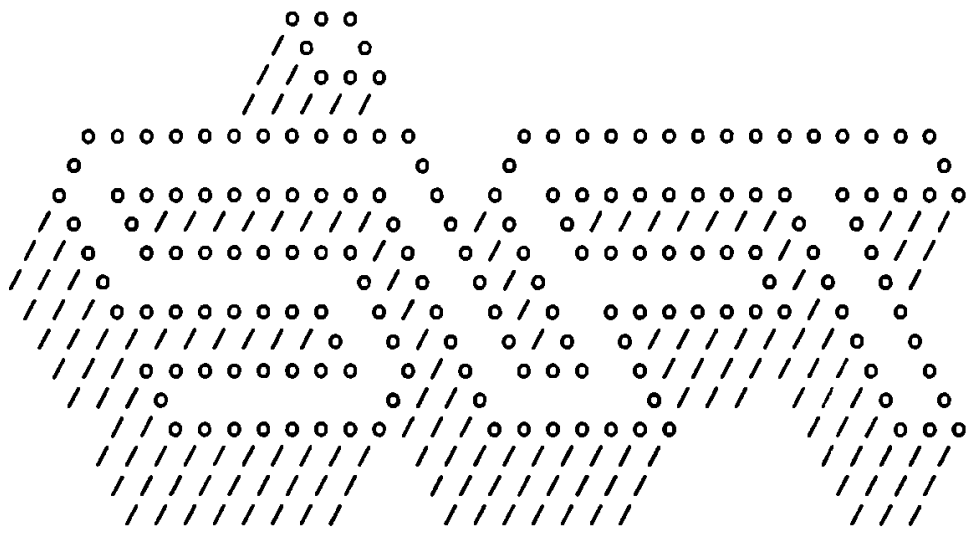
Øystein Reigem
NAVF's EDB-senter for humanistisk forskning



SIFT (Søking I Fri Tekst) - ET GENERELT INFORMASJONSSØKESYSTEM

Generelt om informasjonssøking og anvendelser.

Informasjonssøking eller information retrieval (IR) er en samlebetegnelse på forskjellige teknikker for gjenfinning av relevante undermengder av en større mengde informasjonsenheter. Disse enhetene kan være vitenskapelige artikler, sammendrag av artikler, brevene i en persons korrespondanse, opplysninger om museumsgjenstander, replikker fra et skuespill, lovtekster, domsavsigelser, overvåkingspolitiets personopplysninger m.m. Tradisjonelt forbinder man gjerne bruken av IR-systemer med søking i ustrukturert informasjon, dvs. vanlig tekst o.l. Det er imidlertid et klart behov for systemer som tilbyr søking i både ustrukturert og strukturert informasjon. (Med strukturert informasjon mener en faste datafelter som forfatter, navn på gjenstand, replikkinnehaver, dato for domsavsigelse, postadresse eller hårfarge.) Et godt IR-system trenger altså først funksjoner for å strukturere informasjonen systemet skal lagre, og dernest muligheter for å identifisere informasjonsenheter både ved ord i fritekstsammenheng og opplysninger i faste felter.

Bruken av IR-systemer spenner over svært store områder. Imidlertid går mange trekk ved anvendelsene igjen. Dette stiller bestemte krav til sentrale funksjoner og egenskaper ved generelle IR-systemer.

Store datamengder er karakteristisk for mange anvendelser. For å sikre en rask gjenfinning av informasjonsenheter - eller "dokumenter" som vi vil kalle dem - må systemet opprette og vedlikeholde ekstra datastruk-

SIFT (Søking I Fri Tekst) - Et generelt informasjonssøkesystem
Øystein Reigem
Proceedings of NODALIDA 1981, pages 194-200

turer. Den vanlige strategien består i å ha en såkalt invertert fil i tillegg til "dokumentfilen". Den inverterte filen inneholder en sortert liste over alle ord som forekommer i dokumentene, samt referanser til de enkelte forekomstene. Det går kvikt å slå opp i en sortert fil. Derfor går søking raskt, men det blir lett konflikter hvis en også ønsker en effektiv oppdatering. Dette ser en ved mange IR-systemer i dag.

