Information retrieval software in bibliotheken

E.G. SIEVERTS

1 Inleiding

Het opzoeken van informatie wordt in het Engels in het algemeen aangeduid met de term 'information retrieval'. Hoewel dat niets zegt over de wijze van zoeken, impliceert deze term in de praktijk meestal het gebruik van computers. De 'information' uit dit begrip is nog sterker voor velerlei uitleg vatbaar. Gewoonlijk (maar niet altijd) wordt het terugzoeken van voornamelijk tekstuele informatie bedoeld. In dit hoofdstuk zal het begrip 'information retrieval' in deze beperkte betekenis gebruikt worden: 'het met de computer terugzoeken van tekstuele informatie'.

Als we het woord informatie letterlijk opvatten, zouden we overigens moeten concluderen dat de meeste 'information retrieval'-systemen helemaal geen 'informatie' opsporen. Het gaat op zijn hoogst om het opsporen van teksten of documenten. Pas na lezing en interpretatie zullen die door de gebruiker tot informatie worden getransformeerd. In het Engels wordt daarom ook wel correcter van 'text retrieval' of 'document retrieval' gesproken. Computerprogramma's hiervoor worden behalve 'information storage & retrieval software' ook wel 'text retrieval software' genoemd.

Hoewel terug te zoeken gegevens uiteraard eerst moeten worden opgeslagen (de *storage*), is het terugzoeken, de *retrieval*, toch de centrale functie waar het bij deze programma's om draait. Retrieval-software wordt al zo'n 30 jaar gebruikt op grote 'main-frame' computers. De belangrijkste toepassing was aanvankelijk het zoeken in grote bibliografische databases, als Chemical Abstracts, Psychological Abstracts, Medline of de Science Citation Index. Deze op hostcomputers geladen databases kunnen al sinds de jaren '70 online geraadpleegd worden. Intussen kunnen op deze wijze ook steeds meer full-text bestanden van krantenartikelen, jurisprudentie of wetenschappelijke artikelen doorzocht worden (Sieverts en De Jong, 1996).

Al ruim 15 jaar bestaan er ook PC-programma's waarmee bibliografische of full-

text databases voor lokaal gebruik kunnen worden opgebouwd en doorzoekbaar gemaakt. Dergelijke bestanden kunnen bedoeld zijn voor persoonlijk gebruik door een enkele onderzoeker, maar ook voor een hele werkgroep, een heel bedrijf of alle bezoekers van een documentatie-instelling of een museum. Deze bestanden zijn meestal aanzienlijk kleiner dan die welke op grote centrale systemen worden aangeboden.
Bovendien worden doorzoekbare bibliografische en full-text databases de laatste tien jaar ook op CD-ROM gedistribueerd, waarbij de zoek-software eveneens op een PC (met CD-ROM-station) geladen moet worden. Daarbij gaat het vaak wel om omvangrijke bestanden, vergelijkbaar met die welke online worden aangeboden.
Ondanks deze tamelijk lange historie, is het gebruik van retrievalsystemen pas de laatste jaren sterk toegenomen, niet in het minst door de snel groeiende belangstelling voor Internet en het World Wide Web, waarop allerlei op moderne retrieval-software berustende zoeksystemen worden aangeboden. De laatste jaren zijn ook voor specifieke toepassingen veel nieuwe functies en technieken in de software geïntroduceerd. Gebruik van retrieval-software hoort daardoor niet langer tot het exclusieve domein van informatiespecialisten en bibliothecarissen. Van de vele toepassingen die buiten de directe bibliotheeksfeer liggen, zullen er in dit hoofdstuk ook enkele worden aangestipt.

Behalve de term 'information retrieval' komen we, zoals we in de voorgaande alinea's al zagen, vaak het begrip 'database' tegen. In zijn algemeenheid worden verzamelingen van in een computer (in bestanden) opgeslagen gegevens meestal zo genoemd. Programma's voor het beheer daarvan heten daarom 'database management systemen' (DBMS). Die benaming wordt echter voornamelijk gebruikt voor programma's die specifiek bedoeld zijn voor beheer van strak gestructureerde administratieve en (vaak) numerieke gegevens.
Vroeger werd bijna altijd automatisch aan dit soort programma's gedacht, ook als het om minder strikt gestructureerde tekstuele gegevens ging. Intussen weten steeds meer computergebruikers die meer tekst dan 'data' verwerken, dat er voor hun tekstuele (of documentaire) informatie andere programma's bestaan, die specifiek zijn toegerust voor de verwerking en het terugvindbaar maken van die informatie, de retrieval-software. Omdat in de praktijk soms toch DBMS'en, of de *relationele* variant daarvan de RDBMS'en worden toegepast waar beter een echt retrievalprogramma gebruikt zou kunnen worden, besteden we in paragraaf 2 van dit hoofdstuk aandacht aan verschillen tussen beide soorten software. Verschillen tussen de soorten gegevens die moeten worden opgeslagen vormen een belangrijk uitgangspunt voor die discussie.

Anderzijds wordt in een bibliotheekomgeving vaak met tekstuele informatie gewerkt, waaraan belangrijke administratieve aspecten zitten. Denk aan de bibliotheek-catalogus die gekoppeld is met gestructureerde gegevens van leners, van bestellingen van nieuwe boeken en van afleveringsoverzichten en circulatieschema's van tijdschriften. Specifieke bibliotheeksoftware vormt daarom vaak een hybride tussenvorm tussen text-retrieval-software en (relationele) database management systemen. Ook dat aspect komt in paragraaf 2 aan de orde.

De laatste jaren zijn de verschillen tussen de retrievalprogramma's die op de eerder genoemde computerplatforms (main-frame, PC, PC + CD-ROM) gebruikt worden, sterk verminderd. De in de rest van dit hoofdstuk besproken functionaliteit en typologie van information-retrieval is dan ook veel minder aan de omvang dan aan de aard van de toepassing gebonden. Hoewel veel van de gegevens waarop deze discussie berust in eerste instantie aan PC-toepassingen zijn ontleend (Sieverts, 1996a) zijn ze dus meer algemeen van toepassing.
In paragraaf 3 van dit hoofdstuk wordt ingegaan op de basismethode waarmee grote hoeveelheden tekstuele informatie met behulp van retrievalsystemen doorzocht kunnen worden. In paragraaf 4 komen tekortkomingen van de veel gebruikte Booleaanse zoekmethode aan de orde en in paragraaf 5 zowel klassieke als moderne methoden om iets aan deze problemen te doen. In paragraaf 6, tenslotte, wordt een aantal in de praktijk vele gebruikte categorieën van retrieval-software nader gekarakteriseerd en besproken.

2 Verschillen tussen soorten informatiesystemen

2.1 *Text-retrieval en database management systemen*

Database management systemen zijn vooral ontwikkeld voor het beheer van strikt gestructureerde gegevens, zoals in een personeelsbestand, een systeem voor binnengekomen bestellingen of een relatiebeheersysteem. Eigenschappen van de daarvoor gebruikte software hangen direct samen met de aard van dit soort gegevens en met de aard van het gebruik ervan.
Een belangrijk kenmerk is dat dat meestal gestandaardiseerde en geformaliseerde gegevens zijn. In een personeelsbestand zal voor het veld waarin het geslacht van een personeelslid geregistreerd wordt, alleen een M of een V als inhoud zijn toegestaan; in het veld voor iemands geboortedatum mag altijd alleen een datum staan volgens een voorgeschreven vaste opmaak. Met zo'n datum moet tevens als numeriek gegeven gerekend kunnen worden, zodat bijvoorbeeld

eenvoudig een selectie gemaakt kan worden van alle medewerkers die op 1 september 1998 40 jaar of ouder zijn. Zowel voor de invoerder als voor de gebruiker van dergelijke systemen biedt die formalisering houvast. De invoerder hoeft zich niet af te vragen hoe hij de in te voeren gegevens het best kan karakteriseren; de zoeker/gebruiker weet exact waarop gegevens geselecteerd of teruggevonden moeten worden.

Een tweede kenmerk is dat het vrijwel altijd korte gegevens zijn. Namen, adressen, geboortedata, aantallen, beschrijvingen van bestelde artikelen en dergelijke zijn slechts bij hoge uitzondering langer dan een enkele regel. Een gevolg is dat vaak met een database-structuur met velden met vooringestelde vaste veldlengten wordt gewerkt.

Een ander belangrijk kenmerk is, dat er vaak relaties bestaan tussen verschillende bestanden. In een bestellingenbestand worden niet bij elke bestelling de complete klant- en artikelgegevens opgenomen. In plaats daarvan worden relaties gelegd met een klantenbestand waarin wel de volledige gegevens van een klant, eenmalig en gestandaardiseerd, zijn opgenomen, en met een artikelen- of voorraadbestand waarin de artikelen zijn beschreven. In relationele systemen worden bestanden, op grond van een formele analysemethode, zogenaamde 'normalisatie', vaak nog verder opgedeeld in onderling gerelateerde deelbestanden. In het jargon van RDBMS'en wordt daarbij veelal gesproken van tabellen, rijen en kolommen, waar we het bij retrievalsystemen over bestanden, records en velden hebben.

Uitgaande van de aard van de informatie en van het gebruik ervan, zijn in dit soort systemen meestal geen zeer uitgebreide zoekmogelijkheden nodig. Veelal hoeft men alleen een bepaald record te kunnen lokaliseren, teneinde daarin een wijziging aan te brengen of het te kunnen koppelen aan een record in een ander bestand. Daartoe kan bijvoorbeeld bij een personeelsbestand vaak worden volstaan met een sortering op achternamen. Elk personeelsrecord is dan door de computer voldoende snel te lokaliseren door die naam als zoeksleutel te gebruiken.

Complexere vragen houden meestal in dat een combinatie van gegevens wordt opgevraagd, waarbij voldaan moet zijn aan een combinatie van selectiecriteria. Bijvoorbeeld bepaalde gespecificeerde gegevens van alle manlijke personeelsleden die ouder zijn dan 40 jaar, die bij een bepaalde afdeling werkzaam zijn en die woonachtig zijn in Amsterdam. In een relationeel systeem kunnen zo opgevraagde gegevens ook afkomstig zijn uit verschillende gekoppelde bestanden. Voor dergelijke vragen zijn vaak voorgedefinieerde 'queries', 'views' of 'filters' in het systeem opgeslagen. Dat de computer met het maken van dergelijke selecties soms betrekkelijk lang bezig kan zijn, wordt meestal niet als een ernstig bezwaar gezien.

De meeste tekstuele informatiesystemen hebben heel andere karakteristieken. Dit betreft zowel de informatie zelf, de aard van het gebruik, als de daarvoor benodigde zoekmogelijkheden. Tekstuele informatie die moet worden opgeslagen kan onder andere bestaan uit titels van documenten, uit daaraan toegevoegde trefwoorden, uit samenvattingen van de inhoud van documenten en steeds vaker ook uit volledige teksten van documenten. Hoewel in deze systemen, zoals ze in een bibliotheekomgeving worden gebruikt, soms ook formelere gegevens voorkomen, zoals een publicatiedatum, auteursnamen of een ISBN, zijn de meeste gegevens niet geformaliseerd, veel minder voorspelbaar, veel langer en veel variabeler in lengte dan de gegevens die we bij de administratieve databasesystemen tegenkwamen.

