

---

# Van Case-Based Reasoning tot Information Retrieval

Case retrieval voor de helpdesk van een webhostingbedrijf

Edgar J. Meij

E.J. Meij  
Studentnummer: 9903143

Afdeling Informatiekunde  
Faculteit der Natuurwetenschappen, Wiskunde en Informatica  
Universiteit van Amsterdam  
September 2005

Afstudeerdocent:  
dr. W.N.H. Jansweijer

Tweede beoordelaar:  
prof. dr. B.J. Wielinga

Experience is that marvelous thing that enables you to recognize a mistake when you make it again. -- Franklin P. Jones



# Samenvatting

De helpdesk van Hostnet, een webhosting provider, ontvangt dagelijks gemiddeld 35 tot 50 vragen van klanten. Er bestaan binnen het vakgebied waarin Hostnet opereert weinig kant-en-klare handleidingen en dit is met name merkbaar op de helpdesk. Vragen en problemen worden hier veelal uit het hoofd opgelost en er is bijna geen ondersteuning om deze kennis te expliciteren. Van de medewerkers wordt verwacht dat ze parate kennis hebben van een verscheidenheid aan mogelijke problemen en dat ze creatief kunnen omgaan met andere, nog niet eerder voorgekomen problemen.

Hostnet gebruikt sinds geruime tijd een ticketing applicatie om dergelijke vragen in goede banen te leiden. Een bijkomend voordeel is dat alle gestelde vragen worden opgeslagen met bijbehorend antwoord. Hierdoor is er inmiddels een weelde aan domein- en vakspecifieke kennis verzameld door dit systeem. Binnen het onderzoeksgebied van case-based reasoning wordt dergelijke informatie op een intuïtieve wijze ontsloten. Case-based reasoning gebruikt eerder opgeloste problemen (cases) als bron van kennis om in de toekomst overeenkomstige cases sneller en wellicht beter op te kunnen lossen. Een van de belangrijkste componenten in ieder case-based reasoning systeem is de retrieval module. Deze zoekt naar meest gelijkende cases, aan de hand van een nieuwe case en een similariteitsmaat. Hiervoor kunnen bijvoorbeeld statistische (syntactische) modellen uit het toepassingsgebied van information retrieval worden gebruikt.

Binnen dit onderzoek is geanalyseerd in hoeverre de reeds opgeslagen vragen kunnen dienen als basis voor de syntactische retrieval module van een case-based reasoning systeem voor Hostnet en wat voor invloed verscheidene information retrieval technieken hebben op de prestaties van een dergelijke module. De beoordeelde technieken zijn *stemming*, *term weighting* (TF.IDF) en de combinatie hiervan. De beschreven omstandigheden zijn overigens niet uniek voor Hostnet. Ieder dienstverlenend bedrijf dat direct contact heeft met zijn klanten is bekend met een dergelijke situatie en zou baat kunnen hebben bij de in dit onderzoek gepresenteerde resultaten.

De belangrijkste conclusie is dat de gebruikte information retrieval aanpak behoorlijke resultaten oplevert. Het hoogst behaalde percentage van correcte antwoorden ligt rond de 60%. Dit betekent dat hetzelfde percentage van nieuwe cases succesvol beantwoord kan worden. De gemiddelde positie van het eerste goede antwoord was aan de hoge kant, namelijk rond 7 van de 10. Wat opvalt is dat de beste resultaten behaald worden zonder enige verfijning middels de genoemde technieken. Het blijkt dat de resultaten voor een toepassing binnen een volwaardig case-based reasoning systeem nog niet optimaal zijn, maar de in dit onderzoek gebruikte aanpak lijkt wel degelijk levensvatbaar.



# Inhoudsopgave

|   |           |
|---|-----------|
| <b>VOORWOORD</b>  | <b>IX</b> |
| <b>1. INLEIDING</b>   | <b>1</b>  |
| 1.1 ACHTERGROND   | 1         |
| 1.2 DOEL- EN VRAAGSTELLING  | 3         |
| 1.3 OPBOUW SCRIPTIE   | 3         |
| <b>2. CASE-BASED REASONING</b>                                    | <b>5</b>  |
| 2.1 BEGRIPPEN BINNEN CASE-BASED REASONING                         | 5         |
| 2.2 HISTORIE  | 6         |
| 2.3 CASE-BASED REASONING ALS PROCES                               | 8         |
| 2.3.1 Retrieve  | 9         |
| 2.3.2 Reuse, revise, retain                                       | 10        |
| 2.4 CASE-BASED LEARNING   | 10        |
| 2.5 TAXONOMIEËN VAN CASE-BASED REASONING SYSTEMEN                 | 11        |
| 2.5.1 Textual, conversational and structural case-based reasoning | 12        |
| 2.5.2 Interpretative and problem-solving case-based reasoning     | 13        |
| 2.5.3 Case-based reasoning als containerbegrip                    | 14        |
| 2.6 VOORBEELDEN VAN CASE-BASED REASONING APPLICATIES              | 15        |
| 2.6.1 FAQFINDER   | 15        |
| 2.6.2 CASCADE   | 17        |
| 2.6.3 HOMER   | 18        |
| 2.6.4 FALLQ   | 19        |
| 2.6.5 Conclusie   | 19        |
| <b>3. INFORMATION RETRIEVAL</b>                                   | <b>21</b> |
| 3.1 HISTORIE  | 21        |
| 3.2 MODELLEN  | 22        |
| 3.2.1 Boolean model   | 23        |
| 3.2.2 Probabilistisch model                                       | 24        |
| 3.2.3 Vector space model  | 25        |
| 3.3 INFORMATION RETRIEVAL TECHNIEKEN                              | 28        |
| 3.3.1 Term extractie  | 29        |
| 3.3.2 Stopwoorden   | 29        |
| 3.3.3 Stemming  | 30        |
| 3.3.4 Term weighting  | 32        |
| 3.4 EVALUATIEMETHODEN   | 33        |
| 3.5 CASE-BASED REASONING VS. INFORMATION RETRIEVAL                | 35        |
| <b>4. ONDERZOEKSMETHODEN</b>                                      | <b>39</b> |
| 4.1 OPERATIONALISATIE   | 39        |
| 4.2 ONDERZOEKSOMGEVING  | 40        |
| 4.3 IMPLEMENTATIE   | 42        |

|  |           |
|--|-----------|
| 4.4 INDEXEREN  | 43        |
| 4.4.1 Term weighting   | 44        |
| 4.4.2 Stopwoorden  | 47        |
| 4.4.3 Stemmer  | 47        |
| 4.5 ZOEKEN   | 47        |
| 4.6 EXPERIMENTELE OPZET                                      | 49        |
| 4.7 EVALUATIEMETHODEN  | 51        |
| <br>   |           |
| <b>5. RESULTATEN</b>   | <b>55</b> |
| 5.1 OVEREENSTEMMING TUSSEN EXPERIMENTEN EN EXPERTS ONDERLING | 55        |
| 5.2 PERCENTAGES CORRECTE RESULTATEN                          | 56        |
| 5.3 MEAN RANK OF FIRST RESULT (MRR)                          | 57        |
| 5.4 FRIEDMAN   | 58        |
| <br>   |           |
| <b>6. CONCLUSIES EN AANBEVELINGEN</b>                        | <b>61</b> |
| <br>   |           |
| <b>7. DISCUSSIE</b>  | <b>65</b> |
| 7.1 EVALUATIEMETHODEN  | 66        |
| 7.2 VOORBEELDEN VAN TESTVRAGEN                               | 70        |
| 7.2.1 Slecht presterende vragen                              | 70        |
| 7.2.2 Goed presterende vragen                                | 74        |
| 7.3 VERDERE PUNTEN TER VERBETERING                           | 76        |
| <br>   |           |
| <b>BIJLAGE</b>   |           |
| <b>A. STOPWOORDENLIJST</b>                                   | <b>78</b> |
| <b>B. MEETRESULTATEN</b>                                     | <b>79</b> |
| <b>C. VOORBEELD OUTPUT</b>                                   | <b>80</b> |
| <b>D. BIBLIOGRAFIE</b>                                       | <b>82</b> |

# Voorwoord

Ik wil allereerst graag Hostnet bedanken. Hostnet heeft me de mogelijkheid geboden dit onderzoek in alle vrijheid uit te voeren en stond me, waar nodig, bij met raad en daad. Met name wil ik daarvoor Yom Schutte, Arjan Schaeffer, Jan Hilverda, Steven Mohammedajoeb, Rik van der Werff en Wing-Kai Poon bedanken. Tijdens de eerste weken van mijn stage realiseerde ik me al dat de mensen die werken bij Hostnet een unieke verzameling van jonge, intelligente en daadkrachtige mensen vormen. Ook mijn begeleider Wouter Jansweijer, die altijd een praktische, kritische en vooral objectieve noot wist te raken dank ik van harte voor al zijn inspanningen. Ook wil ik de mensen bedanken die het dichtst bij me staan. Ze hebben bijna een jaar lang mijn overwegingen moeten aanhoren en hebben al die tijd een luisterend oor geboden: José Meij, Jip Visser en Daniël Warmerdam.

Maanden van lezen, schrijven, herschrijven, redigeren, programmeren en evalueren hebben uiteindelijk geleid tot het werk dat nu voor u ligt. Deze scriptie is de voltooiing van mijn studie Sociaal Wetenschappelijke Informatica aan de Universiteit van Amsterdam en in zekere zin de kroon op mijn werk. In praktische en in theoretische zin heb ik, voor mijn gevoel althans, het afgelopen jaar het meest geleerd. Ik kijk desalniettemin met veel plezier terug op de afgelopen jaren en verheug me op de spannende tijd die komen gaat.



# 1 Inleiding

In de zich tegenwoordig steeds sterker ontwikkelende diensteneconomie zijn bereikbaarheid en ondersteuning cruciale succesfactoren voor een dienstverlenend bedrijf. De meeste klanten zien klantondersteuning als een van de belangrijkste criteria waarop ze producten en/of diensten beoordelen (Lenz, Burkhard et al. 1996). Niet alleen een tijdig, maar ook een kwalitatief goed antwoord op vragen van de klant zijn vereisten voor een positieve indruk. Deze scriptie is het resultaat van een onderzoek dat is uitgevoerd naar de haalbaarheid van een geautomatiseerde aanpak met betrekking tot het beantwoorden van vragen voor een webhosting provider. In het vervolg van dit hoofdstuk zullen de hoofdlijnen van het onderzoek en scriptie uiteengezet worden.

## 1.1 Achtergrond

De helpdesk van Hostnet, een middelgrote webhosting provider, ontvangt dagelijks gemiddeld 35 tot 50 e-mails en telefoontjes van klanten, die voornamelijk uit vragen, maar ook verzoeken bestaan. Het werkgebied waarin de helpdesk medewerkers dagelijks opereren is erg kennisintensief en het aantal mogelijke problemen groot en divers. Een gedeelte van de gestelde vragen is echter wel repetitief van aard. Oplossingen worden door de helpdesk medewerkers meestal 'uit het hoofd' gegeven. Van hen wordt verwacht dat ze parate kennis hebben van een ruim gedeelte van alle mogelijke problemen en dat ze creatief kunnen omgaan met mogelijke, nog niet eerder voorgekomen problemen. De kennisondersteunende mogelijkheden die daarbij aangeboden worden zijn beperkt. Hun dagelijkse werkzaamheden worden op dit moment uitsluitend ondersteund door een FAQ (Frequently Asked Questions) lijst, waar maar in beperkte mate gebruik van wordt gemaakt. De FAQ wordt daarnaast handmatig, ongestandaardiseerd en slechts op ad hoc basis bijgehouden. Indien blijkt dat een bepaalde vraag moeilijk te beantwoorden is geweest, wordt deze door de betreffende medewerker in de FAQ gedocumenteerd voor toekomstig gebruik. Tevens vindt er regelmatig (in)formeel overleg plaats tussen de medewerkers. In gevallen waar niet direct uit te komen valt, wordt een vraag doorgespeeld naar een tweede of derde lijn medewerker. Deze handelen de betreffende vraag vervolgens verder af.

Indien een organisatie als Hostnet alle mogelijke problemen zou kunnen expliciteren, vastleggen en methoden ontwikkeld om deze met succes te kunnen ontsluiten, zou deze combinatie in theorie een menselijke medewerker overbodig maken. De eerste voorwaarde is echter praktisch onmogelijk te realiseren, vanwege de constante, voortschrijdende ontwikkeling van nieuwe technologieën en progressie in de wetenschap en de dagelijkse praktijk. Het adequaat representeren en vastleggen van deze informatie zou tevens



resulteren in een systeem met groteske omvang. Slechts in zeer specialistische en afgebakende toepassingsgebieden zou dit een realistisch doel zijn.

Het zou echter al een substantiële vorm van ondersteuning voorstellen indien een systeem een gedeelte van alle mogelijke problemen terug kan vinden en de mogelijkheid zou hebben om nieuwe bij te leren. Een dergelijk systeem kan dan tevens fungeren als een vorm van collectief geheugen (corporate memory) voor een bedrijf of organisatie (Lenz, Auriol et al. 1998). Voor een bedrijf als Hostnet zou dit in eerste instantie een belangrijke vorm van (kennis)ondersteuning betekenen. Middels een geautomatiseerd systeem dat ondersteuning biedt bij question-answering (QA) kunnen vaker voorkomende problemen eerder en wellicht beter opgelost worden, ook door minder ervaren medewerkers.

Case-based reasoning (CBR) wordt inmiddels al meer dan tien jaar toegepast bij onder andere het (semi-)automatisch beantwoorden van vragen, beslissingsondersteuning en in geautomatiseerde processen als planning en allocatie. Het is een verzameling van technieken uit de Kunstmatige Intelligentie (AI), waarmee getracht wordt eerder opgeloste problemen (cases) te gebruiken als bron van kennis en informatie. Al sinds de eerste implementaties leek case-based reasoning een veelbelovende technologie voor bovenstaande processen en taken te zijn omdat zij ontwikkeld is om optimaal te werken in domeinen waar veel voorkomende problemen of situaties eerder regel dan uitzondering zijn. Blijkens het grote aantal case-based reasoning systemen dat tegenwoordig commercieel toegepast wordt is deze veronderstelling gerechtvaardigd. Het TOLTEC systeem lost bijvoorbeeld complexe fabricage problemen op door middel van case-based planning (Tsatsoulis and Kashyap 1993). Caper is een voorbeeld van een op case-based reasoning gebaseerd planning systeem dat de logistiek bij het fabriceren van auto's afhandelt (Kettler, Hendler et al. 1994). Deze systemen functioneren volledig autonoom en leren van eerder genomen beslissingen en uitkomsten. FALLQ is een case-based reasoning QA systeem dat gebruikt wordt door de helpdesk van een telecommunicatie-software bedrijf (Lenz, Hubner et al. 1998).

Zoals in de volgende hoofdstukken zal blijken, is een van de belangrijkste componenten in ieder case-based reasoning systeem de retrieval module. Deze zoekt, naar aanleiding van een nieuw probleem, naar eerder voorgekomen cases. Vervolgens worden aan de hand van een similariteitsmaat de meest gelijkende cases opgehaald. Het succes van een case-based reasoning systeem valt of staat uiteindelijk met de prestaties van deze module (Leake 1996). In het geval van een QA systeem wordt hiervoor de syntactische en/of semantische similariteit bepaald. Als syntactische basis kunnen bijvoorbeeld statistische modellen uit het toepassingsgebied van Information Retrieval (IR) worden gebruikt. Bij semantiek kan bijvoorbeeld gedacht worden aan domeinspecifieke ofwel grammaticale kennis.

## 1.2 Doel- en Vraagstelling

Dit onderzoek beoogt te analyseren in hoeverre de reeds opgeslagen vragen, die ooit gesteld zijn aan en beantwoord door de helpdesk van Hostnet, kunnen dienen als basis voor de retrieval module van een case-based reasoning QA systeem (opererend op syntactische basis) en wat voor invloed verscheidene information retrieval technieken hebben op de prestaties van een dergelijke module. Ondanks eerder onderzoek naar de ontwikkeling en implementatie van case-based reasoning systemen in velerlei kaders en toepassingsgebieden lijkt er tot op heden nog weinig bekend te zijn over de kwantitatieve retrieval resultaten van dergelijke systemen binnen dit specifieke domein. Meestal worden ze slechts aan de hand van kwalitatieve gebruikerservaringen beoordeeld (Leake 1996).

Binnen het onderzoeksgebied van information retrieval is er wel degelijk een lange en succesvolle geschiedenis van onderzoek voorhanden, waar in verregaande mate op gestoeld kan worden. Door inzicht te krijgen in hoeverre verschillende retrieval technieken invloed hebben op de resultaten van een syntactische retrieval module in een case-based reasoning context kan worden bijgedragen aan huidig onderzoek ten aanzien van case-based reasoning. Dit onderzoek beoogt daarnaast een eerste stap te zijn richting een eventuele ontwikkeling van een case-based reasoning systeem en/of corporate memory voor Hostnet. De beschreven omstandigheden zijn niet uniek voor Hostnet. Ieder dienstverlenend bedrijf dat direct contact heeft met zijn klanten is bekend met de eerder beschreven situatie en zou baat kunnen hebben bij de resultaten die voortkomen uit dit onderzoek.

De vraagstelling luidt: "Wat zijn de resultaten van een op statistische basis opererende retrieval module van een case-based reasoning QA systeem met en zonder gebruikmaking van verscheidene information retrieval technieken, toegepast op vragen gesteld aan de helpdesk van Hostnet". Om deze onderzoeksvraag te kunnen beantwoorden is het eerst noodzakelijk een overzicht te geven van relevante literatuur op het gebied van case-based reasoning en information retrieval. Vervolgens zal de onderzoeksvraag worden beantwoord door het uitvoeren van een tweetal experimenten.

## 1.3 Opbouw scriptie

**Hoofdstuk 2.** In het volgende hoofdstuk zullen de theoretische achtergronden van case-based reasoning uiteen worden gezet. Geldende opvattingen, definities en taxonomieën uit de literatuur zullen een theoretische basis vormen voor de achtergronden van dit onderzoek en deze scriptie. Ten slotte zullen enkele concrete voorbeelden van case-based reasoning systemen in een QA context gegeven worden.

**Hoofdstuk 3.** Een belangrijk gedeelte van een case-based reasoning toepassing is de retrieval module. Wetenschappers binnen het vakgebied van information retrieval houden zich al geruime tijd bezig met het terugvinden van informatie in het algemeen, en tekstuele informatie in het bijzonder. Actuele ontwikkelingen, geldende paradigma's, evaluatiemethoden en retrieval modellen en technieken zullen in dit hoofdstuk aan de orde komen. Voor het feitelijke onderzoek zullen er voornamelijk keuzes gemaakt worden op grond van de in dit hoofdstuk gepresenteerde informatie. Tevens zal op theoretische gronden een brug geslagen worden tussen information retrieval en case-based reasoning.

**Hoofdstuk 4.** De experimenten die uitgevoerd zijn om eerdergenoemde onderzoeksvraag te beantwoorden, worden in hoofdstuk 4 uiteengezet. De onderzoeksomgeving, implementatie, gebruikte data en evaluatiemethoden worden besproken, alsmede de operationalisatie van de onderzoeksvraag.

**Hoofdstuk 5.** De feitelijke resultaten van de experimenten zullen worden behandeld aan de hand van enkele evaluatiemethoden.

**Hoofdstuk 6.** De onderzoeksvraag zal hier, aan de hand van de resultaten, worden beantwoord. Ook zal er dieper worden ingegaan op de implicaties van deze resultaten en er zullen enkele punten ter verbetering worden aangemerkt.

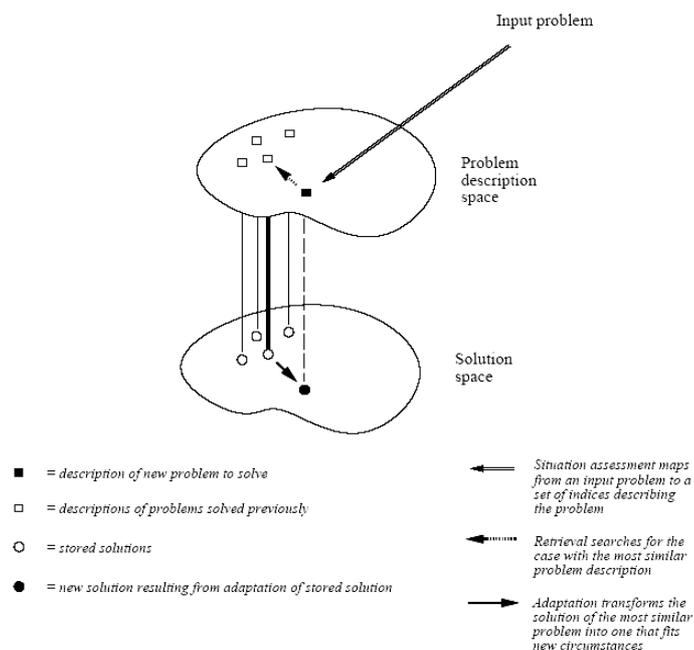
**Hoofdstuk 7.** Tot slot zullen er enige punten van dit onderzoek ter discussie worden gesteld, zoals methodologische keuzes en afwegingen met de bijbehorende gevolgen. Tevens zullen enkele concrete voorbeelden van vragen uit de experimenten worden behandeld.

## 2 Case-based reasoning

Case-based reasoning is een technologie, voortkomend uit het onderzoeksgebied van de Kunstmatige Intelligentie die wordt toegepast bij het genereren van probleemoplossingen en daarbij gebruik maakt van eerdere opgeloste problemen (cases). Deze aanpak is volgens (Aamodt and Plaza 1994) fundamenteel verschillend van andere technologieën uit de Kunstmatige Intelligentie, omdat er geen gebruik wordt gemaakt van algemene kennis en/of regels maar van specifieke informatie die besloten ligt in eerder opgeloste problemen. Het toepassingsgebied is omvangrijk, omdat er geen veronderstellingen gedaan worden over de problemen zelf, situatie, context of taak. Case-based reasoning heeft daarnaast geen expliciet kennismodel nodig om te functioneren.

### 2.1 Begrippen binnen case-based reasoning

Problemen en oplossingen kunnen vele gedaantes aannemen, maar het basisprincipe van case-based reasoning is telkens hetzelfde: zoek voor een bepaald, nieuw probleem (case) in de verzameling van eerder opgeloste cases en vindt daar de meest gelijkende terug. Vervolgens wordt de daarbij behorende oplossing als basis gebruikt om een oplossing te genereren (Lenz, Bartsch-Spörl et al. 1998). Case-based reasoning is daarom in feite te omschrijven als een technologie om een verzameling problemen te koppelen aan een verzameling oplossingen (Leake 1996). Onderstaande figuur geeft dit schematisch weer.



Figuur 2.1 De manier waarop case-based reasoning een nieuwe oplossing genereert, uit (Leake 1996).



Voorbijgaand aan definities wat een probleem of een oplossing precies is of hoe deze gerepresenteerd dienen te worden, kan gesteld worden dat problemen en oplossingen vrij algemene noties zijn. Bij een classificatie taak zijn het probleem en de oplossing duidelijk en afgebakend (een object en een klasse), net zoals bij een diagnose taak (een situatie en de diagnose). Een oplossing bij een beslissingsondersteunende taak is al minder eenduidig te definiëren. Een oplossing kan hierbij bestaan uit een handeling, plan of advies. De kracht van case-based reasoning schuilt erin dat het er niet toe doet welke taak voorhanden is of welke vorm een probleem dan wel oplossing aanneemt, het principe blijft gelijk. Andere toepassingen behelzen bijvoorbeeld ook het beoordelen van een door de gebruiker voorgestelde oplossing of het destilleren van een verwachte uitkomst uit een set waarnemingen (Lenz, Bartsch-Spörl et al. 1998).

Deze weidse notie van een probleem binnen case-based reasoning komt overeen met de definitie die toegepast wordt in andere kennisgebaseerde technologieën, zoals knowledge engineering (CommonKADS, (Schreiber, Akkermans et al. 2000)). Problemen zijn ook in CommonKADS algemene noties en kunnen velerlei specifieke vormen aannemen. Enkele voorbeelden zijn classificatie, diagnose en planning. Het CommonKADS raamwerk biedt echter uitsluitend handreikingen over het genereren van een kennisgebaseerde aanpak van het oplossen van deze problemen en is in die zin niet te vergelijken met case-based reasoning.

De overlap in theoretische benadering tussen knowledge engineering en case-based reasoning is wel aanwezig en velerlei auteurs pogen een brug tussen de twee te slaan. De auteurs in (Cunningham and Bonzano 1999) gebruiken bijvoorbeeld case-based reasoning als basis voor het inferentie model van een kennisgebaseerd systeem. Een meer theoretische aanpak van het vinden van een overeenstemming is terug te vinden in (Bergmann and Schaaf 2003). De auteurs betogen dat ontologiegebaseerde knowledge management en case-based reasoning theorieën gelijkgestemde methodologieën hanteren, maar verschillen in redeneerparadigma's. Voorts plaatsen ze een theoretisch kader om deze twee samen te laten smelten, wat helaas buiten het kader van dit onderzoek valt.

## 2.2 Historie

Redeneren door eerdere problemen te hergebruiken is volgens cognitief psychologisch onderzoek ook bij mensen een krachtige en veelgebruikte methode om met problemen om te gaan. Verschillende studies onderbouwen de rol van het hergebruiken van eerder beleefde situaties als mensen met nieuwe situaties of problemen in aanraking komen (Ross 1989). R.C. Schank introduceert bijvoorbeeld in (Schank 1982) de notie dat menselijk redeneren gebaseerd is op het verzamelen van ervaringen en cases in een zogenaamd dynamisch

geheugen. Dit geheugen is niet rigide georganiseerd en verbanden tussen eerdere ervaringen/cases zijn, op een onbewust niveau, gemakkelijk te leggen. Volgens Schank put de mens hier constant informatie uit om in de dagelijkse wereld te kunnen functioneren. Volgens deze theorie is leren en herinneren bij mensen gebaseerd op het opslaan van ervaringen in dit dynamische, evoluerende geheugen.

G. Lakoff beschrijft in (Lakoff 1987) eenzelfde notie van menselijk geheugen, alleen ligt de nadruk bij deze auteur meer op classificatie en categorisatie. Lakoff beziet de werking van menselijk redeneren als een vorm van doorlopende classificatie, gebaseerd op eerdere ervaringen. Volgens hem is de mens constant bezig met infereren van typische naar atypische situaties, ongeacht de omstandigheid (Lakoff 1987). De vorm van leren, zoals toegepast in case-based reasoning, is dus ook om cognitief-psychologische redenen zeer intuïtief en case-based reasoning systemen worden, naarmate ze meer cases kunnen bijleren en gebruiken, daadwerkelijk intelligenter. Deze inzichten vormden mede een theoretische basis voor het ontstaan van case-based reasoning.

Case-based reasoning is verder ontsproten uit het toepassingsgebied van machine learning. Bij onderzoek naar machine learning werd getracht middels de analyse van menselijk denken en redeneren deze te kunnen laten imiteren door een systeem. Door eerdergenoemde ontwikkelingen in de cognitieve psychologie werd case-based reasoning gezien als een levensvatbare aanpak van deze tak van onderzoek. Case-based reasoning staat van oorsprong echter haaks op eerdere rule-based (regelgebaseerde) approaches binnen machine learning. Hierin zijn alle mogelijkheden vooraf bepaald en afgebakend en het oplossen van problemen vindt plaats door middel van het invullen van variabelen. Deze systemen kunnen daardoor uitsluitend bijleren door handmatig nieuwe regels in te voeren. De regelgebaseerde oplossingen zijn daarom met name geschikt voor minder dynamische toepassingsgebieden.

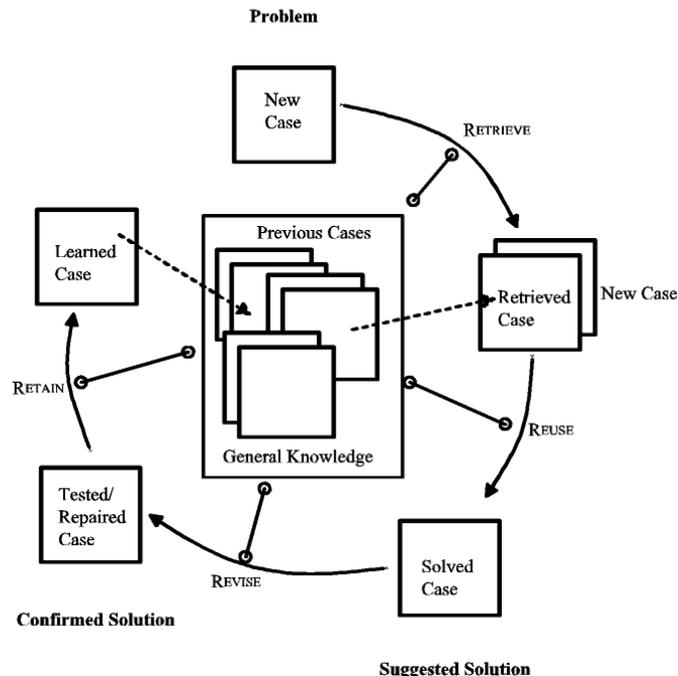
Het proces van het expliciteren van regels ten bate van een dergelijk regelgebaseerd systeem is gebaseerd op knowledge engineering technieken en veelal arbeidsintensief. Interviews met en observaties van experts vormen hierbij doorgaans de belangrijkste bron van informatie. Door deze manier van eliciteren is zij tevens gevoelig voor fouten. Als de geldende wetten of regels van een bepaald toepassingsgebied niet volkomen duidelijk zijn of niet volledig of juist geëliciteerd worden, kunnen er onvolledige regels of fouten ontstaan. Dergelijke problemen kennen case-based reasoning systemen in mindere mate en de oplossingen die met een case-based reasoning aanpak gegeneerd worden presteren dan ook vaak beter dan regelgebaseerde aanpak (Leake 1996). Eerdere cases vormen volgens Leake een betere afspiegeling van wat er daadwerkelijk is gebeurd (of juist niet is gebeurd). Een medisch voorbeeld hiervan is terug te vinden in (Hunter 1991), waarin specifieke medische gevallen de gecodeerde regels voorbij schieten. Deze worden door de auteur treffend "the as-

yet-unorganized evidence at the forefront of clinical medicine” genoemd. Omdat deze regels (nog) niet gecodeerd zijn, faalt het systeem.