Likevel er søkingen den kritiske faktoren i et IR-system. Siden IR-databaser pleier å være statiske, blir søking den hyppigste aktiviteten i systemet. Søking foregår gjerne vha. et søkespråk der man kan kombinere relevante ord med diverse operatorer. Typiske operatorer er logiske (og, eller, ikke), avstands- (søker på ord innen en viss avstand fra hverandre) og trunkeringsoperatorer (søker på starten / slutten av ord). Siden søking er så viktig og vanlig, bør søkespråket være kraftig, dvs. inneholde et variert reportoar av operatorer og funksjoner. Språket må også være lett å lære i sin enkleste form, slik at systemet kan nyttes av uerfarne brukere.

Den som en gang har brukt et IR-system, vet at gjenfinning av relevante dokumenter lett blir en prøve- og feileprosess. En forsøker seg gjerne fram med varianter av samme spørsmål, og en legger skjerpende betingelser på tidligere spørsmål. Også dette stiller krav til søkespråket, men når en skal kontrollere resultatet av hvert spørsmål, er det viktig å ha gode og brukervennlige funksjoner for "blading" i dokumentene. Spesielt i anvendelser der en har store dokumenter (større enn en dataskjerm, f.eks.), bør en ha muligheter for fokusering av avsnitt som inneholder søkebegrep eller visuell framheving av søkebegrep på skjermen / papiret.

Men uansett hvor sofistikert man lager et IR-system, vil det være spesielle anvendelser som ikke får alle sine behov dekket. Det er derfor viktig at systemet lett lar seg modifisere eller at det kan kommunisere med applikasjonsorienterte systemer. Dette krever en modulær oppbygning med veldefinerte grensesnitt utad og mellom delene. Dette er sjelden kost blant dagens IR-systemer. En slik oppbygning gir også muligheter for reduserte mikromaskinutgaver, eller oppdelte versjoner som kan kjøres på et nett av mikromaskiner.

Blant annet fordi en IR-database ofte har lengre levetid enn maskinutstyret den kjøres på, er portabilitet et viktig krav til et IR-system. (Et system sies å være portabelt dersom det ikke kreves mye arbeid å flytte det til en annen maskintype.) Også her lar eksisterende IR-systemer mye tilbake å ønske. Portabilitet fordrer disiplinert og gjennomtenkt programmering.

SIFT-prosjektet.

SIFT-prosjektet ble startet 1-1-80 og pågår under ledelse av Statens Rasjonaliseringsdirektorat. Prosjektet støttes av NTNF, og følgende institusjoner deltar:

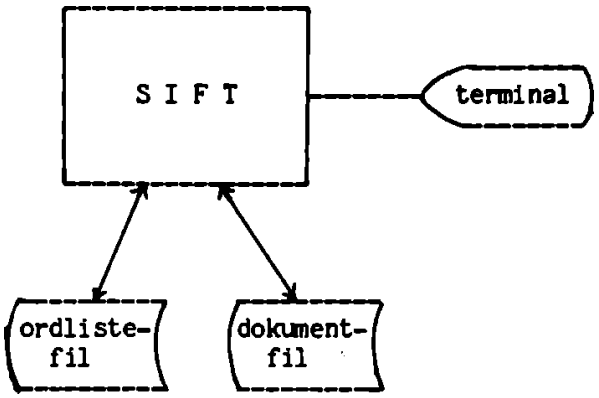
Norsk Data A/S
NAVF's EDB-senter for humanistisk forskning
LOVDATA
Institutt for privatrett, avdeling for EDB-spørsmål, UiO.

Målet med prosjektet er å utvikle et slagkraftig og fleksibelt IR-system som skal stilles til disposisjon for interesserte brukere. Under utviklingen bygger en på erfaringer med andre IR-systemer, spesielt den norske versjonen av det britiske STATUS-systemet, NOVA*STATUS. NOVA*STATUS har funnet anvendelse ved alle norske universiteter og i en rekke offentlige institusjoner.

SIFT-systemet.

