Content Retrieval: Desktop Zoekmachines

Frank Schoep

17-08-2005

Samenvatting

De laatste jaren wordt, door een toenemend gebruik van computers in privé- en bedrijfssfeer, steeds meer informatie opgeslagen op *desktop*- en *workstation*computers. Zoekmachines voor 'gewoon gebruik', vergelijkbaar met internetzoekmachines, zullen in de toekomst hierdoor onmisbaar worden. Kernvragen zijn: welke mogelijkheden en voordelen kan een desktop zoekmachine bieden aan een eindgebruiker.

1 Introductie

1.1 Zoeken op Internet

Google, *Yahoo*, *AltaVista* en *MSN Search*, klinkende namen voor zoekmachines op het Internet. Ze bieden gebruikers een gigantische geïndexeerde database aan met daarin de inhoud van een groot aantal websites die op trefwoorden te doorzoeken is.

De inhoud van die database is afkomstig van zogenaamde *spiders* of *bots*, eenvoudige programma's die pagina voor pagina op zoek gaan naar gewijzigde inhoud en eventuele nieuwe pagina's. De inhoud van opgevraagde pagina's sturen ze door naar databases die later door eindgebruikers worden bevraagd.

Deze versimpelde weergave geeft aan hoe het zoeken op Internet plaatsvindt: op gezette tijden wordt een lange lijst met websites afgegaan op zoek naar nieuwe en gewijzigde inhoud. De beschikbare pagina's worden inclusief hun inhoud opgeslagen in de zoekdatabase. Eindgebruikers bevragen via een *webinterface* de database, waarbij de resultaten van hun zoekactie op relevantie gesorteerd wordt.

1.2 Zoeken op de Desktop

Terwijl zoeken op Internet als vanzelfsprekend wordt ervaren, is het zoeken op de lokale computer altijd een redelijk beperkte ervaring geweest. Gebruikers van *UNIX*-achtige besturingssystemen zoals *Linux* en *BSD*, hebben altijd enige krachtige mogelijkheden in huis gehad om bestanden aan de hand van criteria te vinden, bijvoorbeeld met *indexservices* zoals *slocate* of programma's zoals *find*.

Gebruikers van *Microsoft* producten hebben altijd redelijk in de kou gestaan en zullen een goede zoekfunctionaliteit de komende tijd zeker nog missen. Vanaf *Windows 2000* is er een indexservice aanwezig [3] die gebruikt kan worden om bestandskenmerken vooraf te indexeren en tijdens een zoekactie te gebruiken.

2 Innovatie

2.1 Google Desktop Search

Zoeken op de desktop werd tot enkele maanden voorheen alleen gezien in de context van het vinden van bestanden aan de hand van enkele simpele criteria. Hier kwam pas verandering in toen Google, van nature een Internet-zoekmachine specialist, een gratis applicatie uitbracht die als desktopzoekmachine fungeerde [2].

Plotseling was zoeken op een lokale computer veranderd van een simpele functie die bestanden doorzocht in een totaaloplossing die ook *e-mail* berichten en bezochte websites meenam in de zoekaanvragen. De bevraagmogelijkheden bleven simpel, maar het resultaat dat gepresenteerd werd, was wel veel rijker qua inhoud en tevens op relevantie gesorteerd.

De introductie van de Google desktopzoekmachine zorgde ervoor dat talloze partijen probeerden mee te liften op het succes van het innoverende

werk van Google. In een paar maanden tijd kondigden onder andere *Apple*, *Yahoo* en *Microsoft* aan om met een 'innovatief' nieuw product te komen.

2.2 Besturingssysteemintegratie

Apple en Microsoft hadden de mogelijkheid om als eerste een oplossing te bieden die geïntegreerd was in hun besturingssysteem, respectievelijk *Mac OS X* en *Windows*. De winnaar van die strijd zou hiermee een mooie meerwaarde kunnen bieden ten opzichte van de concurrent.

Op het moment van schrijven is duidelijk wie de eerste strijd heeft gewonnen, Apple heeft onlangs versie 10.4, codenaam *Tiger*, van haar besturingssysteem Mac OS X uitgebracht met daarin een zeer geavanceerde geïntegreerde index- en zoekservice, *Spotlight* [1].

3 Ontwerpbeslissingen

3.1 Introductie

Het maken van een zoekmachine voor gebruik op Internet of de desktop gaat gepaard met het nemen van ontwerpbeslissingen. Er zijn specifieke problemen die komen kijken bij het implementeren van een zoekmachine voor een toepassingsgebied.