Op dit punt kunnen de twee technologieën uit de Kunstmatige Intelligentie elkaar dus aanvullen en versterken. De structuur van een regelgebaseerd systeem kan profiteren van de dynamiek van een case-based reasoning aanpak en vice versa. Een dergelijke, hybride oplossing wordt gepresenteerd in (Gardie 1993). De auteur gebruikt afgebakende decision trees om de kenmerken te expliciteren, waarop in een latere fase een case-based reasoning systeem zoekt. Toegepast op een classificatie taak zijn de resultaten hiervan volgens de auteur, vergeleken met een case-based reasoning en een decision tree systeem apart, significant beter. Het nadeel blijft dat de gebruikte decision tree niet dynamisch uit te breiden is en daarmee het lerend vermogen op de lange termijn in zekere zin ontkracht.

### 2.3 Case-based reasoning als proces

De auteurs in (Aamodt and Plaza 1994) beschrijven een generiek procesmodel van een case-based reasoning systeem, die door vele andere auteurs erkend wordt als de grondslag van elk case-based reasoning systeem (Watson 1995; Lenz, Hubner et al. 1998). Hierin zijn vier concrete stappen te onderscheiden, namelijk retrieve, reuse, revise en retain. Zoals uit onderstaande figuur duidelijk wordt is het een cyclisch proces, dat voor elke nieuwe case herhaald wordt.



Figuur 2.2 Procesmodel CBR cyclus, naar [Aamodt en Plaza, 1994]

### 2.3.1 Retrieve

De retrieve stap vindt eerdere cases terug, op basis van bepaalde eigenschappen van het huidige probleem en een similariteitsmaat. Dit terugvinden kan op velerlei wijzen, maar in de literatuur wordt er onderscheid gemaakt tussen twee insteken; een kennisgebaseerde (semantische) en een kennisarme (syntactische) aanpak (Lenz, Hubner et al. 1998). Bij een semantische aanpak kan de kennis bijvoorbeeld vastgelegd zijn in een ontologie, semantisch net of ander kennismodel en bestaan uit regels en relaties tussen concepten. Syntactische systemen opereren veelal op statistische grondslag. Op semantiek gebaseerde case-based reasoning systemen kunnen echter, in tegenstelling tot syntactische systemen, ook redeneren met voorhanden zijnde informatie. Dit type vergt een grotere initiële investering vanwege de noodzaak een kennismodel te ontwikkelen (Lenz, Hubner et al. 1998).

Case-based reasoning systemen die uitsluitend gebruik maken van syntactische analyses zijn gemakkelijker te realiseren en daarom ook eerder geschikt voor commerciële doeleinden. Zonder de structuur van een kennismodel is er geen sprake meer van (geëxpliciteerde) kennis, maar wordt er in feite slechts informatie toegankelijk gemaakt. Hierdoor raken deze syntactische systemen aan traditionele information retrieval toepassingen. Information retrieval is een onderzoeks- en toepassingsgebied waarin het opslaan, terugzoeken en weergeven van informatie centraal staat. Informatie kan hierbij allerhande vormen aannemen, variërend van tekst en plaatjes, tot complete webpagina's met beeld en geluid (Witten, Moffat et al. 1999). Vooral binnen de tekstuele information retrieval is er een lange geschiedenis van onderzoek voorhanden, waar in het volgende hoofdstuk dieper op zal worden ingegaan.

Ook bij syntactische systemen kan er tijdens de retrieve stap in meer of mindere mate gebruik gemaakt worden van additionele achtergrondkennis; kennis die niet direct uit eerdere cases af te leiden is. Deze wordt in figuur 2 weergegeven met 'General knowledge'. Hierbij moet in het geval van een syntactisch systeem gedacht worden aan een synoniemenlijst of een lijst met afkortingen van bepaalde termen, specifiek voor het toepassingsdomein (Lenz, Burkhard et al. 1996). Dit mag echter niet gezien worden als een kennisgebaseerde aanpak, omdat de similariteitsbepaling nog steeds op statistische gronden geschiedt.

Voor sommige probleemoplossende taken zoals classificatie, kan tijdens de retrieve stap tevens de notie van correctheid van een oplossing geïntroduceerd worden. Deze notie impliceert ook een rangschikking op basis van mate van correctheid. In andere, minder streng afgebakende taken als beslissingsondersteuning zijn deze noties moeilijker te definiëren en is het problematisch een ordening aan te brengen in de gevonden cases. Wel kan er een rangschikking gemaakt worden op basis van andere attributen, zoals een verwachting van te maken kosten bij de gekozen oplossingen (Lenz, Bartsch-Spörl et al. 1998). Een dergelijke



rangschikking kan vervolgens worden gebruikt om een mate van relevantie weer te geven. Indien er geen rangschikking mogelijk is, zal hier tijdens de volgende stap rekening mee gehouden moeten worden.

### 2.3.2 Reuse, revise, retain

Tijdens de reuse stap wordt alle relevante informatie uit de teruggevonden case(s) en oplossing(en) geëxtraheerd en gecombineerd met het huidige probleem. Hieruit wordt een mogelijke nieuwe oplossing gevormd. Op welke wijze dit precies gebeurt, is afhankelijk van het toepassingsgebied en de input uit de vorige stap. Tijdens de revise stap wordt het voorgestelde antwoord uitgevoerd in de praktijk en gekeken of deze werkt. Middels feedback met de gebruiker wordt bepaald of er eventueel aanpassingen nodig zijn. De uiteindelijke, werkende oplossing wordt vervolgens opgeslagen tijdens de retain fase (Aamodt and Plaza 1994).

Normaliter voert het systeem al deze processtappen uit, en vindt de enige interactie met de gebruiker plaats tijdens de revise fase. In een minder geavanceerd case-based reasoning systeem zou de gebruiker de reuse en revise stap zelf kunnen uitvoeren. Het systeem draagt dan alleen zorg voor een correcte opslag, representatie en mogelijkheid tot terugvinden van de cases tijdens de retrieve en retain fases.

In (Göker and Roth-Berghofer 1999a; Göker and Roth-Berghofer 1999b) worden een aantal kanttekeningen bij dit procesmodel geplaatst, bekeken vanuit het perspectief van een praktische implementatie van een case-based reasoning systeem. Ten eerste moet volgens de auteurs de impact van verschillende typen eindgebruikers niet onderschat worden. Met name het onderlinge verschil in kennisniveau kan leiden tot aanzienlijke verschillen in kwaliteit van opgeslagen cases.

In een bedrijfsmatige omgeving dient er volgens dezelfde auteurs daarnaast ook rekening gehouden te worden met het verloop van de tijd. Oplossingen die ooit correct waren, hoeven na enige tijd niet meer relevant te zijn en dienen te worden verwijderd of aangepast. Dit zijn volgens de auteurs zaken waar het model uit (Aamodt and Plaza 1994) geen rekening mee houdt.

## 2.4 Case-based learning

Het leren en hergebruiken van eerdere oplossingen draagt bij aan het probleemoplossende vermogen van een case-based reasoning systeem doordat tijdens de retain processtap eerdere redeneerstappen in plaats van slechts oplossingen dienen te worden opgeslagen.

Hiermee kan bij nieuwe problemen een antwoord 'op maat' gegeven worden, iets wat kenmerkend is voor een case-based reasoning systeem (Aamodt and Plaza 1994).

Een traditioneel case-based reasoning systeem zou daarom ook niet uitsluitend succesvolle oplossingen moeten opslaan. Indien gefaalde oplossingen ook worden meegenomen, kunnen eventuele misstappen in de toekomst worden herkend en vermeden (Leake 1996). Binnen case-based reasoning zijn er volgens Leake twee manieren van leren te definiëren; success-driven learning en failure-driven learning. Een volledig case-based reasoning systeem zou beide vormen moeten beheersen. Indien een probleem succesvol opgelost wordt en als zodanig opgeslagen, zou deze in het systeem voorrang moeten krijgen boven minder of niet succesvolle oplossingen, indien de eigenschappen van het oorspronkelijke probleem gelijk zijn. Een niet succesvolle oplossing geeft voorts aan dat en waar er iets te leren valt.

Case-based reasoning systemen leren volgens Leake verder van zogenaamde task failures en expectation failures, zoals eerder gedefinieerd door (Schank 1982). Task failures treden op als de voorgestelde oplossing niet werkt en expectation failures als de verwachte uitkomst afwijkt van de daadwerkelijke uitkomst. Dit kan zowel positief als negatief zijn. Indien bijvoorbeeld een case-based planning systeem een plan voorstelt dat niet werkt zijn beide fouten aan de orde. De task failure zou het systeem aan moeten sporen een soortgelijke, succesvolle oplossing te vinden, terwijl de expectation failure het systeem zou moeten leren hoe in de toekomst soortgelijke fouten te herkennen en vermijden. Indien het plan uitermate succesvol was en additionele voordelen zoals tijdswinst opleverde, is er geen sprake van task failure, maar leert de expectation failure hoe deze voordelen in de toekomst opnieuw te bereiken (Hammond 1989). Deze noties van leren zijn binnen de context van een case-based reasoning systeem op theoretische gronden zinvol, maar in de praktijk erg moeilijk te realiseren. Om deze reden vallen ze buiten het bereik van dit onderzoek.

## 2.5 Taxonomieën van case-based reasoning systemen

In de literatuur wordt door verscheidene auteurs getracht een eenduidige definitie van een typisch case-based reasoning applicatie te geven. Zoals uit het eerste gedeelte van dit hoofdstuk blijkt, is dit slechts tot op zekere hoogte realiseerbaar. In de literatuur wordt tevens getracht de velerlei verschillende case-based reasoning systemen, doelen en toepassingen op enigerlei wijze te categoriseren. In deze paragraaf zullen een drietal van dergelijke indelingen besproken worden.

### 2.5.1 Textual, conversational and structural case-based reasoning

Case-based reasoning kan pas succesvol zijn als cases op correcte wijze gerepresenteerd worden. In (Bergmann, Breen et al. 1999) wordt een scheiding voorgesteld in de manier waarop de bronnen, materialen en kennis in een case-based reasoning systeem worden gerepresenteerd en verkregen. De auteurs onderscheiden drie varianten: textual, conversational and structural, die in (Bergmann and Schaaf 2003) theoretisch worden uitgediept.

Textual case-based reasoning is met name geschikt voor domeinen, waar kennis in grote hoeveelheden documenten en voornamelijk in vrije tekst vorm beschikbaar is. Deze aanpak werkt het best wanneer er per document een korte samenvatting bestaat van niet meer dan enkele zinnen, en er niet meer dan een paar honderd mogelijke cases teruggevonden hoeven te worden. Indien van een van deze voorwaarden afgeweken wordt, is volgens de auteurs de kans groot dat er veel irrelevante cases teruggevonden worden. Deze aanpak vergt, vanwege zijn ongestructureerde opzet, tevens een grote mate van inspanning om te onderhouden (Bergmann, Breen et al. 1999).

Conversational case-based reasoning heeft een meer gestructureerde vorm dan textual case-based reasoning en is met name geschikt voor toepassingsgebieden waar er een groot aantal relatief simpele problemen zijn die herhaaldelijk voorkomen. De basis is een dialoogvorm, waarbij er een aantal variabelen worden ingevuld aan de hand van vragen. De structuur en inhoud van deze vragen moet vooraf worden bepaald en bijgehouden door een domeinexpert en daardoor raakt deze aanpak aan de meer traditionele rule-based systemen. Er is echter geen kennismodel nodig; alle kennis wordt gehaald uit eerder opgeslagen dialogen.

Hierdoor kunnen echter ook ongewenste afwijkingen in de vragen ontstaan indien deze niet worden gecorrigeerd voor ze worden opgeslagen. Een systeem kan bijvoorbeeld tijdens een dialoog vragen naar elektrische problemen, terwijl het huidige probleem mechanisch van aard is (Bergmann, Breen et al. 1999). Het systeem dient dus te worden bijgehouden door een expert. Dit versterkt nogmaals de indruk dat dit type systeem een grote gelijkenis vertoont met een rule based expert systeem. Ook dergelijke systemen vereisen handmatige aanpassing aan nieuwe situaties. Het verschil is echter dat de inhoudelijke behandeling niet primair op regels en gangbare wetten is gebaseerd, maar op eerdere problemen.

Structural case-based reasoning gebruikt expliciete kennis uit een kennismodel van het betreffende domein. Naast de informatie die in eerdere problemen opgeslagen ligt, wordt ook additionele kennis over het domein gebruikt bij het redeneren. Deze aanpak levert de beste resultaten, maar het maken van een kennismodel vereist een aanzienlijke initiële inspanning. Het kennismodel zorgt er tevens voor dat de informatie uit nieuwe problemen van hoge

kwaliiteit is en houdt de kosten van onderhoud laag, omdat alle problemen in een consistente, gestructureerde manier worden opgeslagen en alle variabelen vastgestelde eigenschappen hebben (Bergmann, Breen et al. 1999).

## 2.5.2 Interpretative and problem-solving case-based reasoning

Door meerdere auteurs wordt er een tweedeling aangebracht in de algemene doelstelling van case-based reasoning systemen; zogenaamde interpretative en problem solving case-based reasoning (Kolodner 1992; Leake 1996). Het doel van eerstgenoemde is evalueren of een nieuw probleem overeenkomt met een eerder gesteld probleem, op grond van overeenkomsten en verschillen. Hierbij zijn er vier stappen te onderscheiden. De eerste stap die genomen moet worden is situation assessment (Kolodner 1992) om te bepalen welke kenmerken van het nieuwe probleem relevant zijn. Vervolgens worden er één of meerdere soortgelijke eerdere problemen teruggevonden, gebaseerd op de uitkomsten van de eerste stap. Als derde stap worden deze eerdere problemen vergeleken met het huidige probleem en wordt het huidige probleem geïnterpreteerd met behulp van deze informatie. Tenslotte wordt het huidige probleem en bijbehorende interpretatie opgeslagen ten bate van toekomstig gebruik (Leake 1996).

In problem solving case-based reasoning is het doel een oplossing te vinden voor een nieuw probleem, op basis van het aanpassen van oplossingen op eerder gestelde problemen. Hierbij worden dezelfde eerdergenoemde stappen doorlopen, maar vindt er vervolgens een adaptatie plaats. Hierin wordt gekeken hoe de oplossing(en) uit de eerdere problemen aangepast kunnen worden aan het huidige probleem.

Deze scheiding is wederom op theoretische gronden zinvol, maar in de praktijk niet altijd even eenduidig. De meeste, huidige case-based reasoning systemen vertonen namelijk beide karakteristieken. Interpretative systemen worden doorgaans gevonden in toepassingsgebieden waar redeneren, en niet zozeer het oplossen van concrete problemen het belangrijkste is. Een voorbeeld van een dergelijk systeem is te vinden in de rechtspraak, en dan met name HYPO (Ashley 1991). Dit systeem zoekt voor een nieuwe rechtzaak eerdere, soortgelijke rechtzaken, en bepaalt de voors en tegens van de bijbehorende uitspraken. Deze kunnen vervolgens worden ingezet als argumenten en tegenargumenten in de nieuwe rechtzaak.

Zoals duidelijk moge worden uit het voorgaande, levert case-based reasoning technologie een breed spectrum aan mogelijkheden om kennis uit eerdere problemen te organiseren, terug te vinden en/of te gebruiken. Er bestaat hierbij een grote variatie in de manier van representeren, het gebruik van domeinkennis en de manier van opslaan van nieuwe

problemen. Ook de mate van intelligentie betreffende het automatisch aanpassen van eerdere oplossingen aan het nieuwe probleem kan variëren.

### 2.5.3 Case-based reasoning als containerbegrip

Case-based reasoning is uitgegroeid tot een containerbegrip en een eenduidige definitie is daarom moeilijk te geven. Er kunnen volledig kennisgebaseerde systemen mee worden aangeduid, maar ook, indien er slechts sprake is van één gebruikte case-based reasoning techniek, kan een systeem al een case-based reasoning systeem genoemd worden. Om deze reden is er getracht in (Aamodt and Plaza 1994) om het kaf van het koren te scheiden. De auteurs onderscheiden een vijftal specifieke variaties van case-based reasoning, te weten:

- Exemplar-based reasoning

Dit komt voort uit de zogenaamde exemplar view, waarin concepten extensioneel worden beschreven. Het leren van concepten is een hoofdmoot uit het machine learning domein en de case-based reasoning systemen die hier gebruik van maken worden als exemplar-based aangemerkt. Hierbij is het oplossen van problemen gekarakteriseerd als een classificatieprobleem. Voor elk nieuw probleem wordt getracht een bijbehorende klasse te vinden, aan de hand van eerdere problemen. De gevonden klasse levert de oplossing op en hierdoor is de verzameling van klassen tevens de verzameling van oplossingen. Het aanpassen van eerdere oplossing aan de nieuwe situatie valt daarmee buiten het bereik van deze aanpak en de verzameling met mogelijke klassen/oplossingen kunnen niet dynamisch worden uitgebreid.

- Instance-based reasoning

Dit is een specialisatie van exemplar-based reasoning richting een meer syntactische aanpak. Indien er geen uitgebreide domeinkennis aanwezig is, zoals bij exemplar-based reasoning, zijn er vele voorbeelden nodig om tot een conceptdefinitie te komen. De nadruk ligt op automatisch leren, zonder tussenkomst van een gebruiker of expert. Hierdoor wordt er veelal gebruik gemaakt van een relatief simpele, kennisarme representaties van cases.

- Memory-based reasoning

Deze aanpak legt de nadruk op de verzameling cases als geheugen en redeneren als het terugzoeken hierin. Het onderscheid met de andere varianten ligt voornamelijk besloten in het gebruik van parallele processen om dit geheugen te benaderen, zoals een menselijk brein ook werkt. De gebruikte methoden kunnen op syntactische basis opereren of gebruik maken van domeinkennis.

- Case-based reasoning

Alhoewel case-based reasoning een generieke term is kan er ook een typische case-based reasoning aanpak gedefinieerd worden. Allereerst gebruikt elke case in deze aanpak kennis uit het domein bij de representatie ervan en wordt hierin een zekere complexiteit veronderstelt. De belangrijkste karakteriserende eigenschap van een case-based reasoning aanpak is echter dat een dergelijk systeem in staat is een gevonden eerdere oplossing aan te passen aan de nieuwe situatie.

- Analogy-based reasoning

Dit wordt veelal gebruikt als synoniem voor de hiervoor genoemde case-based reasoning aanpak. Het wezenlijke verschil is echter dat de typische case-based reasoning aanpak gebruik maakt van cases uit hetzelfde domein en analogy-based reasoning ook kan redeneren tussen domeinen. De hoofdmoot van onderzoek bestaat uit het op automatische wijze vinden van overeenkomstige relaties tussen concepten binnen verschillende toepassingsgebieden. Hiermee wordt getracht algemenere, abstractere kennis te vergaren over de structuur van concepten en hun omgeving (Leake 1996).

## 2.6 Voorbeelden van case-based reasoning applicaties

Case-based reasoning systemen behalen, zoals eerder al aangehaald, betere resultaten dan model- of regelgebaseerde systemen. Met name in toepassingsgebieden waar praktijkervaring belangrijker is dan theorie bij het oplossen van problemen, presteren case-based reasoning systemen beter dan regelgebaseerde varianten (Allen 1994). Het oplossen van problemen in de QA context van een klantenservice of helpdesk is een goed voorbeeld van een dergelijke toepassing. Met name bij hightech bedrijven, waarbij de ontwikkeling van nieuwe technologieën constant doorgaat, is het opzetten en onderhouden van een traditioneel kennis-/regelgebaseerd expertsysteem vaak te kostbaar en onpraktisch (Simoudis 1992). In plaats daarvan leent case-based reasoning zich dus bij uitstek voor toepassing binnen deze context en dat heeft geresulteerd in een aantal al dan niet commerciële toepassingen. Een aantal voorbeelden van dergelijke systemen worden in het vervolg van deze paragraaf gegeven. Helaas zijn er slechts bij enkele van deze systemen concrete resultaten publiekelijk beschikbaar gemaakt.

### 2.6.1 FAQFINDER

FAQFINDER wordt beschreven in (Lenz, Hubner et al. 1998). In (Burke, Hammond et al. 1997) wordt ingegaan op de technische achtergronden en resultaten. Het is een systeem dat tracht antwoorden te vinden op vragen die gesteld worden in natuurlijke taal. De antwoorden worden gehaald uit berichten (postings) die verzameld worden in een online nieuwsgroepen



community (USENET). Voor bijna elk denkbaar onderwerp is er hierin een nieuwsgroep te vinden. De berichten kunnen door iedereen geplaatst, gelezen en beantwoord worden. In een zeer beperkt aantal nieuwsgroepen worden de geplaatste berichten eerst bewerkt door een of meerdere zogenaamde moderators. Dit is echter eerder uitzondering dan regel. Vanwege de diverse aard van deze nieuwsgroepen en postings, worden de vragen die gesteld kunnen worden aan het systeem niet beperkt tot één domein.

De werking van het systeem is gebaseerd op twee fasen. Allereerst worden met een syntactische information retrieval module (SMART, (Buckley 1985; Salton and McGill 1997)) de meest waarschijnlijke nieuwsgroepen behorende bij de gestelde vraag gezocht, waar de gebruiker de beste match(es) uit kiest. FAQFINDER gebruikt hierop volgend een module om in de door de gebruiker geselecteerde nieuwsgroep(en) overeenkomstige vragen te vinden, deels gebaseerd op de grammaticale functies van WordNet. WordNet is een ontologie van engelse termen en hun onderlinge relaties, waaronder synoniemen, hyper-/hyponiemen en meroniemen (Fellbaum 1998). Dergelijke ontologieën kunnen worden gezien als een netwerk van concepten, gekoppeld door onderlinge relaties. Hiermee is het mogelijk zogenaamde 'shallow lexical semantics' te gebruiken, door de afstand in dit netwerk tussen twee termen te bepalen (Burke, Hammond et al. 1997). Deze afstand is het aantal stappen via de onderlinge relaties tussen termen, dat nodig is om van één term bij een andere te komen. Indien woorden een korte afstand tot elkaar hebben in dit netwerk zijn ze hoogstwaarschijnlijk semantisch verwant.

Middels deze methode kan een relevantiegraad voor de gestelde vraag ten opzichte van ieder van de opgeslagen vragen worden bepaald die evenredig is met deze afstand. Tevens tracht FAQFINDER de grammaticale structuur van een zin te ontleden en kunnen de gebruikte termen onderverdeeld worden in bijvoorbeeld zelfstandige naamwoorden, werkwoorden, etcetera. Hiermee verbeteren volgens de auteurs ook de resultaten van het gebruik van WordNet. Gelijke termen kunnen namelijk een andere betekenis hebben in een andere grammaticale variant.

Er wordt getracht domeinspecifieke woorden (die niet in WordNet voorkomen) automatisch uit berichten in de nieuwsgroepen te extraheren en te interpreteren. Dit is echter tot op zeer beperkte hoogte mogelijk, grotendeels vanwege het inherent ongestructureerde formaat van de berichten. (Lenz, Hubner et al. 1998). De module levert per vraag naast de eerder genoemde semantische score tevens een syntactische score (TF.IDF). TF.IDF is een statistische wegingmethode uit het onderzoeksgebied van information retrieval, gebaseerd op de frequentie waarin een bepaalde term voorkomt (*term weighting*). In het volgende hoofdstuk zal hier dieper op worden ingegaan. Nieuwe vragen worden automatisch geïndexeerd door een applicatie genaamd FAQGRINDER. Hiermee is de combinatie van deze systemen in staat autonoom te opereren en nieuwe cases bij te leren.

Het ontbreken van kennis over de diepere semantische betekenis van woorden en hun relaties is volgens (Burke, Hammond et al. 1997) een manco bij dit systeem. Ze vermoeden dat de resultaten van het systeem zullen verbeteren met een uitgebreidere semantische structuur dan beschikbaar is in WordNet. Omdat FAQFINDER bedoeld is om vragen uit iedere willekeurige nieuwsgroep te kunnen vinden, is het toevoegen van domeinspecifieke kennis per nieuwsgroep echter in de praktijk onhaalbaar.

Uit de resultaten van FAQFINDER komt naar voren dat de combinatie van syntactische en semantische analyse leidt tot een verbetering van de kwaliteit van de resultaten, vergeleken met de resultaten van de los toegepaste methoden (Burke, Hammond et al. 1997).

### 2.6.2 CASCADE

CASCADE is een case-based reasoning systeem ter ondersteuning van de helpdesk van Digital Equipment Corporation dat specifiek geënt is op het oplossen van problemen bij crashes van besturingsystemen, ontwikkeld door dit bedrijf zelf. Het gebruikt een zogenaamd validated retrieval algoritme om eerdere cases terug te vinden (Simoudis 1992). Dit algoritme bestaat uit twee stappen. Het zoekt eerst op syntactische basis naar overeenkomstige cases. Vervolgens gaat het systeem uit van kenmerken van de gevonden resultaten om nadere informatie over het huidige probleem te verkrijgen en zo het probleem in te sluiten.

Het onderscheid wat gemaakt wordt tussen de stappen is dat de eerste, syntactische stap gebruik maakt van relatief gemakkelijk te achterhalen informatie, terwijl de tweede fase complexere informatie gebruikt. Laatstgenoemde informatie kan echter pas gevonden worden door middel van relatief kostbare methoden. Omdat er eerst een aantal mogelijke cases worden gevonden waarop deze methoden worden toegepast blijft de netto overhead per zoekactie beperkt.

De informatie die het syntactische algoritme gebruikt is gebaseerd op rapporten die helpdesk medewerkers geschreven hebben, gecombineerd met interviews en bestaat voornamelijk uit een aantal zogenaamde if then regels. Hierdoor is dit systeem niet dynamisch en zullen nieuwe regels en cases handmatig in het systeem ingevoerd moeten worden.

Dit validated retrieval algoritme werkt volgens de auteur het best binnen een domein waarin het aantal gemakkelijk te achterhalen eigenschappen groot is en waarin zij ook een voldoende mate van onderscheidend vermogen hebben. Tevens moeten er complexere eigenschappen van cases bestaan, die als input dienen voor de tweede fase (Simoudis 1992). Volgens de auteur zou een traditionelere aanpak zonder dit algoritme binnen een dergelijk domein te kostbaar zijn vanwege de complexe aard van de problemen. In dat geval zou er namelijk per case uitgebreid gekeken moeten worden naar een groot aantal complexe

eigenschappen. Dit is een kostbare en tijdrovende activiteit en kan met behulp van dit algoritme aanzienlijk beperkt worden.

De resultaten van dit systeem zijn erg goed, 100% van alle mogelijke cases worden teruggevonden (Simoudis 1992). De kanttekening die hierbij geplaatst moet worden is dat het domein erg beperkt is, het aantal mogelijke problemen klein en het gebruikte algoritme veel kenmerken heeft van een regelgebaseerd kennissysteem.

### 2.6.3 HOMER

HOMER is een case-based reasoning systeem, bedoeld om de interne ICT helpdesk van DaimlerChrysler te ondersteunen bij de dagelijkse taken. In (Göker and Roth-Berghofer 1999a) wordt het globale systeem en de impact van het systeem op de organisatie beschreven en in (Göker and Roth-Berghofer 1999b) wordt de technische implementatie verder uitgewerkt. Het maakt gebruik van een objectgeoriënteerd kennismodel van het domein, wat grotendeels door interviews met de helpdesk medewerkers is verkregen. Dit model is top-down gegenereerd.

In eerste instantie dachten de auteurs dat het mogelijk zou zijn om bottom-up alle eerder gestelde vragen, opgeslagen in vrije tekst vorm, te kunnen gebruiken als kennisbron. Dit bleek volgens de auteurs echter onhaalbaar vanwege het ontbreken van additionele informatie en kennisstructuren in de representatie van deze vragen. (Göker and Roth-Berghofer 1999a).

Het systeem kent twee modi waarin het opereert. In de zogenaamde system-driven mode stelt het systeem vragen aan de gebruiker aan de hand van eerder opgeslagen cases om zo het probleem in sluiten, terwijl in de user-driven mode de gebruiker zelf door de verschillende categorieën kan bladeren om tot een match te komen. Hiermee valt HOMER zowel binnen conversational als structural case-based reasoning.

Alle nieuwe cases worden tijdelijk opgeslagen in een case-buffer. Een menselijke *case editor* leest en beoordeelt deze, waarna ze al dan niet bewerkt worden opgeslagen in de *case (data)base*. Hiermee wordt tevens getracht redundantie te voorkomen en kwaliteit te waarborgen, maar is het systeem tegelijkertijd niet volledig autonoom.

Van HOMER zijn helaas geen concrete resultaten beschikbaar gemaakt. De auteurs in bovengenoemde artikelen geven slechts aan dat het systeem naar behoren werkt. Hierbij baseren ze zich uitsluitend op kwalitatieve gebruikerservaringen.

#### 2.6.4 FALLQ

FALLQ is een systeem gebaseerd op het CBR-ANSWERS project (Lenz, Hubner et al. 1998). In CBR-ANSWERS wordt tijdens het terugzoeken van cases gebruik gemaakt van zogenaamde information entities (IEs). Dit zijn domeinspecifieke termen (keywords) die gekoppeld worden aan synoniemen, afkortingen en alternatieve schrijfwijzen. Zij worden opgeslagen in een ontologie, door de auteurs semantisch netwerk ofwel case retrieval net genoemd. Tijdens het zoeken wordt uitsluitend gezocht op deze, vooraf gedefinieerde IEs. Er is echter geen afstand tussen termen in dit case retrieval net gedefinieerd zoals in FAQFINDER. Het netwerk is slechts een mapping van domeinspecifieke termen op meer generieke begrippen of synoniemen. Hiermee kunnen dus syno-, hypo- en hyperniem relaties worden achterhaald, wat volgens de auteurs de effectiviteit van het systeem ten goede komt.