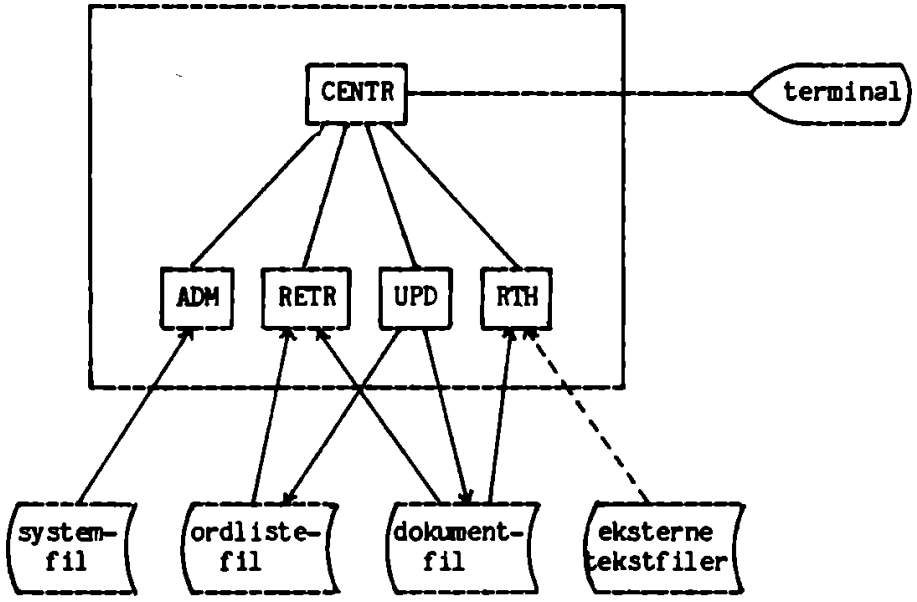
Basisversjonen av SIFT vil være et maskinuavhengig flerbrukersystem. Systemet styres av kommandoer som brukerne gir. (Det motsatte ville være at systemet til enhver tid presenterte valgmuligheter.) Kommandoene har posisjonsbestemte parametre, dvs. at rekkefølgen er viktig. (For eksempel er SLETT DOKUMENT 2-5 en riktig kommando, mens SLETT 2-5 DOKUMENT er gal.) Imidlertid er endel egenskaper lagt inn for å hjelpe uerfarne brukere. Systemet tilbyr en relativt omfattende HELP-funksjon (forklaringer på alle kommandoer og parametre), muligheten for å bli spurt etter en og en parameter i en dialog med systemet, og en omfattende bruk av "defaultverdier" (standardverdier) for de parametrene som brukeren ikke gir.

Slik kan man tenke seg SIFT (og for så vidt mange andre IR-systemer):



Dokumentfilen inneholder selve dokumentene, opplysninger om relasjoner mellom disse og om intern dokumentstruktur. Ordlistefilen inneholder en sortert liste over ord som forekommer i dokumentene, samt referanser til selve forekomstene. Søking foregår stort sett ved oppslag i ordlistefilen.

Her er en mer detaljert og omfattende figur:



I tillegg til den interne strukturen til SIFT, er det kommet flere filer med på figuren. Systemfilen inneholder mest mulig av systemparametre, slik at optimalisering av systemet for en bestemt anvendelse skal kunne foretas med få eller ingen inngrep i selve programmene. Her ligger også alle kommandonavn og meldinger slik at en lettvint kan generere versjoner for brukergrupper med annen språkbakgrunn. De eksterne tekstfilene er tatt med for å antyde at SIFT også skal kunne håndtere dokumenter på filer utenfor selve systemet. Det er et viktig poeng hvis man tenker på utvidelser eller integrering med andre systemer.

SIFT er fullstendig modulært oppbygd. "På toppen" sitter sentralmodulen (CENTR) og styrer de 4 andre hovedmodulene - administrasjonsmodulen (ADM), gjenfinningsmodulen (RETR), oppdateringsmodulen (UPD) og modulen som tar seg av søkeresultater (RTH). Hovedmodulene er oppbygd av undermoduler på samme hierarkiske måte som systemet selv, disse undermodulene av undermoduler igjen, osv.