Text-retrievalsystemen werken daarom vrijwel altijd met velden met variabele veldlengte. Een belangrijker verschil is echter dat dit soort systemen bij praktisch gebruik meestal meer een informerende functie hebben, dan dat ze een directe functionele rol spelen bij de uitvoering van dagelijks te verrichten handelingen, zoals het geval is bij veel DBMS'en. Dat houdt in dat meer op inhoudelijke dan op formele elementen wordt gezocht. Ook de zoekacties zullen daarom minder voorspelbaar en minder geformaliseerd zijn. Er wordt niet naar tevoren bekende gegevens gezocht ('ik weet dat Jansen bij ons werkt, wat is zijn geboortedatum?'), maar er wordt gevraagd of er wellicht informatie over een bepaald onderwerp te vinden is, waarbij het waarschijnlijk is dat we iets zullen vinden, maar nog allerminst zeker, laat staan dat we weten hoeveel we zullen vinden.

Niet alleen valt vaak veel minder exact te formuleren wat we met een retrievalsysteem precies willen vinden, maar meestal is ook niet te voorspellen waar de woorden waarop we een document willen vinden, precies staan. Dat betekent in de praktijk dat we niet alleen op enkele toegekende trefwoorden willen kunnen zoeken, maar ook op woorden uit het beschikbare deel van de tekst zelf, zoals een titel of een samenvatting. Bovendien zijn daarin de laatste woorden als zoekelement net zo belangrijk als de eerste.

Verder kan meestal niet worden volstaan met zoeken op een enkel karakteristiek zoekwoord. Voor het zoeken naar complexere onderwerpen moet ook op combinaties van zoektermen gezocht kunnen worden. In een titel moet bijvoorbeeld een combinatie van twee woorden voorkomen, die in willekeurige volgorde en op willekeurige afstand van elkaar mogen staan. In een ander geval willen we juist meteen zoeken op een aantal (zelf in te tikken) synoniemen van het gewenste zoekbegrip, waarbij de resultaten van die afzonderlijke termen moeten worden samengevoegd. Op dit essentiële kenmerk van retrievalsystemen gaan we in paragraaf 3 uitgebreider in.

De eigenschappen van text-retrievalsystemen zijn dus als volgt samen te vatten:

- we hebben te maken met variabele (en soms extreem lange) veld- en recordlengten;
- de zoeker heeft behoefte om gegevens in principe op elk woord, ongeacht de positie in een zin, veld of record, terug te kunnen vinden;
- er bestaat behoefte gegevens terug te vinden op allerlei combinaties van woorden, waarbij de te combineren woorden vaak in hetzelfde veld mogen of moeten voorkomen;
- omdat je tevoren nooit weet of je wellicht niets of juist te veel zult vinden, bestaat de behoefte om onbevredigende zoekresultaten eenvoudig, interactief bij te stellen.

2.2 *Text-retrieval of online catalogus*

Geautomatiseerde bibliotheekcatalogi zijn aanvankelijk ontstaan als een directe vertaling van de kaartenbak naar de computer. In eerste instantie werd dus vooral uitgegaan van de gebruikelijke wens een boek te kunnen terugzoeken op de naam van een auteur en op de (volledige) titel. Net als bij een kaartenbak met titelbeschrijvingen, is sortering van een database op auteursnamen en op titels daartoe al voldoende. Dat toont veel gelijkenis met de in de vorige paragraaf beschreven werking van database management systemen. Zoeken naar een boek op grond van het vierde woord van de titel was in die eerste bibliotheeksystemen (net als in DBMS'en) dus meestal niet mogelijk.
In kaartenbakken werd ook al lang de mogelijkheid geboden om op onderwerp te zoeken, met behulp van trefwoorden. Ook dat werkte in eerste instantie niet veel anders. De database hoefde alleen ook nog eens op trefwoordveld gesorteerd te worden. Voor samengestelde begrippen werden bovendien vaak precoördinatief samengestelde trefwoorden gebruikt (zoals 'schilderkunst – Frankrijk – 19e eeuw'), waarvoor ook met sortering kon worden volstaan. Dat betekent dat de meeste geautomatiseerde bibliotheekcatalogi aanvankelijk nog absoluut geen retrievalsystemen waren.
Ondanks het feit dat echte retrievalsystemen voor het online doorzoeken van grote bibliografische databases ook al bestonden, heeft de bibliotheeksoftware toch lang een daarvan onafhankelijke ontwikkeling doorgemaakt. Pas de laatste jaren mogen veel (maar nog zeker niet alle) online catalogi eindelijk met enig recht ook information retrievalsystemen genoemd worden, omdat ze de daarvoor karakteristieke zoekmogelijkheden gekregen hebben.
Anderzijds is voor bibliotheeksystemen een ingewikkelder architectuur nodig dan voor gewone bibliografische bestanden. Bibliotheeksystemen dienen name-

lijk ook voor de directe ondersteuning van een aantal administratieve uitvoerende handelingen, zoals registratie van boekbestellingen, administratie van uitgeleende en terugbezorgde boeken of het vastleggen welke afleveringen van een tijdschrift zijn binnengekomen. Dat vraagt juist weer om geformaliseerde beschrijvingen in een relationeel model, waarin een uitleenbestand is gekoppeld met de catalogus en met een lenersbestand. Moderne geïntegreerde bibliotheekpakketten zijn daarom vaak tamelijk complexe hybride systemen, met zowel kenmerken van een RDBMS, als van text-retrievalsystemen waar het zoeken in de catalogus betreft (Oude Groeniger, 1996).

3 Terugzoeken van informatie via indexen en Booleaanse combinaties

Om in een text-retrievalsysteem naar gewenste informatie te zoeken, zul je meestal proberen die op grond van een karakteristiek zoekwoord terug te vinden. De eenvoudigste manier om een zoekprogramma te laten kijken of een gevraagd woord in het systeem voorkomt, is de computer het hele bestand op schijf sequentieel te laten doorzoeken op het moment dat de vraag wordt gesteld. Die methode leidt voor grote bestanden al snel tot onaanvaardbaar lange wachttijden. Toch zijn er zoekprogramma's voor PC's die standaard deze methode hanteren. Meestal zijn dan wel maatregelen genomen om het sequentiële zoeken te versnellen, bijvoorbeeld door een groot deel van het bestand in het sneller toegankelijke interne geheugen van de computer te laden.

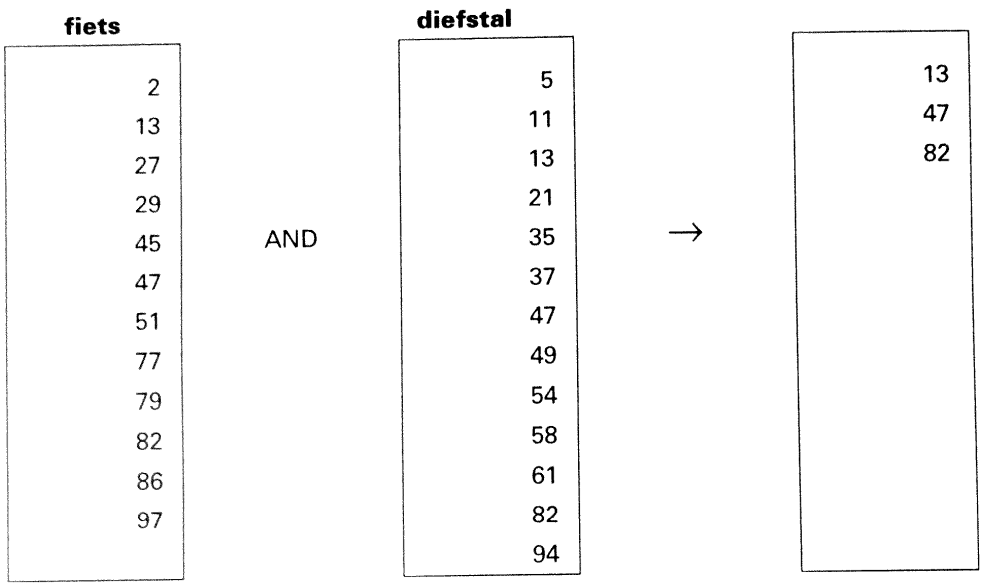
Gelijk op met het sneller worden van computers en met de toename van de hoeveelheid beschikbaar intern geheugen, groeit in de meeste gevallen ook de omvang van te doorzoeken gegevensbestanden, zodat de behaalde winst snel tenietgedaan wordt. Daarom is er behoefte aan een methode die onder alle omstandigheden een snel antwoord garandeert. Dat kan door via indexen te zoeken.

Bij deze al meer dan 30 jaar oude methode legt het computerprogramma een interne alfabetisch geordende woord-index aan, van alle woorden die in de opgeslagen tekstrecords voorkomen. Hooguit wordt een stopwoordenlijst gebruikt om echt betekenisloze woorden als lidwoorden, voorzetsels, persoonlijk voornaamwoorden en hulpwerkwoorden uit de index te houden. In de index wordt ook van elk woord bijgehouden in welke records het voorkomt. Wanneer een index zulke verwijzingen naar records bevat, wordt meestal van een *inverted file* gesproken.

Doordat elke ingang in de index meestal vermeldt hoe vaak die term voorkomt en via zogenaamde *pointers* naar de werkelijke vindplaatsen in het bestand ver-

wijst, is snel een antwoord te krijgen op elke willekeurige zoekvraag. Zelfs in heel grote bestanden kan op deze wijze bij elke ingetikte zoekterm vrijwel ogenblikkelijk worden gevonden of, hoe vaak en waar hij voorkomt en kan meteen elke tekst die die term bevat op het scherm worden opgeroepen. De grote online retrievalsystemen passen soortgelijke methoden toe op bestanden van vele miljoenen records.

In de dagelijkse praktijk bestaat bijna altijd de behoefte op complexere onderwerpen te kunnen zoeken dan alleen die welke je in een enkel woord kunt weergeven. Daartoe wordt in zoek-software al van oudsher gebruik gemaakt van Booleaanse operatoren.
Daarmee geef je door de combinatie 'fiets AND diefstal' aan dat je iets over het stelen van fietsen wilt weten (een inperking door te eisen dat beide woorden tegelijk aanwezig moeten zijn), met 'fiets OR rijwiel' dat je zowel teksten wilt hebben waar over rijwielen wordt gesproken, als die waar het over fietsen gaat (een verruiming, doordat maar één van beide termen aanwezig hoeft te zijn) en met 'fiets NOT diefstal' dat je alles over fietsen wilt weten, zolang het maar niet over het stelen ervan gaat (alle keren dat het eerste woord voorkomt, mits niet samen met het tweede). Bij systemen waarin ook volledige teksten worden opgeslagen, kan dit vaak nog worden verfijnd met de mogelijkheid te zoeken op twee woorden die samen moeten voorkomen en bovendien in een bepaalde volgorde en/of binnen een bepaalde afstand van elkaar moeten staan, met zogenaamde nabijheidsoperatoren.
Ook voor het uitvoeren van dergelijke zoekacties zijn indexen uitermate geschikt. Zoekprogramma's kunnen namelijk allerlei bewerkingen uitvoeren met de zoekresultaten. Dat zijn de in de index gevonden vindplaatsen, in de vorm van rijtjes recordnummers of pointers. Die worden als sets (letterlijk: verzamelingen – namelijk van de vindplaatsen) naar het werkgeheugen van de computer gekopieerd en daar tijdelijk bewaard.



