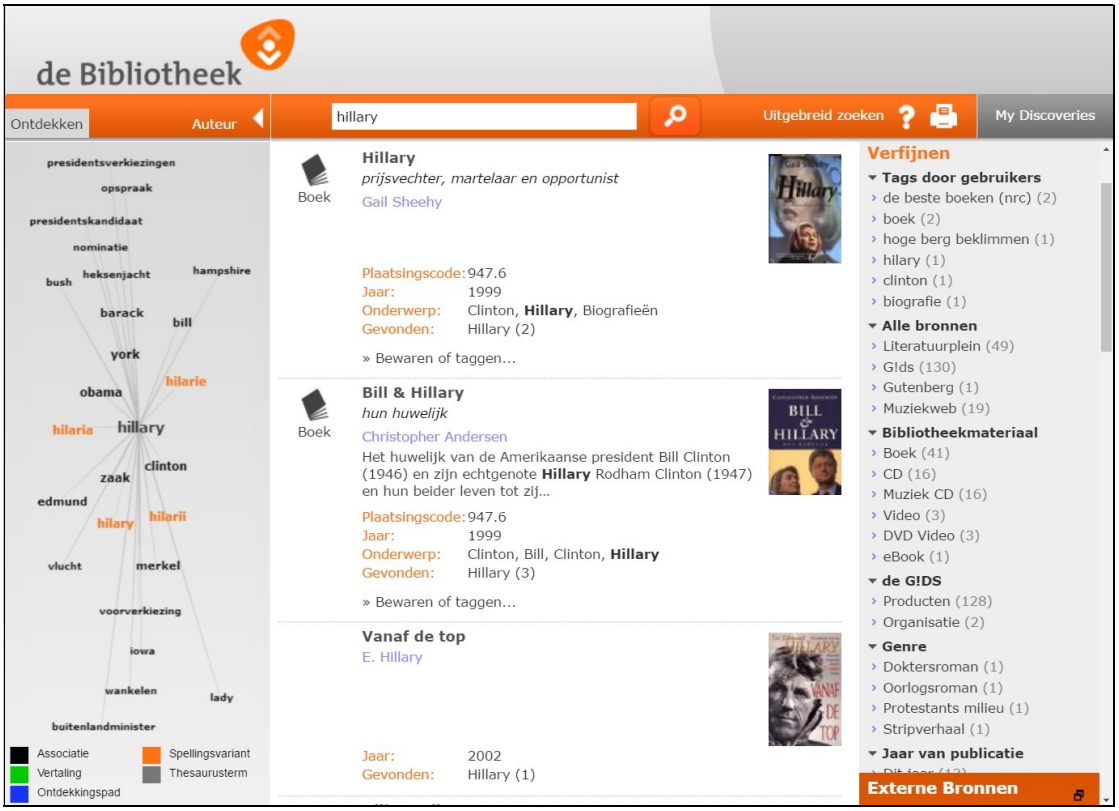
4.3 Splitsen van samengestelde woorden

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 afbreekroutines geven hier soms hilarische voorbeelden van. Denk bijvoorbeeld aan kerst-omaatjes of park-eerst-rook.

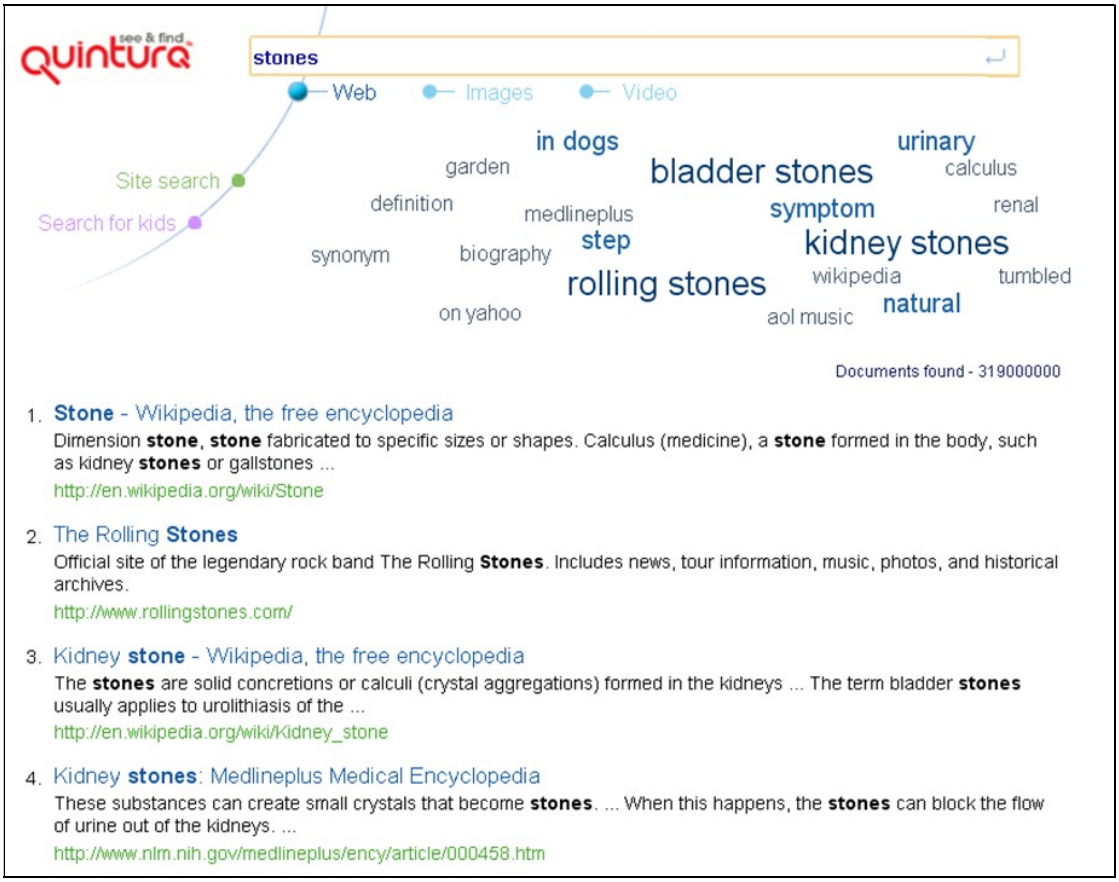
4.4 Genereren van termen voor vraagverfijning

Een statistische analyse van de woorden in een zoekresultaat kan 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 8). 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 dergelijke analyses maakt op basis van de zoekresultaten van een gekoppelde zoekmachine, is hier een voorbeeld van (zie afbeelding 9).



[Afbeelding 8: Zoekresultaat in een Aquabrowser. Links een woordenwolk met statistisch gegenereerde woorden (zwart), verwante woorden uit een thesaurus of een semantisch netwerk (grijs), alsmede fuzzy gelijkende woorden (oranje). Rechts is het zoekresultaat opgedeeld op basis van formele facetten.]