FALLQ is uit dit onderzoeksproject ontwikkeld om de administratie en de helpdesk van een telecommunicatie-software bedrijf te ondersteunen. Het gebruikt als basis de eerder gestelde vragen en antwoorden aan en van de helpdesk, opgeslagen in een database in vrije tekst vorm. De IEs zijn vooraf gedefinieerd aan de hand van technische rapporten, documentaties, handleidingen en FAQ bestanden. Naast de genoemde kenmerken van CBR-ANSWERS wordt in FALLQ ook een additionele semantische analyse in de vorm van een beperkte natural language processing (NLP) module gebruikt en is getracht het bepalen van nieuwe IEs semi-automatisch te laten verlopen. De NLP module hanteert ongeveer dezelfde 'shallow lexical semantics' als semantische module uit FAQFINDER. Hiermee kan op dezelfde wijze de grammaticale structuur van een zin en de woordsoort van de gebruikte termen bepaald worden. Het bepalen van nieuwe IEs gebeurt op basis van statistische methoden; nieuwe termen worden geëxtraheerd en er wordt getracht deze automatisch te koppelen aan synoniemen of hypo-/hyperniemen (Lenz, Burkhard et al. 1996).

Van FALLQ zijn helaas geen concrete resultaten openbaar gemaakt. Binnen het CBR-ANSWERS project is er wel een zogenaamde ablation study uitgevoerd, waarin het effect van de IEs is bepaald. Volgens de auteurs zijn de resultaten significant beter wanneer het systeem gebruik maakt van de IEs (Lenz, Hubner et al. 1998).

#### 2.6.5 Conclusie

Aan de hand van deze beschrijvingen zijn een aantal dimensies te onderscheiden waarop de systemen ingedeeld kunnen worden, te weten: onderhoudbaarheid, geschiktheid voor queries in natuurlijke taal, of het algoritme primair op syntactische (statistische) dan wel semantische basis opereert en het gebruikte bronmateriaal. In tabel 2.1 worden deze samengevat. FALLQ blijkt op grond hiervan het meest overeen te komen met de doelstelling en omstandigheden van dit onderzoek.



|           | ONDERHOUDBAARHEID | NATUURLIJKE TAAL | BRONMATERIAAL   | ALGORITME                             |
|-----------|-------------------|------------------|-----------------|---------------------------------------|
| FAQFINDER | automatisch       | ja               | USENET FAQ's    | statistisch,<br>semantisch* (WordNet) |
| CASCADE   | handmatig         | nee              | rapporten       | logisch                               |
| HOMER     | handmatig         | ja               | kennismodel     | semantisch                            |
| FALLQ     | automatisch       | ja               | tekstuele cases | statistisch<br>semantisch* (NLP, IEs) |

\* FAQFINDER en FALLQ gebruiken beide slechts een beperkte vorm van semantische analyse

**Tabel 2.1 Dimensies binnen case-based reasoning toepassingen**

## 3 Information Retrieval

Het raakvlak van case-based reasoning met information retrieval werd in de vorige hoofdstukken al aangehaald. De belangrijkste processtap in het generieke procesmodel van case-based reasoning, waar uiteindelijk het gehele case-based reasoning proces mee valt of staat, is de retrieve stap, waarin relevante cases worden teruggevonden. Information retrieval is van oorsprong een techniek die gebruikt wordt als men relevante tekstuele documenten wil terugvinden op een gegeven vraag.

### 3.1 Historie

In tegenstelling tot wat de naam doet vermoeden, kunnen information retrieval systemen volgens (Rijsbergen 1979) er niet voor zorgen dat de zogenaamde information need van een gebruiker volledig wordt bevredigd. De information need is het antwoord op een probleem of vraag van een gebruiker van een dergelijk systeem. Een klassiek information retrieval systeem kan, volgens Rijsbergen, de gebruiker slechts informeren over het bestaan van een aantal documenten of feiten, in antwoord op een gegeven vraag (query) aan het systeem. De gebruiker dient vervolgens zelf de gevonden resultaten door te nemen om de information need te bevredigen. De gestelde query is in de meeste gevallen een vertaling van de information need in een aantal sleutelwoorden (keywords) en is om die reden ook armer aan informatie.

Deze vertaalslag aan de kant van de gebruiker wordt in (Baeza-Yates and Ribeiro-Neto 1999) en in (Rijsbergen 1979) aangemerkt als het verschil tussen data retrieval en information retrieval. Data retrieval is het terugvinden van objecten, die voldoen aan strikte regels en voorwaarden, zoals zoeken op keywords in een relationele database. Databases hanteren strikte (programmeer)talen om gegevens terug te vinden, zogenaamde query languages. SQL (Structured Query Language) is hier een welbekend voorbeeld van. Deze talen kennen slechts een beperkt aantal operatoren en zijn om die reden semantisch arm.

Volgens laatstgenoemde auteurs kan information retrieval, vergeleken met data retrieval, dus nader gekenmerkt worden door een semantisch rijkere doch abstractere representatie van queries, waar bij voorkeur natuurlijke taal voor gebruikt wordt. Door deze abstractere vorm van representeren is het ook mogelijk om bij toepassing van information retrieval een mate van relevantie aan te geven in de gevonden resultaten, bijvoorbeeld op syntactische en/of semantische gronden. Dit is bij data retrieval niet mogelijk, een object beantwoordt simpelweg wel of niet aan een gegeven query. Data retrieval werd van oorsprong beschouwd als information retrieval, maar naarmate de rekenkracht van computers toenam en ook de

gebruikte technieken en algoritmes evolueerden ontstond de noodzaak een verschil in definities aan te brengen. Klassieke information retrieval wordt daarom in de hedendaagse context beschouwd als data retrieval.

Deze verschuiving in definities is ook terug te vinden in de verbreding van toepassingsgebieden van information retrieval. Information retrieval werd in eerste instantie uitsluitend ingezet in gespecialiseerde omgevingen waar al een lange geschiedenis van indexerende voorhanden was, zoals in bibliotheken. Mede door de groei van het internet en ICT in het algemeen is ook het toepassingsgebied verruimd (Baeza-Yates and Ribeiro-Neto 1999) en kan information retrieval tegenwoordig op velerlei terreinen en voor velerlei doeleinden ingezet worden. Ook verschuift de aandacht naar andere vormen van informatie. Zowel beeld als geluid vinden bijvoorbeeld hun plaats binnen het onderzoeksgebied van de multimedia retrieval (Witten, Moffat et al. 1999).

Door deze ontwikkelingen is er inmiddels een brede en solide basis van onderzoek binnen het domein van information retrieval voorhanden, met name binnen het tekstuele subdomein. Huidige, veelbelovende onderzoeksrichtingen van tekstuele information retrieval behelzen onder meer de toepassing van eerdergenoemde NLP technieken (Croft 2000; Chowdhury 2004) en daarmee een steeds dieper begrip van betekenis van zinnen, teksten en documenten door geautomatiseerde systemen. De gedachte hierachter is dat het steeds meer van belang wordt om volledige teksten betekenisvol te doorzoeken, in plaats van slechts een aantal keywords (Jones and Willett 1997a).

Met name de toepassing van dergelijke technieken in een context van het automatisch beantwoorden van vragen lijkt veelbelovend (Monz and Rijke 2001). De toepassing van hedendaagse natural language processing technieken ten bate van information retrieval staat echter nog in de kinderschoenen, blijkt (Allan 2000). Volgens de auteur blijkt dat fouten in moderne natural language processing technieken de resultaten van een information retrieval systeem meer schaden dan dat deze ze verbeteren. Daarnaast is natural language processing veelal domeinspecifiek en is volgens de auteur een meer generieke oplossing noodzakelijk om een effectieve bijdrage te kunnen leveren aan information retrieval.

## 3.2 Modellen

Vooralsnog is het onmogelijk voor information retrieval systemen om gehele documenten te doorzoeken en op grond daarvan te bepalen of een het onderwerp van een document overeenkomt met een gestelde vraag (Maron and Kuhns 1997). Dit doel wordt steeds dichter benaderd, mede door de ontwikkeling van NLP technieken en een toenemend gebruik van semantische kennis bij retrieval. Ook de toename in het gebruik van semantische *mark-up*

*languages* zoals SGML en XML dragen bij aan deze ontwikkeling. Tot dit zover is zullen statistische retrieval methoden de dienst uitmaken. In deze paragraaf worden enkele van de meest invloedrijke, statistische retrieval modellen behandeld.

### 3.2.1 Boolean model

Klassieke information retrieval systemen gebruiken indexeerbare termen (keywords), waarop elk document doorzocht en geïndexeerd wordt. Deze termen worden a-priori bepaald en er kan slechts gezocht worden door de verzameling van geïndexeerde documenten aan de hand van deze termen. Het feitelijke indexeren van de documenten aan de hand van keywords gebeurde in eerste instantie handmatig, maar vindt tegenwoordig automatisch plaats. Ook de bepaling van relevante keywords kan automatisch geschieden, op grond van statistische methoden, zoals de bepaling van termfrequenties.

Deze aanpak is settheoretisch van aard en wordt in de literatuur, zoals in (Jones and Willett 1997b; Baeza-Yates and Ribeiro-Neto 1999; Chowdhury 2004), veelal aangeduid als het boolean model, vanwege de boolean logica die gebruikt wordt in het formuleren van queries. Deze vorm van logica (en daarmee dit model) gebruikt als basis een drietal operatoren, namelijk AND, OR en NOT. De similariteit tussen twee documenten of tussen een document en een query wordt bepaald op grond van het al dan niet voorkomen van gegeven keywords, verbonden door deze operatoren.

Op grond van dit model is het echter niet mogelijk een rangschikking en daarmee een mate van relevantie aan te duiden, omdat ze een binaire grondslag heeft. Een document is hierin wel of niet relevant op basis van de aan- of afwezigheid van keywords, zonder tussenliggende mogelijkheden (Jones and Willett 1997c). De enige rangschikking die aangebracht kan worden is het aantal keywords uit de query dat terugkomt in de documenten en de eventuele frequentie daarvan (William 1997). Ten grondslag aan dit model ligt tevens de fundamentele veronderstelling dat de betekenis en inhoud van een willekeurig document accuraat samengevat kan worden middels een verzameling keywords. Dit lijkt een te gesimplificeerde aanname, omdat er een groot deel van de semantiek van een document verloren gaat bij het definiëren van dergelijke keywords (Baeza-Yates and Ribeiro-Neto 1999).

Het boolean model is slechts één voorbeeld van een mogelijk model om information retrieval systemen mee te karakteriseren. (Jones and Willett 1997c; Baeza-Yates and Ribeiro-Neto 1999; Chowdhury 2004) onderscheiden allen naast het boolean model verder nog het probabilistische en het vector space model. Ieder van deze modellen hanteert zijn eigen theoretische aannames en similariteitsmaten en zullen in de volgende paragrafen behandeld worden.



### 3.2.2 Probabilistisch model

In dit model wordt uitgegaan van Bayesiaanse principes. Het retrieval systeem INQUERY maakt bijvoorbeeld gebruik van dergelijke, probabilistische methoden (Callan, Croft et al. 1997). Het wiskundige element dat gebruikt wordt om relevante van de niet relevante documenten te scheiden is een vergelijkingsfunctie, of eigenlijk waarschijnlijkheidsfunctie. Deze functie benadert en kwantificeert de waarschijnlijkheid, dat bepaalde keywords relevant zijn voor een bepaald document en daarmee dat een document relevant is voor een gegeven query.

Er bestaan een aantal verschillende manieren om tot deze maat van waarschijnlijkheid te komen. De auteurs in (Robertson and Jones 1976; Jones 1979; Jones 1997) gebruiken hiervoor historische gronden, 'indien een gebruiker keyword  $x$  in een query opnam, vond diegene document  $y$  relevant'. Per mogelijk keyword (en combinaties daarvan) en document wordt zo een waarschijnlijkheidsfunctie bepaald. Het verzamelen van deze feedback informatie veronderstelt een aanzienlijke inspanning, omdat gebruikers per zoekopdracht aan moeten geven of documenten relevant zijn. Indien zulke informatie opgeslagen en gebruikt kan worden, werkt de probabilistische methode echter uiterst effectief.

Door dergelijke, zogenaamde *relevance feedback* van gebruikers kunnen er tevens in eenzelfde zoekactie nieuwe documenten gevonden worden, die met de oorspronkelijke query misschien lager gewaardeerd werden. Er worden in die situatie namelijk nieuwe keywords uit de door de gebruiker als relevant bestempelde documenten gehaald, die als nieuwe, aanvullende input dienen. Deze uitbreiding van de query wordt door verscheidene auteurs als een vorm van query expansion gezien, die tevens toegepast kan worden in andere information retrieval modellen (Baeza-Yates and Ribeiro-Neto 1999; Chowdhury 2004).

In (Jones 1979) gaat de auteur dieper in op de initiële situatie waarin *relevance feedback* informatie niet beschikbaar is. Indien zulke informatie niet voorhanden is, kan er volgens de auteur teruggerepen worden op statistische methoden, zoals de bepaling van termfrequentie. De waarden in de waarschijnlijkheidsfunctie worden dan bepaald door de verdeling van indexeerbare termen in de gehele set van documenten (Rijsbergen 1979). Deze laatste aanpak komt overeen met de manier waarop de auteurs in (Salton and McGill 1983) normaliter de waarden voor de waarschijnlijkheidsfunctie bepalen. Zij gebruiken dus überhaupt geen *relevance feedback*.

Volgens (Robertson 1997) kan deze methode echter uitsluitend worden toegepast indien de relevantie van één document op een gegeven query, onafhankelijk is van de andere documenten in de verzameling. Dit is volgens Robertson inherent aan de eigenschappen van de Bayesiaanse principes, maar ook aan de eigenschappen van de set van documenten.

Deze onafhankelijkheid kan dus uitsluitend gerealiseerd worden indien gewaarborgd wordt dat alle documenten over verschillende onderwerpen gaan en er geen overlap bestaat. In paragraaf 3.3.4 zal overigens nog dieper op de notie van termfrequentie worden ingegaan.

### 3.2.3 Vector space model

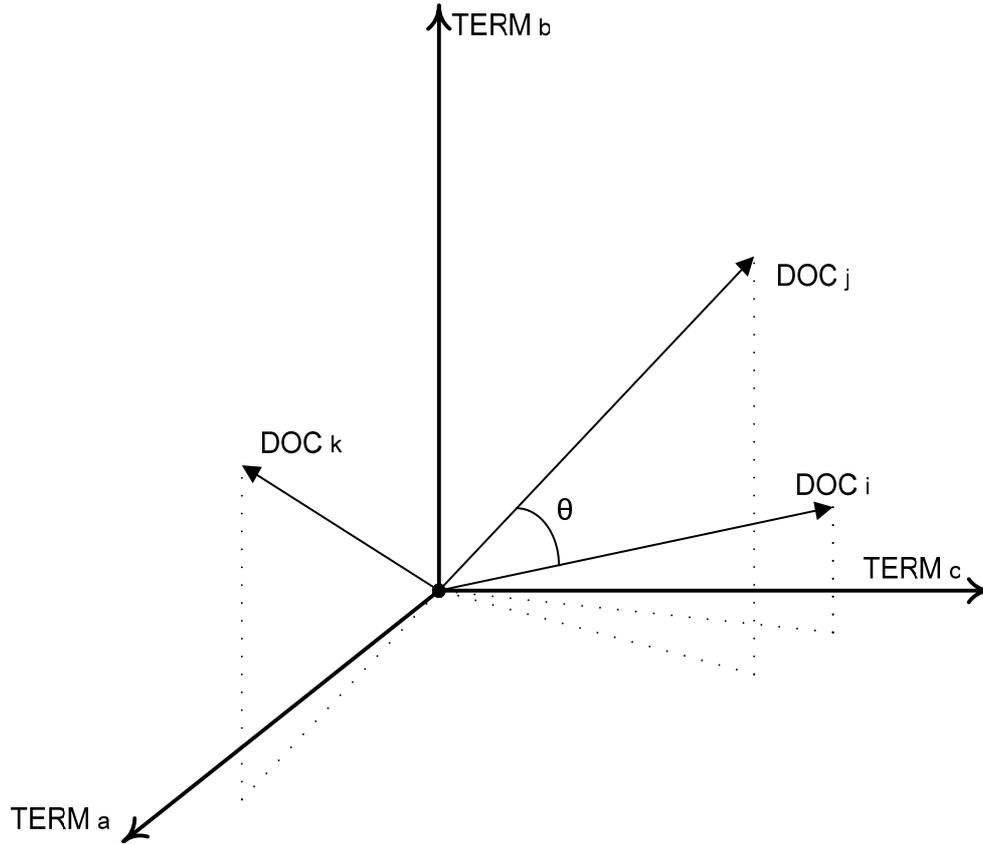
Het model dat, naast het boolean model, de meeste invloed heeft gehad binnen information retrieval is het vector space model (Jones and Willett 1997c). Het al eerder genoemde SMART information retrieval systeem hanteert bijvoorbeeld dit model (Salton and Lesk 1968). Het principiële idee achter het vector space model is dat alle indexeerbare termen kunnen worden gezien als dimensies in een meerdimensionale ruimte. Voor ieder document en query kan met coördinaten langs deze dimensies een vector in een meerdimensionale ruimte worden bepaald.

Een document wordt dan gerepresenteerd als  $DOC_i = [TERM_{i1}, TERM_{i2}, TERM_{i3}, TERM_{i4} \dots TERM_{it}]$ , uitgaande van  $t$  unieke termdimensies (Chowdhury 2004). De vectoren kunnen binair van aard zijn; de lengte van de vector langs elke termdimensie is dan maximaal 1. Hierbij wordt er geen rekening gehouden met de frequentie waarin een term in een document voorkomt, slechts of een term voorkomt. Normaliter wordt er echter wel degelijk rekening gehouden met de frequentie waarin een term in een document voorkomt (Jones and Willett 1997c; Baeza-Yates and Ribeiro-Neto 1999; Chowdhury 2004). De lengte van de vector langs elke termdimensie is dan gelijk aan het aantal maal dat de betreffende term voorkomt in het document, al dan niet genormaliseerd tot een waarde van 0 tot 1.

Beschouw als voorbeeld drie documenten ( $DOC_i$ ,  $DOC_j$  en  $DOC_k$ ) die elk  $t$  verschillende termen bevatten, in wisselende frequenties. Elk document kan gerepresenteerd worden door hun respectievelijke vector:

$$\begin{aligned} DOC_i &= [TERM_{i1}, TERM_{i2}, TERM_{i3}, TERM_{i4} \dots TERM_{it}] &= X \\ DOC_j &= [TERM_{j1}, TERM_{j2}, TERM_{j3}, TERM_{j4} \dots TERM_{jt}] &= Y \\ DOC_k &= [TERM_{k1}, TERM_{k2}, TERM_{k3}, TERM_{k4} \dots TERM_{kt}] &= Z \end{aligned}$$

In het geval van maximaal 3 verschillende termen kan dit visueel worden weergegeven door de figuur op de volgende pagina (3.1). Elke term kan tevens vermenigvuldigd worden met een factor. Indien deze factor binair van aard is, geeft zij slechts aan of de betreffende term voorkomt in het document. In de overige gevallen is zij afhankelijk van een wegingschema, zoals de al eerder genoemde termfrequentie of TF.IDF (Salton and McGill 1983). Een dergelijk wegingschema wordt uitvoerig behandeld in paragraaf 3.3.4.



Figuur 3.1 Visuele representatie van drie documenten in een 3-dimensionale ruimte (t=3)

Doordat ieder document (en query) een vector in een meerdimensionale ruimte voorstelt, kan een similariteitsmaat gedefinieerd worden aan de hand van geometrische principes. Deze principes hebben volgens Jones en Willett een aantal belangrijke voordelen (Jones and Willett 1997c). Ze zijn intuïtief te benaderen (ook voor een niet-specialist) en bieden mogelijkheden voor tal van geometrische operaties. De similariteit tussen twee vectoren kan bijvoorbeeld uitgedrukt worden in het product van de twee vectoren, de Euclidische afstand of in de cosinus van de hoek tussen de twee vectoren. Indien twee documenten uitsluitend gelijke termen bevatten, overlappen de twee vectoren elkaar en levert dit voor elke operatie een maximum similariteit op (Chowdhury 2004). Afgeleid uit bovenstaand voorbeeld kan de similariteit tussen  $DOC_i$  en  $DOC_j$  bijvoorbeeld bepaald worden door de cosinus van de hoek  $\theta$  te bepalen. Deze is te definiëren als (Witten, Moffat et al. 1999):

$$\cos \theta = \frac{X \cdot Y}{|X||Y|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

Vergelijking 3.1

$|X|$  staat hierbij voor de Euclidische lengte van vector  $X$ .

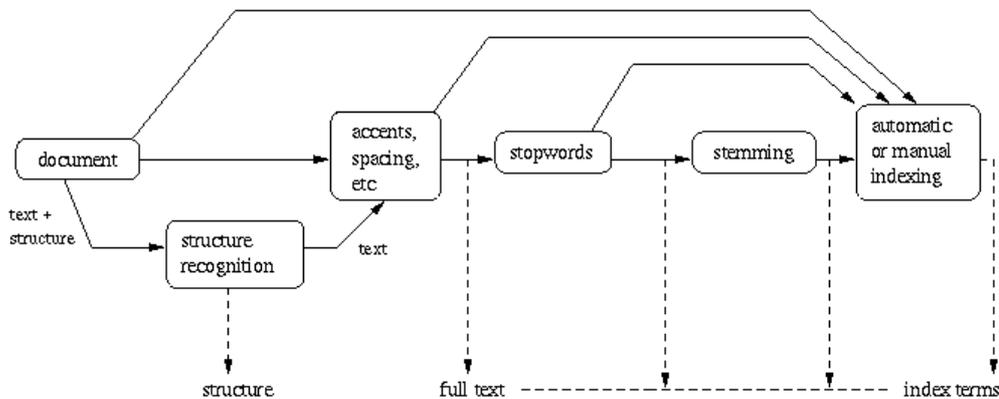
De similariteit wordt hierbij tegelijkertijd genormaliseerd naar een waarde tussen 0 en 1. Hiermee wordt gecorrigeerd voor de versturende invloed van de lengte van de documenten. Binnen een langer document bestaat er immers een hogere kans op het voorkomen van meer verschillende termen en in hogere termfrequenties. Door de niet-binaire aard van deze similariteitsmaat is de lijst met resultaten, als antwoord op een gegeven query, nauwkeuriger te rangschikken en sluit deze volgens (Baeza-Yates and Ribeiro-Neto 1999) dichter aan bij de information need van de gebruiker. Tevens is het met dit model mogelijk de meerdimensionale ruimte op te delen in kleinere ruimtes (clusters), waarin de documenten een bepaalde mate van gelijkheid vertonen op grond van de gekozen similariteitsmaat (Salton, Wong et al. 1997). Hiermee is het vervolgens mogelijk de documenten te classificeren naar overeenstemmende inhoud. Dit principe wordt onder meer beschreven in (Baeza-Yates and Ribeiro-Neto 1999; Weiss and Apte 2002). Naast deze positieve eigenschappen is dit model volgens dezelfde auteurs vooral relatief gemakkelijk te implementeren.

Het vector space model kent dus een groot aantal voordelen en wordt door een groot aantal auteurs daarom gezien als superieur aan andere modellen (Jones and Willett 1997c; Allan 2000; Chowdhury 2004). Volgens Jones en Willett zijn er echter wel degelijk een aantal negatieve praktische aspecten aan dit model. Volgens de auteurs is het met name moeilijk om synoniemen van keywords te expliciteren in een query. Dit kan bij het boolean model relatief gemakkelijk door middel van toepassing van de OR operator. In (Raghavan and Wong 1986) wordt de theoretische grondslag voor dit probleem onderbouwd. Volgens hen is de assumptie van orthogonaliteit inherent aan de toepassing van dit model, maar is dit in de praktijk veelal niet het geval. Hiermee wordt bedoeld dat de gebruikte termen, die de meerdimensionale ruimte bepalen lineair onafhankelijk van elkaar en dus paarsgewijs orthogonaal dienen te zijn. Indien twee documenten uitsluitend verschillende keywords bevatten, staan de vectoren inderdaad haaks op elkaar en hebben zij een onderlinge similariteit die gelijk is aan nul. Indien twee verschillende termen, die voorkomen in de documenten echter synoniemen van elkaar zijn, zou de similariteit hierbij ongelijk aan nul moeten zijn (Lenz, Hubner et al. 1998). Bij gebruik van natuurlijke taal in de documenten zal dit regelmatig voorkomen. De assumptie van de onafhankelijkheid van termen wordt dan geschaad en dit feit zorgt er volgens de auteurs voor dat het gebruik van dit model op theoretische gronden bekritiseerbaar is (Raghavan and Wong 1986). Om deze reden wordt er een alternatieve, gegeneraliseerde variant van het vector space model (GVSM) voorgesteld in (Wong, Ziarko et al. 1987), die helaas buiten het bestek van dit onderzoek valt. Dit model gaat namelijk uit van het herschrijven van alle termen uit de collectie in (gegeneraliseerde) concepten, zodat wordt voldaan aan de assumptie van orthogonaliteit. In de praktijk is dit echter een uiterst tijdrovende bezigheid.

Afhankelijk van het gekozen model zal elk information retrieval systeem een bepaalde wijze hanteren om de documenten te indexeren en representeren. De methoden die worden gebruikt bij het indexeren worden in de volgende paragraaf behandeld. De manier van representeren ligt in het geval van het vector space model voor de hand, namelijk als vectoren in een t-dimensionale ruimte, waarbij t gelijk is aan het aantal indexeerbare termen. Ook voor het probabilistische model wordt veelal eenzelfde soort representatie gehanteerd. In het boolean model kan er gebruik gemaakt worden van zogenaamde inverted file indexes (Baeza-Yates and Ribeiro-Neto 1999). Dit is een index van alle geïndexeerde keywords met een koppeling naar het document waar de betreffende term in voorkomt. Op deze wijze is het gemakkelijk logische bewerkingen toe te passen op de verzameling van documenten.

### 3.3 Information retrieval technieken

Ongeacht de manier van representeren en het toegepaste model zijn er een aantal algemene technieken binnen information retrieval te onderscheiden. Deze technieken en/of algoritmes worden gebruikt door een information retrieval systeem om de inspanningen van een gebruiker te verbeteren of aan te vullen (Jones and Willett 1997a). Hiermee wordt dus getracht beter te voldoen aan de information need van de gebruiker, zonder dat de gebruiker daar zelf extra handelingen voor hoeft te verrichten.



Figuur 3.2 Procesmatige kijk op de pre-processing stappen van een verzameling documenten.

Naar een afbeelding uit (Baeza-Yates and Ribeiro-Neto 1999)

De literatuur betreffende tekstuele information retrieval vermeldt veelal overeenkomstige technieken, die in feite allemaal afgeleid zijn uit het onderzoeksgebied van natural language processing (Allan 2000). Zij omvatten normaliter het verwijderen van stopwoorden, het terugbrengen van woorden tot hun woordstam en/of zogenaamde lemma en het toekennen van gewichten aan termen, op basis van de frequentie waarin deze termen voorkomen (Rijsbergen 1979; Jones and Willett 1997a; Baeza-Yates and Ribeiro-Neto 1999; Witten,

Moffat et al. 1999). Een lemma is een grammaticaal correcte schrijfwijze van de linguïstische stam van een woord, en kan daarom tevens gebruikt worden in een eventuele morfologische analyse (Xu and Croft 1998). Inherent aan deze technieken is het feit dat ze slechts eenmalig uitgevoerd hoeven te worden mits de inhoud en omvang van de set documenten niet verandert. Om deze reden worden ze volgens Baeza-Yates en Ribeiro-Neto beschouwd als pre-processing stappen. Aan de hand van de figuur op de vorige pagina (3.2) zullen deze stappen worden beschreven.

### 3.3.1 Term extractie

Allereerst wordt uit ieder document de feitelijke tekst geëxtraheerd. De structuur van de tekst wordt, met de relatief recente komst van standaarden als SGML, HTML en XML, echter steeds belangrijker. Er bestaan een aantal information retrieval systemen die gebruik maken van de structuur van documenten bij het zoeken, zoals beschreven in (McLeod 1990; Kilpelainen and Mannila 1993). Deze systemen vallen echter buiten het bereik van dit onderzoek, omdat er als bronmateriaal uitsluitend gebruik gemaakt kan worden van e-mails en gelogde calls in platte tekst, zonder enige opmaak.

Vervolgens zal de feitelijke tekst opgedeeld moeten worden in termen. Dit kan gebeuren met een variërende mate van intelligentie. In het simpelste geval worden alle leestekens vervangen door spaties en de termen gedefinieerd als alle resulterende woorden, gescheiden door spaties. Deze aanpak scheidt echter ook samengestelde woorden zoals 'macro-economie' in twee aparte termen. In hoeverre de keuzes die tijdens deze stap gemaakt worden van invloed zijn op de resultaten van het information retrieval systeem is afhankelijk van het toepassingsgebied (Fox 1992); (Baeza-Yates and Ribeiro-Neto 1999).

### 3.3.2 Stopwoorden

Zoals al beschreven geschiedde het indexereren van documenten bij de allereerste information retrieval systemen handmatig. Luhn was een van de eersten die publiceerde over het automatiseren van dit proces aan de hand van statistische methoden (Luhn 1961) (opnieuw gepubliceerd in (Jones and Willett 1997b)). Hierdoor werd het mogelijk alle woorden in de set van documenten te gebruiken als indexeerbare termen en bepaalde termen alleen uit te sluiten op grond van frequentie van voorkomen. Met name aan de termen met een relatief erg hoge frequentie van voorkomen wordt een lage semantische en daardoor weinig discriminerende waarde toegekend (Luhn's zogenaamde *upper cut-off* (Luhn 1961)).