Figuur 1. Voorbeeld van de werking van de AND-operator via de index

Stel dat bij zoekacties op de twee woorden *fiets* en *diefstal* via de index de in figuur 1 getoonde rijtjes recordnummers zijn opgehaald. Het toepassen van een AND-relatie betekent dan dat de computer die record-nummers uitfiltert die in allebei de rijtjes voorkomen. Het feit dat de records 13, 47 en 82 zowel in het rijtje *fiets* als in het rijtje *diefstal* voorkomen, impliceert immers dat die records beide termen bevatten. Een OR-relatie is dan het samennemen van alle in de twee rijtjes voorkomende recordnummers.

Wanneer je achtereenvolgens gemaakte resultaatsets tijdens een hele zoeksessie kunt bewaren, biedt dit de mogelijkheid om die eerder gemaakte sets ook achteraf weer met Booleaanse operatoren met elkaar of met nieuwe zoektermen te combineren. De zo geboden flexibiliteit om zoekresultaten interactief bij te stellen, is bij een sequentieel zoekend programma nauwelijks mogelijk.

De benodigde indexen en/of inverted files moeten wel tevoren door de computer zijn aangemaakt en moeten telkens na het invoeren van nieuwe gegevens weer worden bijgewerkt. Dat kost de computer tijd en bovendien schijfruimte. Vooral met dat laatste dien je bij het zelf opzetten van een retrievalsysteem rekening te houden, want sommige programma's zijn weinig efficiënt en leggen een index aan die nog meer ruimte inneemt dan de oorspronkelijke gegevens.

Bij retrieval van tekstuele informatie bestaat vaak de behoefte om op woordstam-

men of zogenaamde getrunkeerde woorden te zoeken. Truncatie met FIETS* levert dan ook records waarin de woorden FIETSEN, FIETSTE, FIETSER, FIETSPAD, FIETSPOMP, FIETSSLOT of FIETSENSTALLING voorkomen. Eigenlijk is zo'n zoekactie voor een sequentieel zoekend programma makkelijker uit te voeren dan voor een programma dat via indexen zoekt. Bij een sequentiële zoekactie zullen records namelijk standaard gescand worden op het voorkomen van de letterreeks F-I-E-T-S onafhankelijk van het feit of er nog letters voor of achter staan. Dat betekent dat zo'n programma zelfs eenvoudig rechts én links kan trunkeren. We vinden dus ook RACEFIETS, BAKFIETS, DOORFIETSEN, GEFIETST en UITGEFIETST.

Via een index zoekend is rechts trunkeren nog niet heel ingewikkeld, want alle termen die aan de truncatie voldoen, volgen in de alfabetische index direct na elkaar. De termen zijn dus snel gevonden, maar vervolgens moeten de vindplaatsen van elk van die termen nog bij elkaar genomen worden. Met andere woorden, het programma moet de via de index gevonden zoekresultaten van de afzonderlijke termen nog met OR combineren. Wanneer veel termen aan een truncatie voldoen, bijvoorbeeld bij een korte woordstam, vergt dat veel van de verwerkingscapaciteit van de computer.

Links of aan beide zijden trunkeren is voor een indexerend programma nog weer moeilijker, omdat de termen die aan zo'n truncatie voldoen over het hele alfabet verspreid kunnen liggen. Dat betekent dat dus eerst de hele index (van A tot Z) sequentieel moet worden afgezocht om alle woorden te vinden die de gevraagde woordstam bevatten, om vervolgens de via de index gevonden zoekresultaten van al die termen weer bij elkaar te nemen. Alleen links trunkeren kan wel nog versneld worden door een additionele zogenaamde retrograde index aan te leggen. Daarin zijn alle woorden achterstevoren opgenomen (en gealfabetiseerd), zodat daarin woorden die hetzelfde eindigen (en dus aan een linkse truncatie voldoen) direct onder elkaar staan. Verder werkt het dan weer als bij rechtse truncatie.

Gezien deze problemen wordt bij heel grote bestanden meestal geen mogelijkheid voor linkse truncatie geboden. Bij software die voor kleinere databases bedoeld is, wordt bij linkse truncatie de index meestal helemaal niet gebruikt, maar wordt het hele bestand zelf in een sequentiële zoekactie doorlopen.

Om nabijheidsoperatoren te kunnen gebruiken, moeten de indexen zelf nog worden uitgebreid. Om te weten of twee woorden in dezelfde zin voorkomen of niet meer dan vier woorden uit elkaar staan, is bij elke vindplaats van een woord in de index ook zogenaamde positie-informatie nodig. Van elk voorkomen van een woord wordt dus niet alleen opgeslagen in welk record dat is, maar ook waar

in dat record. Het hangt dan af van de mate van detail waarmee de positie van elk woord in de index wordt vastgelegd (nummer van de alinea binnen het veld, nummer van de zin binnen de alinea, nummer van het woord binnen de zin), hoe verfijnde nabijheidsoperatoren gebruikt kunnen worden. De positie-informatie maakt wel dat de omvang van de index wat groter wordt.

Samenvattend kan gezegd worden dat zoeken via een index de volgende voordelen biedt:

- snelle zoekacties, met zoektijden die bijna onafhankelijk zijn van de bestandsgrootte;
- meestal flexibele mogelijkheden om zoekresultaten bij te stellen en combinaties met eerdere resultaten te maken.

Nadelen zijn daarentegen:

- het bouwen en bijwerken van de indexen kost soms veel tijd;
- de indexen nemen extra ruimte in op de geheugenschijf.

Voor programma's die sequentieel zoeken ligt dit op bijna alle punten andersom. Het hangt dus van de omstandigheden af – de grootte van het bestand, de aard van het gebruik en dergelijke – welke van de voor- en nadelen in de praktijk het zwaarst wegen. Overigens werken bijna alle professionele text-retrievalprogramma's tegenwoordig met indexen en bieden ze allemaal enigerlei vorm van Booleaans zoeken, direct door het combineren van zoektermen, door het inperken of uitbreiden van het laatste zoekresultaat, of achteraf door het combineren van sets.

4 Problemen met het terugvinden van informatie in Booleaanse systemen

In de praktijk blijkt dat gebruikers van retrievalsystemen vaak moeite hebben met het juiste gebruik van AND-, OR- en NOT-operatoren. Door die operatoren in een schermtekst, anders en verduidelijkend te omschrijven, valt dat probleem wel enigszins te omzeilen. Toch vereist ook dan de Booleaanse methode nog altijd een sterk analytische houding tegenover de eigen informatiebehoefte. Om tot een goede vraagformulering te komen moet de zoeker die in duidelijk onderscheidbare concepten kunnen ontrafelen.

Dit hangt samen met het feit dat deze methode voortkomt uit een deterministische aanpak die vaak niet gerechtvaardigd wordt door de inhoud van het te doorzoeken bestand of de aard van de informatievraag. Bij het werken met zoeksets en Booleaanse combinaties wordt de inhoud van een bestand wel heel radicaal verdeeld in een groep records die exact aan de ingetikte vraagformulering

voldoen (maar daarmee nog niet noodzakelijkerwijs aan de daarin vertaalde informatiebehoefte) en een andere groep – de hele rest van het bestand – die daar niet exact aan voldoet en wordt uitgesloten (maar heel goed nog voor de vraag relevante informatie zou kunnen bevatten).

Eigenlijk komt deze methode voort uit de werkwijze met database management systemen. Een aan een klantenbestand voorgelegde vraag naar alle manlijke klanten uit Amsterdam die het afgelopen jaar meer dan *f* 250,- besteed hebben (woonplaats= *Amsterdam* AND geslacht= *M* AND bedrag > *250*), zal altijd exact het gewenste resultaat geven. Er zijn geen relevante resultaten die zo gemist worden; alles wat zo gevonden wordt voldoet ook werkelijk aan de zoekvraag. De formele opbouw van zowel de structuur als de inhoud van dat soort databases maakt een eenduidige tweedeling tussen wel en niet gevonden gegevens mogelijk en zinnig.

Bij de meeste tekstuele databases is noch de structuur van de database noch de inhoud van de records zodanig geformaliseerd, dat een dergelijke zoekwijze volledig gerechtvaardigd is. Vooral het feit dat 'taal' van nature zo weinig eenduidig is, speelt ons parten. Woorden kunnen voorkomen in verschillende vormen en schrijfwijzen (zogenaamde morfologische variatie), eenzelfde begrip kan door heel verschillende woorden worden beschreven (synonymie), eenzelfde woord kan heel verschillende betekenissen hebben (homonymie) en het samen voorkomen van woorden in een tekst of record impliceert nog helemaal niet dat ze de onderlinge inhoudelijke (syntactische) relatie hebben die bij het zoeken was verondersteld.

Door de combinatie 'fiets AND diefstal AND nederland' in te tikken, worden alle documenten gemist die over fietsendiefstal in Amsterdam gaan (tenzij in het document toevallig ook het woord Nederland voorkomt), waarin men het over 'rijwielen' (in plaats van fietsen) heeft of waar sprake is van het stelen of ontvreemden daarvan. Anderzijds kan deze zoekopdracht ook teksten opleveren over dieven die er na een inbraak op hun eigen fiets vandoor gaan.

Het zonder meer toepassen van de Booleaanse zoekmethode op tekstsystemen kan er dus makkelijk toe leiden dat een deel van de relevante informatie gemist wordt, terwijl een deel van de gevonden informatie niet relevant is. Deze problemen treden het sterkst op in bestanden waarin geen of maar een beperkte veldstructuur is aangebracht en waarin alleen 'vrije tekst' als zoekmiddel kan worden gebruikt, zoals bij veel full-text bestanden.

Een klassiek onderzoek dat Blair en Maron (1985, 1990) bij een Amerikaans advocatenkantoor uitvoerden, toonde dit voor het eerst aan. Gebruikers die ervan overtuigd waren dat ze in een intern gebruikt full-text retrievalsysteem altijd zeker 80% van de voor een zaak relevante teksten konden vinden, bleken

in de praktijk gemiddeld maar 20% te vinden. We moeten natuurlijk oppassen deze resultaten niet al te makkelijk van toepassing te verklaren op alle Booleaanse zoeksystemen. Voor wat meer gestructureerde en geformaliseerde tekstuele gegevens zoals in bibliografische bestanden, waaraan vaak gecontroleerde trefwoorden zijn toegekend, zullen de gesignaleerde problemen al veel minder spelen.

5 Methoden om zoekresultaten te verbeteren

Een ervaren zoeker zal zich de in de vorige paragraaf geschetste problemen meestal wel realiseren en zo nodig ingewikkelder zoekstrategieën ontwikkelen, met veel AND- en OR-relaties en gebruik van nabijheidsoperatoren. Van meer incidentele gebruikers die zich niet in de eigenaardigheden van te gebruiken informatiesystemen willen verdiepen, mag 'deskundig' zoekgedrag eigenlijk niet worden verwacht. Daarom zijn er technieken ontwikkeld die in informatiesystemen kunnen worden ingebouwd, waarmee getracht wordt aan de genoemde nadelen te ontkomen (Evans, 1994). Daarbij kunnen onder meer de volgende methoden worden onderscheiden:

- woordcontrole in trefwoordvelden;
- aanleggen van hyperlinks;
- best match zoeken met bijbehorende relevantie-ordening;
- natuurlijke-taalverwerking.

De laatste drie kunnen voor een deel als alternatieven of aanvullingen voor de Booleaanse methode worden beschouwd.