Daarnaast is het nodig om te kijken welke bevragingsmogelijkheden een eindgebruiker moet krijgen voor het raadplegen van de zoekmachine en welke broninformatie nodig is voor het kunnen verwerken van aanvragen.

3.2 Internet zoekmachines

Het maken van een zoekmachine voor Internet bestaat voornamelijk uit het verzorgen van een grote dataopslag die op een goede manier geïndexeerd is, maar voor eindgebruikers telt vooral de kwaliteit van het resultaat dat de zoekmachine oplevert. Een belangrijk sleutelwoord hierbij is *relevantie*, de mate waarin een pagina of document voldoet aan de trefwoorden die een eindgebruiker zocht.

De relevantie van een pagina kan op verschillende manieren berekend worden, simpele zoekmachines kunnen bijvoorbeeld uitgaan van het aantal keren dat trefwoorden voorkomen op een pagina. Geavanceerde *algoritmes* zoals bijvoorbeeld Google's *PageRank* bekijken ook in hoeverre er naar een pagina verwezen wordt door andere partijen en hoe vaak er *updates* plaatsvinden op de site.

Een ander groot probleem voor Internet-zoekmachines zijn de zogenaamde *link-spammers*, commerciële jongens die duizenden websites opzetten die allemaal naar elkaar verwijzen om op die manier hoger in de zoekmachine notering terecht te komen. Met behulp van veelgezochte sleutelwoorden proberen ze nietsvermoedende bezoekers naar hun *link-farms* te lokken om ze vervolgens met reclame te bestoken.

Het maken van een goede zoekmachine voor de inhoud van het Internet blijkt door deze factoren een moeilijke klus. De snelheid waarmee een gigantische hoeveelheid data te bevragen moet zijn en het proberen te omzeilen van de trucjes van *spammers*, vormen een struikelblok bij het opstarten van een nieuwe service.

3.3 Desktop zoekmachines

Kijken we naar de toepassing van een zoekmachine op de lokale computer, een *desktop* voor thuisgebruik of een *workstation* in een bedrijfssituatie, dan zijn er een aantal duidelijke verschillen te noemen in vergelijking met een Internet-zoekmachine.

Ten eerste valt op dat de hoeveelheid informatie veel kleiner is, er is dus niet direct behoefte aan een grote dataopslag en een bijbehorend zeer snel zoekmechanisme, het zoeken in een kleinere database vereist van nature al minder tijd.

Een ander verschilpunt is dat er op de lokale machine bijna geen ongewenste resultaten afkomstig van spammers opgeslagen zijn. Alle teksten, e-mails, fotobestanden en videofragmenten, kortom alle documenten die aanwezig zijn, zijn interessant als resultaat voor de eindgebruiker, omdat hij met al deze informatie persoonlijk te maken heeft.

Een gelijkenis met de Internet-zoekmachines is de simpele wens van de gebruiker dat resultaten relevant zijn, als hij op zoek is naar een e-mail die hij aan een collega 'John' heeft gestuurd, is hij niet geïnteresseerd in een CD van John Denver of een videofragment van een film waarin Johnny Depp meespeelt.

Voor een eindgebruiker is ook de *context* van documenten belangrijk, een mogelijk scenario zou zijn dat een gebruiker alle correspondentie met een bepaalde persoon wil opvragen die afgelopen jaar

plaatsvond. Hierbij zijn resultaten die buiten deze periode vallen niet relevant als zoekresultaat.

4 Indexering

4.1 Introductie

Als een desktopzoekmachine effectieve en relevante resultaten wil kunnen presenteren voor een bevraging van een eindgebruiker, moet wel duidelijk zijn op welke gegevens die resultaten gebaseerd moeten worden, we moeten kortom identificeren welke documenteigenschappen essentieel zijn voor het aanbieden van een goede zoekmachine.

4.2 Semantische inhoud

Een eerste belangrijke bron van gegevens voor een zoekmachine is de menselijk leesbare inhoud van een document, ookwel *semantische inhoud* genoemd. Dit lijkt een voor de hand liggende eigenschap, maar door de wildgroei aan bestandsformaten voor documenten blijkt al snel dat de *binaire* inhoud van een bestand geen goede afspiegeling is van de leesbare inhoud.