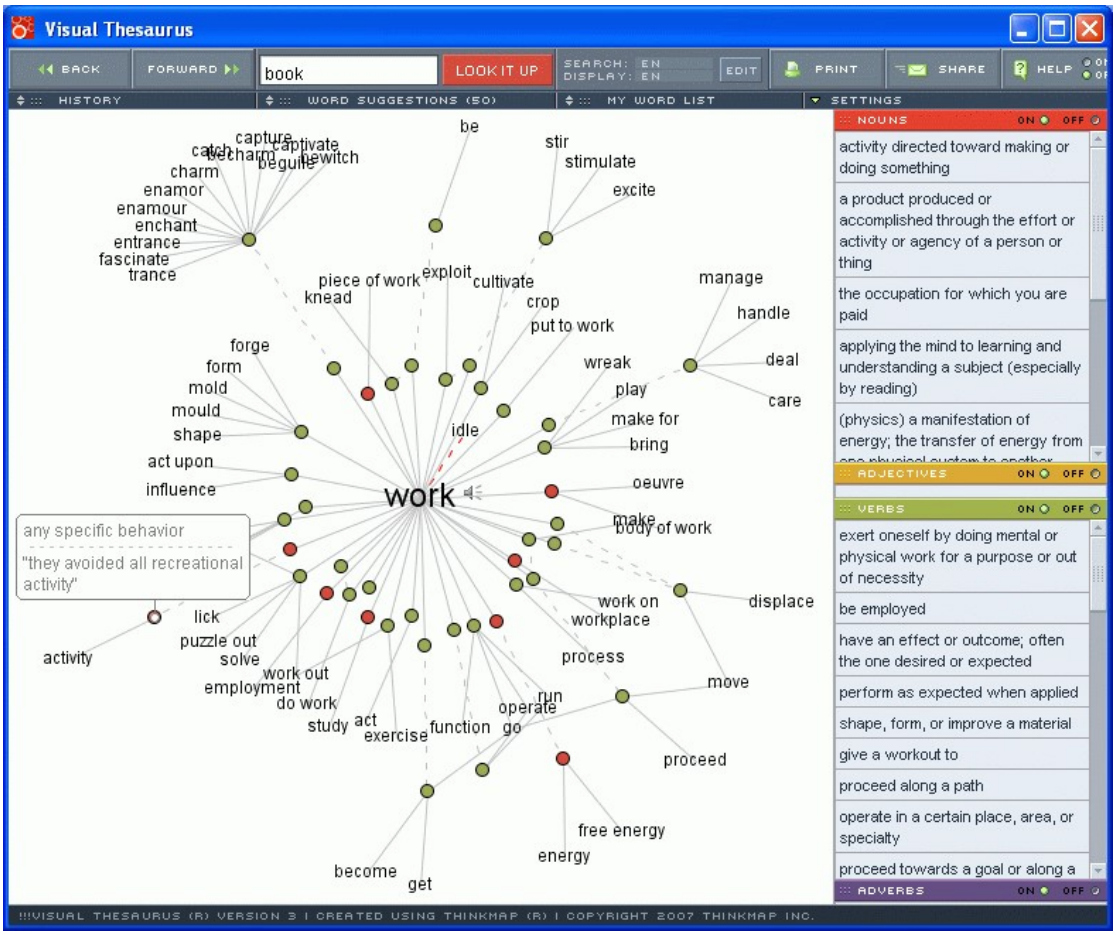
*[Afbeelding 9: Woordenwolk van statistisch uit een zoekresultaat gegenereerde woorden en begrippen in de Quintura webzoekmachine. De grootte van het font van de getoonde begrippen is gerelateerd aan hun TF*IDF-waarde. (Op dit moment niet meer in deze vorm beschikbaar)]*

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 opeenvolgende woorden, is tamelijk geavanceerde taalkundige analyse van de teksten nodig.

4.5 Semantisch netwerk en knowledge graph

De technieken in de vorige paragrafen waren vooral gebaseerd op tamelijk mechanische manipulatie met letters en onderdelen binnen woorden en op statistiek. De betekenis van woorden speelde daarbij nog vrijwel geen rol. Door kennis over de betekenis van woorden toe te voegen kan zoeken ook verder worden verbeterd. Dat kon altijd al door gebruik te maken van vakgerelateerde thesauri. Maar om daarvan gebruik te kunnen maken, moeten vooraf al termen aan documenten zijn toegekend. Er zijn echter ook systemen die pas tijdens het zoekproces een rol kunnen spelen. Dat is onder meer het geval bij gebruik van een semantisch netwerk of van een Knowledge Graph.

Een semantisch netwerk kun je beschouwen als een digitale versie van een woordenboek, met in principe alle woorden uit een taal, waarin niet alleen (verschillende) betekenissen van de woorden zijn vastgelegd, maar ook betekenisrelaties tussen woorden (Miller, 1995). Die laatste beperken zich meestal tot relaties van taalkundige of semantische aard, zoals die ook in woordenboeken voorkomen. 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 relaties met een heleboel tussenliggende woorden. Het aantal tussenstappen dat nodig is voor een willekeurig tweetal woorden, noemt men wel de semantische afstand van die twee woorden. Zo zal de semantische afstand tussen de woorden 'voetbal' en 'stadion' waarschijnlijk kleiner zijn dan die tussen 'voetbal' en 'viool'. Door een zoekvraag (automatisch) aan te vullen met verwante woorden binnen een gegeven afstand, kan de recall van een zoekactie vergroot worden (paragraaf 4.6).



[Afbeelding 10: 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 betekenissen van een woord onderscheiden. Door de in de omgeving van een woord voorkomende tekst in een document (of zelfs in een zoekvraag) te vergelijken met de omgeving van de verschillende betekenissen van dat woord in het semantisch netwerk, wordt het mogelijk te onderscheiden in welke betekenis een dubbelzinnig woord is gebruikt. In paragraaf 4.7 komen we daar nader op terug.
Afbeelding 10 toont een visualisatie van een stukje van zo'n semantisch netwerk voor de Engelse taal.

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 deel van uit. Als dat van belang is, zal men die woorden nog moeten (laten) toevoegen.

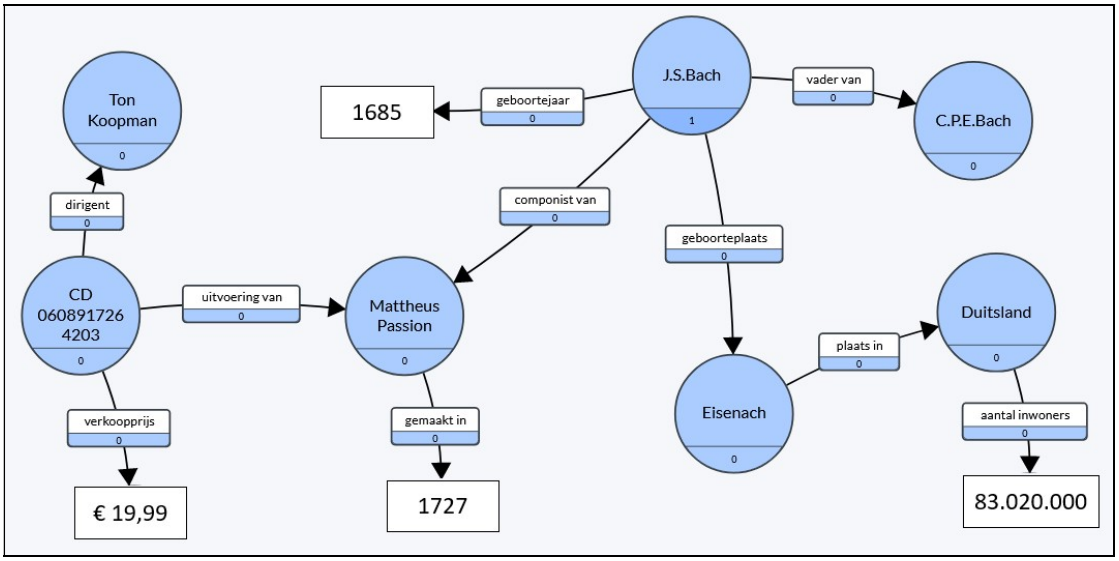
Knowledge Graphs zijn netwerken van knooppunten (nodes) die staan voor concepten of entiteiten en die onderling door relaties (edges) verbonden zijn (Van Aalten, 2017). Daarbij kunnen de relaties van veel gevarieerder en meer inhoudelijke aard zijn dan bij een semantisch netwerk. Wel worden die relaties gestandaardiseerd en vaak ook ontleend aan wereldwijd ondersteunde standaarden, zoals "Dublin Core" of de "friend-of-a-friend" ontologie. In een knowledge graph zullen zeker niet alle woorden uit het woordenboek worden opgenomen, maar zal men zich beperken tot concepten die voor een bepaald domein of een bepaalde toepassing van belang zijn. Voor gebruik in een knowledge graph worden de onderlinge relaties tussen concepten (of entiteiten) in de vorm van zogenaamde tripels vastgelegd, zoals:

J.S.Bach {vader van} C.P.E.Bach
Mattheus Passion {gecomponeerd door} J.S.Bach
J.S.Bach {geboorteplaats} Eisenach
Eisenach {plaats in} Duitsland
CD 0608917264203 {uitvoering van} Mattheus Passion
CD 0608917264203 {dirigent} Ton Koopman

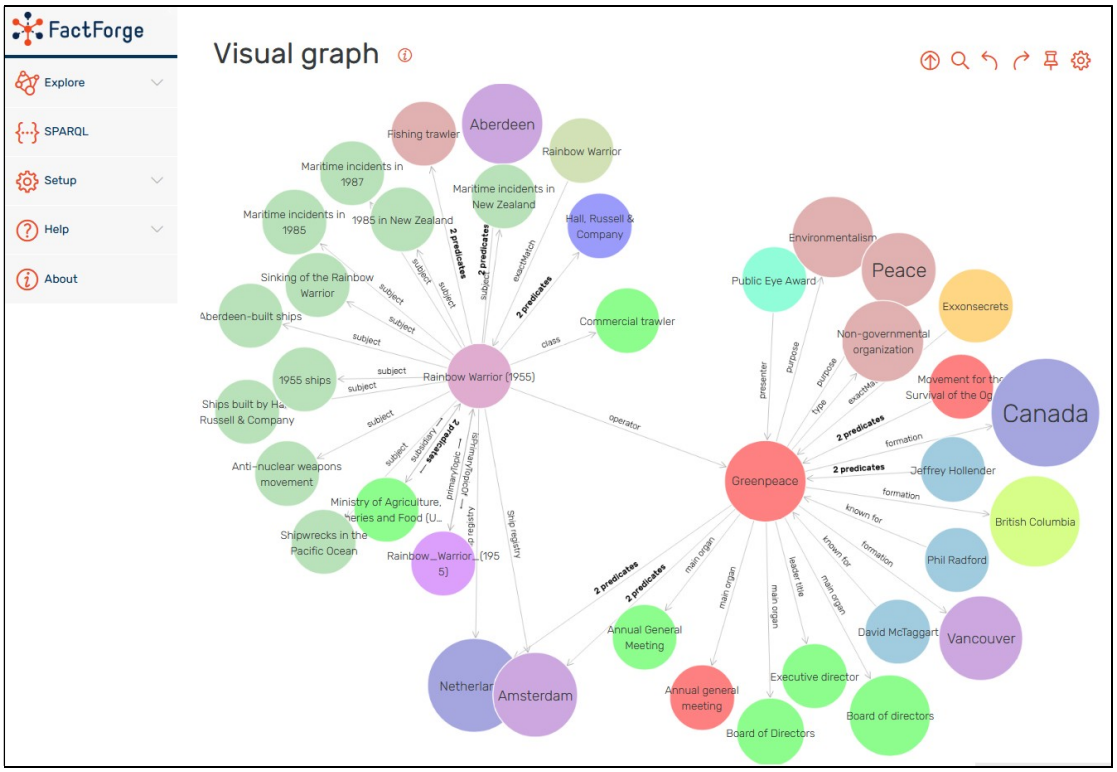
Daarnaast kunnen ook specifieke eigenschappen van die concepten worden vastgelegd, zoals:

Duitsland {aantal inwoners} 83.020.000
J.S.Bach {geboortejaar} 1685
Mattheus Passion {gemaakt in} 1727
CD 0608917264203 {verkoopprijs} € 19,99

Deze voorbeelden laten zien dat een knowledge graph, afhankelijk van zijn scope, een zich eindeloos vertakkend netwerk kan vormen. De visualisatie als graph-netwerk in afbeelding 11, op basis van het beperkte aantal voorbeeld-tripels hierboven, maakt al duidelijk dat grotere echte knowledge graphs nooit in hun geheel op deze wijze gevisualiseerd worden (dit in weerwil van het gebruik van het woord *graph*). Hoogstens kan dat gedaan worden met een geselecteerd stukje van een Knowledge graph, zoals afbeelding 12 laat zien. Dat voorbeeld is gehaald uit de zeer omvangrijke DBpedia die in Factforge (http://factforge.net/) als demo-systeem wordt aangeboden.
Overigens kunnen afzonderlijke knowledge graphs makkelijk gekoppeld worden, omdat alle knooppunten gekenmerkt worden door Uniform Resource Identifiers die in principe via het web herleid kunnen worden. Dat betekent dat in een muziek-graph zoals in ons voorbeeld, niet ook nog de hele geografie toegevoegd hoeft te worden.
De belangrijke rol die knowledge graphs spelen in recente ontwikkelingen in information retrieval komen uitgebreid aan de orde in een overzichtswerk van Reinanda. Meij en De Rijke (Reinanda, 2020)



[Afbeelding 11: Visualisatie van een stukje knowledge graph op basis van het voorbeeld uit de tekst]



[Afbeelding 12: Visualisatie van een stukje van de DBpedia knowledge graph, zoals die door Factforge wordt aangeboden (http://factforge.net/)]

Voorbeelden van zeer grootschalige systemen die als knowledge graphs beschouwd kunnen worden en in principe alle domeinen omvatten, zijn:

- De Google Knowledge Graph. Daarin meer dan 1 miljard concepten met meer dan 70 miljard onderlinge relaties en kenmerken. De gegevens zijn afkomstig uit de Wikipedia (DBpedia) en andere gestructureerde gegevensbronnen, alsmede in toenemende mate uit de inhoud van door Google's software geanalyseerde

webpagina's. Deze knowledge graph wordt alleen intern binnen het Google zoeksysteem gebruikt. Meer hierover in paragraaf 7 over semantisch zoeken.

- Wikidata (https://www.wikidata.org/). Daarin meer dan 80 miljoen concepten met meer dan 10 miljard onderlinge relaties en kenmerken. Ruim 20.000 mensen werken mee aan het "vullen" van Wikidata. Daarbij maken ze soms gebruik van scripts om in bepaalde bronnen beschikbare gegevens automatisch te laden. Wikidata doet steeds meer dienst als centrale bron waaraan andere systemen gestructureerde feitelijke gegevens ontlenen.
- DBpedia (https://wiki.dbpedia.org/). Daarin meer dan 6 miljoen concepten met meer dan 10 miljard onderlinge relaties en kenmerken. De inhoud is - grotendeels geautomatiseerd - ontleend aan de Wikipedia. Als zodanig zou je het een linked-dataversie van de Wikipedia kunnen noemen.

Een knowledge graph kan in zoeksystemen op allerlei manieren worden toegepast. Zo kun je een vollediger zoekresultaat krijgen, op basis van aan de knowledge graph ontleende extra zoektermen. Anderzijds kan de graph extra informatie bieden om de betekenis van dubbelzinnige woorden op basis van hun context te onderscheiden. Op die manier kan een knowledge graph zelfs gebruikt worden als hulpmiddel om automatisch trefwoorden of tags aan documenten toe te kennen (Revenko, 2019).
In de praktijk kom je veelvuldig toepassingen tegen waarin open systemen als Wikidata of DBpedia worden gebruikt om zoekfuncties te verbeteren. Daarnaast worden door steeds meer organisaties enterprise knowledge graphs voor het eigen domein en productaanbod ontwikkeld, om specifiek de zoekmogelijkheden voor eigen medewerkers en klanten te verbeteren. Enkele voorbeelden daarvan zijn Zalando (Kari, 2018), AirBNB (Wei. 2019) en Walmart (Deivasigamani, 2020).