5.1 Woordcontrole

Woordcontrole is een heel gebruikelijke methode om documentaire informatie op meer geformaliseerde wijze terug te kunnen zoeken (Magrijn, 1997). Bij het invoeren van tekstdocumenten in de computer kunnen daaraan trefwoorden worden toegekend, waarvan de toelaatbaarheid gecontroleerd wordt op grond van een in de computer aanwezige validatie- of autorisatielijst van toegestane termen. Dat helpt de documenten over fietsen altijd met de term RIJWIEL te karakteriseren. Ook bij het zoeken kan zo'n lijst dan worden opgevraagd om de goede (gestandaardiseerde) zoekterm te kiezen.

Een stap verder gaan thesauri, waarin ook betekenisrelaties worden gelegd tussen de begrippen die in de lijst met toegestane termen zijn opgenomen. Voorbeelden hiervan zijn hiërarchische *ruimere term* ↔ *nauwere term* relaties, algemene

verwante term relaties of synoniemrelaties tussen *voorkeurs-* en *niet-voorkeurstermen.* Bij het terugzoeken van informatie in een retrievalsysteem kan dan automatisch gebruik gemaakt worden van deze tevoren vastgelegde relaties tussen de termen. Als op een synonieme niet-voorkeursterm wordt gezocht, kan het systeem de goede voorkeursterm suggereren of daar al automatisch op zoeken. Met de zoekterm FIETS komen we dan automatisch op RIJWIEL terecht. Bij het zoeken op een algemene thesaurusterm kan retrieval-software desgewenst onderliggende specifiekere termen automatisch in de zoekactie betrekken, waarbij ze in een OR-relatie bij elkaar genomen worden. Zoekend op bijvoorbeeld 'Europa', kan het systeem dan ook alle (geografisch) specifiekere termen, zoals de namen van alle Europese landen, in de zoekactie betrekken.

Een nadeel van deze woordcontrole-systemen is dat het opzetten en onderhouden ervan veel tijd vergt en dat bovendien elk ingevoerd record inhoudelijk door een menselijke indexeerder bekeken moet worden om het met de juiste termen te kunnen karakteriseren. Het kan dan goedkoper zijn om alleen aan de zoekkant enige hulp te bieden. Zo kan men het zoekprogramma gebruik laten maken van rijtjes min of meer synonieme begrippen die in een hulpbestand zijn opgenomen. Wanneer een gebruiker een woord uit zo'n rijtje als zoekterm intikt, bijvoorbeeld het woord *fiets,* zal de computer automatisch ook op de andere woorden uit het rijtje zoeken, dus bijvoorbeeld op *fietsen, fietsje, rijwiel, rijwielen, sportfiets* en *mountain bike.*

Toch moeten ook deze rijtjes weer tevoren door iemand worden bedacht en in de computer ingevoerd. Bovendien werken de synoniemenlijsten van veel programma's niet reciproque. Dat wil zeggen dat bij het intikken van *rijwiel* niet automatisch op *fiets* en alle andere zojuist genoemde termen wordt gezocht, omdat het lijstje alleen kan worden opgeroepen op grond van de daarin als eerste of hoofdwoord genoemde term. Dat betekent dat de beheerder van het systeem gedwongen is rijtjes synoniemen vele malen in telkens andere volgorde in de synoniemenlijst te herhalen.

Het tevoren opzetten van lijsten met synoniemen is niet nodig bij technieken die het probleem van een andere kant benaderen, door zelf varianten te genereren. De techniek van automatische woord-'*stemming*' houdt in dat van een zoekterm een standaard suffix wordt afgekapt en dat automatisch gezocht wordt op alle woorden die ontstaan door dit door alle mogelijke andere suffixen te vervangen. Dit gaat dus verder dan een gewone truncatie, want *computer* levert dan ook *computing, computers, computation* enzovoort. Uit dit voorbeeld blijkt al dat *'stemming'*-regels en ingebouwde mogelijke suffixen taal-afhankelijk zijn.

Andere hulpmiddelen zijn *fuzzy* zoeken en een variant daarvan, *sound-alike* zoeken. De *fuzzy* zoektechniek houdt rekening met spellings-, verbuigings-, vervoe-

gings-, tikfout- en andere varianten van zoektermen, door die automatisch als zoekterm te genereren. Bij *sound-alike* zoeken wordt gezocht op termen die *klinken als* de ingetikte zoekterm. Hiervoor wordt meestal een standaard *soundex* algoritme gebruikt, waarin (weer meestal voor het Engels) is vastgelegd welke groepen letters verwante klanken opleveren en dus onderling mogen worden uitgewisseld.
Deze technieken vormen overigens maar een beperkt alternatief voor echt gecontroleerd zoeken, want synoniemen (en verwante begrippen) moet de gebruiker nog altijd zelf bedenken. Vaak worden deze technieken ook verwerkt in modules die bij natuurlijke-taalverwerking worden gebruikt.

5.2 *Hypertext*

Een heel andere benadering van informatieproblemen bieden de bij het hypertext-principe geïntroduceerde *hyperlinks*. Hiermee kunnen tekstfragmenten of tekst-records die een inhoudelijke relatie met elkaar hebben, rechtstreeks aan elkaar gekoppeld worden (Franklin, 1989). Wie in zo'n systeem enige interessante informatie gevonden heeft, kan vandaar uit verder navigeren naar gerelateerde documenten, zonder dat eerst gerelateerde zoektermen aangeklikt hoeven te worden. Vooral de laatste paar jaar is deze techniek, mede door het succes van het World Wide Web op Internet, snel populair geworden.
Formeel is hypertext (of hypermedia) een methode om informatie, in tekst, beelden of geluid, op niet-lineaire wijze te organiseren. De lezer wordt daarin de mogelijkheid geboden hyperlinks via een druk op een toets direct te activeren en zo naar eigen believen een navigatiepad door het systeem te kiezen. Het kan daarbij zowel gaan om losse informatie-eenheden waartussen inhoudelijke relaties bestaan, als om samenhangende documenten waarbinnen diverse interne kruisverwijzingen zijn aangelegd (zoals in leerboeken veelvuldig voorkomen).
Programma's voor hypertext moeten gegevens van hyperlinks dus in de computer kunnen opslaan. De voor het activeren van de hyperlinks benodigde gegevens zijn gekoppeld aan aanhechtingspunten in de tekst of in een plaatje. In feite kunnen ook die worden beschouwd als een soort pointers naar de juiste informatie, maar nu niet vanuit een centrale index maar vanuit zorgvuldig gekozen punten in de informatie zelf.
Toch kunnen pure hypertext-systemen zeker geen volledig alternatief bieden voor klassieke retrievalsystemen. Men dient zich namelijk te realiseren dat via goed aangebrachte hyperlinks weliswaar bepaalde relevante gegevens op het scherm gebracht kunnen worden, maar dat dat niet noodzakelijkerwijs *alle* voor een onderwerp relevante gegevens hoeven te zijn. Hyperlinks verwijzen stan-

daard namelijk maar naar één ander punt in het systeem en om ze te kunnen volgen moeten ze bovendien eerst door menselijke tussenkomst bedacht en aangebracht worden.
Gezien het arbeidsintensieve karakter van het ontwerpen ervan, vindt hypertext in pure vorm vooral toepassing voor selectieve informatiebronnen, zoals handleidingen, toeristische informatiesystemen of educatieve media. Hypertext biedt echter wel een interessante aanvulling op klassieke zoeksystemen, zodat je naast gericht zoeken (retrieval) ook door de bouwers van het systeem aangebrachte links kunt volgen (browsen). Steeds meer retrievalprogramma's bieden daarom behalve de klassieke Booleaanse zoekmethoden ook mogelijkheden om hyperlinks aan te leggen.
Overigens heeft het basis-idee van hypertext – klik op een woord en je krijgt er meer informatie over – wel invloed gehad op interfaces van gewone zoeksystemen. Daarin kunnen steeds vaker woorden in al gevonden tekst met de muis worden aangeklikt, om met die woorden als gewone zoektermen verder te zoeken. Deze techniek levert een makkelijke en gebruiksvriendelijke manier om snel extra termen aan een zoekstrategie toe te voegen, maar met *echte* hypertext met hyperlinks naar voorbedachte punten elders in het systeem heeft het niets te maken.

5.3 Best-match en Relevance Ranking

Bij systemen die gebruik maken van de 'best-match' zoektechniek hoeft enerzijds de gebruiker niet na te denken over het verschil tussen AND en OR, terwijl anderzijds niet zo'n rigide opsplitsing wordt gemaakt tussen niet en wel gevonden documenten. Een zoekresultaat dat een aanzienlijk deel of soms zelfs de hele inhoud van het doorzochte bestand omvat, wordt daartoe in een volgorde geplaatst waarbij die documenten die vermoedelijk beter aan de zoekvraag voldoen eerder gepresenteerd worden. Voor de hiervoor noodzakelijke computerberekeningen worden ruwweg twee methoden toegepast: het zogenaamde *vectormodel* en de puur *probabilistische* methode. Omdat het vectormodel vaker wordt toegepast en eenvoudiger beschreven kan worden, beperken we ons hier tot deze techniek (Salton, 1989).
Bij 'best-match' zoeken wordt van de gebruiker meestal verwacht dat hij een rijtje zoektermen intikt, net zoveel als in relatie tot het betreffende onderwerp bedacht kunnen worden. Het systeem zoekt dan naar alle documenten die 'enige' overeenkomst vertonen met die reeks termen, zonder dat ze allemaal hoeven voor te komen. Dat gebeurt in feite door eerst een OR-combinatie tussen alle opgegeven termen uit te voeren. Binnen dat vaak omvangrijke zoekresultaat

berekent de computer een mate van gelijkenis, die als de mate van relevantie wordt opgevat en de presentatievolgorde bepaalt. Zo krijgt de zoeker de vermoedelijk belangrijkste documenten het eerst te zien en kan hij zelf bepalen tot hoeveel documenten of tot welk niveau van relevantie hij nog wil doorbladeren of het zoekresultaat wil afdrukken. In plaats van een ingetikte rij termen, kan vaak ook een al gevonden en als relevant aangemerkt document als geheel als zoekargument genomen worden, of kan een relevant stukje tekst uit een gevonden document via 'knippen en plakken' naar een zoekvenster gekopieerd worden.

Bij de berekening van de mate van gelijkenis en de op grond daarvan verondersteld relevantiegraad speelt meestal een aantal factoren mee. In eerste benadering wordt de gelijkenis bepaald door het aantal van de gevraagde termen dat in een gevonden document voorkomt. Een document met negen van de tien gevraagde termen zou dan een relevantiegraad van 90% krijgen en één dat er maar zeven bevat een relevantie van 70%.

Deze wel heel grove methode kan op allerlei manieren worden verfijnd, onder meer door aan de gevraagde en aan de in de documenten voorkomende termen gewichten toe te kennen. Voor gebruikers is het het eenvoudigst, als dat automatisch door de computer gebeurt. Daarbij kunnen de volgende factoren meespelen:

- een gebruikte zoekterm krijgt een hoger gewicht naarmate die term zeldzamer is in het bestand als geheel (een zeldzame term is sterker onderscheidend dan een term die in bijna elk record voorkomt);
- in een gevonden document krijgt een term een groter gewicht naarmate hij daarin vaker voorkomt, waarbij meestal wordt gecorrigeerd voor de lengte van het document (als een term vaak in een tekst voorkomt, is dat een aanwijzing dat hij karakteristiek is voor de inhoud ervan, mits hij althans niet in bijna elk record van de database zo vaak voorkomt);
- een term krijgt een groter gewicht als hij in één van de eerste regels van een tekst staat (belangrijke dingen worden het eerst gezegd) of in een tevoren als belangrijk aangemerkt gedeelte voorkomt (een onderdeel 'Conclusions' is belangrijk, ook al staat het pas aan het eind).