Deze termen worden stopwoorden genoemd en normaliter niet meegenomen bij het indexereren (Rijsbergen 1979; Fox 1992; Jones and Willett 1997a; Baeza-Yates and Ribeiro-Neto 1999). Aangezien deze verwijdering onzichtbaar is voor de eindgebruiker van het

systeem, kan de situatie zich voordoen waarin een query met alleen stopwoorden niet het gewenste resultaat oplevert (Baeza-Yates and Ribeiro-Neto 1999; Witten, Moffat et al. 1999). Hoewel de toepassing van deze techniek voordelen biedt in het kader van de feitelijke resultaten van een information retrieval systeem, kan zij de interpretatie van het systeem door de eindgebruiker hierdoor bemoeilijken.

### 3.3.3 Stemming

Zodra alle termen uit de verzameling documenten zijn gehaald, kunnen deze al naar gelang het gekozen information retrieval model geïndexeerd worden. Meestal zijn de gebruikte documenten echter geschreven in natuurlijke taal en bestaan er verschillende morfologische variaties van specifieke termen. Om hiermee om te gaan kan er een zogenaamd conflatie algoritme toegepast worden. Hiermee worden woorden teruggebracht tot hun woordstam. De achterliggende assumptie is dat gelijke woorden, hoewel morfologisch verschillend, veelal een gelijke betekenis hebben. Als de verschillende schrijfwijzen van hetzelfde woord teruggebracht kunnen worden tot dezelfde woordstam, zal dit volgens Jones en Willett de resultaten verbeteren (Jones and Willett 1997a). Met name het aantal relevante documenten dat teruggevonden wordt (de zogenaamde recall) zal hierdoor volgens Xu en Croft verhoogd worden (Xu and Croft 1998).

Er bestaan een groot aantal verschillende algoritmes om stemming te bewerkstelligen. Een veelgebruikte indeling die, bijvoorbeeld door (Rijsbergen 1979; Jones and Willett 1997a; Allan 2000), aangebracht wordt in de verschillende methodes, bestaat uit het feit of het betreffende algoritme gebruik maakt van een woordenlijst uit de betreffende taal en/of op regels gebaseerd is. Regelgebaseerde stemming algoritmes halen recursief verbuigingsvormen van individuele woorden af, tot dit niet meer mogelijk is (suffix stripping). De hieruit ontstane termen zijn daarom vaak niet grammaticaal correct. Het algoritme dat het meest voor deze vorm van stemming gebruikt wordt binnen information retrieval is ontwikkeld door M.F. Porter (Porter 1980; Jones and Willett 1997b). Dit algoritme levert de beste prestaties in verhouding tot de vereiste rekenkracht (Xu and Croft 1998; Allan 2000). In (Kraaij and Pohlmann 1994) wordt een Nederlandse implementatie van het algoritme van Porter beschreven.

Regelgebaseerde stemmers hebben als negatieve eigenschap dat ze, vanwege de regelgebaseerde aanpak, in sommige gevallen fouten opleveren. Als het achtervoegsel '-den' bijvoorbeeld wordt verwijderd van 'gulden' blijft er 'gul' over. In dit geval blijft er geen unieke stam over en hierbij worden twee verschillende concepten herleid worden tot eenzelfde stam. Dit verschijnsel wordt overstemming genoemd. De tegenhanger, understemming, leidt tot verschillende woordstammen voor hetzelfde concept. Bijvoorbeeld 'matrices' wordt niet teruggebracht tot 'matrix' vanwege de onregelmatige meervoudsvorm. Bij onregelmatige werkwoorden is dit ook duidelijk zichtbaar, Alle vormen van de tegenwoordige tijd van het

werkwoord 'drinken' worden dan wel terug gebracht tot 'drink', maar omdat 'drinken' een onregelmatig werkwoord is, worden de verleden en voltooid tijd terug gebracht tot andere stammen. Deze fouten zijn inherent aan het algoritme, waarin geen ruimte is voor uitzonderingen en alleen generieke regels toegepast worden (Xu and Croft 1998; Allan 2000; Gaustad and Bouma 2002).

Naast suffix stripping is het ook mogelijk om te stemmen aan de hand van een woordenlijst. Bij deze methode worden termen teruggebracht tot hun lemma. Deze vorm wordt bijvoorbeeld toegepast door KSTEM en beschreven in (Krovetz 1993). KSTEM is in rekenkundig opzicht een stuk kostbaarder dan het algoritme van Porter, omdat een gehele woordenlijst opgeslagen en voor elke term doorzocht moet worden. Volgens Xu en Croft zijn de resultaten echter vergelijkbaar. Helaas is er geen Nederlandse, regelgebaseerde stemmer vrijelijk beschikbaar. Ongeacht welke vorm wordt gekozen zullen stemming algoritmes uitsluitend correct opereren met de taal voor welke ze bedoeld zijn.

Frakes onderscheidt binnen stemmers verder nog het gebruik van n-grammen en zogenaamde successor variety (Frakes 1992). Laatstgenoemde is een complexe regelgebaseerde variant, die gebruik maakt van morfologische grenzen in een woord. Deze grenzen worden bepaald met behulp van structureel linguïstische kennis en is daardoor in termen van rekenkracht kostbaarder dan beide eerder genoemde varianten. Het bepalen van n-grammen is feitelijk geen conflatie algoritme, maar meer een term-clustering principe. Hierbij worden woorden opgedeeld in strings van n letters groot. Deze worden afzonderlijk opgeslagen en gebruikt bij de similariteitsbepaling, zoals beschreven in (Frakes 1992; Jones and Willett 1997a; Baeza-Yates and Ribeiro-Neto 1999). Het gebruik van n-grammen leidt tot de noodzaak een grote hoeveelheid opslagruimte ter beschikking te hebben. De vergelijkende experimentele resultaten van deze varianten onderling zijn volgens (Frakes 1992; Krovetz 1993; Hull 1996) echter niet onverdeeld positief.

In (Xu and Croft 1998) wordt tenslotte een alternatief beschreven, waarin het meerdere malen voorkomen van termen in dezelfde context wordt gebruikt als basis om woordklassen te onderscheiden. Aan de hand van deze woordklassen kan vervolgens een vorm van stemming plaatsvinden. Volgens (Anjewierden, Huijsen et al. 2005) veroorzaakt het gebruik van natuurlijke taal in documenten een hoge mate van ruis, die de correcte retrieval van dergelijke documenten negatief beïnvloedt. Onbekende conflaties dragen hier met name aan bij. Door de auteurs worden een aantal methoden beschreven, die in grote lijnen overeenkomen met de ideeën van Xu en Croft, om deze te herkennen en automatisch te koppelen aan concepten. Gebruikmakend van deze methoden is TOKO ontwikkeld, een applicatie waarmee getracht wordt onbekende conflaties automatisch te herkennen.

### 3.3.4 Term weighting

Term weighting ontbreekt in de eerder gepresenteerde figuur uit (Baeza-Yates and Ribeiro-Neto 1999) over information retrieval technieken. Volgens (Jones and Willett 1997a) is dit echter wel degelijk een indexeer- en retrievaltechniek, omdat toepassing ervan de resultaten van information retrieval systemen in de meeste gevallen verbetert. Het wegen van termen definieert de significantie van een bepaalde term, ten opzichte van het document of query waar de term in voorkomt en/of ten opzichte van de gehele set van documenten. Deze termsignificantie kan vervolgens gebruikt worden om een additionele rangschikking aan te brengen in de set van resultaten op een query. Volgens eerder gehanteerde notatie levert de toepassing van een wegingschema onderstaande, aangepaste vector representatie op. Met  $w_{kt}$  wordt daarin het gewicht van term  $t$  voor document  $k$  aangeduid.

$$\begin{aligned} \text{DOC}_i &= [w_{i1} * \text{TERM}_{i1}, w_{i2} * \text{TERM}_{i2}, w_{i3} * \text{TERM}_{i3}, w_{i4} * \text{TERM}_{i4} \dots w_{it} * \text{TERM}_{it}] = X' \\ \text{DOC}_j &= [w_{j1} * \text{TERM}_{j1}, w_{j2} * \text{TERM}_{j2}, w_{j3} * \text{TERM}_{j3}, w_{j4} * \text{TERM}_{j4} \dots w_{jt} * \text{TERM}_{jt}] = Y' \\ \text{DOC}_k &= [w_{k1} * \text{TERM}_{k1}, w_{k2} * \text{TERM}_{k2}, w_{k3} * \text{TERM}_{k3}, w_{k4} * \text{TERM}_{k4} \dots w_{kt} * \text{TERM}_{kt}] = Z' \end{aligned}$$

Middels een wegingschema kan de semantische waarde van een term op een statistische wijze tot uitdrukking worden gebracht. Er komen een groot aantal verschillende wegingschema's voor in de literatuur, maar het merendeel gaat uit van een termfrequentie en een documentfrequentie component, zoals beschreven in (Salton 1988b). Met termfrequentie (TF) wordt de frequentie van voorkomen van een unieke term in één document aangeduid en met documentfrequentie (DF) de frequentie van voorkomen van een unieke term in de gehele collectie. Volgens Salton levert het product van deze twee (TF.IDF) de beste resultaten op. Het achterliggende idee is dat woorden, die veel voorkomen in veel verschillende documenten, een lage significantie en daarmee onderscheidend vermogen hebben. Woorden die slechts in enkele documenten voorkomen hebben daarentegen een groot onderscheidend vermogen, en zouden daarom een zwaarder gewicht moeten krijgen (Rijsbergen 1979). De meest gebruikte formules zijn:

$$\text{TF}_{d,t} = \frac{\log(f_{d,t})}{\log(l_d)}$$

Vergelijking 3.2

Hierbij is  $f_{d,t}$  de frequentie van voorkomen van term  $t$  in document  $d$  en  $l_d$  de lengte van document  $d$  (het totale aantal termen in document  $d$ ).

$$\text{IDF}_t = \log\left(\frac{N}{n_t}\right)$$

Vergelijking 3.3

Hierbij is  $N$  het totale aantal documenten in de collectie en  $n_t$  het aantal documenten waarin term  $t$  voorkomt. IDF kan ook worden bepaald door de volgende formule (Witten, Moffat et al. 1999):

$$\text{IDF}_t = \log\left(\frac{l}{f_t}\right)$$

Vergelijking 3.4

Hierbij is  $l$  de totale lengte van alle documenten in de collectie (het totale aantal termen in de collectie) en  $f_t$  de frequentie van voorkomen van term  $t$  in de collectie.

### 3.4 Evaluatiemethoden

In de lange geschiedenis van tekstuele information retrieval zijn er een groot aantal methoden ontwikkeld om de prestaties van information retrieval systemen mee te evalueren. Reeds in 1966 gaven de auteurs in (Cleverdon, Mills et al. 1966) een zestal kwantificeerbare maten. De belangrijkste hiervan, die sindsdien gehanteerd worden om de effectiviteit van een information retrieval systeem te meten, zijn zogenaamde precision en recall waardes. Deze kunnen het best belicht worden aan de hand van onderstaande tabel (Rijsbergen 1979).

|               |                  |                        |           |
|---------------|------------------|------------------------|-----------|
|               | relevant         | niet relevant          |           |
| gevonden      | $A \cap B$       | $\bar{A} \cap B$       | $B$       |
| niet gevonden | $A \cap \bar{B}$ | $\bar{A} \cap \bar{B}$ | $\bar{B}$ |
|               | $A$              | $\bar{A}$              | $N$       |

Hierbij is  $N$  het totaal aantal documenten in het systeem.

De precisie  $P$  is de verhouding tussen het totale aantal gevonden en relevante documenten, en het totale aantal gevonden documenten:

$$P = \frac{|A \cap B|}{|B|}$$

Vergelijking 3.5

De recall  $R$  is de verhouding tussen het totale aantal gevonden en relevante documenten, en het totale aantal relevante documenten:

$$R = \frac{|A \cap B|}{|A|}$$

Vergelijking 3.6

$|\cdot|$  staat hierbij voor het aantal documenten in de betreffende set.

Deze waarden zijn te bepalen voor de gehele set met gevonden resultaten, of voor een subset daarvan. Normaliter worden voor een aantal zogenaamde cut-off points bovenstaande waarden bepaald (bijvoorbeeld de top-5, top-10, top-20 en/of top-35 van alle gevonden resultaten).  $P$  kan gezien worden als een functie van  $R$  voor bepaalde cut-off points. Deze functie wordt gebruikt om een diagram te maken die deze relatie expliciteert, een zogenaamde recall-precisie curve. Een dergelijke curve laat precisie dus zien als functie van recall. De waarden die deze curve definiëren zijn de geïnterpoleerde waarden van  $P$  en  $R$ , voor verschillende cut-off points. Indien de curve van een bepaald algoritme of systeem zich boven een andere bevindt, kan gesteld worden dat deze beter presteert.

Er wordt bij het gebruik van bovenstaande evaluatiematen aangenomen dat hoe beter het systeem presteert, hoe dichter het systeem aansluit bij de information need van de gebruiker. Tevens wordt veelal aangenomen dat deze twee waarden voldoende zijn om de effectiviteit te meten, zoals in (Rijsbergen 1979). Er bestaat een levendige discussie in de literatuur over deze aannames en of andere maten niet wellicht geschikter zijn. Deze discussie is echter te breed om volledig te behandelen binnen het bestek van dit onderzoek.

Eén opvallend resultaat uit die discussie is dat, hoewel bovenstaande maten een solide historisch fundament hebben, ze niet meer aansluiten bij de huidige information need. Met name de komst en explosieve groei van het internet hebben bijgedragen aan de ontwikkeling van alternatieve, modernere evaluatiemethoden. Zoals eerder genoemd zijn information retrieval systemen in eerste instantie ontwikkeld om gebruikt te worden in een afgebakende omgeving (zoals een bibliotheek). Indien een gebruiker in een dergelijke omgeving iets zoekt, wil diegene alle of op zijn minst zoveel mogelijk relevante documenten gepresenteerd krijgen. Precisie en recall zijn uitstekende maten om te evalueren hoe goed alle relevante documenten in een verzameling (boeken in een collectie) worden teruggevonden, maar schieten tekort in situaties waar dergelijke systemen tegenwoordig ingezet worden. Zoekmachines op het internet opereren op een verzameling van miljarden documenten, terwijl ze resultaten op dezelfde wijze rangschikken en weergeven als bij een traditioneel information retrieval systeem. Wat het meest is veranderd in de tussentijdse tijd, is de verwachte information need van de gebruiker (Lawrence and Giles 1999). De meeste gebruikers zullen niet honderden documenten doorbladeren op zoek naar de gewenste informatie. Deze grens, waarop gebruikers niet meer verder willen zoeken wordt futility point genoemd in (Blair 1980). Na dit punt zullen de meeste gebruikers het opgeven en hun query herformuleren.

Nauwkeurigheid en precisie worden daarom tegenwoordig belangrijker geacht dan recall (Croft 1995). Recall is over grote collecties als het internet in de praktijk überhaupt niet te bepalen. Met name bij de toepassing bij het eerder genoemde automatisch beantwoorden van vragen of in een case-based reasoning systeem, is een gebruiker typisch op zoek naar

slechts enkele relevante resultaten. De effectiviteit van een dergelijk systeem kan daarom beter bepaald worden door te kijken naar hoe hoog deze resultaten gerangschikt worden, dan naar feitelijke precisie of recall (Shah and Croft 2004).

Indien de eerder genoemde futility point in acht wordt genomen bij het bepalen van het cut-off point heeft precisie wel degelijk evaluatieve betekenis, omdat ze dan overeenkomt met het gedrag van de gebruiker. In het geval van een zoekactie op het internet betekent dit dat de precisie wordt bepaald over bijvoorbeeld de top-10 van de gevonden resultaten.

Een relatief nieuwe evaluatiemaat voor moderne information retrieval systemen wordt mean reciprocal rank (MRR) genoemd en beschreven in (Lin and Katz 2003). Reciprocal rank is gedefinieerd als de inverse van de positie van het eerste relevante resultaat, bij een bepaald cut-off point. Door deze waarde te bepalen op een aantal queries, kan het gemiddelde en daarmee de MRR bepaald worden voor het betreffende systeem en/of algoritme. Lage scores duiden op een effectieve werkwijze, omdat deze in staat is correcte resultaten op een hoge positie te retourneren.

### 3.5 Case-based reasoning vs. information retrieval

Zoals moge zijn gebleken, hanteren zowel case-based reasoning als information retrieval als centraal proces het beheren, opslaan en terugvinden van informatie. Er zijn echter een aantal wezenlijke verschillen aan te merken. Case-based reasoning toepassingen maken per definitie in meer of mindere mate gebruik van kennis (domeinspecifiek of algemeen) of contextinformatie bij het zoeken (Aamodt and Plaza 1994). Tijdens het selecteren van indexeerbare termen wordt in case-based reasoning veelal een semantische basis gebruikt, terwijl dit in information retrieval van oudsher op puur syntactische basis geschiedt (Kamp, Lange et al. 1998). Er zijn inmiddels een aantal initiatieven getoond om ook het aandachtspunt binnen information retrieval te verschuiven van klassieke, traditionele information retrieval richting "intelligente information retrieval", door gebruik te maken van kennis bij het zoeken naar informatie. De vorm van gebruikte kennis kan hierbij verscheidene vormen aannemen, bijvoorbeeld ontologieën en thesauri of de toepassing van het al eerder genoemde NLP (Allan 2000; Croft 2000). Een aantal specifieke voorbeelden van de toepassing van ontologieën en thesauri bij information retrieval is het gebruik van WordNet als semantisch netwerk zoals in (Kazman and Kominek 1997) of de poging een similariteitsthesaurus automatisch te genereren zoals in (Qiu and Frei 1994).

Idealiter wordt informatie uit de documenten niet door keywords gerepresenteerd, maar door een kennisgebaseerde notatie. Middels deze relatief nieuwe aanpak binnen information retrieval wordt onder meer getracht een tweetal typische problemen op te lossen, namelijk het ambiguïteits- en het parafraze probleem (Salton, Singhal et al. 1996; Lenz, Hubner et al.

1998). Twee documenten kunnen veel dezelfde termen bevatten en dus een hoge similariteit hebben, maar inhoudelijk over verschillende zaken gaan (het zogenaamde ambiguïteitprobleem). Tevens kan hetzelfde concept door middel van verschillende termen uitgedrukt worden, wat voor een lage similariteit zorgt (het parafraze probleem). Een nadeel is dat met de huidige stand van de wetenschap en techniek een dergelijke representatie nog niet gerealiseerd kan worden voor grotere domeinen.

In (Salton, Singhal et al. 1996) wordt een andere, mogelijke oplossing gepresenteerd, waarbij vaker voorkomende tekst segmenten worden geïdentificeerd als concepten. Deze segmenten kunnen in grootte variëren van woordparen tot hele zinnen. De auteurs beargumenteren dat deze zogenaamde lokale similariteit de eerder beschreven globale similariteit kan verbeteren. De globale similariteit wordt alleen als voldoende hoog aangemerkt indien er voldoende lokale similariteit is. Dat wil zeggen dat globale similariteit alleen niet voldoende is om tot een hoge score te komen; in de twee vergeleken documenten dienen daarbij ook een voldoende aantal overeenstemmende concepten voor te komen. Deze concepten worden per document tevens voorzien van hyperlinks naar de andere documenten waar de betreffende concepten in voorkomen, om op deze wijze de gebruiker te gidsen naar andere, eventueel relevante documenten.

Het feitelijke retrieval proces binnen case-based reasoning verschilt met die binnen information retrieval in het feit dat deze bij case-based reasoning een meer actieve rol heeft. Information retrieval systemen laten het veelal aan de gebruiker over om de correcte vraag te stellen, terwijl een case-based reasoning systeem adaptief zoekt, op basis van kenmerken die zonder tussenkomst van een gebruiker kunnen worden bepaald (Barletta 1993; Leake 1996). In (Leake 1996) wordt daarnaast het punt gemaakt dat information retrieval systemen van oudsher vooral gebruik maken van een exacte match tussen query en de opgeslagen informatie, terwijl case-based reasoning met name de meest gelijkende case(s) terug dient te vinden. Enkele eigenschappen van deze meest gelijkende cases kunnen conflicteren met de eigenschappen van de query, maar verder toch voldoende overeenstemmen om een match op te leveren. Hierdoor kan gesteld worden dat case-based reasoning systemen dichter trachten aan te sluiten bij de information need van de gebruiker.

Tenslotte ligt het belangrijkste onderscheid besloten in het feit dat case-based reasoning systemen eerdere cases moeten kunnen aanpassen aan een nieuw probleem (Aamodt and Plaza 1994; Kamp, Lange et al. 1998). Dit vereist echter een kennisgebaseerde representatie van de opgeslagen problemen.

Een systeem (SPIRE) dat op het scharnierpunt tussen case-based reasoning en information retrieval opereert wordt beschreven in (Rissland and Daniels 1995b) en verder uitgediept in (Rissland and Daniels 1997). Hierin wordt case-based reasoning gebruikt om in een

kennisrijke database met cases een set met relevante cases terug te vinden. Vervolgens worden de cases uit deze set gebruikt als zogenaamde seed documents in het eerdergenoemde full-text search systeem INQUERY. Dit systeem zoekt in een kennisarme, doch omvangrijkere database met additionele cases naar matches. Hiermee beogen de auteurs ten eerste in een omvangrijke database met cases te kunnen zoeken, zonder te hoeven investeren in een kennisrijke representatie van alle cases. Ten tweede worden volgens de auteurs met deze aanpak de prestaties van het retrieval systeem danig verbeterd vergeleken met de prestaties zonder toepassing van het case-based reasoning systeem.



## 4 Onderzoeksmethoden

De onderzoeksvraag “Wat zijn de resultaten van een op statistische basis opererende retrieval module van een case-based reasoning systeem met en zonder gebruikmaking van verscheidene information retrieval technieken, toegepast op vragen gesteld aan de helpdesk van Hostnet” wordt geoperationaliseerd door middel van een aantal experimenten, gebaseerd op relevante data en expertise van Hostnet. In dit hoofdstuk worden deze experimenten besproken die, op basis van de literatuur uit de vorige hoofdstukken en de praktische omstandigheden voorhanden, ontworpen zijn. Naast de achtergronden van de experimenten worden ook het retrievalmodel, de gebruikte data, algoritmes en implementatie daarvan behandeld.

### 4.1 Operationalisatie

Allereerst is er, in het licht van de vragen die het uiteindelijke case-based reasoning systeem te verwerken zal krijgen, een passend information retrieval model gekozen. Het is de bedoeling dat het systeem moet kunnen functioneren zonder een domein- of ander kennismodel, dus op statistische wijze. Bij toepassing van de module in een case-based reasoning systeem zal deze nieuwe vragen als input krijgen. Ook is het de bedoeling dat de module de meest gelijkende vragen terugvindt, zonder uitsluitend keywords te gebruiken. Om deze redenen valt het boolean model af. Het probabilistische model is overwogen, maar een waarschijnlijkheidsfunctie is moeilijk te definiëren, zonder *relevance feedback* van (toekomstige) gebruikers. Zoals in paragraaf 3.1 beschreven kunnen termfrequenties uitsluitend gebruikt worden voor de waarschijnlijkheidsfunctie indien alle eerdere cases over unieke onderwerpen gaan (Robertson 1997). Dit is niet te garanderen zonder uitvoerige controle van alle opgeslagen cases. Er is gekozen voor het vector space model omdat het gemakkelijkst te implementeren bleek en ook de aard van de vragen die de module te verwerken zal krijgen het meest in lijn ligt met dit model.

Vervolgens is er een similariteitsmaat gekozen die gebruikt zal worden om de mate van overeenkomst tussen de gestelde vragen en de opgeslagen cases te bepalen. Omdat een kennis- of domeinmodel ontbreekt, zal deze maat op statistische (syntactische) gronden moeten opereren. In de literatuur wordt de cosinus similariteit uitvoerig beschreven, behandeld en toegepast (Rijsbergen 1979; Salton and McGill 1983; Buckley 1985; Raghavan and Wong 1986; Salton 1988a; Jones and Willett 1997b; Baeza-Yates and Ribeiro-Neto 1999; Witten, Moffat et al. 1999; Greenwood 2001; Chowdhury 2004). Er is voor deze maat gekozen omdat hij, zoals in de vorige paragraaf beschreven, als belangrijke eigenschap heeft dat hij relatief gemakkelijk te interpreteren en implementeren is.

## 4.2 Onderzoeksomgeving

De vragen waarmee de retrieval module zal gaan werken zullen bestaan uit de inhoud van e-mails, gericht aan de helpdesk van Hostnet, en gelogde calls. Calls worden door medewerkers van de helpdesk samengevat en opgeslagen als platte tekst. De vragen belanden momenteel via een ticketing systeem<sup>1</sup> in een MySQL database. Vanwege het gebruik van dit systeem is het mogelijk een complete dialoog te achterhalen. Per vraag kan tevens in het ticketing systeem worden aangegeven en opgeslagen of deze al dan niet succesvol is beantwoord. In de praktijk wordt er echter meestal net zolang aan een vraag of probleem gewerkt, totdat er een bevredigend antwoord is gevonden. Hierdoor komt het bijna niet voor dat vragen onsuccesvol gesloten worden en worden de effecten, zoals beschreven in paragraaf 2.4, grotendeels vermeden. Het systeem hoeft namelijk geen rekening te houden met een eventuele negatieve uitkomst van een eerder gestelde oplossing. Indien een nieuwe vraag in voldoende mate overeenkomt met een eerdere vraag, kan het antwoord dat op die vraag gegeven is dus in theorie direct worden hergebruikt.

Hostnet gebruikt het ticketing systeem inmiddels al geruime tijd en bevat daarom een aanzienlijk aantal vragen (ongeveer 10.000) met bijbehorende antwoorden. De opgeslagen e-mails worden in de originele staat waarin ze ontvangen zijn opgeslagen in de database. Hierdoor bestaan er verschillende formaten, (bijvoorbeeld platte tekst, HTML, etcetera). De e-mails zijn eerst verwerkt tot één uniform (tekst) formaat door alle HTML en/of andere opmaakgerelateerde code, voor zover aanwezig, te verwijderen.

Vervolgens is getracht om per e-mail op automatische wijze eventuele zogenaamde signatures te verwijderen. Signatures zijn bijvoorbeeld contactgegevens, ondertekeningen of andere geautomatiseerde toevoegingen aan e-mailberichten. Dit is erg voorzichtig gebeurd, teneinde niet per ongeluk tevens de inhoud van een vraag te verwijderen. Ook geciteerde stukken tekst, bijvoorbeeld uit eerdere e-mails worden verwijderd. Deze zijn veelal te herkennen aan het feit dat elke geciteerde regel begint met een vishaakje (>) of door een specifieke zinsnede (bijvoorbeeld '----- Original Message -----').

Tenslotte is de overgebleven inhoud van alle geëxtraheerde vragen opgeslagen in een nieuwe database. De inhoud van deze database wordt uiteindelijk gebruikt door de retrieval module. Op de volgende pagina's staat een uitgewerkt voorbeeld van alle genomen bewerkingsstappen, toegepast op een ingezonden e-mail. In dit voorbeeld is alle vertrouwelijke informatie overigens verwijderd.

---

<sup>1</sup> OTRS; <http://www.otrs.org>



```

<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#FFFFFF" text="#000000" leftmargin="0" topmargin="0" marginwidth="0"
marginheight="0">
<table width="500" border="0" cellspacing="0" cellpadding="0">
<tr>
<td class="text"> <span style = "font-family: Verdana, Arial, Helvetica, sans-
serif; font-size: 12px; color: #000000">
<p>
L.s., <br>
In de PDF documentatie, op pagina 20 (Handleiding Linux MKB versie 2004-8)
wordt een hoofdstuk gewijd aan het gebruik van voorgeïnstalleerde scripts.
in 5.1.1 Noemt u FormMail<br>
Hierbij wordt aangegeven dat het script te vinden is in de cgi-bin
directory.<br>
<br>
Helaas, het staat er niet.<br><br>
Waar kan ik dit vinden?<br>
</p>
<span style = "font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 12px;
color: #000000">
<p>
<br>Met vriendelijke groet,<br><br>
...<br>
</p>
</span>
<span style = "font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px;
color: #000000">
<p>
=====<br>
Deze e-mail is uitsluitend bestemd voor de geadresseerde(n). Gebruik door
derden of openbaarmaking van dit bericht zonder toestemming van de afzender
of de geadresseerde is niet toegestaan en kan onrechtmatig zijn. Uw reactie
kunt u richten aan het e-mail adres van deze afzender. Ik sluit iedere
aansprakelijkheid uit die voortvloeit uit elektronische verzending. Dit
bericht en eventuele bijlagen zijn gescand op de ons bekende virussen. Wij
adviseren u alle berichten en bijlagen op virussen te scannen voor ze te
openen.
</p>
</span>
</td>
</tr>
</table>
</body>
</html>

```

Na verwijdering van alle HTML-code blijft het volgende bericht over:

```
L.s.,  
In de PDF documentatie, op pagina 20 (Handleiding Linux MKB versie 2004-8) wordt  
een hoofdstuk gewijd aan het gebruik van voorgeïnstalleerde scripts. in 5.1.1 Noemt  
u FormMail  
Hierbij wordt aangegeven dat het script te vinden is in de cgi-bin directory.  
  
Helaas, het staat er niet.  
  
Waar kan ik dit vinden?  
  
Met vriendelijke groet,  
  
...  
  
=====
```

Deze e-mail is uitsluitend bestemd voor de geadresseerde(n). Gebruik door derden of openbaarmaking van dit bericht zonder toestemming van de afzender of de geadresseerde is niet toegestaan en kan onrechtmatig zijn. Uw reactie kunt u richten aan het e-mail adres van deze afzender. Ik sluit iedere aansprakelijkheid uit die voortvloeit uit elektronische verzending. Dit bericht en eventuele bijlagen zijn gescand op de ons bekende virussen. Wij adviseren u alle berichten en bijlagen op virussen te scannen voor ze te openen.