4.6 Vraaguitbreiding

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. Aanvullen van een zoekvraag met specifiekere termen zal dus vrijwel altijd tot verbetering van de vangst leiden. 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 zulke 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 niet-thesaurusterm als 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 te verwachten viel, echter vooral ten koste van de precisie. Voor die termen kan de gebruiker dus beter zelf keuze worden gelaten welke daarvan hij toch aan de zoekvraag wil toevoegen. De gebruiker moet die dus als keuzelijstje gepresenteerd krijgen.

Ter verbetering van de vangst kan een zoekactie ook automatisch geëxpandeerd worden door verwante (of zelfs synonieme) begrippen uit een semantisch netwerk in de zoekvraag mee te nemen (Voorhees, 1994). 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. Ook aan een knowledge graph kunnen aanvullende zoektermen ontleend worden, maar het effect daarvan is wat onvoorspelbaar, als gevolg van de veelsoortigheid van de relaties die begrippen met de uitgangsterm kunnen hebben. Hieraan ontleende termen zullen dus vaker worden ingezet als suggesties om de gebruiker in een bepaalde richting te sturen.

4.7 Disambiguering

In paragraaf 4.5 gingen we in op semantische netwerken en knowledge graphs als hulpmiddelen waarmee zoeksystemen hun zoekresultaten kunnen verbeteren. Een van die mogelijke verbeteringen is het disambigueren van woorden die meerdere betekenissen kunnen hebben of in verschillende context kunnen voorkomen. En dat liefst zowel in zoekvragen als in gevonden zoekresultaten.
Bij het vastleggen van de betekenisrelaties in een semantisch netwerk worden de verschillende betekenissen van een woord onderscheiden. Door in een document de woorden die in de omgeving van een zoekwoord voorkomen, te vergelijken met de omgeving van de verschillende betekenissen van dat woord in het semantisch netwerk, kan 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.

Ook al worden in een knowledge graph verschillende betekenissen van woorden niet zo formeel onderscheiden - het gaat daarbij immers vooral om concepten, los van de daarvoor gebruikte woorden - kan vergelijking met de omgeving van de concepten ook hier wel degelijk uitsluitsel geven over de betekenis van woorden in een tekst (Revenko, 2019). Technieken van kunstmatige intelligentie en machine learning spelen daarbij vaak een belangrijke rol. In de paragraaf over semantisch zoeken (par. 7) komen we daar nog op terug.

Ook zonder dat zulke inhoudelijke kennis al a priori in een semantisch netwerk of een knowledge graph 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 4.8 nog afzonderlijk in. Een ander voorbeeld zijn statistisch gegenereerde extra zoektermen, waarmee vragen verfijnd en daarmee tot een specifieke betekenis ingeperkt kunnen worden. Dat kwam in paragraaf 4.4 al aan de orde. In de afbeeldingen die daar ter illustratie werden getoond, zien we overigens dat niet alle uit statistiek afgeleide onderscheiden inhoudelijk zinvol zijn.

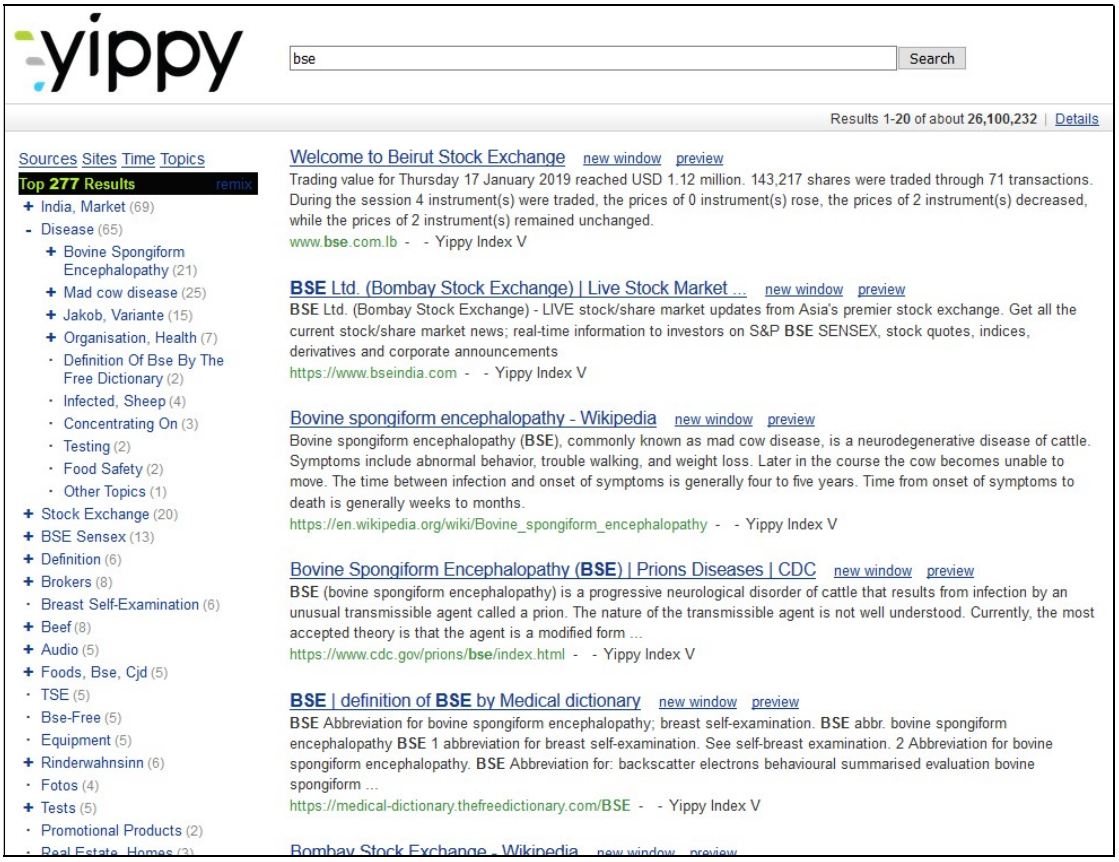
4.8 Clusteren van resultaten

Een manier om de precisie van een al verkregen zoekresultaat te verbeteren is automatische opdeling van dat resultaat in clusters. 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. Dit soort clusters kunnen gebaseerd zijn op tevoren gedefinieerde categorieën of tot stand komen op basis van statistische analyse van de tekst van de gevonden documenten.

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 met behulp van *machine learning* technieken uit voorbeelddocumenten zijn afgeleid, vergelijkbaar met hoe dat gebeurt bij automatisch categoriseren of metadateren van documenten. Bij het fulltext indexeren van 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.

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.



[Afbeelding 13: Een in clusters opgedeeld zoekresultaat van de metazoekmachine Yippy (https://www.yippy.com/)]

Een vergelijkbare methode is een statistische analyse van de tekst van gevonden documenten, waarbij het samen voorkomen van groepen karakteristieke woorden een maat is voor de gelijkenis tussen documenten. Dit laatste 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 (afbeelding 13). De benaming van clusters wordt automatisch gegenereerd op grond van karakteristieke woorden of zinsneden die daarin voorkomen. Dat leidt overigens niet altijd tot even zinnige benamingen.

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 bijvoorbeeld de aard van gevonden documenten (webpagina, pdf, worddocument, boek, krantenartikel), van de taal van de gevonden publicaties, van de namen van daarin frequent voorkomende auteurs, van de titels van de tijdschriften waaruit gevonden artikelen afkomstig zijn, van globale onderwerpscategorieën en dergelijke. Die uitsplitsing vermeldt vaak meteen al hoeveel na inperking overblijft van het oorspronkelijke resultaat. Het voorbeeld van de Aquabrowser in afbeelding 8 toont een dergelijke opdeling in diverse facetten. 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.