Toepassing van zulke weegfactoren kan maken dat een document waarin vier heel zeldzame termen voorkomen, hoger scoort dan een document dat vijf heel algemene termen uit de zoekvraag bevat.

In sommige systemen kan de gebruiker ook zelf aan zoektermen gewicht toekennen. Van de gebruiker vereist dat meer inzicht in de formele formulering van zijn informatiebehoefte. Een vereenvoudigde versie hiervan volstaat met de mogelijkheid om bij bepaalde essentiële termen aan te geven dat ze verplicht in

het zoekresultaat moeten voorkomen of om Booleaanse AND-combinaties in de zoekopdracht te verwerken.

Het berekenen van de getallen die de relevantiegraden van de documenten voorstellen, vindt meestal plaats door de computer rijtjes getallen met elkaar te laten vermenigvuldigen. Het ene rijtje representeert telkens een document, waarbij van elk woord uit de index met een getal staat aangegeven of het al dan niet in dat document voorkomt en wat zijn gewicht is. Het andere representeert de zoekopdracht, waarbij op dezelfde manier wordt aangegeven welke woorden daarin met welk gewicht voorkomen. Met deze rijtjes getallen worden dezelfde soort berekeningen uitgevoerd als met wat in de wiskunde *vectoren* genoemd worden. Daarom wordt de aan dit *best-match* mechaniek ten grondslag liggende theorie het *vectormodel* genoemd.

Om snel de in de vectoren voorkomende gewichten te kunnen berekenen, maakt de computer gebruik van wiskundige formules die, bijvoorbeeld als functie van de frequentie waarmee een woord in het bestand voorkomt, globaal het gewenste dalende of stijgende verloop hebben. Bouwers van een systeem zullen daarvoor empirisch geschikt gebleken functies (meestal met logaritmen) uitkiezen.

Systemen die gebruik maken van deze zoekmethode kunnen makkelijk de indruk wekken natuurlijke-taalverwerking toe te passen. In de meeste gevallen zal namelijk standaard gebruik gemaakt worden van stopwoordenlijstjes met betekenisloze woorden die niet in de index worden opgenomen. Als een gebruiker dan als zoekvraag een gewone volzin intikt, wordt die door het weglaten van de stopwoorden gereduceerd tot een rij woorden waarvan de meeste betrekking hebben op de inhoud van de vraag. Als iemand intikt: *'ik zou wel eens willen weten hoeveel fietsen er eigenlijk jaarlijks in Amsterdam gestolen worden'*, wordt dat automatisch gereduceerd tot: *weten, hoeveel, fietsen, eigenlijk, jaarlijks, Amsterdam, gestolen.*

Met uitzondering van de woorden *weten* en *eigenlijk* (die misschien ook beter in een stopwoordenlijst opgenomen kunnen worden) hebben die woorden inderdaad betrekking op het onderwerp. Om het zoekresultaat nog wat te verbeteren zal bovendien vaak een mechanisme zijn ingebouwd om woorden te reduceren tot hun woordstam, zodat ook op het enkelvoud *fiets*, het zelfstandig naamwoord *jaar* en de onbepaalde wijs *stelen* gezocht wordt. Daarbij spelen technieken uit de natuurlijke-taalverwerking wel al een rol. Gezien de taalafhankelijkheid van toe te passen regels zal een programma dat alleen Engelse regels toepast voor het Nederlands niet goed werken.

De *best-match* zoekmethode volgens het vectormodel, met gebruikmaking van de meeste zojuist geschetste details, wordt onder meer toegepast in de programma's Clarit, Personal Librarian en Smart. Ook berust de retrieval van veel WWW-zoekmachines, als AltaVista, HotBot en ExCite op deze methode.

5.4 *Natuurlijke-taalverwerking*

Onder 'natuurlijke-taalverwerking' wordt een breed scala aan geautomatiseerde hulpmiddelen verstaan, die zijn bedoeld om een door een gebruiker in gewone (natuurlijke) taal ingetikte zoekvraag om te zetten in een voor het computerprogramma zinvolle zoekopdracht. Daarbij kunnen alle door het zoekprogramma geboden technische retrievalmogelijkheden benut worden. Andersom worden die technieken ook toegepast om uit aanwezige tekstdocumenten (per definitie bijna altijd in natuurlijke taal) automatisch de meest betekenisvolle woorden te extraheren. Zelfs kan zo getracht worden documenten met gecontroleerde trefwoorden (bijvoorbeeld uit een al bestaande thesaurus) te laten karakteriseren (Van Gent, 1997).

Bij gebruikte methoden kan geformaliseerde kennis van een bepaald beperkt vakgebied een rol spelen, waarbij technieken op het terrein van Expert Systemen worden toegepast. Vaker zal echter algemene linguïstische kennis worden ingezet. Zo kan morfologische informatie worden gebruikt om zoekwoorden terug te brengen tot hun basis-stam (het zogenaamde vrije morfeem) door het strippen van voor en achter woorden geplakte pre- en suffixen. Ook kan decompositie van samengestelde woorden plaatsvinden. Anderzijds kunnen zoekwoorden juist worden uitgebreid tot reeksen van eveneens te zoeken varianten door kennis van verbuigingen, vervoegingen en dergelijke.

Semantische informatie kan kennis opleveren over de betekenis van woorden, bijvoorbeeld op grond van de context van woorden in een zoekvraag (maar ook in de doorzochte teksten). Zo kan getracht worden dubbelzinnigheid te elimineren *(disambiguation)*. Ook de functionele rol van woorden in een zin, dus zinsontleding en syntactische informatie kunnen daarbij gebruikt worden.

In het eerder gegeven fiets-voorbeeld zou ook een beter zoekresultaat kunnen worden verkregen, als in teksten voorkomende samengestelde begrippen *(noun phrases)* zoals *gestolen fietsen* of *gestolen Amsterdamse fietsen* als bijeenhorende concepten herkend kunnen worden. Hiervoor zullen door de computer veelal grammaticale en ontleedkundige regels moeten worden toegepast.

Behalve linguïstische regels, worden vaak ook domweg uitputtende woordenlijsten en zuiver statistische technieken toegepast om de hier genoemde methoden te implementeren. Samenvattend kun je zeggen dat in natuurlijke-taalsystemen in principe bijna altijd de elementen van interpretatie en van vertaling zijn terug te vinden. Interpretatie van wat bedoeld wordt, wat de juiste betekenis van een gebruikte term is, tot welke woordstam een term gereduceerd kan worden, enzovoort. Anderzijds moeten geïnterpreteerde zoektermen of zoekvragen vaak vertaald worden in te gebruiken trefwoorden, in met OR te combineren varian-

ten en synoniemen, in op de juiste wijze met AND-operatoren gecombineerde concepten, in voor het gebruikte retrievalprogramma specifieke opdrachten, enzovoort.
Aan systemen die dit soort mogelijkheden toepassen, wordt al vele jaren onderzoek gedaan. Zulke modules beginnen echter pas nu in commerciële producten beschikbaar te komen. Een probleem bij bijna al deze technieken is overigens hun taalafhankelijkheid. Zowel taalkundige en grammaticale regels als statistische gegevens over het voorkomen van woorden en lettercombinaties verschillen per taal. Sommige methoden werken daardoor voor bepaalde talen zelfs helemaal niet.

6 Typologie van retrieval-software

Bij de programma's die onder de noemer text retrieval vallen vinden we zeer uiteenlopende soorten. Functies, mogelijkheden en werking van de programma's hangen natuurlijk voor een groot deel samen met de toepassingen waarvoor ze in eerste instantie zijn bedoeld. Hoewel een strenge indeling in categorieën niet altijd recht doet aan individuele aspecten van ieder programma, gaan we hier toch uit van een in de praktijk als nuttig ervaren combinatie van toepassingsgebieden en werkingswijzen. Op grond van die uitgangspunten kwamen we in een andere publicatie (Sieverts, 1996a) tot een opdeling van retrievalsoftware in zeven categorieën:

- administratieve database software;
- klassieke retrieval-software;
- eindgebruikers-software;
- full-text retrieval-software;
- indexeerprogramma's;
- elektronische boeken;
- personal information managers.

De eerste drie categorieën hebben gemeen dat ze met meer of minder gedetailleerde veldstructuren werken. Onder de andere vier categorieën vallen systemen waar de gegevens standaard niet in velden zijn opgedeeld. Hoogstens worden bij full-text software bij uitzondering wel eens velden gebruikt die voor enkele aanvullende gegevens gebruikt worden. In het algemeen kun je zeggen dat programma's met een veldstructuur hogere eisen stellen aan de invoer van data, maar meer mogelijkheden bieden bij het zoeken en de gegevensuitvoer. Bij programma's zonder veldstructuur ligt dat andersom: ze stellen minder eisen aan de invoer, maar je kunt ook minder gericht zoeken en de mogelijkheden

van sorteren en formatteren van de uitvoer zijn beperkter. Een globaal overzicht van de functionaliteit van deze categorieën retrieval-software is weergegeven in tabel 1.

Tabel 1. Globale functionaliteit van verschillende types retrieval-software. (Ontleend aan Sieverts, 1996a)

	administratief	klassiek	eindgebruikers	full text	indexeer	elektron. boek	PIM
gebaseerd op records	ja	ja	ja	vaak	vaak *	zelden	vaak
velden-structuur	ja	ja	ja	soms	nee **	soms	soms
hoofdstuk-structuur	nee	nee	nee	soms	nee **	ja	soms
handmatige invoer	ja	ja	ja	nee	nee	soms	ja
strikte invoer-controle	ja	soms	ja	nee	nee	nee	nee
flexibele invoer	nee	ja	nee	ja	ja	ja	ja
ongeformatteerde elektronische invoer	nee	nee	nee	vaak	ja	vaak	ja
plaatjes in bestanden	soms	soms	nee	soms	soms	ja	vaak
zoekt via index	soms	ja	soms	ja	ja	ja	soms
meer (soorten) indexen	soms	ja	soms	nee	nee	nee	nee
Booleaanse combinaties	vaak	ja	ja	ja	ja	ja	ja
afstandsoperatoren	nee	soms	nee	ja	ja	ja	nee
sets combineren	nee	ja	soms	soms	soms	nee	soms
hypertext linking	nee	soms	nee	soms	soms	ja	vaak
variabele uitvoerformaten	ja	ja	ja	nee	nee	nee	nee
uitvoer volgens bibliografische stijlen	nee	nee	ja	nee	nee	nee	nee

* Hier spelen afzonderlijke tekst-files de rol van database-records.
** Niet door het programma op te leggen, wel te gebruiken als al in geïndexeerde tekst aanwezig.

Van de zeven genoemde categorieën komt de administratieve database software in deze paragraaf niet meer apart aan de orde, omdat daarvan, in de gedaante van database management systemen, in paragraaf 2.1 al karakteristieken gegeven zijn. Ook de personal information managers worden hier verder niet besproken. Dit zijn programma's voor persoonlijk gebruik, voor mensen die met veel losse aantekeningen van zeer uiteenlopende aard werken, van notities over

een afspraak tot concepten voor een brief of ideeën voor een lezing. Met deze weinig homogene categorie zal men in een bibliotheekomgeving maar zelden in aanraking komen. Van de overige vijf categorieën geven we wel een uitgebreider karakterisering.