Als voorbeeld kunnen we het bestandsformaat van de bekende tekstverwerker *Word* nemen. Een poging tot het openen van een bestand in dit formaat in een simpele teksteditor levert op dat de bestandsinhoud geen relatie heeft tot de visuele presentatie en tekstuele inhoud van het document.

Voor het doorzoeken van documenten op leesbare tekst is het nodig om een vertaalslag per bestandsformaat te hebben, zodat de onsamenhangende binaire tekens omgezet kunnen worden in een tekstuele representatie van de inhoud. Deze representatie kan dan later gebruikt worden voor het zoeken op trefwoorden.

Het 'vertaalprobleem' wordt in bestaande zoekmachines opgelost door gebruik te maken van een zogenaamde *plugin-architectuur*. Ieder bestand waarvan de inhoud vertaald moet worden, wordt aan één of meerdere kleine programma's, de plugins, gegeven, die ieder aangeven of ze de vertaling kunnen verrichten. Als een plugin de inhoud kan vertalen, wordt doorgegeven wat de semantische inhoud en kenmerken van het bestand zijn.

4.3 Metadata

Voor een desktop zoekmachine is het van belang dat er, meer nog dan bij een Internet zoekmachine het geval is, een grote hoeveelheid informatie *over* documenten wordt opgeslagen. Dit soort informatie, die zelf ook weer informatie beschrijft, wordt *metadata* genoemd.

Metadata kan opgeslagen worden in documenten zelf, indien het bestandsformaat dit ondersteunt. Een goed voorbeeld hiervan is het *JPEG* bestandsformaat voor afbeeldingen, waar metadata opgeslagen kan worden in zogenaamde *EXIF-headers*. Mogelijke toepassingen hiervan zijn het opslaan van fotokenmerken zoals sluitertijd, diafragma, oriëntatie en witbalans.

Voor bestandsformaten die geen metadata ondersteunen, zal een universele oplossing gebruikt moeten worden. Hierbij valt de denken aan een simpele centrale opslagruimte in de vorm van een catalogus, waar documenten aan toegevoegd kunnen worden inclusief extra metadata.

Een elegantere oplossing, die ook door Apple voor Spotlight wordt gehanteerd, is het opslaan van metadata kenmerken in het *bestandssysteem*. Het bestandssysteem verzorgt de opslag van bestanden en kan dus op zichzelf ingeschakeld worden voor het bijhouden van metadata.

4.4 Context

Het opslaan van simpele metadata kenmerken bij een bestand kan maar in beperkte mate voorzien in de vraag naar geavanceerde zoekmogelijkheden waarbij de context van een document betrokken is. Context betekent hier letterlijk de omgeving waaruit het document afkomstig is en welke relaties het heeft met andere documenten.

Om een eenvoudig voorbeeld te noemen: als een bijlage uit een e-mailbericht opgeslagen wordt op de harde schijf, dan gaat hier bij de huidige besturingssystemen een aanzienlijke hoeveelheid contextuele informatie verloren. Het document wordt vanaf het moment van opslaan namelijk losgemaakt van het originele bericht, waardoor de relatie tot de afzender en de datum van versturen verloren gaat.

Het opslaan van context zou dus voor een deel gerealiseerd worden door het opnemen van verwijzingen naar andere documenten en het opslaan van

relaties met andere eenheden van informatie, ookwel *entiteiten* genoemd.

Zoeken op contextinformatie bij documenten staat momenteel nog in de kinderschoenen, er is geen geïntegreerde oplossing beschikbaar voor de veelgebruikte besturingssystemen. Zelfs Apple's geavanceerde Spotlight zoekmachine bevat geen doorzoekbare contextinformatie.

De eerste bruikbare oplossing voor contextueel zoeken zal waarschijnlijk komen uit het *open-source* kamp. Voor de populaire desktopomgeving *KDE*, het *K Desktop Environment*, wordt momenteel vol overgave gewerkt aan het ontwerpen en realiseren van *Tenor*. Tenor wordt gezien als 's werelds eerste zogenaamde *context linking engine* en moet de desktopzoekmachine als fenomeen naar een hoger niveau brengen.

4.5 Labels

Als we naast de natuurlijke context waarin een bestand voorkomt ook de gebruiker zelf de mogelijkheid willen bieden om zijn bestanden te categoriseren, zodat hij eventueel nog extra informatie kan toevoegen, dan is een mogelijke oplossing het introduceren van *labels* of *tags*. Een document kan meerdere labels bevatten, dus ook in meerdere categorieën zijn opgenomen.