5. Terugkoppeling door gebruikers

Behalve de in paragraaf 4.4 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 (zie paragraaf 2.2). 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, bijvoorbeeld door Google dat daarvoor linkpatronen tussen webpagina's gebruikt. Hoewel je zou verwachten dat het resultaat van deze methoden altijd tot goede precisie zal leiden, blijkt dat in de praktijk toch nog vaak 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, door te registreren welke resultaten de gebruiker metterdaad opvraagt en hoe lang die bekeken worden. 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. Zeker voor systemen die ook voor recommandatie van nieuwe informatie bedoeld zijn, is dit natuurlijk onontbeerlijke functionaliteit.

6. Relevantieordening van resultaten

De relevantieordening 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 in de paragraaf over probabilistische zoektechnieken (2.3) en in sommige over zoekverbeteringen. Hier sommen we die factoren nog wat systematischer op:

1. Waar een gevraagde term in 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 meer gewicht dan algemene, omdat zeldzame begrippen meer onderscheidend zijn en belangrijker, want specifieker, delen van de zoekvraag representeren (par. 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.

Daarnaast gebruiken grote webzoekmachines als Google of Bing nog een veelheid aan andere factoren - soms wel honderden - die vaak heel specifiek zijn voor bepaalde toepassingen, zoals e-commerce, local search en dergelijke. Hierbij valt bijvoorbeeld te denken aan correct gebruik van *schema.org* markering binnen webpagina's, aan beschikbaarheid van goede mobiele versies van de pagina's of aan snelheid van laden van pagina's. Deze factoren evolueren regelmatig mee met nieuwe ontwikkelingen. Sommige ervan zijn ook 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. De snelle opkomst van technieken voor kunstmatige intelligentie (AI) en in het bijzonder *machine learning* (ML) hebben dat recent nog verder versterkt. Dat "semantisch" houdt in dat zoek- en analysetechieken worden toegepast waarbij de computer iets weet over de betekenis van de woorden in zowel zoekvragen als gevonden documenten. Wat simplistisch gesteld betreft dat technieken die niet meer primair gericht zijn op het vinden van documenten, maar op het geven van antwoorden, omdat de zoekvraag begrepen wordt, en omdat in het systeem een grote hoeveelheid kennis beschikbaar is - vaak wel afgeleid uit documenten - waarin dat antwoord gevonden kan worden. Vooral bij de grote webzoekmachines is deze ontwikkeling heel prominent. Daar horen uitspraken bij als:

- "Liever een antwoord dan 10 blauwe links": refererend aan de aanklikbare blauwe titels van de klassieke zoekresultaten bij de meeste webzoekmachines.
- "Not strings, but things": het gaat niet om de toevallige reeksen letters (strings) van woorden, maar om de betekenis ervan; om de "dingen" die ermee worden aangeduid.

In het voorafgaande zijn we al wat semantische aspecten tegengekomen, waar gebruik van thesauri, semantische netwerken en knowledge graphs aan de orde kwam. Echt semantische zoeksystemen horen echter nog een stapje verder te gaan met dat begrijpen. Daarbij valt bijvoorbeeld te denken aan het trekken van beredeneerde gevolgtrekkingen uit gevonden informatie. Of alle systemen die van het etiket semantisch worden voorzien, ook werkelijk dat extra stapje maken, 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; Berasategi, 2019; Kasenchak, 2019). Doelen en methoden die vaak genoemd worden in overzichten van hetgeen semantisch zoeken inhoudt, vallen globaal in de volgende drie categorieën:

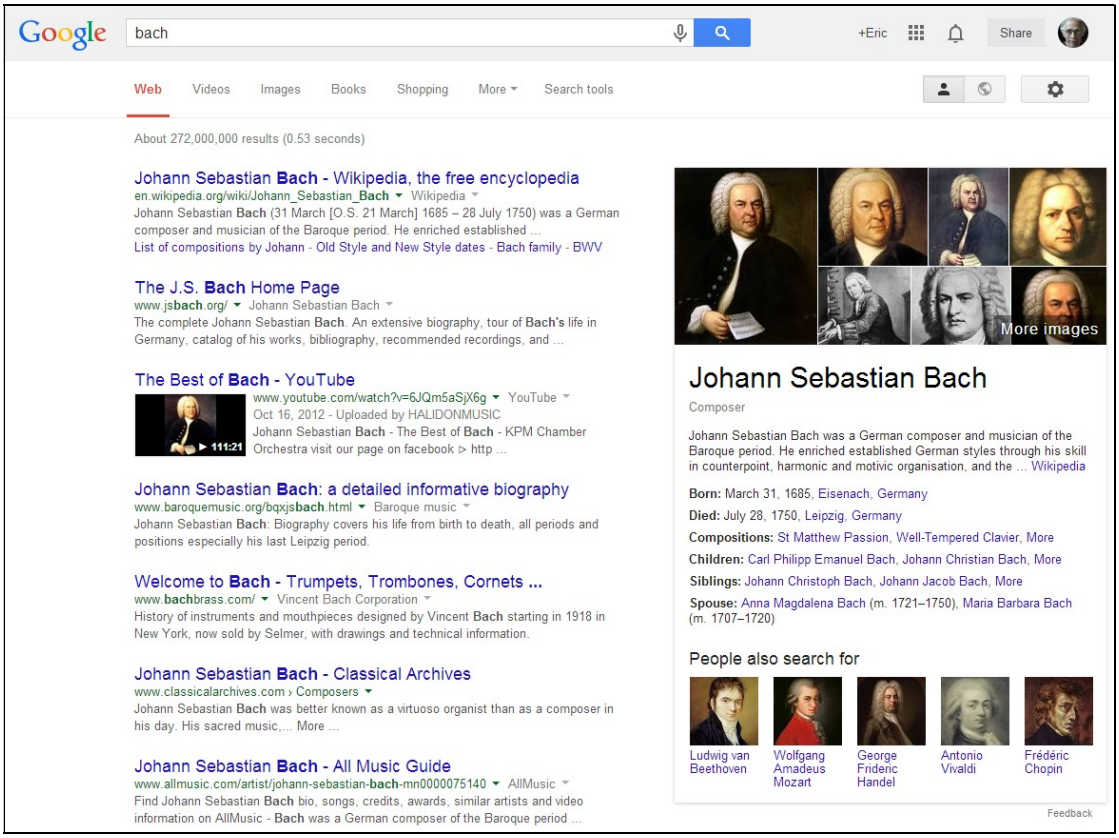
- inschatten van doel of context van zoekvragen,
- vooraf (bij indexeren) of achteraf (na een zoekactie) analyseren van tekst,
- automatisch aanpassen van zoekvragen.

Over elk van die categorieën hieronder iets meer details. Omdat vooral in het kader van het zogenaamde semantisch web het gebruik van ontologieën een centrale rol speelt, schenken we daar in een aparte paragraaf nog aandacht aan.

7.1 Doel en context van zoekvragen

Sommige zoekmachines - vooral op het web - proberen een inschatting te maken van het meest waarschijnlijke doel waartoe een zoekvraag wordt gesteld, de *query intent*.. Daarvoor 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 locatie-informatie kan zich beperken tot de landenversie van de zoekmachine die wordt gebruikt. Wie op *google.be* zoekt krijgt bij veel vragen andere antwoorden dan wie *google.nl* gebruikt of standaard naar *google.com* gaat. 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 zoekactie naar een recept), waarbij informatie over restaurants in de onmiddellijke nabijheid van de zoeker getoond kan worden. Eerder zoekgedrag van de gebruiker en eerder geraadpleegde websites kunnen ook een indicatie geven van de waarschijnlijke context waarbinnen een vraag gesteld wordt. Dit lijkt erg op de in de paragrafen 2.3 en 6 genoemde personalisatie van zoekresultaten.

Verder kan herkend worden of in een zoekvraag bepaalde entiteiten, personen, plaatsen, gebeurtenissen, voorkomen, op dezelfde manier als dat in documenten gebeurt (zie paragraaf 7.2). 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 om een jaartal te doen is, 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 referencecollectie. Google's Knowledge Graph is een voorbeeld van een systeem waarbij dit gebeurt (afbeelding 14).