6.1 Klassieke retrieval-software

Deze software is bedoeld voor het opslaan van gestructureerde gegevens die voornamelijk uit tekst bestaan, zoals we die kennen bij bibliografische databases. Deze gegevens worden opgeslagen in zelf te definiëren veldstructuren met variabele veldlengten. Vaak zijn daarin ook herhaalbare velden mogelijk. Gebruik van velden is handig, omdat elementen die vast in elk record voorkomen, op die manier altijd in hetzelfde hokje gezet kunnen worden. Bij het zoeken biedt dat het voordeel dat men specifiek op gegevens in een bepaald veld kan zoeken. Ook bij uitvoer van gevonden gegevens kan selectief alleen informatie uit gewenste velden worden afgedrukt.
Meestal wordt ervan uitgegaan dat records in een bestand ook allemaal dezelfde recordstructuur moeten hebben. Voor sommige toepassingen is het echter handig of noodzakelijk om records met uiteenlopende structuur toch in één bestand te kunnen onderbrengen. Lang niet met elk programma kan dat echter. Als het wel kan, gebeurt dat meestal door velddefinities te maken voor alle velden die maar kunnen voorkomen. Verschillende record-types worden dan gedefinieerd als bepaalde selecties uit al die mogelijke velden. Ook bij de in de volgende paragraaf besproken eindgebruikers-software wordt die methode veel toegepast.

Invoer van gegevens in deze programma's kan via het toetsenbord plaatsvinden of in elektronische vorm via zogenaamde *batch*-invoer. Om bij handmatige invoer de toelaatbaarheid of de spelling van termen in bepaalde velden te controleren, zijn verschillende methoden in gebruik. In de eerste plaats kan soms de index van de al in het bestand aanwezige gegevens op het scherm geroepen worden, teneinde consistentie met al aanwezige gegevens te vereenvoudigen. Daarnaast kan soms echte controle op de aard, de lengte of het patroon van ingevoerde gegevens plaatsvinden, bijvoorbeeld om te garanderen dat in een datumveld cijfers en leestekens volgens voorgeschreven patroon zijn ingevoerd en dat in een numeriek veld geen letters worden ingetikt. Verder kan bij sommige programma's voor bepaalde velden met autorisatielijsten van toegestane termen gewerkt worden.

Via *batch*-invoer kunnen grote hoeveelheden ('stapels') gegevens die uit andere bronnen (hostcomputers, CD-ROM's, lokale bestanden) zijn gedownload, direct worden ingelezen, zonder dat gegevens opnieuw ingetikt hoeven te worden. Meestal is dat alleen mogelijk in de vorm van ASCII-files, dat wil zeggen zuivere tekst-files zonder bijzondere controletekens. In de meeste gevallen moet die ASCII-file een door het retrievalprogramma voorgeschreven *format* hebben. Dat wil zeggen dat de structuur en uiterlijke kenmerken van records en velden aan bepaalde eisen moeten voldoen. Om gedownloade gegevens in die vorm te krijgen zijn soms aparte conversieprogramma's nodig (Sieverts, 1996a).

Voor het zoeken bieden deze programma's uitgebreide zoek- en combineermogelijkheden met truncatie, AND-, OR- en NOT-operatoren en dergelijke, zoals die ook van de grote online retrievalsystemen bekend zijn. Vooral de mogelijkheid te zoeken in gespecificeerde velden en zo expliciet gebruik te maken van de formele structuur van de records, is kenmerkend voor deze programma's. Sommige programma's werken niet met de gebruikelijke operatoren, maar bieden alleen de mogelijkheid het laatste zoekresultaat met een volgend criterium in te perken (*select* of *narrow* = AND), met een extra zoekterm uit te breiden (*include* of *broaden* = OR), of door uitsluiting te verkleinen (*exclude* = NOT). Deze bewerkingen bieden doorgaans iets minder flexibiliteit dan wat een ervaren gebruiker met ingewikkelde combinaties van AND, OR, NOT en haakjes kan bereiken.
Om ook in grote hoeveelheden gegevens snel te kunnen zoeken wordt standaard gebruik gemaakt van één of meer indexen op de verschillende velden. De precieze vorm van de indexen heeft direct invloed op de zoekmogelijkheden. Zo kun je vaak alleen op afzonderlijke velden zoeken, wanneer daarvoor ook afzonderlijke indexen gemaakt zijn. Toch zijn er ook programma's die maar één index maken, maar daarin registreren uit welke velden de vindplaatsen van de termen afkomstig zijn. Om standaard op een aantal inhoudelijk belangrijke velden (titel, samenvatting, trefwoorden) tegelijk te kunnen zoeken, kun je soms een gezamenlijke zogenaamde *basic index* voor die velden definiëren.
Verder zijn in principe verschillende soorten indexen mogelijk: een woord-index, een term-index of een veld-index. Bij een woord-index komen alle losse woorden in de index maar ook vaak de losse delen van samengestelde begrippen die door koppeltekens of andere leestekens gescheiden worden. Bij een term-index kunnen samengestelde begrippen (bijvoorbeeld 'online information retrieval') als geheel in de index komen. Om zulke begrippen tijdens het indexeerproces als zodanig te herkennen moet het programma óf vaste koppeltekens tussen de samenstellende woorden herkennen, óf juist scheidingstekens

tussen afzonderlijke termen. Bij een veld-index wordt de inhoud van een veld als geheel in de index gezet.
Nog meer dan bij het invoeren van gegevens, is het tijdens het zoeken handig als indexen op het scherm kunnen worden opgeroepen. Vooral als daarbij ook informatie wordt gegeven over aantallen vindplaatsen, kan dat helpen bij het kiezen van de juiste zoektermen. Bovendien kunnen termen daaruit meestal rechtstreeks als zoekterm worden aangeklikt.

Gevonden gegevens kunnen meestal op verschillende, zelf te definiëren manieren worden afgedrukt. Zo kunnen zowel beknopte titellijstjes als volledige gegevens op scherm of papier getoond worden. Bovendien kan gedacht worden aan afdruk-formats voor aanwinstenlijsten of zelfs voor adreslabels. Daarbij moeten de gegevens liefst ook op allerlei criteria gesorteerd kunnen worden, en ook op meer termen uit één record zodat dat record op meer plaatsen in een lijst herhaald kan worden. Dat laatste kan zogenaamde KWOC *(keyword out of context)* indexen opleveren.

In veel moderne software uit deze categorie kunnen ook plaatjes worden opgenomen. Daarvoor wordt meestal een apart grafisch veld gedefinieerd. Opslag van de plaatjes kan dan op verschillende manieren plaatsvinden. In de eerste plaats kan elk plaatje een apart bestand met eigen file-naam blijven, waarbij het grafische veld alleen die bestandsnaam bevat. Voor het tonen van het plaatje in het record wordt het dan via die naam opgevraagd en meestal via een geïntegreerd hulpprogramma (een viewer) op het scherm zichtbaar gemaakt. Een andere mogelijkheid is dat de gegevens van alle plaatjes bij een bestand tezamen in één apart plaatjesbestand worden opgeslagen en het grafische veld alleen een interne pointer naar het juiste plaatje bevat. Een derde mogelijkheid is dat het plaatje zelf in de vorm van binaire gegevens ín het betreffende veld in het record wordt opgeslagen.

Klassieke retrieval-software kan onder andere worden toegepast voor bibliografische databases (dus ook voor bibliotheekcatalogi wanneer geen koppeling met administratieve functies vereist is) en voor beschrijving van collecties van objecten zoals in een museum, waarbij het opnemen of koppelen van afbeeldingen een nuttige extra functie is. In Nederland veel gebruikte pakketten die tot deze categorie behoren zijn onder meer: BRS/Search, Cardbox-Plus, CDS-ISIS, DB/-Textworks (de Windows-opvolger van Inmagic), Idealist en Strix.

6.2 Eindgebruikers-software

Eindgebruikers-software hebben we zo genoemd, omdat hieronder vallende programma's vooral zijn bedoeld om informatie-eindgebruikers te helpen bij het beheer van hun literatuurdocumentatie. Hoewel dit soort programma's dus niet in eerste instantie op informatiespecialisten gericht is, zal men er in veel bibliotheken toch direct of indirect mee te maken hebben, omdat de in deze bestanden op te nemen informatie bijna altijd via een informatiecentrum of bibliotheek bij de eindgebruikers binnenkomt (Sieverts, 1996b).
Zij die veel literatuurinformatie verzamelen, doen dat vaak om zelf ook weer aan de productie van nieuwe literatuur bij te dragen. Dat betekent dat zij vaak artikelen schrijven, waarin selecties uit de verzamelde literatuur zijn opgenomen, in de vorm van literatuurverwijzingen of referenties. Speciaal voor dit doel wordt in deze programma's veel aandacht besteed aan functies voor het automatisch genereren van referentielijsten volgens de 'stijl'-voorschriften die door uitgevers van (vooral wetenschappelijke) tijdschriften aan auteurs worden opgelegd. Om die reden wordt dit type software in het Engels wel *Bibliography Formatting software* genoemd.
De hier bedoelde stijlvoorschriften kunnen inhouden dat in kopij voor het ene tijdschrift de voorletters van auteurs achter de achternaam moeten komen en voor het andere ervóór en dat voor de eerste auteur van een publicatie soms een ander voorschrift kan gelden dan voor de volgende. Soms mogen alle acht auteurs van een artikel worden vermeld en in een ander geval dient na twee namen *'et al.'* te volgen. Tijdschriftnamen moeten soms voluit en soms volgens standaard afkorting worden vermeld. Volumenummers van tijdschriften moeten nu eens vet en dan weer cursief gedrukt worden, waarbij ze nu eens voor en dan weer na het jaar van publicatie moeten staan. Soms moet niet alleen de beginbladzijde, maar ook de slotpagina van een artikel worden vermeld, enzovoort. Bovendien hangen details van de opmaakregels vaak af van de aard van het document waarnaar verwezen wordt; voor een tijdschriftartikel gelden andere regels dan voor een congresbijdrage of een boekhoofdstuk.

Om in de uitvoer aan deze voorschriften te kunnen voldoen, moet daarmee in structuur en inhoud van het bestand al rekening zijn gehouden. Dat betekent dat gegevens op gecontroleerde en tot in details gestructureerde wijze opgeslagen en dus ingevoerd moeten worden. Een karakteristiek van dit soort software is daarom dat met voorgedefinieerde, meestal nauwelijks aanpasbare recordstructuren wordt gewerkt. Bovendien zullen er afzonderlijke recordstructuren zijn voor de verschillende soorten documenten waaraan gerefereerd kan wor-

den. Dat betekent dat die verschillende structuren naast elkaar in één database toegelaten moeten zijn.
Ten behoeve van de noodzakelijke consistentie wordt de gebruiker bij handmatige invoer vaak op bepaalde invoer-conventies gewezen of kunnen gegevens van bijvoorbeeld tijdschriftnamen in de juiste spelling worden ontleend aan meegeleverde autorisatie-bestanden of aan de index van al ingevoerde gegevens.