All kommunikasjon med brukerne går gjennom sentralmodulen. Internt i systemet blir alle kommandoer omformet til "kommandopakker" som starter opp aktiviteter i en eller flere av de 5 hovedmodulene. Kommunikasjonen mellom sentralmodulen og de andre modulene går gjennom helt standardiserte grensesnitt. Derfor kan deler av systemet lett skiftes ut med eller modifiseres til applikasjonsorienterte versjoner. De ulike delene av systemet vil til og med kunne operere på forskjellige maskiner.

Av hovedmodulene er gjenfinnings- og oppdateringsmodulene de mest sentrale. Det er disse som leser og skriver på ordlistefilen. Denne filen er i SIFT organisert som et såkalt B-tre. Nodene i treet ("sidene" i ordlisten) er gitt en nokså sammensatt struktur. Denne organiseringen

løser langt på vei konflikten mellom effektiv søking og oppdatering, og gir dessuten mulighet for å optimalisere systemet med hensyn på en av de to prosessene.

Strukturer.

En samling sammenhørende dokumenter utgjør det vi vil kalle en database. Når et dokument innlemmes i en database, blir teksten organisert i felter, avsnitt, setninger, fraser og ord etter de kriterier brukeren har satt for den typen dokument. (En database kan inneholde flere typer dokumenter.) Referanser til lite meningsbærende ord kan sløyfes i ordlisten for å spare plass. Uønskede dokumentavsnitt kan utelukkes fra ordlisten og forstyrrer dermed ikke søkingen.

I en database kan brukeren gruppere dokumentene i mengder, og dokumenter kan defineres som historiske versjoner av hverandre. Søking kan foretas i flere databaser samtidig.

I tillegg til strukturering av databasene og de enkelte dokumentene, kan brukeren bygge opp tesauri til bruk i søkingen. Hun kan definere egne relasjoner og siden knytte sammen ord vha. disse relasjonene. Søkingen på en term kan så utvides til også å omfatte termene som står i bestemte relasjoner til denne. Tesaurusrelasjonene kan være av 4 typer: 1) Nettverksstrukturer, dvs. relasjoner av typen beslektede termer. 2) Trestrukturer, dvs. relasjoner av typen mer og mindre generell term. 3) Likeverdige, dvs. relasjoner av synonymtype. 4) Relasjoner mellom ord og forklaring på hvordan de brukes.

Søking.

Søkespråkets elementer er ord, maskete ord, fraser, operatorer og navn på felter og brukerdefinerte mengder av dokumenter.

En frase er en sekvens av ord definert som en sammenhørende enhet. Brukeren kan velge om de enkelte ordene i en frase skal være søkbare.

Masking av søkeord vil si at søkeordene inneholder spesielle tegn som står for mer eller mindre vilkårlige tegnstrenger. Eksempler: Dersom * er definert til å stå for en hvilken som helst tegnstreng, vil en søking etter *problem* finne alle dokumenter med ord som inneholder tegnstrengen problem, f.eks. problematikk, kommunikasjonsproblem, problemenes, problem osv. Dersom £ står for ett siffer, vil en søking etter 19££ finne alle dokumenter med firsifrede tall som begynner på 19.

Operatorene faller i 4 grupper: Logiske operatorer, avstandsoperatoren, operatorer som refererer til innholdet av navngitte felt og klasse-operatoren.

Blant de logiske finner vi de vanlige ELLER, OG, OGIKKE og IKKE. I tillegg kommer AVSN og IKKEAVSN, SETN og IKKESETN som er OG- og OGIKKE-operatorer innen samme avsnitt / setning. Eksempel: Spørsmålet usa SETN

nøytronbombe gir oss alle dokumenter der disse to ordene forekommer i samme setning.

Avstandsoperatoren setter grenser for hvor langt ord kan stå fra hverandre. Eksempler: Spørsmålet produ* /3/ nøytron* gir oss alle dokumenter hvor to ord som begynner på produ og nøytron står i høyst 3 ords avstand fra hverandre. Spørsmålet produ* nøytron* krever at ordene står etter hverandre med produ... først.

Innhold av felt kan gjøres til søkekriterium vha. operatorene LIK, MI, ML, ST, SL og MELLOM. Operatorene står for henholdsvis "lik", "mindre enn", "mindre eller lik", "større enn", "større eller lik" og "mellom". Sammenligningen foretas tegn for tegn i tekstfelt, og for tallet som helhet i tallfelt.