[Afbeelding 14: Voorbeeld van automatisch getoonde feitelijke informatie uit Google's "Knowledge Graph" bij een daarvoor in aanmerking komende zoekvraag]

Zo zal ook bij een zoekvraag als "smeltpunt ijzer" meteen de betreffende temperatuur in graden Celsius (of Fahrenheit) getoond worden. Zelfs bij kortere vragen naar alleen een persoonsnaam, stad, land of bedrijf, zullen vaak meteen al feitelijke gegevens uit Google's Knowledge Graph getoond worden, overigens nog wel naast een lijst gewone zoekresultaten. Bij dubbelzinnige woorden of namen worden soms ook gegevens met betrekking tot de verschillende betekenissen getoond. Op deze manier zijn zoekmachines dus geleidelijk tot antwoordmachines geëvolueerd. 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.

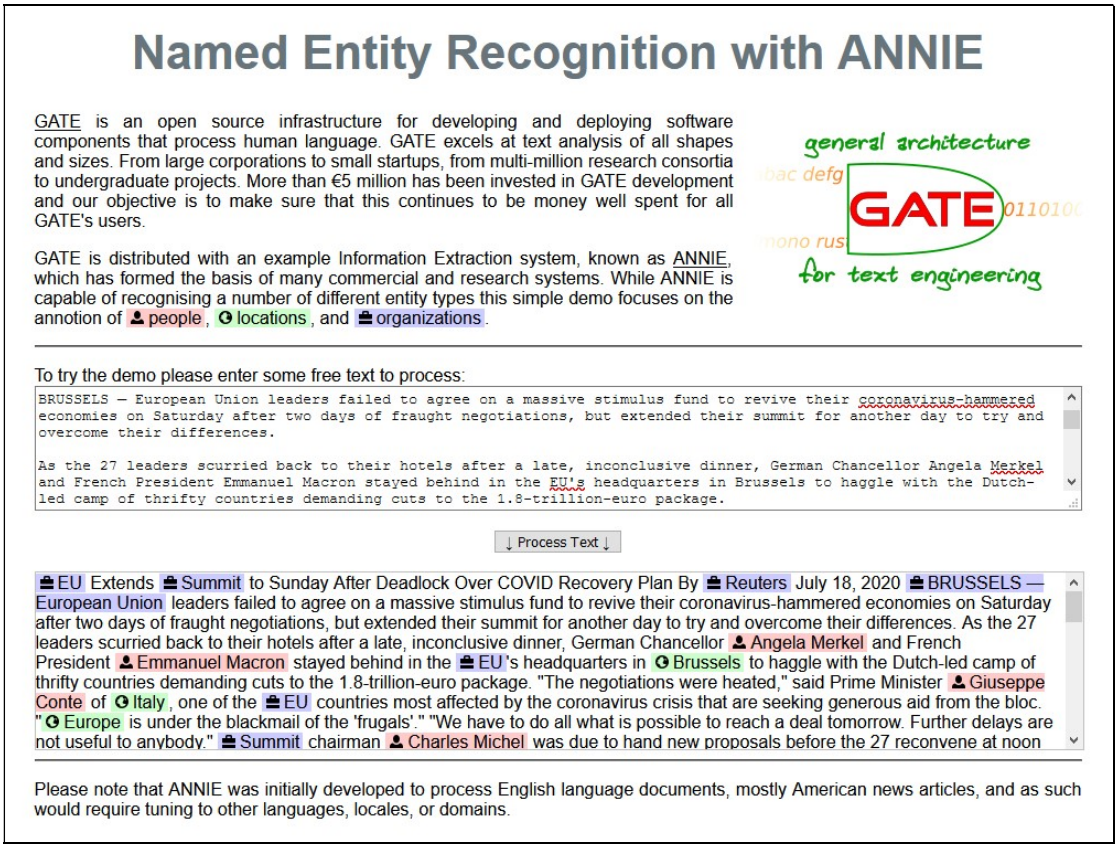
De natuurlijke taaltechnieken die worden gebruikt bij vraaginterpretatie zijn door toepassingen van Kunstmatige Intelligentie en *Machine Learning*-methoden sterk verbeterd. Eerst zorgde het door Google ontwikkelde *Rankbrain* algoritme al voor een belangrijke doorbraak, specifiek voor het analyseren van *query intent* (Enge, 2016; Harry,

2016). Meer recent is dat opnieuw gebeurd door de BERT-technologie die voor een ware revolutie gezorgd heeft (Anderson, 2019; Wiggers, 2019; Kurita, 2019). Dit acroniem staat voor de voor leken wat raadselachtige omschrijving *Bidirectional Encoder Representations from Transformers*. Simplistisch gezegd komt deze techniek erop neer dat, nadat het systeem getraind is op onwaarschijnlijk grote hoeveelheden digitaal beschikbare tekst, voorspellingen gedaan kunnen worden welke woorden het meest waarschijnlijk voor en achter een gegeven woord - bijvoorbeeld een zoekterm - kunnen voorkomen.

7.2 Analyse van de informatie

De inhoudelijke analyse van tekstdocumenten in semantische zoeksystemen kan allerlei doelen dienen. Zo kunnen bijvoorbeeld concepten, woorden of namen - zogenaamde entiteiten - 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 filtermogelijkheden 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 kan automatisch plaatsvinden of op basis van vooraf al in de tekst aangebrachte coderingen, een soort interne (of *embedded*) metadata.

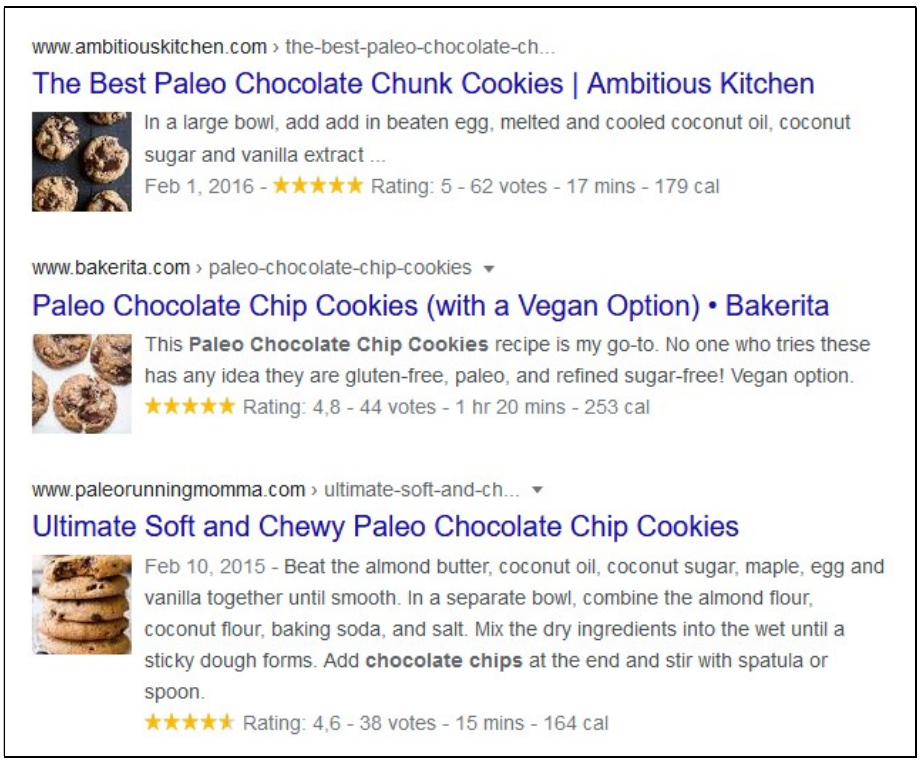
Voorbeelden van systemen die de inhoud van beschikbare teksten kunnen scannen op het daarin voorkomen van dergelijke entiteiten zijn onder meer OneCalais van Refinitiv (voorheen Thomson Reuters) en het wat beperktere Annie (afbeelding 15).