Bijna meer nog dan bij de voorgaande categorie van de klassieke retrievalprogramma's, is batch-invoer van gedownloade gegevens uit online en CD-ROM-databases hier van belang. CD-ROM's met bibliografische bestanden zijn voor eindgebruikers populaire bronnen van literatuurinformatie. Aangezien de standaarduitvoer van de meeste bibliografische databases aanzienlijk minder gedetailleerde record-structuren gebruikt dan de eindgebruikers-software vereist, is batch-invoer bijna nooit zonder meer mogelijk. Vaak moeten complexe conversies worden uitgevoerd. Daarbij moeten onder meer gegevens uit bepaalde velden in afzonderlijke elementen worden uitgesplitst, zoals paginanummer, jaar van publicatie, volume-nummer en tijdschrifttitel, die in gedownloade records vaak bij elkaar in één veld staan, maar in de eigen database in afzonderlijke velden moeten komen.
Om te voorkomen dat de gebruiker zelf ingewikkelde conversies moet specificeren, bestaan er voor veel van deze programma's specifieke conversie- of invoermodules, waarin deze bewerkingen zijn voorgedefinieerd. Deze modules kunnen in het programma geïntegreerd zijn of dienen apart te worden aangeschaft. In deze modules zijn vaak al vele tientallen tot zelfs honderden conversies vanuit verschillende bestanden uit diverse online- en CD-ROM-systemen voorgeprogrammeerd.
Omdat dit soort software vooral in biomedische hoek veel wordt gebruikt, zijn conversies uit de meeste in aanmerking komende bestanden en systemen op dat terrein vrijwel altijd al standaard aanwezig. In andere disciplines kan het echter voorkomen dat bepaalde, voor een gebruiker belangrijke bestanden, hosts of CD-ROM-formaten, in de module ontbreken. Soms kunnen aanvullende conversies dan tegen een geringe vergoeding bij de leverancier worden aangevraagd. Ook bieden sommige programma's de gebruiker zelf de mogelijkheid aanvullende conversies te specificeren, al blijkt dat vaak nogal ingewikkeld, gezien de strenge eisen die eindgebruikers-software aan de conversie stelt.

Zoek- en combineermogelijkheden zijn in deze programma's vaak wat beperkter, enerzijds omdat ze niet voor informatiespecialisten bedoeld zijn, anderzijds omdat het zoeken van informatie (op onderwerp) vaak niet de belangrijkste

functie van het programma geacht wordt. Omdat van persoonlijke documentatiebestanden aanvankelijk niet werd verwacht dat ze erg groot zouden worden, werd vaak volstaan met sequentiële zoektechnieken. Pas de laatste jaren wordt dit bij de meeste programma's aangevuld met mogelijkheden op bepaalde velden (maar meestal niet op alle) indexen te laten maken, zodat ook nog snel gezocht kan worden wanneer de bestanden groter worden.
Om de mogelijkheden van onderwerp zoeken in deze programma's te verbeteren wordt soms een eenvoudige vorm van automatische trefwoordtoekenning toegepast. Daartoe vergelijkt het programma de woorden in titels en samenvattingen van nieuw ingevoerde records met de termen die al in de index van het trefwoordenveld aanwezig zijn. Blijkt een woord of woordcombinatie in titel of abstract van een nieuw record al als trefwoord te bestaan, dan wordt het automatisch ook aan dat nieuwe record toegekend.

Bij deze programma's ligt de nadruk op gestandaardiseerde uitvoerformaten voor bibliografieën of referentielijsten. Daartoe worden vaak tientallen voorgedefinieerde tijdschriftstijlen met het programma meegeleverd. Bij de betere programma's zal elke stijl nog afzonderlijke regels kennen voor de verschillende voorkomende documentsoorten. Veel programma's bieden bovendien de mogelijkheid een stijl aan te passen of zelf gedefinieerde nieuwe stijlen toe te voegen.
Referentielijsten kunnen worden samengesteld door de gewenste records successievelijk in de database op te zoeken en te markeren en dan de hele gemarkeerde set uit te printen. Daarnaast bieden de meeste programma's ook de mogelijkheid een referentielijst volautomatisch uit een tekstverwerkingsbestand van een manuscript te laten genereren. De programma's kunnen daartoe bestanden lezen, die met WP-, Word- of andere tekstverwerkers zijn gemaakt. Als literatuurverwijzingen daarin volgens voorschrift zijn gecodeerd, worden de bijbehorende records automatisch in de database opgezocht, zodat gegevens daaruit in de literatuurlijst verwerkt kunnen worden. Bovendien worden de in het manuscript aanwezige coderingen vaak meteen vervangen door de echte verwijzingen, ook in de vorm zoals het betreffende tijdschrift voorschrijft. Dat betekent voor het ene tijdschrift als oplopende referentie-nummers [14,15], voor het andere als aanduidingen in de vorm *(Jansen, 1995)*. Andersom kan tijdens het schrijven van het manuscript meestal vanuit de tekstverwerker de database geraadpleegd worden.

Toepassing van deze programma's beperkt zich tot opslag van bibliografische gegevens (in de nieuwste versies ook voor beschrijving van WWW-documenten). Doelgroep vormen wetenschappers die zelf onderzoek doen en frequent publi-

ceren. In veel universitaire en speciale bibliotheken heeft men dan ook met gebruikers van deze programma's te maken. Vooral in de Verenigde Staten zijn er vele tientallen van op de markt (Stigleman, 1994, 1996). De bekendste die in Nederland gebruikt worden, zijn EndNote, Papyrus, ProCite en Reference Manager.

6.3 Full-text retrievalprogramma's

Met deze software kunnen bestanden met volledige teksten van artikelen, rapporten, boekhoofdstukken en dergelijke worden opgebouwd. Om meer gericht zoeken mogelijk te maken kan vaak wel enige structuur in de informatie worden aangebracht, in de vorm van enkele toegevoegde velden, of bij uitzondering in een meer gedetailleerde veldenstructuur zoals klassieke retrievalprogramma's die kennen. Om alle mogelijke voorkomende soorten full-text documenten te kunnen opnemen, worden uiteraard geen praktische begrenzingen aan recordlengten gesteld.
Bij software die alleen voor dit soort toepassingen is bedoeld, ontbreken meestal handmatige invoer- en bewerkingsfaciliteiten. Er wordt van uitgegaan dat de teksten al op andere wijze (meestal met een tekstverwerker) zijn aangemaakt en daarna geen onderhoud meer vergen. Dat betekent dat altijd in elk geval batchinvoer mogelijk is. Overigens worden bestanden uit een tekstverwerker daarbij niet altijd rechtstreeks geaccepteerd. Soms is eerst nog conversie naar een puur ASCII-bestand nodig.

Ten behoeve van het zoeken wordt bijna altijd alleen met indexen op losse woorden gewerkt, waarbij een stopwoordenlijst wordt gebruikt. De zoekmogelijkheden zijn gericht op grote, weinig gestructureerde hoeveelheden tekst. Daartoe kunnen behalve de Booleaanse AND-operator ook nabijheidsoperatoren worden gebruikt.
In deze programma's worden vaak verschillende technieken toegepast, die het ontbreken van structuur en gecontroleerde ontsluiting in full-text bestanden moeten compenseren. Zo wordt, naast het gewone truncatie-zoeken, soms gebruik gemaakt van automatische woord-*'stemming'* en worden *fuzzy* en *sound-alike* zoektechnieken toegepast (zie paragraaf 5.1). Verder bieden sommige van deze programma's mogelijkheden van best-match zoeken en presenteren zoekresultaten op grond daarvan in relevantie-volgorde (paragraaf 5.3). Ook kunnen vaak hypertext-relaties worden toegepast (paragraaf 5.2).
Aangelegde hyperlinks kunnen zowel voorkomen binnen hetzelfde record of dezelfde tekst, als tussen verschillende records/teksten in hetzelfde bestand.

Hyperlinks kunnen ook naar bijvoorbeeld plaatjes verwijzen, zij het dat die in moderne programma's ook in de tekst zelf kunnen worden opgenomen. Bij Windows-programma's zorgt bijvoorbeeld de OLE techniek *(Object Linking and Embedding)* ervoor dat plaatjes echt ín de tekst zichtbaar gemaakt worden.
Vaak kan binnen één tekst zeer gedetailleerd naar bepaalde passages gezocht worden. Meestal wordt de gevonden informatie dan ook niet als een heel record op het scherm getoond, beginnend bij het begin ervan, maar kunnen meteen die tekstgedeelten op het scherm gevraagd worden, waar de in de zoekvraag gebruikte woorden voorkomen, de context van de zoektermen. Dat kunnen vaak meer passages uit dezelfde tekst zijn.

Bij ontbreken van een recordstructuur zijn er uiteraard geen verschillende presentatieformaten, zoals klassieke retrieval-software die kent. Hoogstens kan selectief een alinea of een passage met de zoektermen geprint worden of voor verder gebruik naar een bestandje of een tekstverwerker worden gekopieerd.
Full-text retrieval-software kan uiteraard voor elk soort 'volledige' teksten worden toegepast. In een bibliotheekomgeving kunnen dat volledige tijdschriftartikelen en rapporten zijn. In een kantooromgeving, waar men met andersoortige tekst-documenten te maken heeft, kan deze software ook worden toegepast. In de praktijk zal men daar echter vaker een indexeerprogramma gebruiken. Deze hierna te bespreken programma's vormen een gespecialiseerde variant van de full-text retrieval-software. Voorbeelden van full-text programma's zijn Personal Librarian en Recall-Plus, maar gezien hun zoekmogelijkheden kunnen ook enkele klassieke pakketten als BRS/Search en DB/Textworks in zekere zin tot deze categorie gerekend worden.

6.4 Indexeerprogramma's

Indexeerprogramma's verschillen van de algemene categorie van full-text programma's doordat ze uitsluitend een index opbouwen op allerlei verschillende tekstbestanden, zonder die in een database op te nemen. Wat in een echte database dus als records beschouwd worden, blijven bij een indexeerprogramma op computerniveau afzonderlijke bestandjes. Toch ziet de presentatie van zoekresultaten er meestal gewoon uit of we met een database met records te maken hebben. De 'index' is dan het overkoepelende geheel dat met ons idee van 'database' correspondeert.
Indexeerprogramma's zijn vooral bedoeld om grote hoeveelheden ongestructureerde tekst te doorzoeken die op harde schijven in PC's en netwerk-servers zijn opgeslagen. Naarmate die schijven groter worden, neemt de behoefte aan dit

soort programma's toe, zeker ook in een kantooromgeving. Vaak ontbreekt hier de tijd om alle stukken, voorzien van trefwoorden en andere toegevoegde karakterisering, in databases op te slaan. Dan is het makkelijk een programma te hebben dat in de ongeordende inhoud van een harde schijf snel en zonder al te veel moeite toch de gezochte informatie of het gezochte document kan vinden.

Deze programma's bieden meestal op zijn minst dezelfde zoekmogelijkheden als de in de vorige paragraaf besproken full-text software, met uitzondering misschien van de volledige mogelijkheden van best-match zoeken.
Aangezien indexeerprogramma's uitgaan van al beschikbare 'elders' aangemaakte gegevens, kunnen ze zelf geen structuur opleggen aan de doorzoekbaar te maken informatie. Als de geïndexeerde informatie toevallig enige herkenbare vaste structuur blijkt te hebben, kan daar vaak wel gebruik van gemaakt worden. Daartoe kunnen een soort pseudo-velden gedefinieerd worden. In een file met gedownloade literatuurreferenties kan dan bijvoorbeeld toch op titelveld gezocht worden, door zo'n veld voor het zoeken te definiëren als die stukken tekst die staan tussen de karakteristieke tekststrings 'TI-' en 'AU-', die in de tekst de aanduidingen voor het titelveld en het daarop volgende auteursveld zijn. Ook in bepaalde teksten aanwezige SGML- of HTML-coderingen kunnen zo voor het definiëren van zoekvelden gebruikt worden.