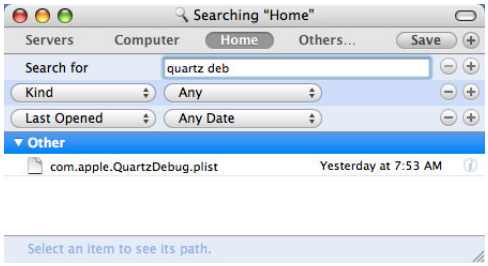
Bij een latere zoekactie kan een gebruiker aangeven dat er rekening gehouden moet worden met bepaalde labels, waardoor bestanden uit die categorieën een hogere, of lagere, relevantie toebedeeld krijgen dan bestanden zonder het label. Op deze manier heeft een eindgebruiker zelf nog een waardevol instrument om toekomstige zoekresultaten gunstig te beïnvloeden.

5 Bevragen

5.1 Gebruikersinterface

Het bevragen van de desktop zoekmachine moet op een manier gebeuren die de gebruiker in staat stelt om op een simpele manier gebruik te maken van de geavanceerde zoekindex. Simpele scenarios waarin alleen een bestandsnaam gegeven wordt moeten ook afgehandeld kunnen worden.

Een mooie oplossing voor dit probleem, waarin een gebruiker stap voor stap zijn zoekvraag kan



Figuur 1: Het opgeven van meerdere zoekcriteria in Spotlight.

formuleren, is de methode die in Spotlight wordt gehanteerd. Een eindgebruiker kan op een visuele manier criteria toevoegen aan de zoekactie, zie figuur 1 op bladzijde 4.

Als er een extra criterium is toegevoegd, verschijnt daaronder direct een invoerregel voor een volgend criterium. Een mooie eigenschap van Spotlight, die vele in ontwikkeling zijnde desktop zoekmachines proberen na te streven, is dat resultaten van de zoekactie in *real-time* worden getoond. Een gebruiker hoeft dus niet te wachten of op een knop te klikken om het resultaat van zijn actie te zien.

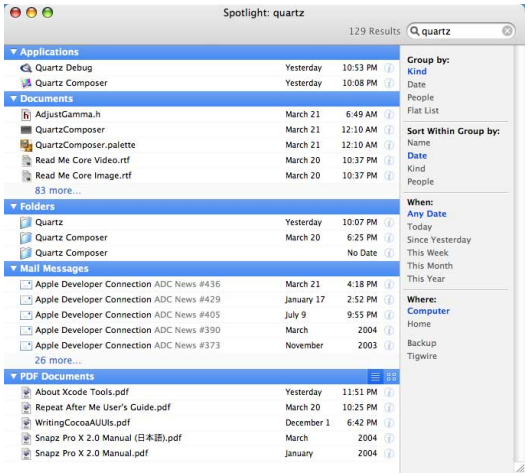
Een gebruikersinterface voor het opvragen van documenten op basis van contextinformatie bestaat op dit moment nog niet, maar het *Tenor* project houdt wel vanaf de start in het oog hoe de gebruiker het systeem moet gaan gebruiken.

5.2 Presenteren resultaat

Belangrijk bij de presentatie van het resultaat is dat de meest relevante resultaten duidelijk getoond worden. Een lijst is de meest natuurlijk representatie van een zoekresultaat en leent zich uitstekend voor het opsommen van gevonden documenten. Relevantie kan in een lijst goed aangegeven worden door ordening, de meest relevante resultaten worden bovenaan getoond.

Naast sorteren op relevantie is het mogelijk om resultaten te categoriseren, of te groeperen. Een goed voorbeeld van flexibiliteit wordt door Spotlight aangeboden, waarbij de gebruiker zelf kan aangeven hoe resultaten gegroepeerd moeten worden en hoe de sortering per groep plaatsvindt.

Een resultatenscherm van Spotlight is weergegeven in figuur 2 op bladzijde 5. Naast het simpelweg



Figuur 2: Spotlight zoekresultaatscherm met gebruikersopties.

bieden van een lijstrepresentatie van het resultaat kan in Spotlight ook aangegeven worden dat plaatjes als miniaturen moeten worden weergegeven.

Vanuit het resultaat van een zoekactie kan direct doorgeklikt worden naar het echte document dat in het resultaat is weergegeven.

6 Overwegingen

6.1 Voordelen

De voordelen van een desktopzoekmachine voor een eindgebruiker zijn duidelijk, ten eerste biedt een desktopzoekmachine een veel breder zoekgebied dan een simpele zoekfunctie voor bestanden. Denk hierbij aan het feit dat een desktopzoekmachine bijvoorbeeld ook e-mails, chatconversaties en bezochte websites mee kan nemen in het zoekresultaat.