[Afbeelding 15: Voorbeeld van de analyse door ANNIE van een artikel uit de NewYork Times]

Dit soort diensten koppelt daar overigens zelf meestal geen zoekfunctionaliteit aan. Wel kunnen ze vaak semantische codering of hyperlinks naar gerelateerde informatie toevoegen aan de herkende entiteiten in de brontekst voor een webpagina. Pas na daarmee verrijkt te zijn, worden de pagina's dan definitief op het web of een intern informatiesysteem gepubliceerd. OneCalais heeft eigen lijsten van te herkennen termen voor geografie, bedrijven, personen, feiten, gebeurtenissen en dergelijke. Andere systemen maken wel gebruik van open beschikbare knowledge graphs zoals Wikidata. OneCalais en Annie hebben demonstratie webviewers, waar je stukken tekst kunt inplakken: https://permid.org/onecalaisViewer en http://services.gate.ac.uk/annie/. Systemen die automatisch trefwoorden of tags aan documenten toekennen bieden vaak ook functionaliteit voor het herkennen van entiteiten in die documenten. Dat is onder meer het geval bij PoolParty, SmartLogic, Aylien en Lexalytics.

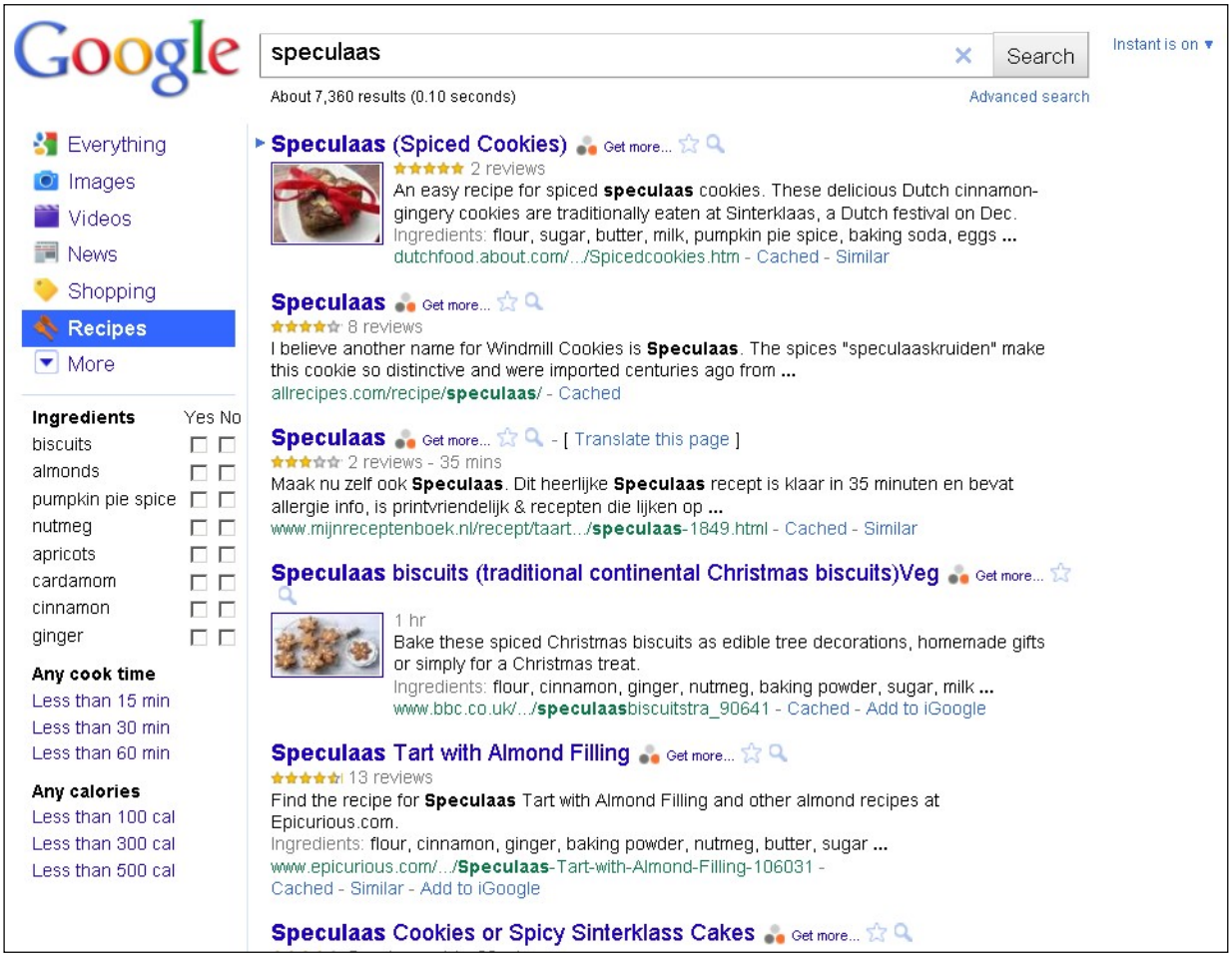
Voor het toevoegen van embedded metadata aan webpagina's zijn verschillende standaarden ontwikkeld. De meest gebruikte op dit moment zijn RDFa, microdata en JSON-LD. Als algemeen metadata-schema om hierbij te gebruiken. hebben de grote zoekmachines Schema.org ontwikkeld (https://schema.org). Dat is een hiërarchie van ruim 850 standaard begrippen en ruim 1350 eigenschappen om betekenis toe te kennen aan elementen in webpagina's. Daarmee kan bijvoorbeeld gecodeerd worden dat iets een verkoopprijs, een openingstijd of een ingrediënt voor een recept is. Deze gegevens worden wel met de benaming *structured data* aangeduid. In feite kan hierbij zelfs al van een soort ontologie worden gesproken (zie 7.4). Op basis hiervan kan een zoekmachine gevonden zoekresultaten verrijken met zogenaamde *rich snippets*. Daarbij worden op de zoekresultatenpagina (de "SERP") bepaalde met *schema.org*-codering gemarkeerde metadata uit de gevonden webpagina's getoond, zodat gebruikers makkelijker keuzes uit het zoekresultaat kunnen maken. Afbeelding 16 toont een voorbeeld hoe dat er bij recepten uit kan zien.



[Afbeelding 16: Voorbeeld van getoonde "rich snippets" in een zoekresultaat van Google]

Google heeft vastgelegd welke zinvolle kenmerken van onder meer recepten, recensies, personen, producten, organisaties, gebeurtenissen en muziek in zoekresultaten getoond worden. Ook kenmerken die niet getoond worden, gebruikt Google intern wel om af te leiden in hoeverre een webpagina een bedoeld antwoord op een door de gebruiker gestelde vraag vormt.

Een leuk voorbeeld van de mogelijkheden, dat Google helaas niet gecontinueerd heeft, was een receptenzoekmachine. De recepten van aanbieders die zich aan de standaarden hielden, konden zo heel gericht gefilterd worden op het al dan niet benodigd zijn van bepaalde ingrediënten, op bereidingstijd en op aantallen calorieën (afbeelding 17).



[Afbeelding 17: Voorbeeld van de toepassing van in webpagina's aanwezige embedded metadata, toen nog volgens de HRecipe standaard voor recepten. In deze toepassing kon daardoor gericht gezocht of gefilterd worden op ingrediënten en andere receptkarakteristieken (in linker kolom). (Bij Google helaas niet meer in deze vorm beschikbaar)]

Het eerder genoemde facetzoeken (paragraaf 4.8) 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 4.8 ook al ter sprake, namelijk de statistische analyse van in een zoekresultaat voorkomende karakteristieke woorden, op basis waarvan dat resultaat geclusterd kan worden. Hierbij weet de computer echter niets over de betekenis van woorden en teksten, zodat het wat ver gaat om dit ook al een semantische zoektechniek te noemen.