In de volgende stap wordt de signature verwijderd:

```
L.s.,  
In de PDF documentatie, op pagina 20 (Handleiding Linux MKB versie 2004-8)  
wordt een hoofdstuk gewijd aan het gebruik van voorgeïnstalleerde scripts.  
in 5.1.1 Noemt u FormMail  
Hierbij wordt aangegeven dat het script te vinden is in de cgi-bin  
directory.  
Helaas, het staat er niet.  
Waar kan ik dit vinden?
```

## 4.3 Implementatie

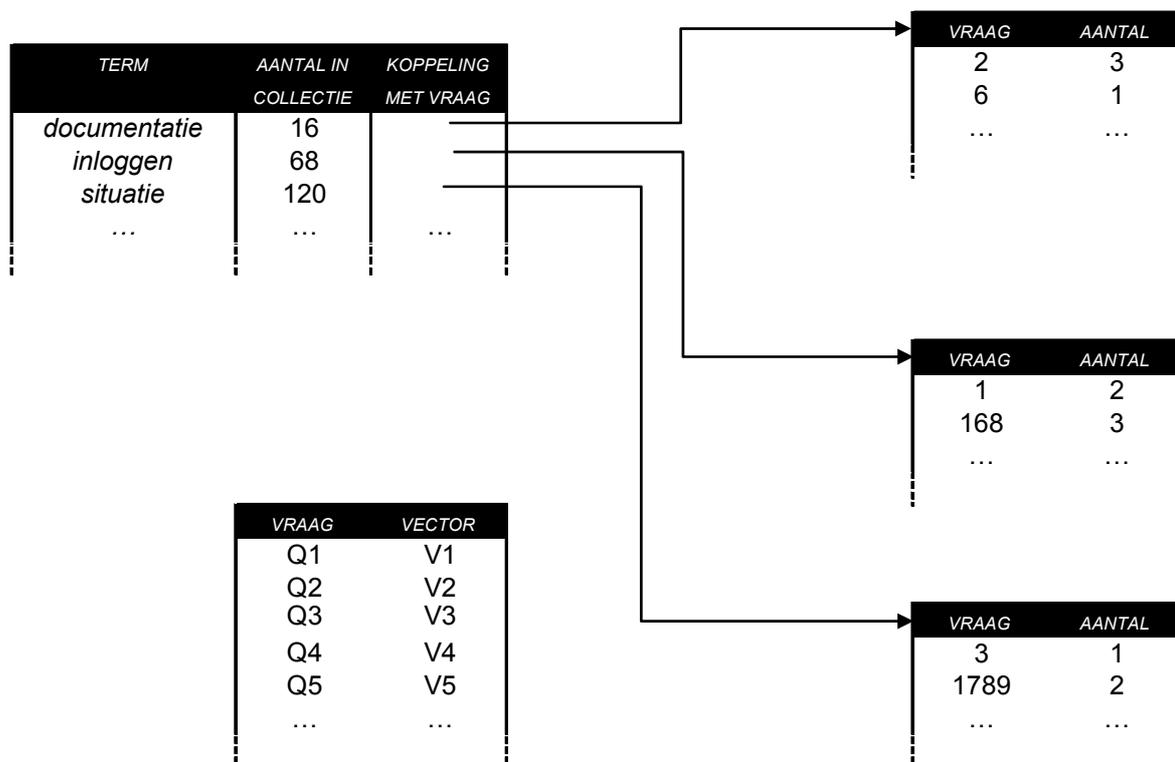
De vectorgebaseerde retrieval module is geschreven in Perl. Perl is een relatief oude programmeertaal en al van oorsprong bedoeld om op gemakkelijke en efficiënte wijze teksten te verwerken (Orwant, Hietaniemi et al. 1999). Perl is met name ontwikkeld om in een UNIX-achtige omgeving tekstuele in- en output met de shell te verwerken. In Perl bestaat het zogenaamde hash datatype, ook wel associatief array genoemd. Dit is een type variabele waarmee aan de hand van een key een of meerdere waarden te definiëren zijn. Vanwege



deze eigenschap is dit type gekozen om als primaire manier van opslag te dienen. Naast hashes wordt er gebruik gemaakt van PDL<sup>2</sup> om de vectoren in een meerdimensionale ruimte te representeren.

## 4.4 Indexeren

Als de module wordt gestart, worden alle vragen eerst uitgelezen uit de database. Vervolgens worden hieruit alle leestekens verwijderd en vervangen door spaties. Accenten op letters worden eveneens verwijderd. Alle cijfers en overblijvende woorden die kleiner zijn dan drie tekens worden vervolgens ook gewist. Tenslotte worden alle hoofdletters vervangen door kleine letters. De resulterende woorden, gescheiden door spaties, worden gezien als termen en opgeslagen in een woordenlijst per vraag. Tevens wordt er een globale woordenlijst van de gehele collectie bijgehouden. Van elke vraag wordt een vector bepaald met als dimensies het totale aantal termen in de globale woordenlijst. De waarde langs elke dimensie komt overeen met het aantal keer dat de betreffende term voorkomt in de vraag. Elke vector wordt tenslotte gekoppeld aan de betreffende vraag. In onderstaande figuur wordt een voorbeeld gegeven van de datastructuur die daarbij gehanteerd wordt.



Figuur 4.1 Datastructuur

<sup>2</sup> Perl Data Language; <http://pdl.perl.org/>

Elke unieke term wordt opgeslagen in een globale lijst met termen (linkerkolom 'TERM'). Voor elke term wordt de globale frequentie van voorkomen als koppeling opgeslagen in de globale term-count hash (tweede kolom). De lokale termfrequentie per vraag wordt bewaard als koppeling met de betreffende vraag in de vraag term-count hash (derde kolom). Tenslotte is er een lijst met vragen die, door middel van een hash, gekoppeld wordt aan de bijbehorende lijst met vectoren. Op de volgende pagina wordt het geoperationaliseerde indexeerproces in pseudo-code (4.1) weergegeven.

#### 4.4.1 Term weighting

Zoals in paragraaf 3.3.4 beschreven is  $TF.IDF$  de vorm van term weighting die het meest toegepast wordt en de beste resultaten oplevert. Een uitgebreide beschrijving van hoe een dergelijk algoritme te operationaliseren en implementeren is terug te vinden in (Witten, Moffat et al. 1999). Voor dit onderzoek is per term  $t$  in document  $d$   $TF.IDF$  bepaald volgens de volgende formules:

$$TF_{d,t} = \frac{\log(f_{d,t})}{\log(l_d)}$$

Vergelijking 3.2

$$IDF_t = \log\left(\frac{l}{f_t}\right)$$

Vergelijking 3.4

$$TF.IDF(d,t) = TF_{d,t} * IDF_t = \frac{\log(f_{d,t}) \log\left(\frac{l}{f_t}\right)}{\log(l_d)}$$

Vergelijking 4.1

Hierbij is  $f_{d,t}$  de frequentie van voorkomen van  $t$  in  $d$ ,  $l$  de lengte van  $d$  (het totale aantal termen in  $d$ ),  $l$  de totale lengte van alle documenten in de collectie (het totale aantal termen in de collectie) en  $f_t$  de frequentie van voorkomen van  $t$  in de collectie.

Indien een term dus vaak voorkomt in een vraag en relatief weinig in de gehele collectie krijgt deze een hoog gewicht en vice versa. Toepassing van  $TF.IDF$  vereist echter een andere manier van indexeren dan bovenstaand. Bij het maken van een vector voor een vraag dient de frequentie van voorkomen van elke term in de vraag en in de gehele collectie meegenomen te worden. Pseudo-code 4.2 (op bladzijde 46) geeft het geoperationaliseerde indexeerproces weer bij toepassing van  $TF.IDF$ .

**Pseudo-code 4.1 (indexeerproces zonder TF.IDF)**

Creëer voor deze collectie een nieuwe, lege globale term-count hash en een nieuwe, lege lijst voor unieke termen.

Haal een vraag op uit de database en lees deze regel voor regel in tot er geen regels meer overblijven. Creëer een nieuwe, lege term-count hash voor deze vraag.

Vervang leestekens door spaties, hoofdletters door kleine letters en verwijder getallen en resulterende woorden die kleiner zijn dan 3 tekens.

Splits de vraag op in termen, gescheiden door spaties. Verwijder stopwoorden uit deze termen (*indien geselecteerd*) en pas het stemming algoritme toe (*indien geselecteerd*) op iedere term.

FOR iedere term in de vraag

    Haal de deze term op uit de globale lijst met termen.

    IF deze bestaat (is de term al eerder tegengekomen in een eerdere vraag)

        -

    ELSE  
        Creëer een nieuwe entry voor deze term in deze lijst

    Haal de waarde voor deze term op uit de term-count hash voor deze vraag.

    IF deze waarde bestaat (is de term al eerder tegengekomen in deze vraag)

        Tel 1 op bij deze waarde in de term-count hash voor deze vraag

    ELSE

        Creëer een nieuwe waarde 1 voor deze term in de term-count hash voor deze vraag

REPEAT tot er geen onbehandelde vragen meer in de database zijn.

Lees iedere opgeslagen vraag nogmaals in (om per vraag een vector te genereren)

    Maak een vector met als lengte het totale aantal unieke entries in de globale lijst met termen en vul deze met nullen.

    FOR iedere unieke term in de vraag

        Bepaal term frequency (TF) door de corresponderende waarde uit de term-count hash van deze vraag in te lezen. Sla TF op voor de corresponderende dimensie van deze term in de vector van de vraag.

REPEAT tot er geen onbehandelde vragen meer in de database zijn.

    Sla de resulterende vector op, gekoppeld aan de oorspronkelijke vraag

### Pseudo-code 4.2 (indexeerproces met TF.IDF)

Creëer voor deze collectie een nieuwe, lege globale term-count hash en een nieuwe, lege lijst voor unieke termen.

Haal een vraag op uit de database en lees deze regel voor regel in tot er geen regels meer overblijven. Creëer een nieuwe, lege term-count hash voor deze vraag.

Vervang leestekens door spaties, hoofdletters door kleine letters en verwijder getallen en resulterende woorden die kleiner zijn dan 3 tekens.

Splits de vraag op in termen, gescheiden door spaties. Verwijder stopwoorden uit deze termen (*indien geselecteerd*) en pas het stemming algoritme toe (*indien geselecteerd*) op iedere term.

Bepaal het totale aantal resulterende termen voor deze vraag en sla deze op. Tel deze waarde tevens op bij het totale aantal termen in de collectie.

FOR iedere term in de vraag

    Haal de deze term op uit de globale lijst met termen.

    IF deze bestaat (is de term al eerder tegengekomen in een eerdere vraag)

        -

    ELSE

        Creëer een nieuwe entry voor deze term in deze lijst

    Haal de waarde voor deze term op uit de globale term-count hash.

    IF deze waarde bestaat

        (is de term al eerder tegengekomen in een eerdere vraag)

        Tel 1 op bij deze waarde in de globale term-count hash

    ELSE

        Creëer een nieuwe waarde 1 voor deze term in de globale term-count hash

    Haal de waarde voor deze term op uit de term-count hash voor deze vraag.

    IF deze waarde bestaat (is de term al eerder tegengekomen in deze vraag)

        Tel 1 op bij deze waarde in de term-count hash voor deze vraag

    ELSE

        Creëer een nieuwe waarde 1 voor deze term in de term-count hash voor deze vraag

REPEAT tot er geen onbehandelde vragen meer in de database zijn.

Lees iedere opgeslagen vraag nogmaals in (om per vraag een vector te genereren)

Maak een vector met als lengte het totale aantal unieke entries in de globale lijst met termen en vul deze met nullen.

FOR iedere unieke term in de vraag

    Bepaal term frequency (TF) door het logaritme van de termfrequentie voor deze term in dit document te delen door het logaritme van het totale aantal termen in dit document.

    Bepaal inverse document frequency (IDF) door het logaritme van het totale aantal globale termen te delen door het logaritme van de globale termfrequentie voor deze term.

    Vermenigvuldig TF met IDF en sla de resulterende waarde op voor de corresponderende dimensie van deze term in de vector van de vraag.

    Sla de resulterende vector op, gekoppeld aan de oorspronkelijke vraag

REPEAT tot er geen onbehandelde vragen meer in de database zijn.



#### 4.4.2 Stopwoorden

Met behulp van een keyword density analyzer<sup>3</sup> is van de verzameling termen uit alle vragen na het indexeren de individuele termfrequentie bepaald. Hierin zijn dus al cijfers, leestekens en woorden die minder dan drie tekens bevatten verwijderd. De termen die een frequentie groter dan 0.15% hebben zijn aangemerkt als stopwoorden en in een apart bestand geplaatst. Dit percentage is bepaald door de frequentieverdeling van deze termen te bekijken en is in overeenstemming met de in het vorige hoofdstuk behandelde *upper cut-off* uit (Luhn 1961). Dit leverde 110 stopwoorden op, die terug te vinden zijn in bijlage A. De inhoud van het bestand met stopwoorden kan worden gebruikt bij het indexeren en het zoeken, door deze termen hierbij niet mee te nemen (zie de respectievelijke pseudo-codes).

Toegepast op eerdergenoemd voorbeeld levert dit het volgende op:

```
pdf documentatie pagina handleiding linux mkb versie hoofdstuk gewijd gebruik
voorgeïnstalleerde scripts noemt formmail hierbij aangegeven het script vinden cgi
bin directory helaas staat waar kan vinden
```

#### 4.4.3 Stemmer

De Nederlandse versie van het stemming algoritme van Porter, ontwikkeld door Kraaij en Pohlmann (Kraaij and Pohlmann 1994) is publiekelijk beschikbaar<sup>4</sup> en tevens als optie geïmplementeerd in de module. Indien geselecteerd wordt stemming toegepast op zowel de geïndexeerde vragen als de uiteindelijke query (zie respectievelijke pseudo-codes).

Toegepast op het voorbeeld geeft toepassing van het stemming algoritme als resultaat:

```
pdf documentatie pagina handleid linux mkb versie hoofdstuk gewijd gebruik
voorgeïnstalleerd script noemt formmail hierbij aangegev het script vind cgi bin
directory helas stat war kan vind
```

### 4.5 Zoeken

Als de index is gecreëerd is de retrieval module klaar voor gebruik. Middels het toetsenbord kan een query ingegeven worden in het systeem. Deze verwijderd alle woorden die niet in de globale woordenlijst voorkomen en maakt een nieuwe vector van alle overblijvende termen. Vervolgens wordt de similariteit van de verkregen vector met elk van de opgeslagen vectoren bepaald, door telkens de cosinus van de hoek tussen de vectoren te berekenen. Uitgaande van vergelijking 3.1 is deze relatief gemakkelijk te bepalen als het inner product van de twee vectoren, mits deze genormaliseerd zijn. Op het scherm wordt tenslotte een gerangschikt

<sup>3</sup> GRKda; [http://www.grsoftware.net/search\\_engines/software/grkda.html](http://www.grsoftware.net/search_engines/software/grkda.html)

<sup>4</sup> Snowball; <http://snowball.tartarus.org/algorithms/dutch/stemmer.html>



overzicht gepresenteerd van de gevonden vragen, gesorteerd op aflopende mate van similariteit, met bijbehorende cosinus waardes. In onderstaande pseudo-code (4.3), wordt de operationalisatie van het zoekproces weergegeven.

**Pseudo-code 4.3 (zoekproces)**

Lees de invoer van het toetsenbord in als query en creëer een nieuwe, lege term-count hash voor deze query.

Vervang leestekens door spaties, hoofdletters door kleine letters en verwijder getallen en resulterende woorden die kleiner zijn dan 3 tekens.

Splits de query op in termen, gescheiden door spaties. Verwijder stopwoorden uit deze termen (*indien geselecteerd*) en pas het stemming algoritme toe (*indien geselecteerd*) op iedere term.

Verwijder woorden uit de query die niet voorkomen in de globale lijst met geïndexeerde termen.

Maak een vector voor deze query met als lengte het totale aantal unieke entries in de globale lijst met geïndexeerde termen en vul deze met nullen.

FOR iedere term in de query

    Haal de waarde voor deze term op uit de term-count hash voor deze query.

    IF deze waarde bestaat (is de term al eerder tegengekomen in deze query)

        Tel 1 op bij deze waarde in de term-count hash voor deze query

    ELSE

        Creëer een nieuwe waarde 1 voor deze term in de term-count hash voor deze query

FOR iedere unieke term in de query

    Bepaal term frequency (TF) door de corresponderende waarde uit de term-count hash voor deze query in te lezen. Sla TF op voor de corresponderende dimensie van deze term in de vector van de query.

Normaliseer de resulterende query vector.

FOR iedere opgeslagen vector in de collectie

    Normaliseer de vector en bepaal de cosinus van de hoek tussen de query vector en de ingelezen vector. Sla het resultaat op in een aflopende, gesorteerde lijst met resultaten.

FOR ieder resultaat in de lijst met resultaten

    Print de waarde van het resultaat met de corresponderende vraag indien het resultaat groter is dan de vooraf gedefinieerde grens (*indien geselecteerd*) en er niet al voldoende resultaten geretourneerd zijn (*indien geselecteerd*).

Het is mogelijk een limiet op de similariteitsmaat op te geven, onder welke het systeem geen resultaten meer retourneert. Indien deze niet of erg laag ingesteld wordt, worden ook vragen weergegeven die een extreem lage mate van overeenkomst vertonen. Als alternatief hiervoor kan er gekozen worden hoeveel vragen de module retourneert, ongeacht de waarde van de similariteitsmaat. Indien voor geen van beide opties gekozen wordt, zullen dus alle opgeslagen vragen weergegeven worden, in aflopende mate van similariteit.



Binnen een query wordt ook term weighting toegepast. Hiermee wordt de significantie van een bepaalde term in een query meegenomen. Termen die vaak in een query voorkomen krijgen hierdoor een hoger gewicht dan minder frequente termen. Indien iemand een bepaalde term vaak gebruikt in zijn vraag zal deze hoogstwaarschijnlijk meer van belang zijn dan een minder frequente term.

## 4.6 Experimentele opzet

Het feitelijke onderzoek maakt gebruik van bovenstaande implementatie(s) en zal toetsen in hoeverre de genoemde technieken invloed hebben op de retrieval resultaten van de module. Omdat het systeem waar de module op draait niet snel genoeg is om real-time antwoorden te genereren, is er voor gekozen dit vooraf (off-line) te doen. Ook is er om dezelfde reden voor gekozen slechts 3000 vragen te indexeren.

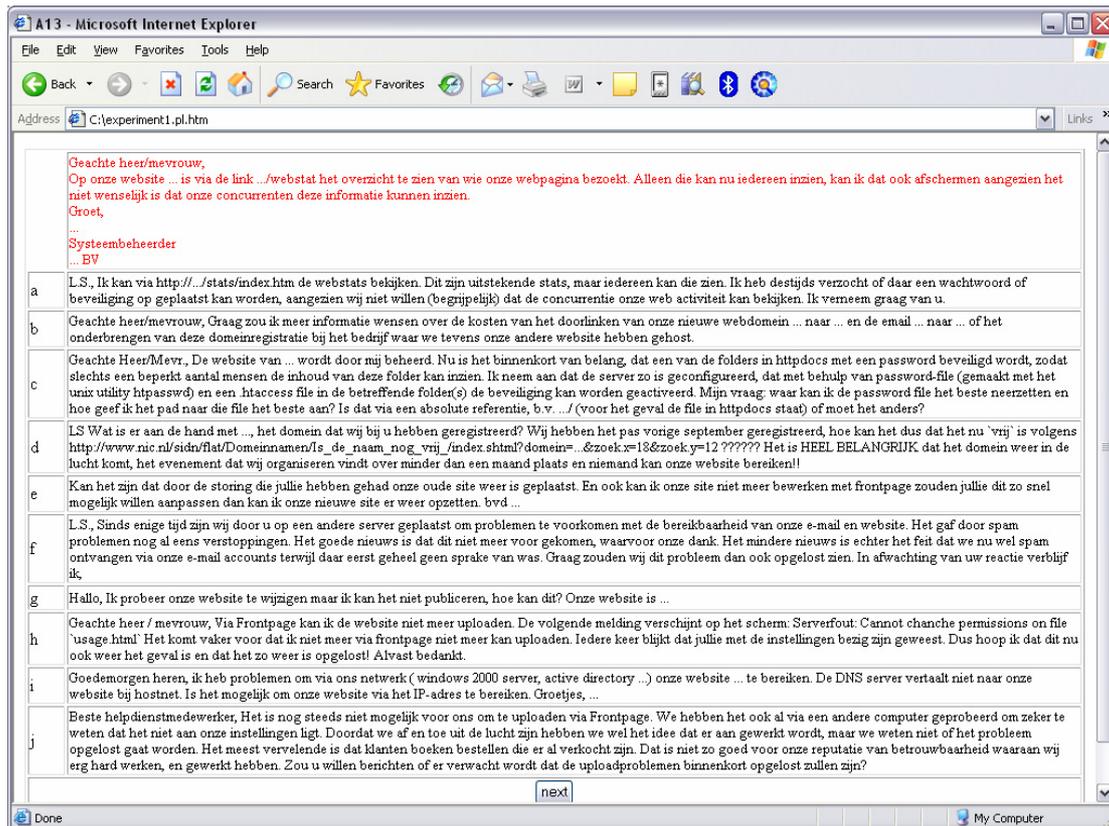
Er zijn vervolgens 35 nieuwe vragen, gesteld aan de helpdesk van Hostnet, willekeurig gekozen uit het ticketing systeem. Tijdens een pilot bleek dat 9 van deze 35 testvragen niet bruikbaar waren als experimentele vragen. Het betrof hier bijvoorbeeld vragen die leeg waren, bedrijfsgerelateerde notities of meldingen bevatten en/of zogenaamde auto-replies. Laatstgenoemde zijn automatisch gegenereerde berichten die per abuis in het ticketing systeem terecht zijn gekomen. Deze 9 vragen zijn niet meegenomen in het uiteindelijke experiment.

De 26 testvragen zijn op geautomatiseerde wijze stuk voor stuk als query ingevoerd in de module en de resultaten per techniek opgeslagen in een nieuwe database met resultaten. Per testvraag en toegepaste techniek worden de 10 meest correlerende vragen en bijbehorende antwoorden door het systeem geretourneerd. Voor dit aantal is gekozen omdat uit eerder onderzoek is gebleken dat de gemiddelde gebruiker na een bepaalde hoeveelheid resultaten niet meer objectief kan en wil beoordelen, het al eerder genoemde futility point (Blair 1980).

Tijdens de pilot bleek eveneens dat het beoordelen van één techniek (26 testvragen met telkens 10 resultaten) drie kwartier tot een uur in beslag kan nemen. Om deze reden is besloten uiteindelijk slechts de volgende information retrieval technieken te beoordelen:

- A. Geen toepassing van IR technieken (BASIC)
- B. Met toepassing van stopwoorden en stemming (STEMMING)
- C. Met toepassing van stopwoorden en TF.IDF (TF.IDF)
- D. Met toepassing van stopwoorden, stemming en TF.IDF (TF.IDF + STEMMING)

De database met resultaten is uiteindelijk gebruikt ter beoordeling van de verschillende technieken. Hiervoor is een Perl webscript ontwikkeld dat de oorspronkelijke testvraag uitleest en in een tabel eronder de lijst met resultaten uit de resultaten database weergeeft. Figuren 4.2 en 4.3 geven een screenshot weer van deze uiteindelijke output. Medewerkers van de helpdesk van Hostnet hebben deze als experts beoordeeld. De inhoud van beide screenshots is tevens, in leesbaarder vorm, terug te vinden in bijlage C. Vertrouwelijke informatie is wederom onherkenbaar gemaakt.

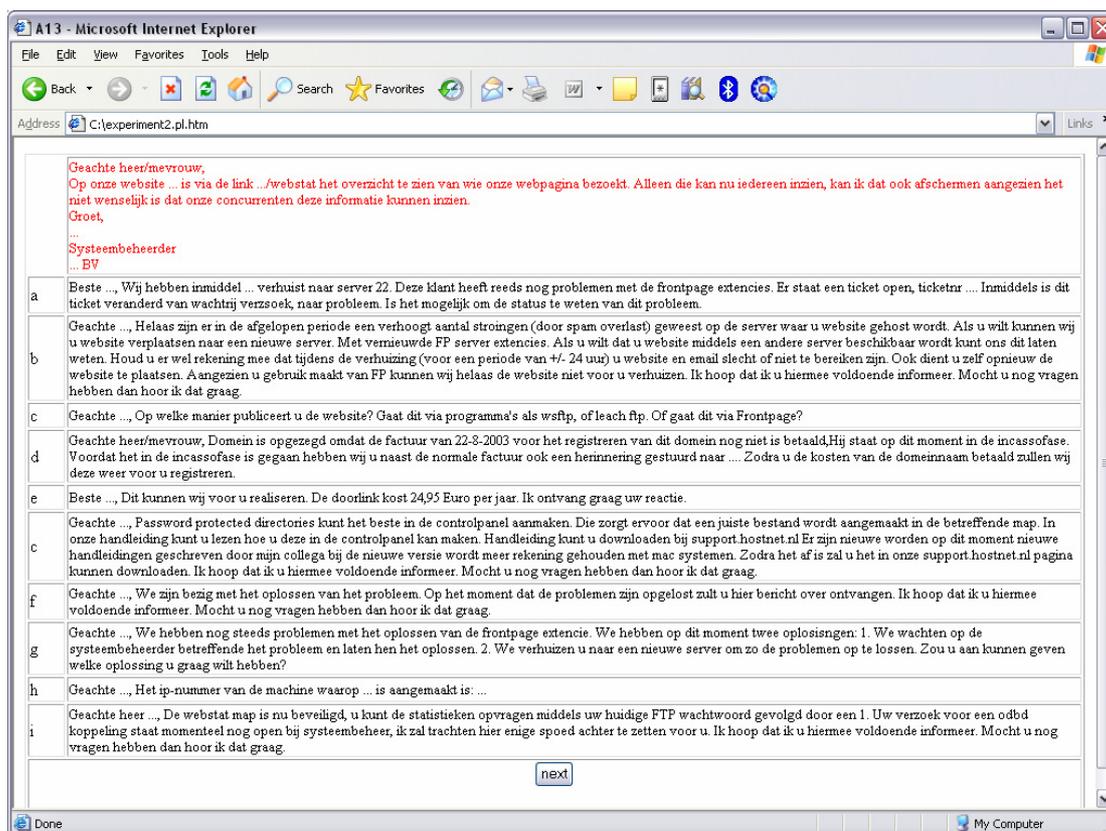


Figuur 4.2 Screenshot output experiment I

Binnen het onderzoek zijn er twee experimenten te onderscheiden. In beide zijn de gebruikte information retrieval technieken gerandomiseerd en geblindeerd beoordeeld. De testvragen werden per techniek in willekeurige volgorde gepresenteerd. De bijbehorende antwoorden werden tevens gerandomiseerd, met een willekeurige letter in plaats van de werkelijke rangorde positie. De experts hebben telkens per testvraag de resultaten beoordeeld en de overeenstemmende letters genoteerd.

In het eerste experiment kregen drie experts bij elke testvraag de, volgens het systeem, 10 hoogst scorende vragen te zien. Het uiteindelijke doel van het systeem is het automatisch genereren van antwoorden op nieuwe vragen. Om deze taak dichter te benaderen kregen in het tweede experiment twee andere experts de antwoorden te zien, behorende bij de 10 hoogst scorende vragen. Dit zijn de antwoorden die eerder door de helpdesk gegeven zijn

aan de klant als antwoord op de gestelde vragen (dus uit experiment 1). Een screenshot van de interface tijdens experiment 2 (voor dezelfde testvraag) is te vinden in de figuur 4.3.



Figuur 4.3 Screenshot experiment II

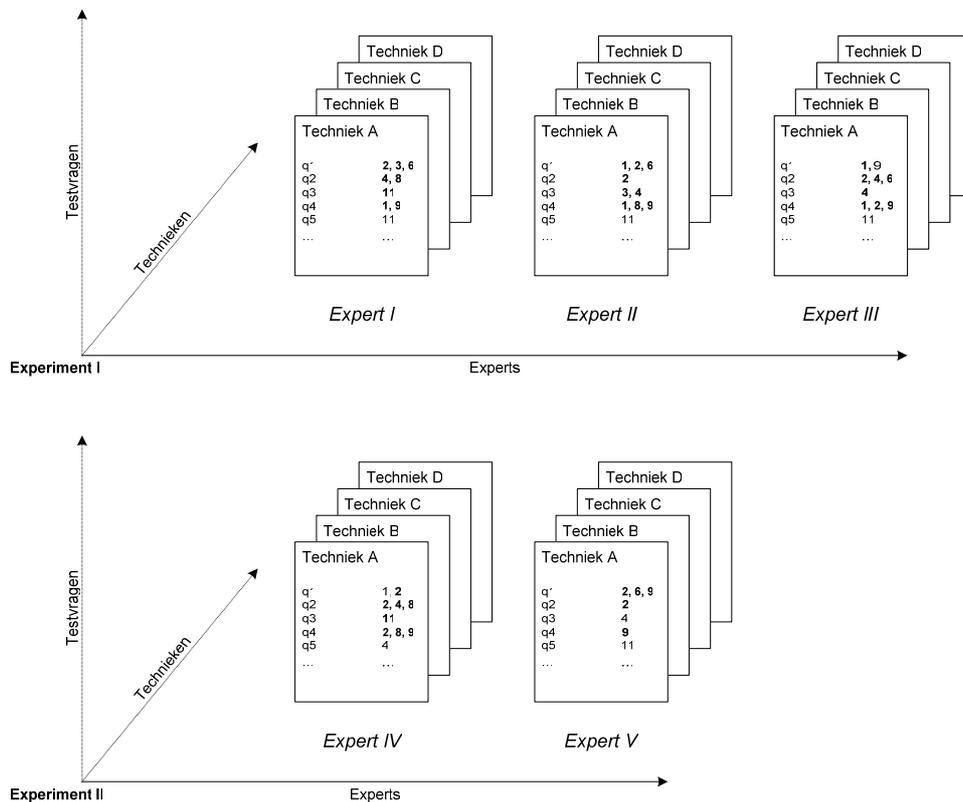
## 4.7 Evaluatiemethoden

Per testvraag en techniek gaven de experts aan welke resultaten volgens hen relevant waren. Elke expert leverde dus vier lijsten met beoordelingen op, één per toegepaste techniek. Elke lijst bestond weer uit een genummerd overzicht, waarbij elk nummer correspondeerde met een testvraag. Per testvraag zijn de letters van elk zinvol resultaat ingevuld, die zijn vertaald naar de werkelijke rangorde positie (van 1 tot 10, of 11 indien er geen zinvol resultaat geretourneerd werd). In figuur 4.4 is een schematische weergave hiervan te zien, met enkele fictieve waarden ingevuld.