Daarnaast kan met een fijnere *granulariteit* opgegeven worden wat er gezocht wordt, waardoor de relevantie van resultaten aanzienlijk toeneemt. Het zoeken veranderd hierdoor, mits een juiste zoekvraag wordt gesteld, in het daadwerkelijk vinden van het gewenste document en verwante entiteiten.

Een ander groot voordeel is dat het resultaat bijna direct gepresenteerd wordt, een goede indexservice gekoppeld aan een goed zoekalgoritme kan, net als Spotlight, in real-time resultaten presenteren terwijl de gebruiker zijn vraag typt. Hierdoor kan een gebruiker gelijk zijn zoekvraag bijwerken en zien of dat het gewenste resultaat oplevert.

Een laatste voordeel, dat momenteel nog aanwezig is, is het het zoeken op context. Hoewel deze term nu misschien nog vaag overkomt, zal deze vorm van informatie in de toekomst zeker een rol spelen in het opgeven van zoekcriteria en het bepalen van relevantie van documenten.

6.2 Nadelen

Voor de gebruiker lijken aan een desktopzoekmachine niet direct nadelen te kleven, er worden immers alleen extra mogelijkheden geboden terwijl een simpele zoekfunctie ook nog steeds mogelijk is.

Er zijn echter wel nadelen te vinden aan de technische implementatie van een zoekmachine voor de desktop. Het belangrijkste punt is dat er een indexservice aanwezig moet zijn voor het kunnen indexeren van documenten. Deze service kan in principe op twee manieren werken:

De eerste methode is *Interval scanning*, waarbij op gezette tijden de hele computer wordt afgezocht naar nieuwe en gewijzigde documenten. Tijdens de zoekactie wordt vooral de harde schijf zwaar belast waardoor de computer enige tijd langzamer wordt en vooral minder snel reageert op gebruikersacties.

Het alternatief is gebruik te maken van *Continue updates*, zodra er nieuwe documenten worden aangemaakt of bijgewerkt worden de wijzigingen direct opgeslagen in de zoekindex. Het voordeel is dat er geen lange periodes van intensieve activiteit zijn, maar het nadeel is dat het opslaan en bijwerken van documenten wel iets langzamer is dan normaal het geval is.

Naast het snelheidsaspect kan ook nog een eventueel nadeel gevonden worden in *privacy*, de zoekmachine moet namelijk wel goed afgeschermd worden tegen onrechtmatig gebruik. In de zoekindex zijn vooral persoonlijke en vertrouwelijke documenten opgenomen, die niet toegankelijk moeten zijn voor derden, ze mogen zelfs niet van het bestaan van die bestanden afweten.

Dit privacyprobleem is echter goed op te lossen door de zoekindex te koppelen aan een losse *gebruikersaccount* op het systeem, waardoor anderen geen toegang kunnen krijgen tot de zoekmachine. Spotlight gebruikt bijvoorbeeld een account gekoppelde zoekindex.

6.3 Conclusie

De desktopzoekmachine is *hot* en dat is niet zonder reden, er zijn een groot aantal voordelen te behalen door het gebruik van een zoekmachine, terwijl er weinig nadelen aan zitten. Het enige concrete systeem dat momenteel verkrijgbaar is, is Apple's Spotlight, maar de concurrentie zal de komende jaren zeker niet stil zitten.

Het *Tenor* project, dat dankzij contextuele informatie een nog rijker resultaat wil bieden, is zeker een kandidaat voor de directe opvolging van Spotlight en verwanten. Het toevoegen van contextinformatie is belangrijk genoeg om het te zien als een stap verder dan metadata geörienteerde zoekmachines.

7 Dankwoord

Ik wil graag de volgende reviewers bedanken voor de beoordeling van het artikel en het aandragen van suggesties ter verbetering:

- Dalen, Maiko van
- Hubregtse, Leendert
- Pol, Hans van de
- Verwerft, Johan

Referenties

[1] Apple Computer Inc. Mac OS X - Spotlight. http://www.apple.com/macosx/features/spotlight/.

[2] Google. Google Desktop Search. http://desktop.google.com/.

[3] Microsoft Corporation. Glossary of Windows 2000 Services. http://www.microsoft.com/windows2000/techinfo/howitworks/management/w2kservices.asp.