7.3 Aanpassen van zoekvragen

De in de paragrafen 4.1 - 4.3 genoemde technieken om ook woordvarianten in het zoekproces te betrekken, worden ook wel eens als semantisch aangeduid. Hierbij speelt de betekenis van woorden echter vrijwel geen rol, 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, knowledge graphs of ontologieën, omdat het daarbij wel om de betekenis van woorden en begrippen gaat. Daarop gingen we in paragraaf 4.6 en 4.7 al in. Behalve om het toevoegen of vervangen van termen, zoals daar besproken, worden soms ook automatisch termen uit een vraag weggelaten als die onnodig te beperkend lijken te werken.

7.4 Ontologieën en semantisch web

Niet alleen semantisch zoeken, maar ook het "semantisch web" staat volop in de belangstelling. In dat semantisch web (Bernstein 2016) zal veel informatie al bij voorbaat met betekenis gelabeld worden, zoals met de hiervoor al beschreven metadata-standaarden. 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" (Van Aalten, 2017). 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 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 ze in verschillende ontologieën voorkomen. Daartoe hoeven niet altijd "echte" ontologieën, volgens de eerdere definitie, beschikbaar te zijn. In het kader van het semantisch web, wordt het begrip ontologie namelijk heel ruim opgevat. Ook woordenlijsten, thesauri, taxonomieën, semantische netwerken, knowledge graphs, metadatamodellen en dergelijke worden vaak als ontologie bestempeld. Daarvan komen knowledge graphs misschien nog het dichtst in de buurt van wat een echte ontologie is. Dergelijke systemen kunnen op verschillende wijze termen inhoudelijk met elkaar in verband brengen, ook tussen verschillende systemen. Dat maakt interpretatie van de bedoeling van zoekvragen, uitbreiding of aanpassing van zoekvragen met andere zoekwoorden, alsook interoperabiliteit tussen afzonderlijke zoeksystemen 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, uit de DBpedia, of uit Wikidata.

Vooral het web is een interessante proeftuin voor het ontwikkelen en uittesten van semantische zoektechnieken. Maar ook in interne zoeksystemen voor bedrijven en organisaties (enterprise search) vinden dergelijke technieken al steeds algemener ingang.

Literatuur:

Joyce van Aalten, Peter Becker, Marjolein van der Linden & Eric Sieverts (2017) - Maak het vindbaar op schijven, sites en Sharepoint, Open Textbook, 329 blz., ISBN 9789492388001 https://hbo-kennisbank.nl/details/sharekit_hh:oai:surfsharekit.nl:ecd91d27-abc2-4c66-a114-ed922c10a834 (geraadpleegd op 24/07/2020)

Dawn Anderson (2019). A deep dive into BERT: How BERT launched a rocket into natural language understanding. *SearchEngineLand,* 5 November 2019 https://searchengineland.com/a-deep-dive-into-bert-how-bert-launched-a-rocket-into-natural-language-understanding-324522 (geraadpleegd op 25/07/2020)

Ana Berasategi (2019). Semantic search; A brief post on semantics, search, and semantic search. *Towards DataScience,* 1 April 2019 https://towardsdatascience.com/semantic-search-73fa1177548f (geraadpleegd op 25/07/2020)

A. Bernstein, J. Hendler,& N. Noy (2016). A New Look at the Semantic Web. *Communications of the ACM*, vol 59, nr 9, p. 35-37 (Sept 2016) https://doi.org/10.1145/2890489 (geraadpleegd op 25/07/2020)

D.C.Blair & M.E. Maron (1985). An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM*, 28(3), 289-299 https://pdfs.semanticscholar.org/91f7/579e660c139cb971f37f87c2dc0d3c86b9c6.pdf (geraadpleegd op 24/07/2020)

M. Braschler & B. Ripplinger (2004) - How effective is stemming and decompounding for German text retrieval? - *Information Retrieval* 7, nr. 3, 291-316 https://core.ac.uk/reader/154369374 (geraadpleegd op 24/07/2020)

Karthik Deivasigamani (2020). Retail Graph - Walmart's Product Knowledge Graph. *Medium*, 13 March 2020 https://medium.com/walmartglobaltech/retail-graph-walmarts-product-knowledge-graph-6ef7357963bc (geraadpleegd op 03/08/2020)

Eric Enge (2016). RankBrain: A Study to Measure Its Impact. StoneTemple - *The Digital Marketing Excellence Blog*, 9 March 2016 https://www.stonetemple.com/rankbrain-a-study-to-measure-its-impact/ (geraadpleegd op 24/07/2020)

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 DOI: 10.1002/1532-2890(2001)9999:9999<::AID-ASI1089>3.0.CO;2-K

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 DOI: 10.1002/asi.1093

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 op 24/07/2020)

David Harry (2016). Getting Your Head Around Google's RankBrain. *The Sem Post*, March 28, 2016 http://www.thesempost.com/getting-head-around-googles-rankbrain/ (geraadpleegd op 24/07/2020)

Tony John (2012) - What is semantic search? - In: *Techulator Resources*, 14 march 2012 http://www.techulator.com/resources/5933-What-Semantic-Search.aspx (geraadpleegd op 24/07/2020)

Katariina Kari (2018). The Art of Ontology - Introducing Semantic Web Technologies at Zalando. In: *Zalando Engineering Blog*, 20 March 2018. https://engineering.zalando.com/posts/2018/03/semantic-web-technologies.html (geraadpleegd op 03/08/2020)

Robert T. Kasenchak (2019). What Is Semantic Search? And Why Is It Important? *Information Services & Use,* vol. 39, no. 3, pp. 205-213 https://content.iospress.com/articles/information-services-and-use/isu190045 (geraadpleegd op 25/07/2020)

Keita Kurita (2019). Paper Dissected: "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" Explained. *Machine Learning Explained*, 7 January 2019 https://mlexplained.com/2019/01/07/paper-dissected-bert-pre-training-of-deep-bidirectional-transformers-for-language-understanding-explained/ (geraadpleegd op 25/07/2020)

G.A. Miller (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39-41. https://www.csee.umbc.edu/courses/graduate/691/spring13/01/papers/p39-miller.pdf (geraadpleegd op 25/07/2020)

Ridho Reinanda, Edgar Meij and Maarten de Rijke (2020), Knowledge Graphs: An Information Retrieval Perspective. *Foundations and Trends® in Information Retrieval*, Vol. 14: No. 4, pp 1-158. DOI:10.1561/1500000063. https://www.nowpublishers.com/article/Details/INR-063 (geraadpleegd op 27/07/2020)

Artem Revenko (2019). Label unstructured data using Enterprise Knowledge Graphs. *Semantic Tech Hotspot*, 16 December 2019. https://medium.com/semantic-tech-hotspot/label-unstructured-data-using-enterprise-knowledge-graphs-9d63f6f85ae1 (geraadpleegd op 27/07/2020)

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://sieverts.pbworks.com/f/zoeken-en-vinden-2011.pdf (geraadpleegd op 24/07/2020)

E.M. Voorhees (1994). Query expansion using lexical-semantic relations. In *SIGIR'94* (pp. 61-69). Springer, London. https://core.ac.uk/download/pdf/189788837.pdf (geraadpleegd op 25/07/2020).

Xiaoya Wei and Yizheng Liao (2019). Contextualizing Airbnb by Building Knowledge Graph. *Medium,* 29 January 2019. https://medium.com/airbnb-engineering/contextualizing-airbnb-by-building-knowledge-graph-b7077e268d5a (geraadpleegd op 27/07/2020).

Kyle Wiggers (2019). Google achieves state-of-the-art NLP performance with an enormous language model and data set. *The Machine*, 24 Oktober 2019 https://venturebeat.com/2019/10/24/google-achieves-state-of-the-art-nlp-performance-with-an-enormous-language-model-and-data-set/ (geraadpleegd op 25/07/2020)