Om tot een zinvolle evaluatie van technieken te komen (langs de z-as) zal er eerst een mate van overeenstemming tussen experts onderling bepaald moeten worden, aangezien waarschijnlijk niet iedere expert hetzelfde zal oordelen. Afhankelijk van deze mate van overeenstemming zal er gekozen kunnen worden om bijvoorbeeld alleen de antwoorden waar alle experts het unaniem over eens zijn als correct te bestempelen of om bijvoorbeeld de beoordeling van de meerderheid van de experts als leidraad te nemen. Vanaf dat moment is

de 'Experts'-dimensie weggenomen en kunnen er uitspraken worden gedaan over de technieken op basis van de beoordelingen van de testvragen.

Tevens zal worden gekeken in hoeverre de twee experimenten hetzelfde meten. Indien dit het geval is zullen de beoordelingen, voortkomend uit beide experimenten, samengenomen mogen worden. Indien zij een onvoldoende mate van samenhang vertonen zullen de beide experimenten afzonderlijk beschouwd moeten worden.



**Figuur 4.4 Grafische weergave van de meetresultaten**

Per techniek zal worden bepaald op welke gemiddelde positie het eerste als correct beoordeelde antwoord terug te vinden is; de in paragraaf 3.4 genoemde mean reciprocal rank (MRR). Deze waarde geeft de een belangrijke indicatie van effectiviteit van de betreffende techniek, variërend op een schaal van 1 (alle goede antwoorden op de eerste positie) tot 11 (geen enkel goed antwoord). Hiertoe wordt per techniek de gemiddelde positie van het eerste, goede antwoord bepaald.

Ook zal, per techniek, het percentage van testvragen worden beschouwd, waarin er een relevant antwoord bij de eerste 10 resultaten werd geretourneerd. Deze waarde geeft de precisie weer, tevens een maat van de effectiviteit van de toegepaste techniek. Indien een techniek in een relatief laag scorende MRR resulteert, maar toch een correct antwoord in de

top-10 retourneert, zal deze terdege van nut blijken in de case-based reasoning context van het uiteindelijke systeem.

Middels een non-parametrische toets voor gepaarde waarnemingen (Friedman toets) zal tenslotte de nulhypothese, dat de resultaten van dezelfde beoordelaars voor dezelfde taak onder verschillende omstandigheden (technieken) gelijk zijn, worden getoetst. Deze toets wordt overigens in de literatuur ook wel aangeduid als non-parametrische twee-factor-variantie-analyse en is voor twee beoordelaars gelijk aan de teken-toets (sign-test).



## 5 Resultaten

In dit hoofdstuk zullen alle resultaten uit de experimenten de revue passeren. Allereerst zal bekeken worden in hoeverre de experts het met elkaar eens zijn en in hoeverre de twee experimenten hetzelfde meten. In de voorgaande hoofdstukken zijn voorts een aantal evaluatiematen gedefinieerd aan de hand waarvan de toegepaste technieken nader zullen worden beoordeeld. In hoofdstuk 7 zal dieper worden ingegaan op de aard van de testvragen en er zullen enkele concrete voorbeelden gegeven worden van goed dan wel slecht presterende vragen, die uiteindelijk leiden tot de gepresenteerde resultaten.

### 5.1 Overeenstemming tussen experimenten en experts onderling

Tussen de doorsneden van beide experimenten is de waarde van Cohen's  $\kappa$  berekend. De doorsneden zijn de correcte antwoorden per experiment waar alle experts het over eens zijn. De waarde van Cohen's  $\kappa$  geeft een indicatie van de mate van overeenkomst tussen beoordelaars, waarbij gecorrigeerd wordt voor de invloed van toeval (Carletta 1996). Normaliter wordt een waarde van  $\kappa$  hoger dan 0,6 gezien als voldoende mate van overeenstemming. De waarde van  $\kappa$  bedraagt tussen de doorsneden van beide experimenten 0,672 en levert daarmee een voldoende mate van overeenstemming. Er kan dus gesteld worden dat beide experimenten in voldoende mate hetzelfde meten. Dit zorgt er voor dat binnen het vervolg van dit hoofdstuk de beoordelingen van de experts uit beide experimenten samengenomen zullen worden.

De beoordelingstaak lijkt, tussen de experts onderling bekeken, echter niet eenduidig te zijn. Waar de ene expert meerdere correcte resultaten op een testvraag vindt, vindt een ander bijvoorbeeld geen enkele. In tabel 5.1 is een kruistabel weergegeven van de onderlinge waarden van Cohen's  $\kappa$  tussen de experts voor alle testvragen en technieken. De gebruikte beoordelingen zijn terug te vinden in bijlage B.

|          | EXPERT 1 | EXPERT 2 | EXPERT 3 | EXPERT 4 | EXPERT 5 | MEAN         |
|----------|----------|----------|----------|----------|----------|--------------|
| EXPERT 1 |          | 0,469    | 0,614    | 0,605    | 0,533    | 0,555        |
| EXPERT 2 | 0,469    |          | 0,503    | 0,656    | 0,537    | 0,541        |
| EXPERT 3 | 0,614    | 0,503    |          | 0,568    | 0,558    | 0,561        |
| EXPERT 4 | 0,605    | 0,656    | 0,568    |          | 0,686    | 0,629        |
| EXPERT 5 | 0,533    | 0,537    | 0,558    | 0,686    |          | 0,579        |
| MEAN     | 0,555    | 0,541    | 0,561    | 0,629    | 0,579    | <b>0,573</b> |

Tabel 5.1 Kruistabel Cohen's  $\kappa$  tussen experts

Vanwege de totale, gemiddelde waarde van Cohen's  $\kappa$  (0,573) en het relatief grote aantal experts zal in het vervolg van dit hoofdstuk telkens uitgegaan worden van de antwoorden, waarover de meerderheid van de experts het eens is. De waarde van  $\kappa$  is te laag om uit te gaan van uitsluitend de als unaniem correct beoordeelde antwoorden. Dit punt en de mogelijkheid dat misschien alle experts gelijk hebben wordt verder verkend in hoofdstuk 7.

Het blijkt helaas niet mogelijk Cohen's  $\kappa$  tussen elke expert onderling per techniek te bepalen. Het aantal beoordelingsmogelijkheden (11) is daarvoor te groot in vergelijking met het aantal testvragen per techniek (26). Helaas bestaat er geen mogelijkheid bij de bepaling van Cohen's  $\kappa$  om hiervoor te corrigeren.

## 5.2 Percentages correcte resultaten

Per techniek is bepaald op welk percentage testvragen er een goed antwoord in de 10 gepresenteerde resultaten geretourneerd werd. Met andere woorden; welk percentage vragen kunnen er met de behandelde techniek correct beantwoord worden, op grond van de 10 hoogst scorende antwoorden. In onderstaande tabel zijn deze waardes nader uitgewerkt. Uit deze gegevens blijkt dat de zoekmodule het best functioneert zonder toepassing van enige information retrieval technieken.

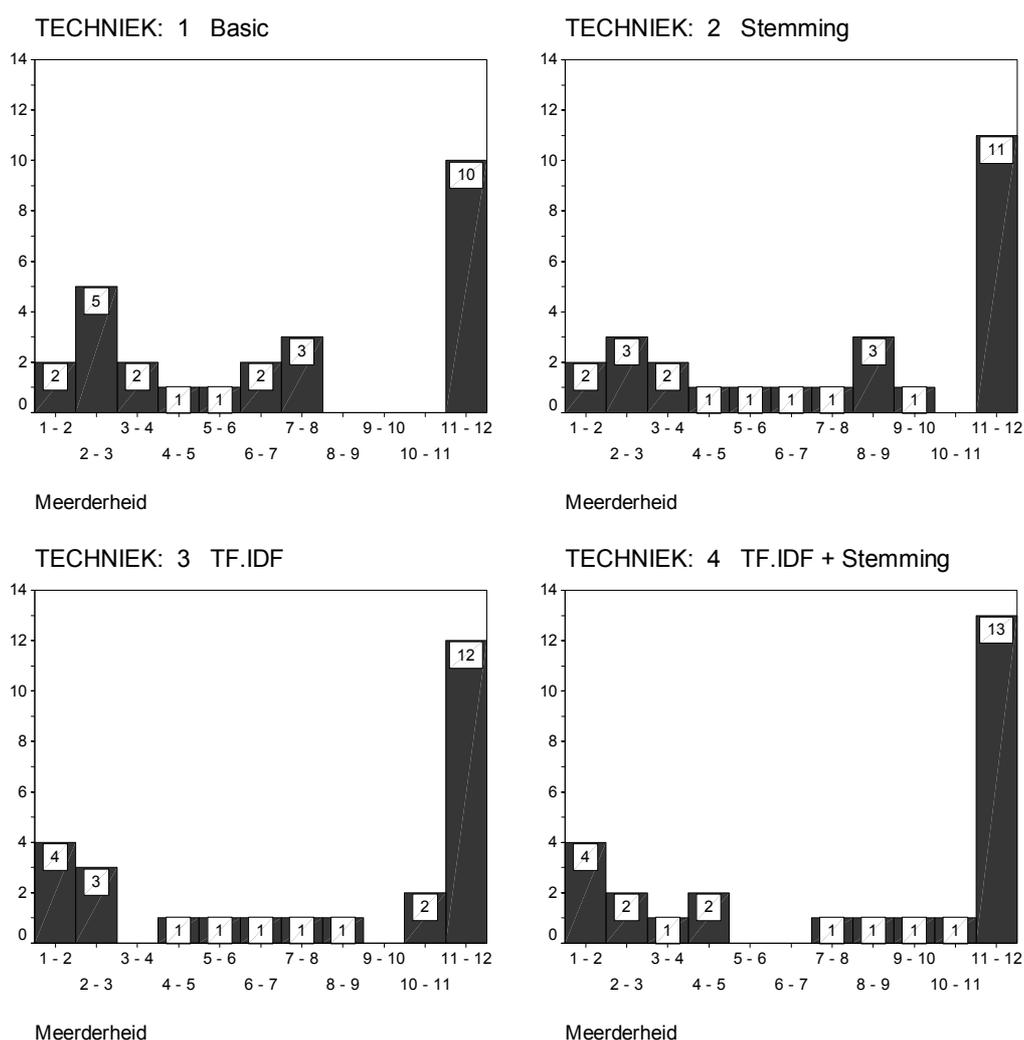
|                   |       |
|-------------------|-------|
| BASIC             | 61,5% |
| STEMMING          | 57,7% |
| TF.IDF            | 46,2% |
| TF.IDF + STEMMING | 50,0% |

Tabel 5.2 Percentage testvragen met een goed antwoord in de top-10, per techniek

Toepassing van TF.IDF lijkt met name een negatieve invloed te hebben. De inhoud van de testvragen lijkt daarom een belangrijke rol lijkt te spelen. Dit is tevens terug te zien in het feit dat bepaalde testvragen bij geen enkele techniek een correct antwoord opleveren, terwijl op andere vragen probleemloos meerdere goede antwoorden geretourneerd worden. Opvallend is dat de combinatie van TF.IDF met stemming betere resultaten oplevert dan zonder stemming, al is dit verschil slechts enkele procenten. In hoofdstuk 7 zal dieper worden ingegaan op de inhoudelijke aspecten van de testvragen, aan de hand van enkele voorbeelden.

### 5.3 Mean rank of first result (MRR)

Bovenstaande waarden zijn nader te specificeren naar de positie waarop het eerste goede antwoord staat, in plaats van uitsluitend te kijken of er een goed antwoord bij de top 10 zit. Hiermee wordt inzichtelijk gemaakt in hoeverre de gebruikte technieken in staat zijn correcte antwoorden hoog te laten eindigen, zoals eerder beschreven in paragraaf 3.4. De histogrammen van deze uitkomsten worden in onderstaande figuur per techniek weergegeven.



**Figuur 5.2** Histogrammen van MRR, per techniek

De meest rechter kolommen (categorie '11-12') geven de aantallen testvragen weer, waarop überhaupt geen correct antwoord werd geretourneerd of waarop de meningen van de experts niet in voldoende mate overeenkomen. Indien de combinatie van de antwoorden van de experts een lege verzameling voor een bepaalde testvraag opleverde, werd dit gezien als een

niet eenduidig en daarmee geen antwoord. Als zodanig werd aan deze testvraag de categorie '11-12' toegekend.

Het valt te beargumenteren dat een dergelijke lege verzameling in zijn geheel niet meegenomen dient te worden of tenminste een aparte categorie zou moeten vormen. Als de experts het niet eens zijn betekent dit namelijk niet noodzakelijkerwijs dat er geen correct antwoord gegeven wordt. In het vervolg van dit hoofdstuk zal hier echter wel van uit gegaan worden. Een verdere verkenning van dit punt is terug te vinden in hoofdstuk 7.

Bij de beoordeling van deze resultaten zijn twee zaken van belang. Ten eerste is een negatief beoordelingspunt de hoeveelheid testvragen waarop geen correct antwoord wordt teruggevonden (categorie '11-12'). De verhouding tussen deze en de overige categorieën is echter reeds in de vorige paragraaf aan de orde gekomen. Belangrijker is de verdeling van de goede antwoorden. Des te meer goede antwoorden een hoge score en daarmee een lage positie krijgen, des te beter presteert de techniek.

Uit de histogrammen wordt duidelijk dat de gebruikte technieken elkaar in dit opzicht niet veel ontlopen. Het blijkt dat wederom de toepassing van geen enkele information retrieval techniek de beste resultaten oplevert. Deze behaalt 11 correcte antwoorden in de top-5, ten opzichte van 9 bij de overige technieken.

Deze resultaten kunnen worden samengevat in een gemiddelde positie (van het eerste, goede antwoord) per techniek. Dit levert de gemiddelden in tabel 5.3 op. In dit geval geldt overigens hoe lager de score, hoe beter. De gegevens in deze tabel bevestigen de eerdergenoemde waarneming dat de toepassing van geen enkele techniek de beste resultaten oplevert.

|                   | MEAN RANK |
|-------------------|-----------|
| BASIC             | 6,54      |
| STEMMING          | 7,54      |
| TF.IDF            | 7,54      |
| TF.IDF + STEMMING | 7,03      |

Tabel 5.3 Gemiddelde positie van het eerste goede resultaat

## 5.4 Friedman

Middels de Friedman toets wordt de nulhypothese, dat de resultaten van dezelfde beoordelaars voor dezelfde taak onder verschillende omstandigheden (technieken) gelijk zijn, getoetst. Hierbij zijn de mean ranks per expert en techniek gebruikt, dus zonder de

meerderheid van de antwoorden te bepalen waar de experts het over eens waren. In onderstaande tabellen 5.4 en 5.5 staan de resultaten.

|                   | N<br>(EXPERTS) | MEAN | STD<br>DEVIATIE | MINIMUM | MAXIMUM |
|-------------------|----------------|------|-----------------|---------|---------|
| BASIC             | 5              | 6,52 | 0,30            | 6,15    | 6,96    |
| STEMMING          | 5              | 6,80 | 0,38            | 6,31    | 7,31    |
| TF.IDF            | 5              | 6,91 | 0,82            | 5,88    | 8,04    |
| TF.IDF + STEMMING | 5              | 6,68 | 0,64            | 6,19    | 7,77    |

**Tabel 5.4 Beschrijvende statistieken**

|                   | MEAN RANK |
|-------------------|-----------|
| BASIC             | 2,6       |
| STEMMING          | 2,6       |
| TF.IDF            | 2,8       |
| TF.IDF + STEMMING | 2         |

**Tabel 5.5 Gemiddelde rang van het eerste goede resultaat volgens Friedman**

Hieruit blijkt dat, op grond van de beoordelingen van de individuele experts (in tegenstelling tot eerdergenoemde resultaten op basis van meerderheid), TF.IDF + STEMMING het beste presteert met de laagste rang. Deze uitspraak is echter niet statistisch significant ( $p = 0,78$ ) en er kan dus gesteld worden dat de gebruikte techniek geen statistisch significante invloed heeft op de beoordelingen.

Het verschil tussen de means in tabel 5.4 en 5.3 wordt overigens veroorzaakt door de manier van berekenen. In tabel 5.3 wordt uitgegaan van de antwoorden, waar de meerderheid van de experts het over eens is. Vervolgens wordt daarover per techniek de gemiddelde rangorde berekend. In tabel 5.4 wordt uitgegaan van de gemiddelden per expert (en techniek), waar vervolgens nogmaals het gemiddelde over bepaald wordt. Hier worden dus ook beoordelingen meegenomen waar geen meerderheid het over eens is. Beide waardes zijn terug te vinden in de tabellen in bijlage B.



## 6 Conclusies en aanbevelingen

Dit onderzoek beoogde te analyseren in hoeverre reeds opgeslagen vragen, gesteld aan en beantwoord door de helpdesk van Hostnet, kunnen dienen als basis voor een case-based reasoning QA systeem, opererend met een statistische similariteitsmaat. De belangrijkste conclusie is dat deze syntactische aanpak behoorlijke resultaten oplevert. Het hoogst behaalde percentage van correct geretourneerde antwoorden in de top 10 van resultaten ligt rond de 60%. Dit betekent dat hetzelfde percentage van nieuwe cases succesvol beantwoord kan worden op grond van de geïndexeerde cases, de hier gebruikte implementaties en de 10 hoogst scorende resultaten. De gemiddelde positie van het eerste goede antwoord was aan de hoge kant, namelijk rond de 7.

Wat opvalt is dat de beste resultaten behaald worden zonder enige verfijning middels information retrieval technieken. Met name de toepassing van TF.IDF leverde niet de gehoopte en verwachte resultaten op. Zoals in paragraaf 7.2 zal worden behandeld is de inhoud van de berichten wellicht debet aan deze tegenvallende resultaten. In het licht van de investering in rekenkracht voor de toepassing van deze retrieval techniek is dit feit terdege opmerkelijk.

Alle geïndexeerde cases worden in de huidige implementatie bewaard in het geheugen. Op het gebruikte systeem was het daardoor niet mogelijk alle eerder voorgekomen cases te indexeren. Omdat elke geïndexeerde case op individuele basis wordt vergeleken met een nieuwe case was de module daarnaast erg langzaam, ook zonder toepassing van enige information retrieval technieken. Het is daarom aanbevelingswaardig de prestaties van de huidige implementatie op een sneller systeem te testen. Om de module in te zetten tijdens de dagelijkse werkzaamheden op de helpdesk van Hostnet zou er ook aandacht besteed moeten worden aan optimalisaties betreffende de opslag en retrieval. Retrieval zou bijvoorbeeld bespoedigd kunnen worden door ook de termen die uiterst weinig voorkomen niet te indexeren. Dit is in overeenstemming met Luhn's zogenaamde *lower cut-off* (Luhn 1961).

Een tweetal concrete verbeteringen van de in dit onderzoek gehanteerde implementaties liggen besloten in de gebruikte stemmer, die in sommige gevallen vreemde woordstammen produceerde. De negatieve resultaten van het toepassen van stemming hebben dus wellicht te maken met het in paragraaf 4.4.3 genoemde over- dan wel onderstemming. In hoeverre de gebruikte stemmer van invloed is geweest is, in een eventueel vervolgonderzoek, alleen na te gaan door middel van het gebruik van een andere stemmer. Helaas waren er tijdens de uitvoering van dit onderzoek geen andere, gratis stemmers voor de Nederlandse taal voorhanden. Een verandering in de wijze van termextractie kan wellicht ook voor verbeterde resultaten zorgen. Met name het herkennen van samengestelde woorden zou een gunstige invloed kunnen hebben op de resultaten. Nader onderzoek zou de invloed hiervan moeten

kunnen kwantificeren. In paragraaf 7.3 zullen meer structurele punten ter verbetering worden beschouwd.

Methodologisch gezien zijn er ook een aantal opmerkingen te plaatsen bij de gebruikte information retrieval technieken. Er werd, op basis van relevante literatuur, aangenomen dat de invloed van het verwijderen van stopwoorden gering zou zijn. Om deze reden en vanwege praktische overwegingen (met name tijd) is besloten deze techniek op zichzelf niet te evalueren. Zoals uit de stopwoordenlijst in bijlage A blijkt, komen hier een aantal domeinspecifieke termen in voor. De tegenvallende resultaten van de toegepaste technieken, vergeleken met de BASIC aanpak kunnen mogelijk dus ook verklaard worden door het verwijderen van de als stopwoorden aangemerkte termen. Verder onderzoek is noodzakelijk om deze vraag te kunnen beantwoorden.

Praktische overwegingen bepaalden ook de hoeveelheid testvragen die beoordeeld werd. Er bestaat een mogelijkheid dat het aantal testvragen te klein en daarom niet divers genoeg is geweest. De testvragen kwamen ook allemaal uit dezelfde maand. Na bestudering van de gebruikte testvragen leek het terdege mogelijk een aantal categorieën te onderscheiden in vragen die goed dan wel slecht presteerden. In het volgende hoofdstuk wordt nader hierop ingegaan, aan de hand van enkele voorbeelden.

De waarde van Cohen's  $\kappa$  en daarmee de overeenstemming tussen de beide experimenten is interessant. Volgens deze statistische waarde maakt het weinig verschil of experts de antwoorden dan wel de vragen beoordelen. De uiteindelijke beoordelingstaak zal er niet door afwijken. Voor de uiteindelijke toepassing van de zoekmodule in een case-based reasoning context is deze uitspraak van belang, omdat de re-use fase uit paragraaf 2.3 dus, in plaats van een overeenstemmende vraag, ook een bijbehorend antwoord als input kan hebben. In een ideale situatie worden beide gepresenteerd, zodat tijdens de re-use fase meer contextinformatie voorhanden is en er uiteindelijk een meer overwogen keus gemaakt kan worden ten aanzien van de relevantie van de gevonden cases.

Het verschil in beoordelingen tussen experts, wederom gekwantificeerd middels Cohen's  $\kappa$ , biedt ruimte voor discussie. Het lijkt er op dat er binnen de beoordelingstaak gradaties hebben bestaan tussen goed en fout. Sommige experts gaven aan dat het ontbreken van contextuele informatie soms een dergelijke ambiguïteit veroorzaakte. Het komt ook tijdens de dagelijkse werkzaamheden voor dat pas na een (e-mail) dialoog de eigenlijke vraag van een klant duidelijk wordt. Dit is mede de oorzaak van het feit dat de beoordelingen van de experts veelal niet op één lijn lagen. In paragraaf 7.1 zullen de consequenties van dit verschil en de gebruikte maat van overeenkomst (meerderheid) op de resultaten nader uiteengezet worden.

Helaas ontbrak het aan een gouden standaard om de resultaten objectief mee te toetsen. Het is wenselijk de in dit onderzoek gebruikte implementaties in een vervolgonderzoek te toetsen met een collectie, waarbij er wel een gouden standaard voorhanden is. Hiertoe kan elke case uit de in dit onderzoek gebruikte case (data)base handmatig worden beoordeeld, zodat deze informatie voor een vervolgonderzoek beschikbaar is. Dit is echter een uitermate tijdrovend proces en het is zeer de vraag of dit, op grond van de genoemde resultaten uit paragraaf 5.1, op objectieve wijze kan geschieden. Een alternatief schuilt in het gebruik van een reeds beoordeelde set van documenten. Er zijn verscheidene van dergelijke collecties beschikbaar, zoals bijvoorbeeld TREC<sup>1</sup> en CLEF<sup>2</sup>. Het nadeel is dat de documenten in deze collecties qua inhoud en stijl niet overeenstemmen met de in dit onderzoek gebruikte vragen. Hierdoor kan niet goed getoetst worden in hoeverre de in dit onderzoek gebruikte implementaties gebruikt kunnen worden binnen een case-based reasoning QA systeem. Slechts de retrieval capaciteiten kunnen op deze wijze worden geëvalueerd.

Het bleek niet mogelijk de gevonden resultaten te vergelijken met andere onderzoeken. Met name de case-based retrieval context zorgde ervoor dat er bijna geen soortgelijke onderzoeken terug te vinden zijn. Het enige onderzoeksproject dat enige gelijkenis vertoont is FALLQ, maar daar zijn helaas geen concrete resultaten van bekend gemaakt.

Met de huidige implementatie zal het vooralsnog niet mogelijk zijn automatisch vragen te beantwoorden, maar de prestaties zijn op zijn minst redelijk te noemen. In kennisondersteunende zin bieden de resultaten wel degelijk ruimte voor optimisme. Dit onderzoek had als doelstelling een eerste stap te zijn richting een eventuele ontwikkeling van een case-based reasoning QA systeem en/of corporate memory voor Hostnet. Zoals blijkt zijn de resultaten voor een dergelijke toepassing nog niet optimaal, maar de in dit onderzoek gebruikte aanpak lijkt wel degelijk levensvatbaar.

---

<sup>1</sup> Text REtrieval Conference; <http://trec.nist.gov/>

<sup>2</sup> Cross-Language Evaluation Forum; <http://www.clef-campaign.org/>



## 7 Discussie

Waar binnen dit onderzoek geen rekening mee is gehouden is de aard van de gestelde vraag. Zoals in paragraaf 7.2 duidelijk zal worden, komt het voor dat een klant een verzoek tot handelen doet in plaats van een concrete vraag te stellen. De intentie van een klant is in dat geval dus ook anders. Dergelijke vragen, met een afwijkende aard en intentie komen voor in de gebruikte cases en kunnen een zekere mate van ruis veroorzaken. In hoeverre dit de uiteindelijke resultaten van dit onderzoek heeft beïnvloed is niet eenduidig te zeggen. Het is noodzakelijk alle geïndexeerde cases handmatig door te lezen en te ontdoen van dergelijke afwijkende berichten om tot een bevredigend antwoord op deze vraag te komen.

Volgens (Kosseim, Beauregard et al. 2001) en (Koton 1988) is het niet mogelijk om op geautomatiseerde wijze de intentie van de auteur van een bericht te achterhalen aan de hand van uitsluitend tekstuele inhoud. Koton spreekt daarnaast ook van de noodzaak een document te interpreteren, voordat er accurate retrieval kan plaatsvinden. Daarom zou er zorg voor gedragen moeten worden dat er uitsluitend cases worden gebruikt met een eenduidige intentie.

Eén expert viel op dat de resultaten op een testvraag over het algemeen sterk op elkaar leken. Hierbij doelde hij niet op inhoud, maar volgens hem kregen voornamelijk overeenkomsten in schrijfstijl een hoge similariteitswaarde. Met het oog op retrieval in een case-based QA context is dit geen wenselijke eigenschap. Ze is echter inherent aan de gebruikte statistische similariteitsmaat. Overigens leken volgens hem niet alleen de goede resultaten qua stijl op elkaar, ook de foute. Indien een testvraag dus weinig inhoudelijk correcte resultaten retourneert, levert deze wel voornamelijk in stijl op elkaar gelijkende foute resultaten op.

De verschillen in beoordelingen tussen de experts hebben ertoe geleid dat er bij de evaluatie van de resultaten voor is gekozen uit te gaan van de antwoorden, waarover de meerderheid van de experts het eens was. In hoeverre dit van invloed is geweest op de uiteindelijke resultaten wordt in de volgende paragraaf behandeld aan de hand van andere maten van overeenstemming tussen experts. In paragraaf 7.2 zullen concrete voorbeelden van vragen gepresenteerd worden. Zoals zal blijken zijn er typen vragen aan te duiden die goed dan wel slecht presteren met de in dit onderzoek gebruikte implementaties. Tenslotte zullen in paragraaf 7.3 enkele structurele punten ter verbetering worden aangedragen.

## 7.1 Evaluatiemethoden

Zoals is gebleken zijn de experts het tijdens de beoordelingstaak het in geringe mate met elkaar eens geweest. De beoordelingstaak lijkt daarom, zoals in het vorige hoofdstuk al aangehaald, onduidelijk geweest te zijn. Het is helaas, vanwege het ontbreken van een gouden standaard, niet na te gaan welke antwoorden daadwerkelijk correct waren. De mogelijkheid bestaat dat er op grond van de gepresenteerde informatie geen eenduidiger keus te maken viel, bijvoorbeeld vanwege het ontbreken van contextuele informatie. Nader onderzoek zou hier meer duidelijkheid over kunnen verschaffen.

Het relatief grote aantal experts dat geraadpleegd werd draagt tevens bij aan deze verschillen. Andere onderzoekers vermijden dit probleem door bijvoorbeeld met slechts één of twee beoordelaars te werken, zoals in (Carmel, Shtalhaim et al. 2000). S. Rydin werkt in haar onderzoek met 4 experts en gebruikt onder meer de onderstaande maten van overeenstemming bij de beoordeling van hypo-/hyperniemparen door experts (Rydin 2002):

- Unaniem

De positie(s) van de goede antwoorden waar alle experts het over eens zijn. Indien de beoordeling van één expert per testvraag bezien wordt als een verzameling van antwoorden, komt deze waarde overeen met de doorsnede tussen de verzamelingen van alle experts. Dit is daarom de meest strikte vorm van beoordelen.

- Meerderheid

De positie(s) van de goede antwoorden waar minimaal de helft van de experts het over eens zijn. Dit is de maat van overeenstemming zoals die in hoofdstuk 5 is toegepast.

- Tenminste één

De positie(s) van de goede antwoorden, die minimaal door één expert wordt genoemd. Dit komt overeen met de vereniging tussen eerdergenoemde verzamelingen en is daarmee het minst streng.