Van belang is verder dat indexeerprogramma's niet alleen ASCII-tekst indexeren, maar ook teksten die zijn gemaakt met allerlei verschillende tekstverwerkingsprogramma's. Standaard zullen zeker altijd Word en WordPerfect bestanden (in diverse versies) herkend worden. Daarnaast worden vaak allerlei minder bekende tekstverwerkers ondersteund, alsook veelgebruikte database- en spreadsheetformaten. Als zoekresultaat gevonden teksten kunnen vrijwel altijd in vereenvoudigde opmaak door het indexeerprogramma zelf getoond worden. Daarnaast biedt het meestal de mogelijkheid om automatisch het bij het gevonden document horende programma op te starten.
Doordat wijzigingen van teksten buiten het indexeerprogramma om (in de betreffende tekstverwerker) worden aangebracht, dient apart zorg besteed te worden aan het up-to-date houden van indexen. Daartoe houdt het programma, aan de hand van de door het besturingssysteem aan elk tekstbestand gekoppelde datum en tijd, bij welke versie het laatst geïndexeerd is. Zo blijven indexeeracties automatisch beperkt tot nieuwe bestanden en die oude bestanden waarin sinds de vorige keer wijzigingen zijn aangebracht. Ook kan vaak een schema worden opgegeven van tijdstippen waarop automatisch herindexering gestart moet worden.

In een kantooromgeving worden indexeerprogramma's vaak gebruikt als eenvoudig Documentair Informatie Systeem, een zogenaamde DIS. De functionaliteit blijft dan beperkt tot het zoeken; er zijn geen functies voor *work flow management,* zoals echte DIS-pakketten die wel hebben. Om alleen op papier aanwezige documenten ook in het systeem te kunnen opnemen bieden steeds meer indexeerprogramma's integratie met software voor optische karakterherkenning (OCR), zodat ook gescande papieren documenten full-text doorzoekbaar gemaakt kunnen worden. Gezien de frequente herkenningsfouten die zelfs goede OCR-software maakt, is het hierbij essentieel dat fuzzy zoeken mogelijk is.
Behalve kantoortoepassingen kent dit soort software ook bibliotheektoepassingen. In een tekstverwerker gemaakte inventarisaties van materialen die zich niet voor de gewone catalogus lenen of teksten waarop men voor inlichtingenwerk terug kan vallen, kunnen zo eenvoudig doorzoekbaar gemaakt worden. Ook geïntegreerd met een OCR-module zijn vele toepassingen mogelijk, zoals het doorzoekbaar maken van inhoudsopgaven van boeken uit een deel van een collectie of van de hele tekst van een gedrukte bibliografie of inventaris.
Enkele bekende programma's uit deze categorie zijn dt/Search, Isys, het in WordPerfect ingebouwde QuickFinder en ZYindex. Bovendien zijn ook veel van de op het World Wide Web gebruikte zoekmachines een soort indexeerprogramma's. Sommige daarvan, zoals AltaVista en OpenText, hebben ook versies die voor lokaal gebruik, voor het doorzoeken van de eigen computerschijven bedoeld zijn.

6.5 Elektronische boeken

We onderscheiden nog een derde categorie retrieval-software voor volledige teksten. Daarbij ontbreken niet alleen veldstructuren, maar er is zelfs geen sprake meer van afzonderlijke records zoals we die bij alle andere categorieën tegenkwamen. Deze software is dan ook bedoeld voor elektronische versies van boekachtige documenten die in feite één geheel vormen. Dat kunnen echte leerboeken zijn, maar ook handleidingen en bedienings- of onderhoudsvoorschriften.
Voor dit type informatie is het niet meer voldoende dat er alleen maar goede full-text retrievalfaciliteiten zijn. In aanvulling daarop zal zo'n programma de structuur van het boek moeten ondersteunen. In plaats van een veldenstructuur is daarvoor een geneste structuur nodig, van hoofdstukken en verscheidene niveaus van paragrafen. Bovendien wordt die structuur zichtbaar gemaakt in automatisch gegenereerde inhoudsopgaven met hoofdstuk- en paragraaftitels. Van daaruit kan dan rechtstreeks naar een bepaald deel van het 'boek' gesprongen worden.

Het vanuit de inhoudsopgave naar bepaalde paragrafen springen doet al aan hypertext denken. Bovendien zal zo'n programma meestal ook echte hypertextfuncties ondersteunen, teneinde gebruik van elektronische verwijzingen mogelijk te maken. Behalve de al door de auteur aangebrachte links, kan de gebruiker vaak ook persoonlijke hyperlinks toevoegen. Hypertextfuncties houden verder in dat het gevolgde pad door de tekst geregistreerd moet worden, zodat de gebruiker dat pad zichtbaar kan maken en op zijn schreden kan terugkeren. Bovendien bieden moderne programma's uit deze categorie vrijwel altijd de mogelijkheid om plaatjes of andere 'media' direct of via hyperlinks in de tekst op te nemen.

Dit soort software kent vaak extra functies die karakteristiek zijn voor het gebruik dat van een elektronisch boek gemaakt wordt. Zo kunnen soms voorgeprogrammeerde paden door een 'hyperboek' worden aangelegd. Die maken het mogelijk om voor speciale toepassingen, taken, problemen of gebruikers vooraf een kortste route uit te zetten langs de daarvoor relevante gegevens. Verder kunnen aan willekeurige punten in de tekst vaak persoonlijke aantekeningen gekoppeld worden, die later weer automatisch zichtbaar worden of aangeklikt kunnen worden. Daarnaast zijn vaak markeringen aan te brengen met een soort elektronische 'highlighters'.
Als een elektronisch boek in een netwerkomgeving door meer mensen gebruikt moet worden, dienen de persoonlijke aantekeningen, markeringen en hyperlinks bovendien uitsluitend in een persoonlijk 'transparant laagje' los van de vaste basistekst geregistreerd te worden, zodat aantekeningen van verschillende gebruikers niet interfereren. Een gedeeltelijk ander probleem moet worden opgelost als de software tevens een middel is, waarmee meer gebruikers aan de opbouw van hetzelfde 'boek' moeten werken.
Voorbeelden van programma's uit deze categorie zijn Druid, Envoy (met in de standaardversie slechts zeer beperkte zoekmogelijkheden), Expanded Book Toolkit en Folio Views.

7 Slotopmerkingen

In dit hoofdstuk is getracht een algemeen overzicht te geven van de werking van retrieval-software, van de verschillende soorten retrieval-software die op dit moment verkrijgbaar zijn en van de belangrijkste toepassingen waarvoor die bedoeld zijn. Bij de werking van de software hebben we ons beperkt tot op dit moment al operationele technieken. Gezien het vele onderzoek dat de afgelopen

jaren op dit terrein begonnen is, valt te verwachten dat spoedig meer nieuwe technieken in commerciële producten verwerkt zullen worden.
Wat de toepassing van retrieval-software betreft, hebben we geen aandacht besteed aan het aspect software-keuze. Daar bestaan al andere tamelijk uitgebreide publicaties over, zoals de reeks waarin resultaten van de VOGIN Projectgroep Programmatuur gerapporteerd worden (Sieverts, 1996a). Daarin zijn uitgebreide gegevens over individuele programma's opgenomen. In dit hoofdstuk werden namen van programma's alleen vermeld als illustratie bij beschreven technieken en categorieën. Dat pretendeert dus geenszins volledigheid.
Een nog onbelicht aspect van de programma's zelf is hun mate van aanpasbaarheid. Verkrijgbare programma's kunnen daar sterk in verschillen. Veel ervan zijn *kant-en-klaar van de plank* geleverde applicaties, waarin de gebruiker, behalve zelf te definiëren recordstructuren en uitvoerformaten, verder eigenlijk niets kan aanpassen. Andere programma's zijn een soort *toolbox* waarmee een ontwikkelaar helemaal naar eigen wensen een op een specifieke doelgroep gerichte applicatie kan bouwen.
Een belangrijke recente ontwikkeling tenslotte, is dat netwerkgebruik steeds minder beperkt blijft tot *local area networks,* waardoor koppeling van verschillende systemen belangrijker wordt. Daartoe wordt steeds vaker gebruik gemaakt van het *client-server* principe, waarbij een deel van een programma op de computer van de gebruiker draait (de *client* die meestal het gebruikersinterface afhandelt) en het andere deel op een centrale computer (de *server* waar de databases en zoeksoftware zich bevinden). Een algemene, open standaard voor client-server communicatie voor bibliotheeksoftware is het *Z39.50* protocol (Oude Groeniger, 1996). Voor retrieval-software wordt dat echter nog zelden toegepast. Wel worden voor steeds meer programma's WWW-versies ontwikkeld, waarmee de gebruiker vanaf een web-browser als Netscape via het *HyperText Transfer Protocol* (HTTP) kan communiceren.

Met dank aan de 20 leden van de VOGIN Projectgroep Programmatuur; in het bijzonder aan Hanneke Smulders en Marten Hofstede, zonder wier inbreng in het VOGIN-project deze tekst niet zo makkelijk had kunnen worden geschreven.

8 Literatuur

Blair, D.C., en M.E. Maron, An Evaluation of Retrieval Effectiveness for a Full-text Document Retrieval System. *Communications of the ACM* 28 (1985) 289-299.

Blair, D.C., en M.E. Maron, Full-text Information Retrieval: further analysis and clarification. *Information Processing & Management* 26 (1990) 437-447.

Evans, R., Beyond Boolean: relevance ranking, natural language and the new search paradigm. In: M.E. Williams (red.), *Proceedings of the National Online Meeting.* Medford: Learned Information, 1994, pp. 121-128.

Franklin, C., Hypertext Defined and Applied. *Online* 13 (1989) nr. 3, 37-49.

Gent, J.J.M. van, Automatische thesauriële ontsluiting. In: H. Magrijn e.a. (red.), *Woordsystemen; theorie en praktijk van thesauri en trefwoordsystemen,* 's Gravenhage: NBLC, 1997.

Magrijn, H. e.a. (red.), *Woordsystemen; theorie en praktijk van thesauri en trefwoordsystemen,* 's Gravenhage: NBLC, 1997.

Oude Groeniger, B., *Software voor bibliotheekautomatisering.* Eelde: Projectgroep Bibliotheeksoftware, 1996.

Salton, G., *Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information.* Reading: Addison-Wesley, 1989.

Sieverts, E.G., en M.W. de Jong-Hofman (red.), *Online opsporen van informatie.* 's Gravenhage: NBLC, 1996.

Sieverts, E.G. (red.), *Text retrieval software; een vergelijking van bijna 50 retrieval-programma's.* 's Gravenhage: VOGIN, 1996a.

Sieverts, E.G., End-user Software. In: A. Kent (red.), *Encyclopedia of Library and Information Science* 57, suppl. 20, New York: Marcel Dekker Inc., 1996b, pp. 154-175.

Stigleman, S., Bibliography Formatting Software: an updated buying guide for 1994. *Database* 17 (1994) nr. 6, 53-65.

Stigleman, S., Bibliography programs do Windows. *Database* 19 (1996) nr. 2, 57-66.