Ofte kan I-operatoren være bedre å bruke enn LIK. Dersom forfatter er et felt, vil spørsmålet asbjørnsen I forfatter gi alle dokumenter hvor ordet asbjørnsen forekommer i forfatterfeltet, selv om det skulle stå noe annet der også.

Dersom et spørsmål gir en viss mengde dokumenter som resultat, kan SAMME-operatoren brukes til å inkludere flere dokumenter i resultatet, nemlig alle dem med samme verdi i et bestemt felt, eller alle dem som ligger i samme brukerdefinerte mengder.

Klasseoperatoren kan gis eksplisitt eller den kan gis implisitt i klassekommandoen. I tillegg til vanlig søking, tilbyr nemlig SIFT såkalt klassesøking. Brukeren gir da et sett med spørsmål. Hvert spørsmål identifiserer en klasse. Dokumentene rangeres etter hvor mange klasser de tilhører, dvs. hvor mange spørsmål de tilfredsstiller. Brukeren kan få en mer eller mindre detaljert oversikt over klassefordelingen. Klassesøking er lett å bruke og har ofte vist seg å gi vel så gode resultat som kompliserte spørsmål basert på bare logiske operatorer.

SIFT kan vise brukeren alle ord eller termer som tilfredsstiller et bestemt uttrykk, f.eks. et masket ord eller en tesaurusrelasjon. Brukeren kan så på sin side bruke et utvalg av ordene som basis for nye spørsmål.

Brukeren kan lettvint referere til tidligere stilte spørsmål og også inkludere disse i nye. Spørsmål eller deler av spørsmål som går igjen kan brukeren definere som såkalte makroer. En makro gis et navn som den refereres ved. En makro kan ha argumenter, dvs. "åpne rom" for varierende deler. Brukeren kan bygge opp sine egne filer med makroer som kan hentes inn etter behov. En kan også ta vare på alle sine vanlige spørsmål på denne måten.

Resultathåndtering.

SIFT har et godt sett med kommandoer for å "bla" i dokumentene. Blading foregår både i mengder av dokumenter og innen lange dokumenter. Aktuelle mengder av dokumenter er: resultatet av et spørsmål, de historiske versjonene av et dokument, en brukerdefinert mengde eller databasen som

helhet. Det finnes en printkommando som gir utskrift på linjeskriver eller fil. Brukeren kan selv definere formater for visning og utskrift ved å spesifisere et utvalg av felter og avsnitt. "Highlight"- og fokuseringskommandoene framhever søkeordene og deres omgivelser, og gjør det dermed lettere å sjekke relevansen av resultatet.

Dokumentene i en database har en innbyrdes ordning bestemt ved generering/oppdatering. Resultatet av et spørsmål vil være en undermengde ordnet i samme rekkefølge. Men brukeren kan også få dokumentene rangert etter hyppighet av søkeord eller sortert på angitte felter.

I tillegg til dette skal den endelige versjonen av SIFT inneholde ikke-interaktive, men mer avanserte sorterings- og utskriftsfunksjoner. Disse skal også omfatte frekvenslistinger på grunnlag av ordlistefilen.

Framdrift. Tilgjengelighet.

SIFT-prosjektet er delt i flere trinn. Det nåværende arbeidet går på en prototyp for NORD-maskin (SIFT-1). Prototypen planlegges ferdig innen utgangen av 1981. Andre trinn er utviklingen av en maskinuavhengig versjon (SIFT-PORTABLE). Disse første versjonene vil ikke inneholde alle egenskapene nevnt her. Tredje trinn består imidlertid i å utvide SIFT-PORTABLE til en fullstendig versjon (SIFT-COMPLETE). Siden kan en tenke seg ytterligere utvidelser som f.eks. skjermorientert kommandohåndtering, spørsmål i naturlig språk, lagring av dokumentrelasjoner vha. et ordinært databasesystem osv.

I samsvar med prosjektets målsetting vil all programvare og alle rapporter bli stilt til disposisjon for interesserte parter gratis.

Norsk Data A/S vil sannsynligvis markedsføre en SIFT-versjon spesielt tilpasset NORD.