Aangezien deze maat van overeenstemming van invloed is op de uiteindelijke resultaten worden hieronder de eerder gepresenteerde uitkomsten vergeleken met verschillende maten van overeenstemming. Uit tabel 7.1 blijkt dat de invloed van deze maat een behoorlijke invloed heeft op de resultaten. Met name tussen 'unaniem' en 'tenminste één' zit een factor twee verschil.

|                   | UNANIEM | MEERDERHEID | TENMINSTE ÉÉN |
|-------------------|---------|-------------|---------------|
| BASIC             | 42,3%   | 61,5%       | 84,6%         |
| STEMMING          | 34,6%   | 57,7%       | 76,9%         |
| TF.IDF            | 38,5%   | 46,2%       | 73,1%         |
| TF.IDF + STEMMING | 38,5%   | 50,0%       | 73,1%         |

Tabel 7.1 Herberekening percentage testvragen met een goed antwoord in de top-10

Ook in tabel 7.2 wordt duidelijk wat de invloed is van de gekozen maat van overeenkomst. In de figuren op de volgende pagina worden de histogrammen behorende bij deze gemiddelden weergegeven (voor 'unaniem' en 'tenminste één'). Ook hier is het effect op de resultaten duidelijk zichtbaar.

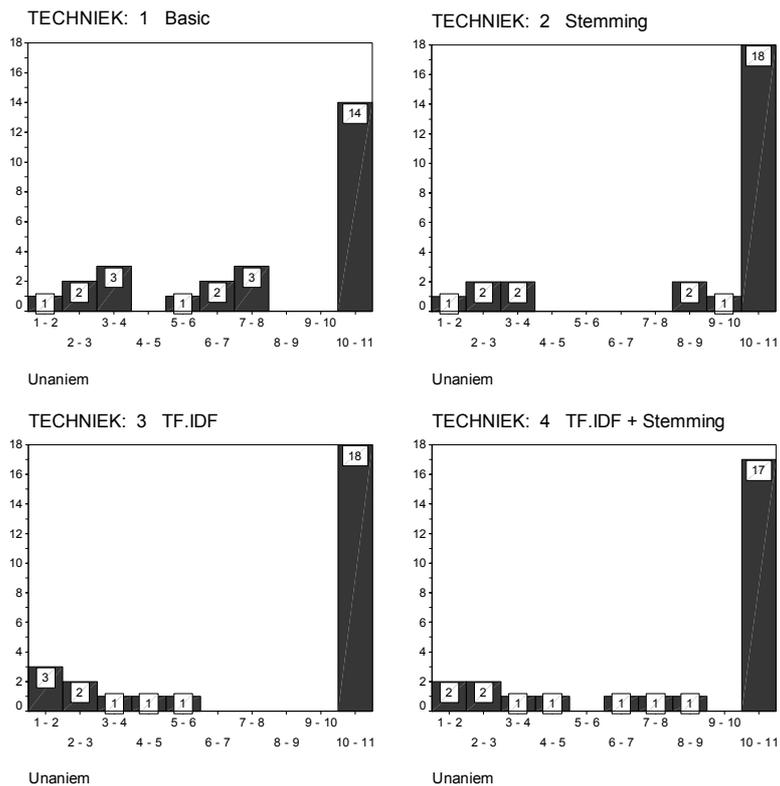
|                   | UNANIEM | MEERDERHEID | TENMINSTE ÉÉN |
|-------------------|---------|-------------|---------------|
| BASIC             | 7,92    | 6,54        | 4,77          |
| STEMMING          | 9       | 7,31        | 4,85          |
| TF.IDF            | 8,31    | 7,38        | 4,88          |
| TF.IDF + STEMMING | 8,46    | 7,54        | 5             |

Tabel 7.2 Herberekening gemiddelde positie van het eerste goede resultaat

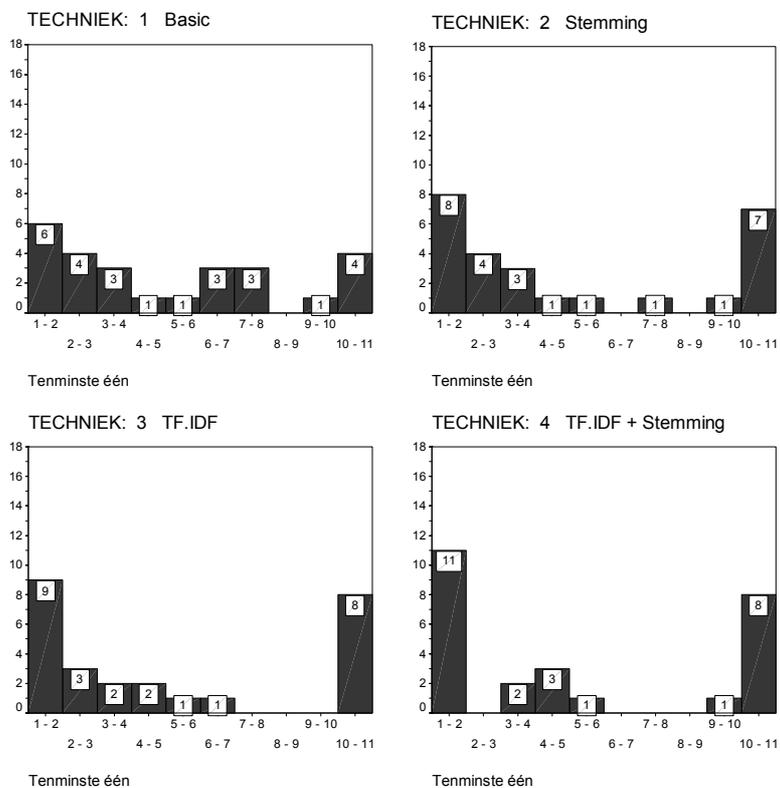
Bij de getallen uit tabel 7.2 is het mogelijk het non-parametrische equivalent van Student's t-toets te gebruiken, om te zien of de gemiddelde positie per techniek en maat van overeenkomst significant afwijkt van wat er, op basis van toeval verwacht kan worden. In tabel 7.3 staan de uitkomsten van deze toets van Wilcoxon. Tussen de gemiddelden bij 'unaniem' en 'tenminste één' en tussen 'meerderheid' en 'tenminste één' levert dit voor elke techniek een significant verschil op ( $p < 0.01$ ). Tussen 'meerderheid' en 'unaniem' zijn de gemiddelden op 0.05-niveau significant, met uitzondering van de toepassing van TF.IDF + STEMMING. Dit betekent dat de maat van overeenkomst een statistisch significante invloed heeft op de uiteindelijke resultaten.

|                   | TENMINSTE ÉÉN -<br>UNANIEM<br>P | MEERDERHEID -<br>UNANIEM<br>P | MEERDERHEID -<br>TENMINSTE ÉÉN<br>P |
|-------------------|---------------------------------|-------------------------------|-------------------------------------|
| BASIC             | 0,0004                          | 0,0116                        | 0,0033                              |
| STEMMING          | 0,0003                          | 0,0180                        | 0,0004                              |
| TF.IDF            | 0,0020                          | 0,0277                        | 0,0070                              |
| TF.IDF + STEMMING | 0,0009                          | 0,2059                        | 0,0032                              |

Tabel 7.3 p-waardes behorende bij de toets van Wilcoxon

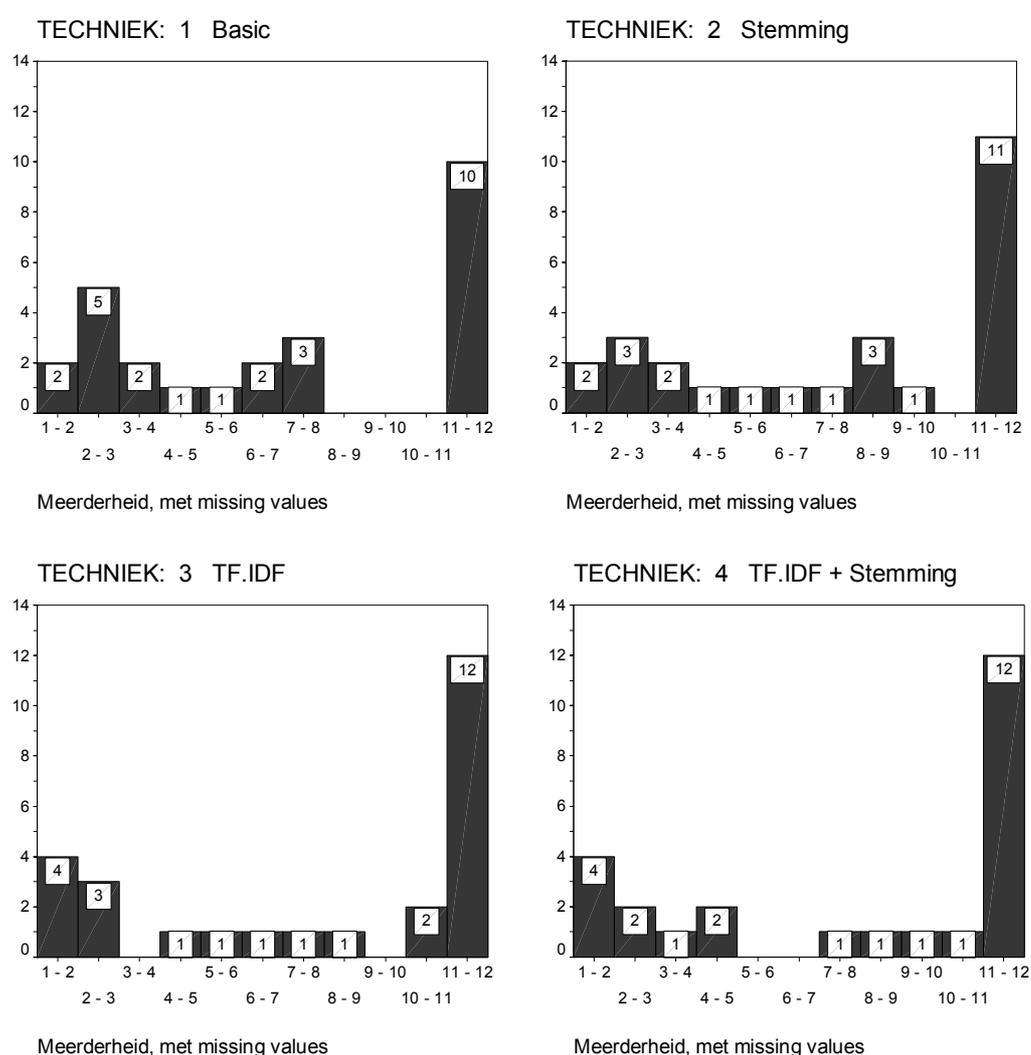


**Figuur 7.1 Histogrammen van MRR, per techniek (Unaniem)**



**Figuur 7.2 Histogrammen van MRR, per techniek (Tenminste één)**

In hoofdstuk 5 werd er vanuit gegaan dat de antwoorden, waar de experts het niet in meerderheid over eens waren, tevens foute antwoorden waren. Als de experts het niet in meerderheid eens zijn betekent dit echter niet noodzakelijkerwijs dat er geen correct antwoord gegeven wordt. Ter referentie worden daarom in figuur 7.3 de resultaten weergegeven indien de testvragen, waar geen meerderheid over bestaat, niet worden meegerekend in de categorie '11-12'. Deze kolom bestaat dus uitsluitend uit de aantallen testvragen, waar de experts het in meerderheid over eens waren dat er geen correct resultaat werd geretourneerd. Het is duidelijk dat deze aanpassing nauwelijks effect heeft op de resultaten. Alleen bij TF.IDF + STEMMING is er een miniem verschil zichtbaar, de overige resultaten zijn identiek.



Figuur 7.3 Histogrammen bij MRR, lege doorsneden buitengesloten

## 7.2 Voorbeelden van testvragen

Voor een aantal testvragen (2 van de 26) wordt er bij geen enkele techniek en expert een correct resultaat in de top-10 geretourneerd. Bij 5 andere testvragen vielen de resultaten ook aanzienlijk tegen. Zij presteerden goed bij één of twee technieken, maar retourneerden over het algemeen geen enkele als correct beoordeeld antwoord. In hoeverre de inhoud van berichten bijdraagt aan een negatieve retrieval performance zal in het vervolg van deze paragraaf, aan de hand van deze 7 slecht presterende vragen, behandeld worden. De nummers van de getoonde testvragen komen overigens overeen met de nummers in bijlage B (meetresultaten). Eventuele namen en/of andere vertrouwelijke informatie zijn uit de hier weergegeven voorbeelden verwijderd.

### 7.2.1 Slecht presterende vragen

Onderstaande vraag (Q9) is een wedervraag op een eerder gegeven oplossing die per abuis als nieuwe vraag in het systeem terechtgekomen is. Hierdoor mist de informatie die in het bericht staat zijn context. Er wordt immers verwezen naar een eerder bericht waarvan de inhoud niet meegenomen wordt bij de uiteindelijke retrieval.

Q9

Oke bedankt ik heb gezien dat de dns verwijzing nu goed staan, maar kan nog steeds geen mail ontvangen maar dat klopt dus, want dat kan wel 48 uur duren dus ??

Voor de vraag op de volgende pagina (Q17) geldt dat de eigenlijke vraag als quote is bijgevoegd. Deze wordt door de module ook als zodanig herkend en daarom niet meegenomen bij de similariteitsbepaling. Bij deze testvraag valt ook nog iets anders op; het systeem heeft de automatische ondertekening niet herkend. Om deze reden is de gehele inhoud van de ondertekening meegenomen, wat voor een aanzienlijke verstoring van de resultaten zorgt. De ondertekening bevat immers meer woorden dan de feitelijke vraag. Dit is ook het geval bij de daaropvolgende testvraag (Q21).

Q17

Hoi ...,

Helaas kan ik niet rechtstreeks naar je mailen, dus doe ik het op deze manier.

Vriendelijke groet,

...

Deze e-mail en de inhoud daarvan is vertrouwelijk. Indien dit bericht niet voor u bestemd is, verzoeken wij u deze e-mail direct aan ons te retourneren en daarna te vernietigen. In dit geval is het ook niet toegestaan deze e-mail en de inhoud daarvan te gebruiken, kopiëren of openbaar te maken aan derden. Onze onderneming sluit elke aansprakelijkheid uit in verband met het niet juist, onvolledig of niet tijdig overkomen van de informatie in deze e-mail.

----- Original Message -----

From: ...

To: ...

Cc: ...

Sent: Tuesday, April 19, 2005 3:02 PM

Subject: Geleden schade t.g.v. verhuizen servers

Beste ...,

Onlangs hebben wij gesproken over de schade die ontstaan is door de verhuizing van jullie servers. Je zou mij nog berichten wat ... als tegemoedkoming zou doen voor deze geleden schade. Ik verzoek je om mij uiterlijk deze week hiervan op de hoogte te brengen.

Met vriendelijke groet,

...

Deze e-mail en de inhoud daarvan is vertrouwelijk. Indien dit bericht niet voor u bestemd is, verzoeken wij u deze e-mail direct aan ons te retourneren en daarna te vernietigen. In dit geval is het ook niet toegestaan deze e-mail en de inhoud daarvan te gebruiken, kopiëren of openbaar te maken aan derden. Onze onderneming sluit elke aansprakelijkheid uit in verband met het niet juist, onvolledig of niet tijdig overkomen van de informatie in deze e-mail.

Q21

LS

Momenteel draait de website van ... bij jullie (...). Nu heb ik soms dat als ik een pagina upload een plaatje dan wel een button niet getoond krijg. De onderliggende link werkt wel alleen zie ik een rood kruisje bij het desbetreffende plaatje of een klein streepje.

Ik gebruik Coffeecup om de pagina's te maken. Weten jullie hoe het komt dat ik de afbeeldingen niet zie?

Bijvoorbaat dank

...

This message contains information that may be privileged or confidential and is the property of ... It is intended only for the person to whom it is addressed. If you are not the intended recipient, you are not authorized to read, print, retain, copy, disseminate, distribute, or use this message or any part thereof. If you receive this message in error, please notify the sender immediately and delete all copies of this message.

De volgende vragen (Q22 en Q5) zijn erg specifiek van aard. Het is zeer de vraag of een overeenkomstige vraag überhaupt al eerder gesteld is aan de helpdesk. De eerste vraag (Q22) bestaat bovendien uit twee subvragen. De tweede subvraag is een verzoek tot handeling, in plaats van een concrete vraag. Omdat de volledige inhoud wordt meegenomen bij de similariteitsbepaling heeft dit waarschijnlijk voor verstoring van de resultaten gezorgd. Dit is een van de eerder genoemde vragen waarop bij geen enkele techniek een als correct beoordeeld resultaat werd behaald.

Q22

geachte heer mevrouw,

ik heb bij u ...eu vastgelegd waarom krijg ik dan een pdf met de informatie dat ik een .nl domein heb vast gelegd. zou u voor me willen kijken of ik ...eu bij u in het bestand staat

Met vriendelijke groeten,

...

Q5

Hallo,

Hier is als bijlage het script voor het versturen van nieuws brieven via mijn site. de berichten komen in hotmail e.d. wel goed aan maar via outlook express in onleesbare HTML code. ook als er een berichtje voor iemand is, krijgt hij een mailtje, die komt ook in HTML aan ik zal een voorbeeld hieronder plakken, hoe het binnen komt in de Outlook Express 6. Ik hoop echt dat je de oorzaak kan vinden, waar dit aan kan liggen.

Vriendelijke groet,

...

Q24

Beste helpdesk,

Ik krijg vanaf gister deze emails van aanmeldingen binnen en constateerde dat ze afkomstig waren van dit email adres.

pemail@server40.hostnet.nl

Kan het zijn dat u zich GRATIS probeert aan te melden in onze database? Als zo dan verzoeken wij u om de gegevens zo volledig mogelijk in te vullen en te mailen naar sales@...nl

Mocht het bovenstaande niet het geval zijn dan verzoek ik om na te gaan of dit soort mailtjes geblokt kunne worden.

Vast met vriendelijke groet,

...

Bij bovenstaande vraag (Q24) komt, naast de uiterst specifieke aard tevens naar voren wat de negatieve invloed is van een grote hoeveelheid aan spelfouten. Hierdoor wordt maar een gering aantal van de woorden uit de vraag meegenomen als zoektermen, omdat verkeerd gespelde woorden niet voorkomen in de collectie. Dit is de andere testvraag waarop geen enkel als correct beoordeeld resultaat werd behaald. Bij de volgende vraag (Q16) is het moeilijk een eenduidige oorzaak aan te geven van de slechte resultaten. De geretourneerde resultaten zijn erg gevarieerd.

Q16

Geachte heer/mevrouw,

Vandaag heb ik via het Plesk Control Panel een mailgroep aangemaakt, bestaande uit 26 e-mailadressen. Nu wil ik via de webmail een e-mail verzenden aan deze mailgroep. Mijn vraag is: hoe gaat dat precies in zijn werk? Ik kan er namelijk niks over terugvinden in de help/FAQ.

Ik hoop dat u mij verder kunt helpen.

Met vriendelijke groet,

...

Op grond van de gebruikte testvragen kunnen er vier categorieën worden aangemerkt, waarin de retrieval resultaten tegenvallen. Deze zijn: te specifieke vragen, onduidelijke (automatische) ondertekeningen, afwijkende taak en spelfouten. Met 'afwijkende taak' wordt bijvoorbeeld een verzoek in plaats van een concrete vraag bedoeld. Deze categorie kwam overigens niet vaak voor in de gebruikte testvragen.

Slechts de automatische ondertekeningen kunnen aan de kant van het systeem opgelost worden. De te specifieke vragen kunnen niet beantwoord worden, omdat een dergelijke vraag hoogstwaarschijnlijk nog niet eerder is voorgekomen. Indien een e-mail een verzoek bevat tot een handeling in plaats van een concrete vraag kan het systeem er niets mee. Dergelijke e-mails zouden dan ook niet opgeslagen dienen te worden. Het is echter, zoals aan het begin van dit hoofdstuk al aangehaald, problematisch deze op een geautomatiseerde wijze te herkennen. Spelfouten zouden aangepakt kunnen worden middels het gebruik van een lexicon, waarmee getracht kan worden fouten automatisch en/of onder menselijke supervisie te corrigeren. Vanwege het grote aantal domeinspecifieke termen en de constante ontwikkeling van nieuwe termen levert dit echter praktische problemen op.

### 7.2.2 Goed presterende vragen

Naast de voorbeelden van tegenvallende resultaten zijn er ook testvragen te onderscheiden die relatief goed presteren. Enkele voorbeelden hiervan zijn hieronder weergegeven. Wat direct opvalt is de lengte van dergelijke testvragen. Hierdoor wordt ruis, veroorzaakt door het gebruik van natuurlijke taal (met name onnodige beleefdheden en/of uitweidingen) tegengegaan. De zoekmodule kan immers geen onderscheid maken tussen betekenisvolle en minder betekenisvolle termen. Hiervoor is een zekere mate van semantiek (bijvoorbeeld in de vorm van een ontologie) vereist. Op grond van de in dit onderzoek gebruikte, statistische methoden is dit echter niet realiseerbaar. Voorts lijkt het er op dat het specifieke, concrete

vragen betreft die in ruime mate eerder zijn voorgekomen, blijktens de gevonden resultaten. Slechts inspectie van de volledige case (data)base kan hier uitsluitel over bieden.

**Q3**

Beste Heer-Mevrouw,

Graag wil ik ook ruimte kopen om de website te bouwen. Kunt u mij verder ook nog iemand aanraden die een mooie website kan bouwen?

Bedankt en groet,

...

**Q10**

Hallo helpdesk,

Alweer enige tijd is mijn site en het controlpanel niet bereikbaar. Is er een storing gaande? Ik kan niets op jullie site terugvinden.

Met vriendelijke groeten,

...

**Q20**

Goedenavond,

Voor een perl script heb ik het absolute path naar een directory nodig, wat is dit pad tot -sitenaam-.nl?

Met vriendelijke groet,

...

**Q26**

Geachte heer/ mevrouw,

Dhr. ... van het bedrijf ...wil graag het domein ... en ... verhuizen. Nu willen wij bij deze het .nl domein opzeggen bij hostnet. Nu moet het .com unlocked worden wil het verhuist kunnen worden. Wat moeten wij daarvoor doen of moeten wij de termijn afwachten?

Graag hierop uw antwoord.

Met vriendelijke groet,

...

### 7.3 Verdere punten ter verbetering

De kennisarme representatie van de gebruikte cases leidde ertoe statistische information retrieval methoden in te zetten. Zoals beschreven in paragraaf 3.1 maakten statistische information retrieval technieken van oudsher gebruik van vooraf gedefinieerde, afgebakende woordenlijsten. Het gebruik van natuurlijke taal daarentegen veroorzaakt een zekere mate van ruis indien dergelijke technieken zonder enige vorm van correctie worden toegepast (Anjewierden, Huijsen et al. 2005). Zoals is gebleken uit de voorbeelden in de vorige paragraaf hebben typefouten en non-relevante woorden in de testvragen (en opgeslagen cases) inderdaad een negatieve invloed op de retrieval resultaten van een op statistische gronden opererend algoritme.

Op grond hiervan ligt een verbetering dus besloten in de kwaliteit van de opgeslagen cases. Indien een menselijke *case editor* deze zou doornemen en ontdoen van dergelijke verstoringen nemen de resultaten hoogstwaarschijnlijk in positieve zin toe. Uiteraard zullen toekomstige cases op gelijke wijze opgeslagen moeten worden, naar voorbeeld van het eerder behandelde case-based reasoning systeem HOMER. De queries die het systeem te verwerken krijgt zullen tevens op enigerlei wijze bewerkt moeten worden. Hiertoe zou een medewerker van de helpdesk bijvoorbeeld het relevante gedeelte uit een bericht moeten kunnen selecteren als query.

Een andere manier van gebruikersinteractie waar de resultaten waarschijnlijk van zullen profiteren, zou het toevoegen van een vorm van *relevance feedback* zijn. Gebruikers van het systeem kunnen op die wijze resultaten selecteren die een zekere mate van overeenkomst met de testvraag vertonen en teruggeven aan het systeem. Het systeem gebruikt vervolgens deze informatie voor een nieuwe zoekopdracht, waarvan de resultaten volgens (Robertson and Jones 1976) zullen verbeteren. In hoeverre deze vormen van interactie de resultaten binnen het huidige toepassingsgebied zullen verbeteren is slechts na te gaan middels een vervolgonderzoek.

Een andere mogelijke verbetering zou de introductie van een beperkte mate van semantiek zijn. Dergelijke semantiek zou bijvoorbeeld kunnen komen uit een ontologie, waarin de domeinspecifieke termen opgenomen zijn, naar voorbeeld van CBR-ANSWERS (paragraaf 2.6.4). Deze kunnen een hoger gewicht krijgen dan niet-relevante termen of er kan voor gekozen worden uitsluitend van domeinspecifieke termen uit te gaan. Hierbij kan een syntactische similariteitsmaat behouden blijven. Indien alleen uitgegaan wordt van de domeinspecifieke termen bij de retrieval, is er tegelijkertijd een snelheidswinst te behalen vanwege het geringere aantal termdimensies. In hoeverre dit invloed heeft op de resultaten is slechts door middel van een vervolgonderzoek te bepalen. Met behulp van een dergelijke ontologie wordt tevens een mogelijke basis gelegd voor de ontwikkeling van een meer

gestructureerde FAQ. Volgens de auteurs in (Carmel, Shtalham et al. 2000) kan een dergelijke gestructureerde aanpak, mits voldoende onderhouden, de kwaliteit van antwoorden op veel voorkomende vragen aanzienlijk verbeteren.

Een diepere vorm van semantiek kan toegevoegd worden indien ook relaties tussen domeinspecifieke termen (concepten) geëxpliciteerd zouden worden. Dit vereist echter een aanzienlijke inspanning om te realiseren. Met een dergelijk semantisch netwerk is het tevens mogelijk een semantische similariteitsmaat te hanteren, zoals in het case-based reasoning systeem HOMER. In hoeverre de kosten van de ontwikkeling van een dergelijk kennisgebaseerd netwerk opwegen tegen de baten zou in een vervolgonderzoek bepaald moeten worden.

Zoals is gebleken lijken sommige vragen überhaupt niet te beantwoorden en het vermoeden rijst dat het aantal gebruikte cases te klein is. Hierdoor vallen de resultaten niet zozeer tegen omdat de juiste cases niet gevonden worden, maar omdat er geen overeenkomstige cases bestaan in de case (data)base. Wat de precieze invloed is geweest van het aantal geïndexeerde cases op de resultaten is moeilijk in te schatten. Vanuit een case-based reasoning perspectief zou een grotere hoeveelheid aan cases verbetering op moeten leveren. Dit is echter geen structurele oplossing; slechts de kans dat een overeenkomstige, nieuwe case eerder is voorgekomen wordt ermee vergroot. Of deze ook daadwerkelijk teruggevonden wordt is mede afhankelijk van andere factoren. Een vorm van longitudinaal onderzoek zou hier meer inzicht in kunnen verschaffen. Daarmee zou het effect van het groeien van de case (data)base op de retrieval resultaten over de tijd gemeten kunnen worden.

De ontwikkeling van het case-based reasoning QA systeem HOMER (paragraaf 2.6.3) bevestigt deze notie en geeft aan dat een groot aantal opgeslagen cases geen garantie is voor een adequate retrieval (Göker and Roth-Berghofer 1999a). Volgens de auteurs heeft het toevoegen van een semantische dimensie aan het zoekproces voorkeur boven het alsmaar uitbreiden van het aantal opgeslagen cases.

## A Stopwoordenlijst

een  
het  
van  
niet  
klant  
dat  
met  
heb  
dit  
naar  
bij  
voor  
mijn  
kan  
www  
mail  
jullie  
server  
zijn  
deze  
aan  
als  
graag  
nog  
hostnet  
maar  
website  
ook  
mij  
worden  
wat  
kunnen  
wil  
belt  
die  
vraag  
mogelijk

via  
domeinnaam  
geen  
site  
heeft  
hebben  
geachte  
zou  
dan  
wij  
meer  
wel  
email  
domein  
moet  
hoe  
hallo  
adres  
beste  
kunt  
wordt  
door  
werkt  
heer  
hij  
anders  
onze  
over  
krijg  
the  
ontvangen  
staat  
probleem  
http  
info  
gaat  
weten

dus  
hier  
dns  
volgende  
helpdesk  
wilt  
willen  
even  
ons  
andere  
problemen  
nieuwe  
bedankt  
gegevens  
was  
ftp  
maken  
goed  
ben  
naam  
dank  
echter  
alvast  
laten  
komt  
sturen  
contact  
daar  
pakket  
weer  
iets  
waar  
gebruik  
zie  
php  
binnen

## B Meetresultaten

In onderstaande tabellen zijn de hoogst beoordeelde posities per expert en testvraag weergegeven. De hoogste beoordelingen, waar de meerderheid van de experts het over eens was, worden telkens als laatste rij (per techniek) vermeld. De gemiddelden uit paragraaf 5.3 worden in de laatste kolom getoond.

| BASIC        |           |   |   |   |    |    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |             |
|--------------|-----------|---|---|---|----|----|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|-------------|
| Expert       | Testvraag |   |   |   |    |    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             | Mean        |
| (experiment) | 1         | 2 | 3 | 4 | 5  | 6  | 7 | 8 | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26          |             |
| 1 (I)        | 3         | 2 | 4 | 5 | 11 | 11 | 6 | 2 | 1  | 1  | 1  | 1  | 11 | 11 | 9  | 11 | 7  | 2  | 11 | 7  | 11 | 11 | 2  | 11 | 11 | 6           | 6,50        |
| 2 (I)        | 3         | 2 | 3 | 7 | 3  | 5  | 6 | 2 | 11 | 2  | 1  | 1  | 11 | 11 | 11 | 11 | 7  | 2  | 5  | 7  | 11 | 11 | 11 | 11 | 11 | 6           | 6,62        |
| 3 (I)        | 3         | 2 | 3 | 3 | 3  | 2  | 6 | 2 | 11 | 2  | 1  | 1  | 11 | 11 | 11 | 11 | 7  | 2  | 5  | 7  | 6  | 11 | 11 | 11 | 11 | 6           | 6,15        |
| 4 (II)       | 3         | 2 | 4 | 7 | 11 | 2  | 6 | 2 | 11 | 5  | 1  | 1  | 11 | 11 | 11 | 11 | 7  | 2  | 5  | 7  | 11 | 11 | 11 | 11 | 11 | 6           | 6,96        |
| 5 (II)       | 3         | 3 | 7 | 7 | 1  | 2  | 6 | 1 | 11 | 1  | 3  | 1  | 11 | 7  | 11 | 4  | 11 | 2  | 5  | 7  | 11 | 11 | 11 | 11 | 11 | 6           | 6,35        |
|              |           |   |   |   |    |    |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>6,52</b> |             |
| Meerderheid  | 3         | 2 | 4 | 7 | 3  | 2  | 6 | 2 | 11 | 2  | 1  | 1  | 11 | 11 | 11 | 11 | 7  | 2  | 5  | 7  | 11 | 11 | 11 | 11 | 11 | 6           | <b>6,54</b> |

| STEMMING     |           |   |   |    |    |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |             |
|--------------|-----------|---|---|----|----|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|-------------|
| Expert       | Testvraag |   |   |    |    |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             | Mean        |
| (experiment) | 1         | 2 | 3 | 4  | 5  | 6 | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26          |             |
| 1 (I)        | 11        | 5 | 8 | 1  | 11 | 9 | 11 | 3  | 1  | 2  | 9  | 3  | 4  | 8  | 11 | 11 | 11 | 11 | 2  | 2  | 11 | 11 | 11 | 11 | 11 | 1           | 7,31        |
| 2 (I)        | 11        | 3 | 8 | 1  | 1  | 9 | 11 | 2  | 1  | 2  | 8  | 3  | 3  | 8  | 11 | 11 | 2  | 6  | 5  | 2  | 11 | 11 | 11 | 11 | 11 | 1           | 6,31        |
| 3 (I)        | 11        | 3 | 8 | 1  | 1  | 9 | 11 | 4  | 2  | 1  | 1  | 3  | 4  | 8  | 11 | 11 | 11 | 7  | 7  | 2  | 11 | 11 | 11 | 11 | 10 | 1           | 6,58        |
| 4 (II)       | 11        | 1 | 7 | 1  | 7  | 5 | 7  | 11 | 11 | 2  | 1  | 3  | 11 | 4  | 9  | 11 | 11 | 6  | 1  | 2  | 11 | 11 | 11 | 11 | 11 | 1           | 6,85        |
| 5 (II)       | 11        | 3 | 4 | 11 | 7  | 5 | 7  | 2  | 11 | 1  | 8  | 3  | 11 | 3  | 11 | 11 | 11 | 2  | 1  | 2  | 11 | 11 | 11 | 11 | 11 | 1           | 6,96        |
|              |           |   |   |    |    |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>6,80</b> |             |
| Meerderheid  | 11        | 3 | 8 | 1  | 7  | 9 | 11 | 5  | 2  | 2  | 8  | 3  | 4  | 8  | 11 | 11 | 11 | 6  | 11 | 2  | 11 | 11 | 11 | 11 | 11 | 1           | <b>7,54</b> |

| TF.IDF       |           |    |   |    |    |    |    |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |             |
|--------------|-----------|----|---|----|----|----|----|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|-------------|
| Expert       | Testvraag |    |   |    |    |    |    |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             | Mean        |
| (experiment) | 1         | 2  | 3 | 4  | 5  | 6  | 7  | 8 | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26          |             |
| 1 (I)        | 1         | 11 | 1 | 6  | 11 | 10 | 11 | 4 | 11 | 1  | 1  | 3  | 1  | 8  | 11 | 1  | 11 | 5  | 1  | 1  | 11 | 11 | 2  | 11 | 2  | 6           | 5,88        |
| 2 (I)        | 1         | 11 | 1 | 11 | 11 | 10 | 11 | 4 | 11 | 1  | 11 | 4  | 1  | 6  | 11 | 3  | 11 | 5  | 3  | 11 | 11 | 11 | 2  | 11 | 2  | 10          | 7,12        |
| 3 (I)        | 1         | 11 | 1 | 11 | 11 | 11 | 11 | 4 | 11 | 3  | 11 | 11 | 1  | 4  | 11 | 11 | 11 | 5  | 11 | 11 | 11 | 11 | 2  | 11 | 2  | 10          | 8,04        |
| 4 (II)       | 1         | 11 | 1 | 1  | 1  | 10 | 11 | 4 | 11 | 1  | 2  | 11 | 1  | 8  | 11 | 11 | 11 | 5  | 11 | 6  | 1  | 11 | 2  | 11 | 2  | 10          | 6,38        |
| 5 (II)       | 1         | 11 | 1 | 11 | 11 | 10 | 11 | 4 | 11 | 1  | 2  | 6  | 1  | 6  | 11 | 11 | 11 | 5  | 11 | 2  | 11 | 11 | 2  | 11 | 2  | 10          | 7,12        |
|              |           |    |   |    |    |    |    |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>6,91</b> |             |
| Meerderheid  | 1         | 11 | 1 | 11 | 11 | 10 | 11 | 4 | 11 | 1  | 2  | 6  | 1  | 8  | 11 | 11 | 11 | 5  | 11 | 7  | 11 | 11 | 2  | 11 | 2  | 10          | <b>7,54</b> |

| TF.IDF + STEMMING |           |    |   |   |    |    |    |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             |             |
|-------------------|-----------|----|---|---|----|----|----|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------|-------------|
| Expert            | Testvraag |    |   |   |    |    |    |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |             | Mean        |
| (experiment)      | 1         | 2  | 3 | 4 | 5  | 6  | 7  | 8 | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26          |             |
| 1 (I)             | 1         | 11 | 1 | 6 | 11 | 11 | 11 | 8 | 11 | 1  | 1  | 11 | 1  | 7  | 4  | 1  | 11 | 5  | 1  | 1  | 11 | 11 | 2  | 11 | 3  | 10          | 6,27        |
| 2 (I)             | 1         | 11 | 1 | 6 | 11 | 11 | 11 | 8 | 11 | 1  | 1  | 11 | 1  | 11 | 4  | 2  | 11 | 5  | 1  | 1  | 11 | 11 | 4  | 11 | 3  | 10          | 6,54        |
| 3 (I)             | 1         | 11 | 1 | 6 | 11 | 11 | 11 | 8 | 11 | 3  | 1  | 11 | 2  | 11 | 4  | 11 | 11 | 7  | 11 | 2  | 11 | 11 | 11 | 11 | 3  | 10          | 7,77        |
| 4 (II)            | 1         | 11 | 1 | 1 | 9  | 11 | 11 | 3 | 11 | 1  | 3  | 11 | 2  | 11 | 4  | 11 | 1  | 7  | 11 | 1  | 11 | 11 | 4  | 11 | 3  | 10          | 6,62        |
| 5 (II)            | 1         | 11 | 1 | 1 | 11 | 11 | 11 | 3 | 11 | 1  | 1  | 11 | 2  | 4  | 4  | 2  | 11 | 7  | 6  | 1  | 11 | 11 | 4  | 11 | 3  | 10          | 6,19        |
|                   |           |    |   |   |    |    |    |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | <b>6,68</b> |             |
| Meerderheid       | 1         | 11 | 1 | 6 | 11 | 11 | 11 | 8 | 11 | 1  | 1  | 11 | 2  | 11 | 4  | 2  | 11 | 7  | 11 | 1  | 11 | 11 | 4  | 11 | 3  | 10          | <b>7,04</b> |

## C Voorbeeld output

### Experiment 1

|   |  |
|---|--|
|   | <p>Geachte heer/mevrouw,<br/>Op onze website ... is via de link .../webstat het overzicht te zien van wie onze webpagina bezoekt. Alleen die kan nu iedereen inzien, kan ik dat ook afschermen aangezien het niet wenselijk is dat onze concurrenten deze informatie kunnen inzien.<br/>Groet,<br/>...<br/>Systeembeheerder<br/>... BV</p>   |
| a | <p>L.S., Ik kan via <a href="http://.../stats/index.htm">http://.../stats/index.htm</a> de webstats bekijken. Dit zijn uitstekende stats, maar iedereen kan die zien. Ik heb destijds verzocht of daar een wachtwoord of beveiliging op geplaatst kan worden, aangezien wij niet willen (begrijpelijk) dat de concurrentie onze web activiteit kan bekijken. Ik verneem graag van u.</p>   |
| b | <p>Geachte heer/mevrouw, Graag zou ik meer informatie wensen over de kosten van het doorlinken van onze nieuwe webdomein ... naar ... en de email ... naar ... of het onderbrengen van deze domeinregistratie bij het bedrijf waar we tevens onze andere website hebben gehost.</p>  |
| c | <p>Geachte Heer/Mevr., De website van ... wordt door mij beheerd. Nu is het binnenkort van belang, dat een van de folders in httpdocs met een password beveiligd wordt, zodat slechts een beperkt aantal mensen de inhoud van deze folder kan inzien. Ik neem aan dat de server zo is geconfigureerd, dat met behulp van password-file (gemaakt met het unix utility htpasswd) en een .htaccess file in de betreffende folder(s) de beveiliging kan worden geactiveerd. Mijn vraag: waar kan ik de password file het beste neerzetten en hoe geef ik het pad naar die file het beste aan? Is dat via een absolute referentie, b.v. .../ (voor het geval de file in httpdocs staat) of moet het anders?</p> |
| d | <p>LS Wat is er aan de hand met ..., het domein dat wij bij u hebben geregistreerd? Wij hebben het pas vorige september geregistreerd, hoe kan het dus dat het nu 'vrij' is volgens <a href="http://www.nic.nl/sidn/flat/Domeinnamen/ls_de_naam_nog_vrij_/index.shtml?domein=...&amp;zoek.x=18&amp;zoek.y=12">http://www.nic.nl/sidn/flat/Domeinnamen/ls_de_naam_nog_vrij_/index.shtml?domein=...&amp;zoek.x=18&amp;zoek.y=12</a> ?????? Het is HEEL BELANGRIJK dat het domein weer in de lucht komt, het evenement dat wij organiseren vindt over minder dan een maand plaats en niemand kan onze website bereiken!!</p>  |
| e | <p>Kan het zijn dat door de storing die jullie hebben gehad onze oude site weer is geplaatst. En ook kan ik onze site niet meer bewerken met frontpage zouden jullie dit zo snel mogelijk willen aanpassen dan kan ik onze nieuwe site er weer opzetten. bvd ...</p>   |
| f | <p>L.S., Sinds enige tijd zijn wij door u op een andere server geplaatst om problemen te voorkomen met de bereikbaarheid van onze e-mail en website. Het gaf door spam problemen nog al eens verstoppingen. Het goede nieuws is dat dit niet meer voor gekomen, waarvoor onze dank. Het mindere nieuws is echter het feit dat we nu wel spam ontvangen via onze e-mail accounts terwijl daar eerst geheel geen sprake van was. Graag zouden wij dit probleem dan ook opgelost zien. In afwachting van uw reactie verblijf ik,</p>  |
| g | <p>Hallo, Ik probeer onze website te wijzigen maar ik kan het niet publiceren, hoe kan dit? Onze website is ...</p>  |
| h | <p>Geachte heer / mevrouw, Via Frontpage kan ik de website niet meer uploaden. De volgende melding verschijnt op het scherm: Serverfout: Cannot change permissions on file `usage.html` Het komt vaker voor dat ik niet meer via frontpage niet meer kan uploaden. Iedere keer blijkt dat jullie met de instellingen bezig zijn geweest. Dus hoop ik dat dit nu ook weer het geval is en dat het zo weer is opgelost! Alvast bedankt.</p>  |
| i | <p>Goedemorgen heren, ik heb problemen om via ons netwerk ( windows 2000 server, active directory ...) onze website ... te bereiken. De DNS server vertaalt niet naar onze website bij hostnet. Is het mogelijk om onze website via het IP-adres te bereiken. Groetjes, ...</p>  |
| j | <p>Beste helpdienstmedewerker, Het is nog steeds niet mogelijk voor ons om te uploaden via Frontpage. We hebben het ook al via een andere computer geprobeerd om zeker te weten dat het niet aan onze instellingen ligt. Doordat we af en toe uit de lucht zijn hebben we wel het idee dat er aan gewerkt wordt, maar we weten niet of het probleem opgelost gaat worden. Het meest vervelende is dat klanten boeken bestellen die er al verkocht zijn. Dat is niet zo goed voor onze reputatie van betrouwbaarheid waaraan wij erg hard werken, en gewerkt hebben. Zou u willen berichten of er verwacht wordt dat de uploadproblemen binnenkort opgelost zullen zijn?</p>                                |

## Experiment 2

|   |  |
|---|--|
|   | <p>Geachte heer/mevrouw,<br/> Op onze website ... is via de link .../webstat het overzicht te zien van wie onze webpagina bezoekt. Alleen die kan nu iedereen inzien, kan ik dat ook afschermen aangezien het niet wenselijk is dat onze concurrenten deze informatie kunnen inzien.<br/> Groet,<br/> ...<br/> Systeembeheerder<br/> ... BV</p>  |
| a | <p>Beste ..., Wij hebben inmiddels ... verhuist naar server 22. Deze klant heeft reeds nog problemen met de frontpage extencies. Er staat een ticket open, ticketnr .... Inmiddels is dit ticket veranderd van wachtrij verzoek, naar probleem. Is het mogelijk om de status te weten van dit probleem.</p>  |
| b | <p>Geachte ..., Helaas zijn er in de afgelopen periode een verhoogt aantal stroingen (door spam overlast) geweest op de server waar u website gehost wordt. Als u wilt kunnen wij u website verplaatsen naar een nieuwe server. Met vernieuwde FP server extencies. Als u wilt dat u website middels een andere server beschikbaar wordt kunt ons dit laten weten. Houd u er wel rekening mee dat tijdens de verhuizing (voor een periode van +/- 24 uur) u website en email slecht of niet te bereiken zijn. Ook dient u zelf opnieuw de website te plaatsen. Aangezien u gebruik maakt van FP kunnen wij helaas de website niet voor u verhuizen. Ik hoop dat ik u hiermee voldoende informeer. Mocht u nog vragen hebben dan hoor ik dat graag.</p> |
| c | <p>Geachte ..., Op welke manier publiceert u de website? Gaat dit via programma's als wsftp, of leach ftp. Of gaat dit via Frontpage?</p>  |
| d | <p>Geachte heer/mevrouw, Domein is opgezegd omdat de factuur van 22-8-2003 voor het registreren van dit domein nog niet is betaald, Hij staat op dit moment in de incassofase. Voordat het in de incassofase is gegaan hebben wij u naast de normale factuur ook een herinnering gestuurd naar .... Zodra u de kosten van de domeinnaam betaald zullen wij deze weer voor u registreren.</p>   |
| e | <p>Beste ..., Dit kunnen wij voor u realiseren. De doorlink kost 24,95 Euro per jaar. Ik ontvang graag uw reactie.</p>   |
| f | <p>Geachte ..., We zijn bezig met het oplossen van het probleem. Op het moment dat de problemen zijn opgelost zult u hier bericht over ontvangen. Ik hoop dat ik u hiermee voldoende informeer. Mocht u nog vragen hebben dan hoor ik dat graag.</p>   |
| g | <p>Geachte ..., We hebben nog steeds problemen met het oplossen van de frontpage extencie. We hebben op dit moment twee oplossingen: 1. We wachten op de systeembeheerder betreffende het probleem en laten hen het oplossen. 2. We verhuizen u naar een nieuwe server om zo de problemen op te lossen. Zou u aan kunnen geven welke oplossing u graag wilt hebben?</p>  |
| h | <p>Geachte ..., Het ip-nummer van de machine waarop ... is aangemaakt is: ...</p>  |
| i | <p>Geachte ..., Password protected directories kunt het beste in de controlpanel aanmaken. Die zorgt ervoor dat een juiste bestand wordt aangemaakt in de betreffende map. In onze handleiding kunt u lezen hoe u deze in de controlpanel kan maken. Handleiding kunt u downloaden bij support.hostnet.nl Er zijn nieuwe worden op dit moment nieuwe handleidingen geschreven door mijn collega bij de nieuwe versie wordt meer rekening gehouden met mac systemen. Zodra het af is zal u het in onze support.hostnet.nl pagina kunnen downloaden. Ik hoop dat ik u hiermee voldoende informeer. Mocht u nog vragen hebben dan hoor ik dat graag.</p>  |
| j | <p>Geachte heer ..., De webstat map is nu beveiligd, u kunt de statistieken opvragen middels uw huidige FTP wachtwoord gevolgd door een 1. Uw verzoek voor een odbd koppeling staat momenteel nog open bij systeembeheer, ik zal trachten hier enige spoed achter te zetten voor u. Ik hoop dat ik u hiermee voldoende informeer. Mocht u nog vragen hebben dan hoor ik dat graag.</p>   |

## D Bibliografie

- Aamodt, A. and E. Plaza (1994). "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches." AI Communications 7(1): 39-59.
- Allan, J. (2000). NLP for IR. Tutorial presented at NAACL/ANLP language technology joint conference.
- Allen, B. P. (1994). "Case-based reasoning: business applications." Communications of the ACM 37(3): 40-42.
- Anjewierden, A., W.-O. Huijsen, et al. (2005). Enhancing knowledge mapping using automatically derived concepts. Third International Conference on Knowledge Capture (K-CAP 2005), Banff, Canada.
- Ashley, K. D. (1991). Modelling legal arguments: Reasoning with cases and hypotheticals, MIT Press.
- Baeza-Yates, R. and B. Ribeiro-Neto (1999). Modern Information Retrieval, AddisonWesley.
- Barletta, R. (1993). "Case-based reasoning and information retrieval: Opportunities for technology sharing." IEEE Expert 8(6): 2-4.
- Bergmann, R., S. Breen, et al., Eds. (1999). Developing industrial case-based reasoning applications. Lecture Notes in Artificial Intelligence 1612. Heidelberg, Springer Verlag.
- Bergmann, R. and M. Schaaf (2003). On the relations between structural case-based reasoning and ontology-based knowledge management. Proceedings German Workshop on Experience Management.
- Blair, D. C. (1980). "Searching biases in large interactive document retrieval systems." Journal of the American society for information science 31(4): 271-277.
- Buckley, C. (1985). Implementation of the SMART Information Retrieval System, Technical Report TR85-686, Cornell University.
- Burke, R. D., K. J. Hammond, et al. (1997). "Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder." AI Magazine 18(2): 57-66.
- Callan, J. P., W. B. Croft, et al. (1997). TREC and TIPSTER experiments with INQUERY. Readings in Information Retrieval. K. S. Jones and P. Willett, Morgan Kaufmann: 436-440.
- Carletta, J. (1996). "Assesing agreement on classification tasks: the kappa statistic." Computational Linguistics 22(2): 249-254.
- Carmel, D., M. Shtalhaim, et al. (2000). eResponder: Electronic Question Responder. Proceedings of the 7th International Conference on Cooperative Information Systems, Springer-Verlag.
- Chowdhury, G. G. (2004). Introduction to modern information retrieval, 2nd edition. London, Facet publishing.
- Cleverdon, C. W., J. Mills, et al. (1966). Factors Determining the Performance of Indexing Systems. Vol. 1: Design, Vol. 2: Test Results. ASLIB Cranfield Research Project.

- Croft, W., Ed. (2000). Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval. The Kluwer International Series on Information Retrieval, Kluwer Academic Publishers.
- Croft, W. B. (1995). "What Do People Want from Information Retrieval?" D-Lib Magazine, from <http://www.dlib.org/dlib/november95/11croft.html>.
- Cunningham, P. and A. Bonzano (1999). Knowledge engineering in a real world casebased reasoning application. Technical Report TCD-CS-1999-36. Dublin, Trinity College.
- Fellbaum, C., Ed. (1998). WordNet, an electronic lexical database, MIT Press.
- Fox, C. (1992). Lexical analysis and stoplists. Information retrieval: data structures & algorithms. W. B. Frakes and R. Baeza-Yates.
- Frakes, W. B. (1992). Stemming algorithms. Information retrieval: data structures & algorithms. W. B. Frakes and R. Baeza-Yates: 131-160.
- Gardie, C. (1993). Using decision trees to improve Case-Based learning. Proceedings of the tenth international conference on machine learning, Morgan Kaufmann.
- Gaustad, T. and G. Bouma (2002). "Accurate stemming of Dutch for Text Classification." Language and Computers **45**(1): 104-117.
- Göker, M. and T. Roth-Berghofer (1999a). Development and Utilization of a Case-Based Help-desk Support System in a corporate environment. Lecture Notes in Artificial Intelligence 1650. K.-D. Althoff, R. Bergmann and L. K. Branting. Heidelberg, Springer-Verlag: 132-146.
- Göker, M. and T. Roth-Berghofer (1999b). "Development and Utilization of the Case-Based Help-desk support system HOMER." Engineering applications of artificial intelligence **12**: 665-680.
- Greenwood, M. A. (2001). "Implementing A Vector Space Document Retrieval System." unpublished.
- Hammond, K. (1989). Case-Based Planning: Viewing Planning as a Memory Task, San Diego Academic Press.
- Hull, D. A. (1996). "Stemming Algorithms: A Case Study for Detailed Evaluation." Journal of the American Society of Information Science **47**(1): 70-84.
- Hunter, K. M. (1991). Doctors' stories: The Narrative Structure of Medical Knowledge, Princeton Univ. Press.
- Jones, K. S. (1979). Search term relevance weighting given little relevance information. Readings in Information Retrieval. K. S. Jones and P. Willett: 329-338.
- Jones, K. S. (1997). Search term relevance weighting given little relevance information. Readings in Information Retrieval. K. S. Jones and P. Willett: 329-338.
- Jones, K. S. and P. Willett (1997a). Techniques. Readings in Information Retrieval. K. S. Jones and P. Willett: 305-310.
- Jones, K. S. and P. Willett, Eds. (1997b). Readings in Information Retrieval. The Morgan Kaufmann series in multimedia information and systems, Morgan Kaufmann.
- Jones, K. S. and P. Willett (1997c). Models. Readings in Information Retrieval. K. S. Jones and P. Willett: 257-265.

- Kamp, G., S. Lange, et al. (1998). Related Areas. Case-Based Reasoning Technology, From Foundations to Applications. Heidelberg, Springer Verlag: 327-352.
- Kazman, K. and J. Kominek (1997). Supporting the Retrieval Process in Multimedia Information Systems. Proceedings of the 30th Hawaii International Conference on System Sciences: Digital Documents - Volume 6, IEEE Computer Society.
- Kettler, B. P., J. A. Hendler, et al. (1994). "Massively Parallel Support for Case-Based Planning." IEEE Expert **9**(1): 8-14.
- Kilpelainen, P. and H. Mannila (1993). Retrieval from hierarchical texts by partial patterns. Proceedings of the 16th annual int. ACM SIGIR conference on research and development in IR, Pittsburgh.
- Kolodner, J. (1992). "An introduction to case-based reasoning." AI review **6**: 3-34.
- Kosseim, L., S. Beaugard, et al. (2001). "Using information extraction and natural language generation to answer e-mail." Data & Knowledge Engineering **38**(1): 85-100.
- Koton, P. (1988). Reasoning about evidence in causal explanations. AAAI-88: Proceedings of the Seventh National Conference on Artificial Intelligence, St Paul, MN.
- Kraaij, W. and R. Pohlmann (1994). Porter's stemming algorithm for Dutch. Informatiewetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie. L. G. M. Noordman and W. A. M. d. Vroomen. Tilburg: 167-180.
- Krovetz, R. (1993). Viewing Morphology as an Inference Process. Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, Pennsylvania, United States, ACM Press.
- Lakoff, G. (1987). Women, fire and dangerous things, The university of Chicago Press.
- Lawrence, S. and C. L. Giles (1999). "Searching the Web: General and Scientific Information Access." IEEE Communications **37**(1): 116-122.
- Leake, D. B., Ed. (1996). Case-based reasoning: experiences, lessons and future directions. Indiana University, AAAI Press/MIT Press.
- Lenz, M., E. Auriol, et al. (1998). Diagnosis and Decision Support. Case-Based Reasoning Technology, From Foundations to Applications. M. Lenz, B. Bartsch-Spörl, H.-D. Burkhard and S. Wess. Heidelberg, Springer Verlag: 64-68.
- Lenz, M., B. Bartsch-Spörl, et al., Eds. (1998). Case-Based Reasoning Technology, From Foundations to Applications. Lecture Notes in Artificial Intelligence 1400. Heidelberg, Springer Verlag.
- Lenz, M., H.-D. Burkhard, et al. (1996). "CBR for diagnosis and Decision Support." AI Communications **9**(3): 138-146.
- Lenz, M., A. Hubner, et al. (1998). Textual CBR. Case-Based Reasoning Technology, From Foundations to Applications. M. Lenz, B. Bartsch-Spörl, H. Burkhard and S. Wess. Heidelberg, Springer-Verlag.
- Lin, J. and B. Katz (2003). Question Answering Techniques for the World Wide Web. Tutorial presentation at EAACL.

- Luhn, H. P. (1961). "The automatic derivation of information retrieval encodements from machine-readable texts." Information retrieval and machine translation **3**(2): 1021-1028.
- Maron, M. E. and J. L. Kuhns (1997). On relevance, probabilistic indexing and information retrieval. Readings in Information Retrieval. K. S. Jones and P. Willett: 39-47.
- McLeod, I. A. (1990). "Storage and retrieval of structured documents." Information processing and management **26**(2): 197-208.
- Monz, C. and M. d. Rijke (2001). Tequesta: The University of Amsterdam's Textual Question Answering System. Proceedings of Tenth Text Retrieval Conference (TREC-10).
- Orwant, J., J. Hietaniemi, et al. (1999). Mastering Algorithms with Perl 1st edition, O'Reilly.
- Porter, M. F. (1980). "An algorithm for suffix stripping." Program **14**(3): 130-137.
- Qiu, Y. and H.-P. Frei (1994). Improving the Retrieval Effectiveness by a Similarity Thesaurus. Technical Report 225. Zurich ETH, Department of Computer Science.
- Raghavan, V. V. and S. K. M. Wong (1986). "A critical analysis of vector space mode for information retrieval." Journal of the American society for information science **5**: 279-287.
- Rijsbergen, C. J. v. (1979). Information retrieval. London, Butterworth & Co.
- Rissland, E. L. and J. J. Daniels (1995b). Using CBR to drive IR. Proceedings of the Thirteenth Inter-national Joint Conference on Artificial Intelligence, Morgan Kaufmann.
- Rissland, E. L. and J. J. Daniels (1997). What you saw is what you want: Using cases to seed information retrieval. Proceedings of the Second International Conference on Case-Based Reasoning Research and Development.
- Robertson, S. E. (1997). The probability ranking principle in IR. Readings in Information Retrieval. K. S. Jones and P. Willett: 281-287.
- Robertson, S. E. and K. S. Jones (1976). "Relevance weighting of search terms." Journal of the American society for Information Science **27**: 129-146.
- Ross, B. H. (1989). Some psychological results on case-based reasoning. DARPA Case-Based Reasoning Workshop, Morgan Kaufmann.
- Rydin, S. (2002). Building a hyponymy lexicon with hierarchical structure. Unsupervised Lexical Acquisition: Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX), Philadelphia, Association for Computational Linguistics.
- Salton, G. (1988a). Automatic text processing, Addison-Wesley Longman Publishing Co., Inc.
- Salton, G. (1988b). "Term-Weighting approaches in automatic text retrieval." Information Processing & Management **24**(5): 513-523.
- Salton, G. and M. E. Lesk (1968). "Computer Evaluation of Indexing and Text Processing." Journal of the ACM **15**: 8-36.
- Salton, G. and M. J. McGill (1983). Introduction to modern information retrieval. Auckland, McGraw-Hill.

- Salton, G. and M. J. McGill (1997). The SMART and SIRE experimental retrieval systems. Readings in Information Retrieval. K. S. Jones and P. Willett, Morgan Kaufmann: 381-400.
- Salton, G., A. Singhal, et al. (1996). "Automatic Text Decomposition and Structuring." Information Processing and Management **32**(2): 127-138.
- Salton, G., A. Wong, et al. (1997). A vector space model for automatic indexing. Readings in information retrieval. K. S. Jones and P. Willett, Morgan Kaufmann Publishers Inc.: 273-280.
- Schank, R. C. (1982). "Dynamic memory: A theory of reminding and learning in computers and people."
- Schreiber, A. T., J. M. Akkermans, et al. (2000). Knowledge Engineering and Management: The CommonKADS Methodology. Cambridge, MIT Press.
- Shah, C. and W. B. Croft (2004). Evaluating high accuracy retrieval techniques. Proceedings of the 27th annual international conference on Research and development in information retrieval, Sheffield, United Kingdom.
- Simoudis, E. (1992). "Using case-based retrieval for customer technical support." IEEE Expert **7**(5).
- Tsatsoulis, C. and R. L. Kashyap (1993). "Case-Based Reasoning and Learning in Manufacturing with the TOLTEC Planner." IEEE Expert **23**(4): 1010-1025.
- Watson, I. D., Ed. (1995). Progress in Case-Based Reasoning. Lecture Notes in Artificial Intelligence 1020. Heidelberg, Springer-Verlag.
- Weiss, S. M. and C. V. Apte (2002). "Automated generation of model cases for help-desk applications." IBM systems journal **41**(3): 421-427.
- William, S. C. (1997). Getting beyond Boole. Readings in information retrieval. K. S. Jones and P. Willett, Morgan Kaufmann: 265-267.
- Witten, I. H., A. Moffat, et al. (1999). Managing Gigabytes: Compressing and Indexing Documents and Images, Morgan Kaufmann.
- Wong, S. K. M., W. Ziarko, et al. (1987). "On modeling of information retrieval concepts in vector spaces." ACM Transactions on Database Systems **12**(2): 299-321.
- Xu, J. and W. B. Croft (1998). "Corpus-based stemming using co-occurrence of word variants." ACM Transactions on Information Systems **16**(1): 61-81.